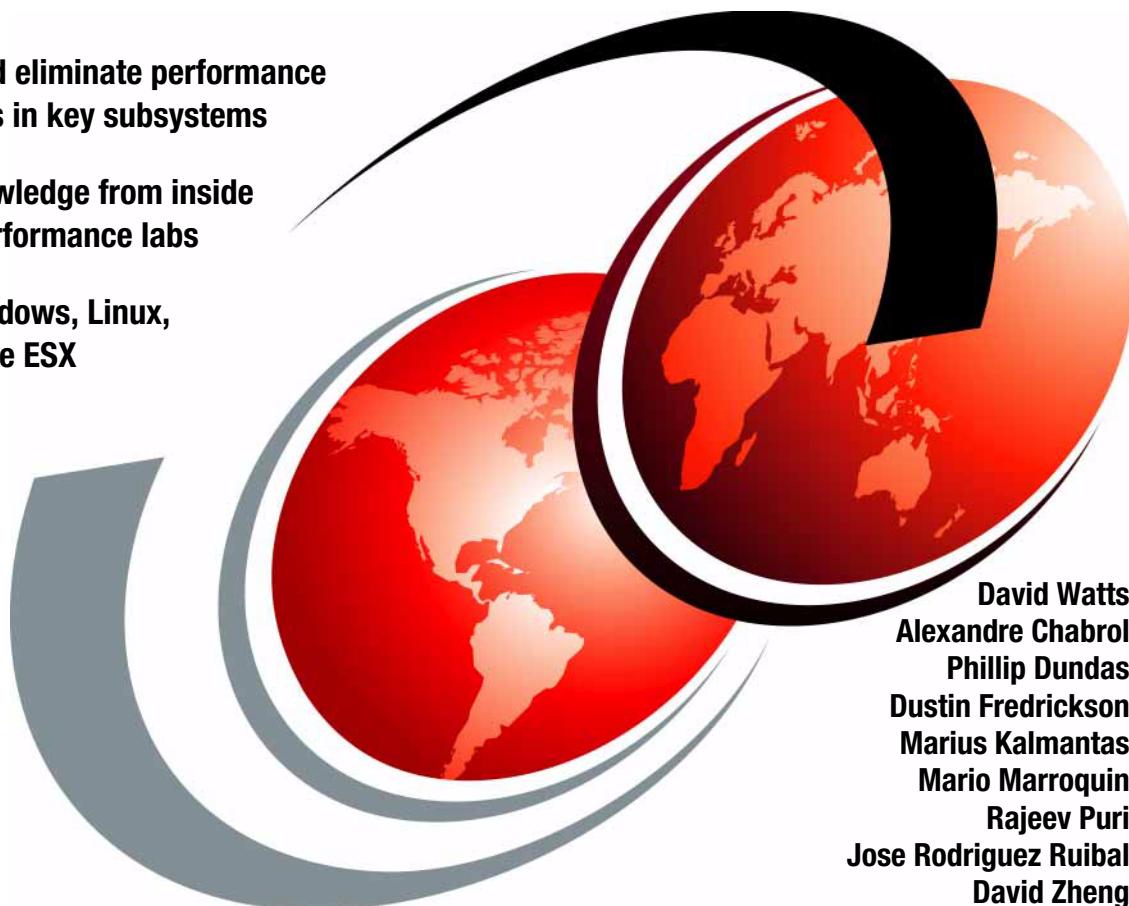


# Tuning IBM System x Servers for Performance

Identify and eliminate performance bottlenecks in key subsystems

Expert knowledge from inside the IBM performance labs

Covers Windows, Linux, and VMware ESX



# Redbooks





International Technical Support Organization

**Tuning IBM System x Servers for Performance**

August 2009

**Note:** Before using this information and the product it supports, read the information in "Notices" on page xvii.

### **Sixth Edition (August 2009)**

This edition applies to IBM System x servers running Windows Server 2008, Windows Server 2003, Red Hat Enterprise Linux, SUSE Linux Enterprise Server, and VMware ESX.

**© Copyright International Business Machines Corporation 1998, 2000, 2002, 2004, 2007, 2009. All rights reserved.**

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP

# Contents

<b>Notices</b> .....	xvii
Trademarks .....	xviii
<b>Foreword</b> .....	xxi
<b>Preface</b> .....	xxiii
The team who wrote this book .....	xxiv
Become a published author .....	xxix
Comments welcome .....	xxix
<b>Part 1. Introduction</b> .....	1
<b>Chapter 1. Introduction to this book</b> .....	3
1.1 Operating an efficient server - four phases .....	4
1.2 Performance tuning guidelines .....	5
1.3 The System x Performance Lab .....	5
1.4 IBM Center for Microsoft Technologies .....	7
1.5 Linux Technology Center .....	7
1.6 IBM Client Benchmark Centers .....	8
1.7 Understanding the organization of this book .....	10
<b>Chapter 2. Understanding server types</b> .....	13
2.1 Server scalability .....	14
2.2 Authentication services .....	15
2.2.1 Windows Server 2008 Active Directory domain controllers .....	15
2.3 File servers .....	17
2.4 Print servers .....	18
2.5 Database servers .....	18
2.6 E-mail servers .....	20
2.7 Web servers .....	21
2.7.1 Web 2.0 servers .....	22
2.8 Groupware servers .....	22
2.9 Multimedia server .....	23
2.10 Communication server .....	24
2.11 Terminal server .....	25
2.12 Infrastructure servers .....	26
2.12.1 DNS server .....	26
2.12.2 DHCP server .....	27
2.12.3 WINS server .....	27

2.13 Virtualization servers . . . . .	28
2.14 High Performance Computing . . . . .	28
<b>Chapter 3. Performance benchmarks for servers . . . . .</b>	<b>31</b>
3.1 IBM and benchmarks . . . . .	32
3.1.1 The different kinds of benchmarks . . . . .	32
3.1.2 Why IBM runs benchmarks on System x servers . . . . .	33
3.2 The main industry standard benchmarks . . . . .	33
3.2.1 Types of information that benchmarks can provide . . . . .	34
3.2.2 System benchmarks . . . . .	35
3.2.3 Product-specific benchmarks . . . . .	39
3.2.4 Industry standard benchmark results on IBM System x . . . . .	40
3.3 Understanding IBM System x benchmarks . . . . .	41
<b>Part 2. Server subsystems . . . . .</b>	<b>43</b>
<b>Chapter 4. Introduction to hardware technology . . . . .</b>	<b>45</b>
4.1 Server subsystems . . . . .	46
<b>Chapter 5. Energy efficiency . . . . .</b>	<b>49</b>
5.1 Importance of finding an energy efficiency balance . . . . .	51
5.2 Server-level solutions . . . . .	53
5.2.1 Processors . . . . .	54
5.2.2 Memory . . . . .	60
5.2.3 Drives . . . . .	64
5.2.4 Fans . . . . .	64
5.2.5 Power supplies . . . . .	65
5.2.6 Operating systems . . . . .	68
5.2.7 Virtualization . . . . .	72
5.3 Rack-level solutions . . . . .	76
5.3.1 IBM products to manage energy efficiency . . . . .	76
5.3.2 IBM intelligent power distribution units . . . . .	81
5.3.3 Hardware consolidation . . . . .	82
5.4 Data center-level solutions . . . . .	84
5.5 Power and performance benchmarks . . . . .	88
5.5.1 The benchmark description . . . . .	88
5.5.2 The benchmark methodology . . . . .	89
5.6 Resources . . . . .	91
<b>Chapter 6. Processors and cache subsystem . . . . .</b>	<b>93</b>
6.1 Processor technology . . . . .	94
6.2 Intel Xeon processors . . . . .	94
6.2.1 Dual-core processors . . . . .	95
6.2.2 Quad-core processors . . . . .	98

6.2.3 Six-core processors .....	101
6.2.4 Intel Core microarchitecture .....	103
6.2.5 Intel Nehalem microarchitecture .....	105
6.3 AMD Opteron processors .....	109
6.3.1 AMD Revision F (1207 socket) Opteron .....	109
6.3.2 AMD quad-core Barcelona .....	111
6.3.3 AMD quad-core Shanghai .....	112
6.3.4 Opteron split-plane .....	112
6.3.5 IBM CPU passthru card .....	113
6.4 64-bit computing .....	116
6.5 Processor performance .....	122
6.5.1 Comparing CPU architectures .....	123
6.5.2 Cache associativity .....	123
6.5.3 Cache size .....	127
6.5.4 Shared cache .....	128
6.5.5 CPU clock speed .....	128
6.5.6 Scaling versus the number of processor cores .....	129
6.5.7 Processor features in BIOS .....	130
<b>Chapter 7. Virtualization hardware assists</b> .....	133
7.1 Introduction to virtualization technology .....	134
7.1.1 Privilege levels .....	134
7.1.2 Binary translation and paravirtualization .....	135
7.1.3 Memory-intensive workload .....	136
7.1.4 Nested paging .....	137
7.2 Virtualization hardware assists .....	138
7.2.1 CPU command interface enhancements .....	138
7.2.2 Intel VT .....	139
7.2.3 AMD-V .....	141
7.3 Support for virtualization hardware assists .....	143
7.4 Resources .....	144
<b>Chapter 8. PCI bus subsystem</b> .....	145
8.1 PCI and PCI-X .....	146
8.2 PCI Express .....	149
8.2.1 PCI Express 2.0 .....	151
8.2.2 PCI Express performance .....	152
8.3 Bridges and buses .....	153
<b>Chapter 9. Chipset architecture</b> .....	157
9.1 Overview of chipsets .....	158
9.2 System architecture design and performance .....	159
9.2.1 Hardware scalability .....	160
9.2.2 SMP .....	160

9.2.3 NUMA . . . . .	161
9.2.4 The MESI protocol . . . . .	165
9.2.5 Software scalability . . . . .	168
9.2.6 Unified Extensible Firmware Interface . . . . .	170
9.3 Memory controller-based chipset . . . . .	171
9.3.1 Intel 5000 chipset family . . . . .	171
9.3.2 Intel 5400 chipset family . . . . .	173
9.3.3 IBM XA-64e fourth-generation chipset . . . . .	175
9.4 PCI bridge-based chipsets . . . . .	177
9.4.1 AMD HyperTransport . . . . .	177
9.4.2 Intel QuickPath Architecture . . . . .	181
<b>Chapter 10. Memory subsystem . . . . .</b>	<b>183</b>
10.1 Introduction to the memory subsystem . . . . .	184
10.2 Memory technology . . . . .	185
10.2.1 DIMMs and DRAMs . . . . .	185
10.2.2 Ranks . . . . .	187
10.2.3 SDRAM . . . . .	188
10.2.4 Registered and unbuffered DIMMs . . . . .	188
10.2.5 Double Data Rate memory . . . . .	189
10.2.6 Fully-buffered DIMMs . . . . .	191
10.2.7 MetaSDRAM . . . . .	195
10.2.8 DIMM nomenclature . . . . .	196
10.2.9 DIMMs layout . . . . .	198
10.2.10 Memory interleaving . . . . .	199
10.3 Specifying memory performance . . . . .	199
10.3.1 Bandwidth . . . . .	199
10.3.2 Latency . . . . .	200
10.3.3 Loaded versus unloaded latency . . . . .	202
10.3.4 STREAM benchmark . . . . .	202
10.4 SMP and NUMA architectures . . . . .	203
10.4.1 SMP architecture . . . . .	203
10.4.2 NUMA architecture . . . . .	204
10.5 The 32-bit 4 GB memory limit . . . . .	207
10.5.1 Physical Address Extension . . . . .	208
10.6 64-bit memory addressing . . . . .	210
10.7 Advanced ECC memory (Chipkill) . . . . .	212
10.8 Memory mirroring . . . . .	213
10.9 Intel Xeon 5500 Series Processors . . . . .	215
10.9.1 HS22 Blade . . . . .	216
10.9.2 System x3550 M2, x3650 M2, and iDataPlex dx360 M2 . . . . .	217
10.9.3 Memory performance . . . . .	219
10.9.4 Memory Interleaving . . . . .	224

10.9.5 Memory ranks . . . . .	227
10.9.6 Memory population across memory channels . . . . .	227
10.9.7 Memory population across processor sockets . . . . .	228
10.9.8 Best practices . . . . .	229
10.10 eX4 architecture servers . . . . .	231
10.11 IBM Xcelerated Memory Technology . . . . .	234
10.12 Memory rules of thumb . . . . .	234
<b>Chapter 11. Disk subsystem . . . . .</b>	<b>237</b>
11.1 Introduction to disk subsystems . . . . .	238
11.2 Disk array controller operation . . . . .	240
11.3 Direct-attached storage . . . . .	241
11.3.1 SAS . . . . .	242
11.3.2 Serial ATA . . . . .	246
11.3.3 NL SAS . . . . .	249
11.3.4 Solid State Drive . . . . .	249
11.4 Remote storage . . . . .	250
11.4.1 Differences between SAN and NAS . . . . .	251
11.4.2 Fibre Channel . . . . .	254
11.4.3 iSCSI . . . . .	255
11.4.4 IBM XIV Storage System . . . . .	257
11.5 RAID summary . . . . .	257
11.5.1 RAID-0 . . . . .	258
11.5.2 RAID-1 . . . . .	259
11.5.3 RAID-1E . . . . .	260
11.5.4 RAID-5 . . . . .	261
11.5.5 RAID-5EE and RAID-5E . . . . .	262
11.5.6 RAID-6 . . . . .	265
11.5.7 RAID-10, RAID-50 and other composite levels . . . . .	266
11.6 Factors that affect disk performance . . . . .	267
11.6.1 RAID strategy . . . . .	268
11.6.2 Number of drives . . . . .	269
11.6.3 Active data set size . . . . .	271
11.6.4 Drive performance . . . . .	273
11.6.5 Logical drive configuration . . . . .	274
11.6.6 Stripe size . . . . .	275
11.6.7 Disk cache write-back versus write-through . . . . .	281
11.6.8 RAID adapter cache size . . . . .	282
11.6.9 Rebuild time . . . . .	284
11.6.10 Device drivers and firmware . . . . .	284
11.6.11 Fibre Channel performance considerations . . . . .	285
11.7 Disk subsystem rules of thumb . . . . .	291

<b>Chapter 12. Network subsystem .....</b>	293
12.1 LAN operations .....	294
12.1.1 LAN and TCP/IP performance .....	296
12.2 Factors affecting network controller performance .....	300
12.2.1 Transfer size .....	300
12.2.2 Number of Ethernet ports .....	304
12.2.3 Processor speed .....	310
12.2.4 Number of processors or processor cores .....	311
12.2.5 Jumbo frame .....	312
12.2.6 10 Gigabit Ethernet adapters .....	313
12.2.7 LAN subsystem performance summary .....	314
12.3 Advanced network features.....	316
12.3.1 TCP offload engine .....	316
12.3.2 I/O Accelerator Technology.....	324
12.3.3 Comparing TOE and I/OAT.....	329
12.3.4 TCP Chimney Offload.....	332
12.3.5 Receive-side scaling.....	334
12.3.6 Operating system considerations .....	339
12.4 Internet SCSI (iSCSI) .....	339
12.4.1 iSCSI Initiators .....	340
12.4.2 iSCSI network infrastructure .....	344
12.5 New trends in networking .....	345
12.5.1 10 Gbps Ethernet .....	345
12.5.2 Converged Enhanced Ethernet.....	346
<b>Part 3. Operating systems .....</b>	349
<b>Chapter 13. Microsoft Windows Server 2003 .....</b>	351
13.1 Introduction to Microsoft Windows Server 2003 .....	352
13.2 Windows Server 2003 - 64-bit (x64) Editions .....	354
13.2.1 32-bit limitations .....	355
13.2.2 64-bit benefits .....	355
13.2.3 The transition to 64-bit computing.....	357
13.2.4 Acknowledgements.....	358
13.3 Windows Server 2003, Release 2 (R2).....	358
13.4 Processor scheduling .....	359
13.5 Virtual memory .....	361
13.5.1 Configuring the pagefile for maximum performance gain .....	363
13.5.2 Creating the pagefile to optimize performance .....	364
13.5.3 Measuring pagefile usage.....	364
13.6 File system cache .....	365
13.6.1 Servers with large amounts of free physical memory .....	370
13.7 Disabling or removing unnecessary services .....	371

13.8	Removing unnecessary protocols and services . . . . .	374
13.9	Optimizing the protocol binding and provider order. . . . .	376
13.10	Optimizing network card settings . . . . .	378
13.11	Process scheduling, priority levels, and affinity. . . . .	382
13.11.1	Process affinity . . . . .	387
13.12	Assigning interrupt affinity . . . . .	388
13.13	The /3GB BOOT.INI parameter (32-bit x86) . . . . .	390
13.14	Using PAE and AWE to access memory above 4 GB (32-bit x86) . . . . .	391
13.14.1	Interaction of the /3GB and /PAE switches . . . . .	393
13.15	TCP/IP registry optimizations . . . . .	394
13.15.1	TCP window size. . . . .	395
13.15.2	Large TCP window scaling and RTT estimation (time stamps) . . . . .	396
13.15.3	TCP connection retransmissions . . . . .	398
13.15.4	TCP data retransmissions. . . . .	399
13.15.5	TCP TIME-WAIT delay . . . . .	399
13.15.6	TCP Control Block (TCB) table . . . . .	400
13.15.7	TCP acknowledgement frequency . . . . .	402
13.15.8	Maximum transmission unit. . . . .	403
13.15.9	Path Maximum Transmission Unit (PMTU) Discovery . . . . .	405
13.16	Memory registry optimizations. . . . .	406
13.16.1	Disable kernel paging . . . . .	407
13.16.2	Optimizing the Paged Pool Size (32-bit x86) . . . . .	407
13.16.3	Increase memory available for I/O locking operations . . . . .	409
13.16.4	Increasing available worker threads . . . . .	410
13.16.5	Prevent the driver verifier from running randomly. . . . .	412
13.17	File system optimizations . . . . .	412
13.17.1	Increase work items and network control blocks. . . . .	412
13.17.2	Disable NTFS last access updates . . . . .	414
13.17.3	Disable short-file-name (8.3) generation. . . . .	415
13.17.4	Use NTFS on all volumes . . . . .	415
13.17.5	Do not use NTFS file compression . . . . .	416
13.17.6	Monitor drive space utilization. . . . .	416
13.17.7	Use disk defragmentation tools regularly . . . . .	417
13.17.8	Review disk controller stripe size and volume allocation units . . . . .	417
13.17.9	Use auditing and encryption judiciously . . . . .	418
13.18	Other performance optimization techniques . . . . .	419
13.18.1	Dedicate server roles . . . . .	419
13.18.2	Run system-intensive operations outside peak times. . . . .	419
13.18.3	Log off the server console. . . . .	419
13.18.4	Remove CPU-intensive screen savers . . . . .	419
13.18.5	Use the latest drivers, firmware, and service packs . . . . .	420
13.18.6	Avoid the use of NET SERVER CONFIG commands. . . . .	420
13.18.7	Monitor system performance appropriately. . . . .	424

<b>Chapter 14. Microsoft Windows Server 2008</b>	425
14.1 Introduction to Microsoft Windows Server 2008	426
14.1.1 Performance tuning for Windows Server 2008	426
14.1.2 What is covered in this chapter	427
14.2 The Windows Server 2008 product family	427
14.2.1 Service Pack 1 and Service Pack 2	429
14.3 New features of Windows Server 2008	431
14.3.1 Server roles	431
14.3.2 Server Core	432
14.3.3 Read-only Domain Controllers	436
14.3.4 Hyper-V	437
14.3.5 Windows System Resource Manager	438
14.4 Networking performance	439
14.4.1 Server Message Block version 2.0	439
14.4.2 TCP/IP stack improvements	440
14.4.3 Tolly Group study	443
14.5 Storage and file system performance	443
14.5.1 Self-healing NTFS	443
14.5.2 Capacity limits	443
14.5.3 Hardware versus software RAID	444
14.5.4 Disk write-caching	444
14.5.5 GPT and MBR disk partitions	446
14.5.6 Partition offset	446
14.5.7 Last-access time stamp	446
14.6 Other performance tuning measures	447
14.6.1 Visual effects	447
14.6.2 System Configuration utility	448
14.6.3 Windows Error Reporting - per process dump files	449
14.7 Windows Server 2008 R2	450
<b>Chapter 15. Linux</b>	453
15.1 Linux kernel 2.6 overview	454
15.2 Working with daemons	455
15.3 Shutting down the GUI	461
15.4 Security Enhanced Linux	464
15.5 Changing kernel parameters	466
15.5.1 Parameter storage locations	467
15.5.2 Using the sysctl commands	468
15.6 Kernel parameters	469
15.7 Tuning the processor subsystem	473
15.7.1 Selecting the right kernel	474
15.7.2 Interrupt handling	475
15.8 Tuning the memory subsystem	476

15.8.1 Configuring bdflush (kernel 2.4 only) . . . . .	476
15.8.2 Configuring kswapd (kernel 2.4 only) . . . . .	478
15.8.3 Setting kernel swap behavior (kernel 2.6 only) . . . . .	478
15.8.4 HugeTLBfs . . . . .	479
15.9 Tuning the file system . . . . .	480
15.9.1 Hardware considerations before installing Linux . . . . .	480
15.9.2 Ext3: the default Red Hat file system . . . . .	482
15.9.3 ReiserFS: the default SUSE Linux file system . . . . .	483
15.9.4 File system tuning in the Linux kernel . . . . .	483
15.9.5 The swap partition . . . . .	490
15.10 Tuning the network subsystem . . . . .	492
15.10.1 Preventing a decrease in performance . . . . .	492
15.10.2 Tuning in TCP and UDP . . . . .	493
15.11 Xen virtualization . . . . .	496
15.11.1 What virtualization enables . . . . .	497
15.11.2 Full virtualization versus paravirtualization . . . . .	498
15.11.3 CPU and memory virtualization . . . . .	500
15.11.4 I/O virtualization . . . . .	500
<b>Chapter 16. VMware ESX 3.5 . . . . .</b>	<b>501</b>
16.1 Introduction to VMware ESX 3.5 . . . . .	502
16.1.1 An approach to VMware ESX performance and tuning . . . . .	502
16.2 Hardware considerations . . . . .	502
16.2.1 VMware ESX network concepts . . . . .	503
16.2.2 VMware ESX Virtualized storage and I/O concepts . . . . .	505
16.2.3 Virtualized CPU concepts . . . . .	510
16.2.4 ESX virtualized memory concepts . . . . .	513
16.2.5 VMware disk partitioning . . . . .	515
16.2.6 Firmware and BIOS settings . . . . .	516
16.3 Tuning activities . . . . .	518
16.3.1 Tuning the VMware kernel . . . . .	518
16.3.2 Tuning the virtual machines . . . . .	520
16.3.3 Tuning the VM memory allocation . . . . .	522
16.3.4 Selecting the right SCSI driver . . . . .	522
16.3.5 Time synchronization . . . . .	523
16.4 VMware ESX 3.5 features and design . . . . .	523
16.4.1 Overview of VMware ESX 3.5 . . . . .	523
16.4.2 Virtual Infrastructure 2.5 with VMware ESX 3.5 Update 4 . . . . .	524
16.4.3 Number of servers and server sizing . . . . .	524
16.4.4 VMotion considerations . . . . .	527
16.4.5 Planning your server farm . . . . .	528
16.4.6 Storage sizing . . . . .	528
16.4.7 Planning for networking . . . . .	529

16.4.8 Network load balancing.....	530
<b>Part 4. Monitoring tools .....</b>	<b>531</b>
<b>Chapter 17. Windows tools .....</b>	<b>533</b>
17.1 Reliability and Performance Monitor console .....	534
17.1.1 Overview of the Performance console window .....	535
17.1.2 Using Performance Monitor .....	541
17.1.3 Using Data Collector Sets.....	546
17.2 Task Manager .....	573
17.2.1 Starting Task Manager .....	573
17.2.2 Processes tab .....	574
17.2.3 Performance tab .....	577
17.3 Network Monitor .....	580
17.3.1 Installing Network Monitor.....	581
17.3.2 Using Network Monitor .....	581
17.4 Other Windows tools .....	588
17.4.1 Microsoft Windows Performance Toolkit.....	588
17.4.2 Microsoft Windows Sysinternals tools.....	589
17.4.3 Others tools.....	592
17.5 Windows Management Instrumentation .....	593
17.6 VTune .....	600
<b>Chapter 18. Linux tools .....</b>	<b>607</b>
18.1 Introduction to Linux tools .....	608
18.2 The uptime command .....	609
18.3 The dmesg command .....	610
18.4 The top command .....	611
18.4.1 Process priority and nice levels.....	612
18.4.2 Zombie processes.....	613
18.5 The iostat command .....	613
18.6 The vmstat command .....	615
18.7 The sar command .....	615
18.8 numastat .....	617
18.9 KDE System Guard .....	617
18.9.1 The KSysguard work space .....	618
18.10 The free command .....	625
18.11 Traffic-vis.....	625
18.12 The pmap command .....	628
18.13 The strace command.....	629
18.14 The ulimit command .....	630
18.15 The mpstat command .....	631
18.16 System x Performance Logger for Linux .....	632
18.16.1 Counters descriptions .....	633

18.16.2 Instructions . . . . .	637
18.16.3 Parameter file . . . . .	638
18.17 The nmon tool . . . . .	642
18.17.1 Using nmon . . . . .	643
18.17.2 The nmon Analyser Excel macro . . . . .	647
<b>Chapter 19. VMware ESX tools . . . . .</b>	<b>649</b>
19.1 Benchmarks . . . . .	650
19.2 The esxtop utility . . . . .	650
19.2.1 Starting esxtop . . . . .	651
19.2.2 Using esxtop . . . . .	655
19.2.3 Exiting esxtop . . . . .	657
19.3 VirtualCenter Console . . . . .	657
<b>Part 5. Working with bottlenecks . . . . .</b>	<b>661</b>
<b>Chapter 20. Spotting a bottleneck . . . . .</b>	<b>663</b>
20.1 Achieving successful performance tuning . . . . .	665
20.2 Step 1: Gathering information . . . . .	667
20.3 Step 2: Monitoring the server's performance . . . . .	669
20.3.1 Where to start . . . . .	673
20.3.2 Disk subsystem . . . . .	675
20.3.3 Memory subsystem . . . . .	681
20.3.4 Processor subsystem . . . . .	683
20.3.5 Network subsystem . . . . .	685
20.4 Step 3: Fixing the bottleneck . . . . .	687
20.5 Conclusion . . . . .	689
<b>Chapter 21. Analyzing bottlenecks for servers running Windows . . . . .</b>	<b>691</b>
21.1 Introduction . . . . .	692
21.2 CPU bottlenecks . . . . .	692
21.2.1 Finding CPU bottlenecks . . . . .	693
21.2.2 Processor subsystem performance tuning options . . . . .	696
21.3 Analyzing memory bottlenecks . . . . .	697
21.3.1 Paged and non-paged RAM . . . . .	698
21.3.2 Virtual memory system . . . . .	699
21.3.3 Performance tuning options . . . . .	700
21.4 Disk bottlenecks . . . . .	703
21.4.1 Analyzing disk bottlenecks . . . . .	704
21.4.2 Performance tuning options . . . . .	705
21.5 Network bottlenecks . . . . .	707
21.5.1 Finding network bottlenecks . . . . .	708
21.5.2 Analyzing network counters . . . . .	709
21.5.3 Solving network bottlenecks . . . . .	712

21.5.4 Monitoring network protocols .....	716
<b>Chapter 22. Analyzing bottlenecks for servers running Linux .....</b>	<b>719</b>
22.1 Identifying bottlenecks .....	720
22.1.1 Gathering information .....	720
22.1.2 Analyzing the server's performance .....	722
22.2 CPU bottlenecks .....	724
22.2.1 Finding bottlenecks with the CPU .....	726
22.2.2 Multi-processing machines .....	726
22.2.3 Performance tuning options for the CPU .....	727
22.3 Memory subsystem bottlenecks .....	728
22.3.1 Finding bottlenecks in the memory subsystem .....	729
22.3.2 Performance tuning options for the memory subsystem .....	732
22.4 Disk bottlenecks .....	733
22.4.1 Finding bottlenecks in the disk subsystem .....	734
22.4.2 Performance tuning options for the disk subsystem .....	738
22.5 Network bottlenecks .....	739
22.5.1 Finding network bottlenecks .....	739
22.5.2 Performance tuning options for the network subsystem .....	741
<b>Chapter 23. Case studies .....</b>	<b>743</b>
23.1 Analyzing systems .....	744
23.2 Case 1: SQL Server database server .....	745
23.2.1 Memory analysis .....	746
23.2.2 Processor analysis .....	747
23.2.3 Network analysis .....	749
23.2.4 Disk analysis on the C: drive .....	751
23.2.5 Disk analysis on the D: drive .....	753
23.2.6 Disk analysis of the V: drive .....	754
23.2.7 SQL Server analysis .....	756
23.2.8 Summary of Case 1 .....	756
23.3 Case 2: File servers hang for several seconds .....	758
23.3.1 Memory analysis .....	759
23.3.2 Processor analysis .....	760
23.3.3 Network analysis .....	761
23.3.4 Disks analysis of the V: drive .....	762
23.3.5 System-level analysis .....	764
23.3.6 Summary of Case 2 .....	765
23.4 Case 3: Database server .....	766
23.4.1 CPU subsystem .....	766
23.4.2 Memory subsystem .....	767
23.4.3 Disk subsystem .....	769
23.4.4 Summary of Case 3 .....	772

<b>Related publications</b> .....	773
IBM Redbooks publications .....	773
Other publications .....	774
Online resources .....	774
How to get IBM Redbooks publications .....	783
Help from IBM .....	783
<b>Abbreviations and acronyms</b> .....	785
<b>Index</b> .....	791



# Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.*

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

## COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

# Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

Active Memory™	IBM Systems Director Active	ServerProven®
AIX®	Energy Manager™	System p®
BladeCenter®	IBM®	System Storage™
Cool Blue™	iDataPlex™	System x®
DB2®	Lotus Notes®	System z®
Domino®	Lotus®	Tivoli®
DPI®	Netfinity®	TotalStorage®
DS4000®	Notes®	X-Architecture®
ESCON®	POWER®	XIV®
eServer™	Redbooks®	xSeries®
	Redbooks (logo) 	zSeries®

The following terms are trademarks of other companies:

Advanced Micro Devices, AMD, AMD Opteron, AMD Virtualization, AMD-V, ATI, Direct Connect, HyperTransport, the AMD Arrow logo, and combinations thereof, are trademarks of Advanced Micro Devices, Inc.

InfiniBand, and the InfiniBand design marks are trademarks and/or service marks of the InfiniBand Trade Association.

Snapshot, and the NetApp logo are trademarks or registered trademarks of NetApp, Inc. in the U.S. and other countries.

Novell, SUSE, the Novell logo, and the N logo are registered trademarks of Novell, Inc. in the United States and other countries.

Oracle, JD Edwards, PeopleSoft, Siebel, and TopLink are registered trademarks of Oracle Corporation and/or its affiliates.

Interchange, Red Hat, and the Shadowman logo are trademarks or registered trademarks of Red Hat, Inc. in the U.S. and other countries.

mySAP, mySAP.com, SAP, and SAP logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

VMotion, VMware, the VMware "boxes" logo and design are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions.

EJB, eXchange, IPX, J2EE, Java, JavaServer, JDBC, JDK, JSP, JVM, Power Management, Solaris, Sun, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Active Directory, Excel, Fluent, Hyper-V, Internet Explorer, Microsoft, MSDN, MS, SQL Server, Visual Basic, Windows NT, Windows Server, Windows Vista, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel Core, Intel NetBurst, Intel Pentium, Intel SpeedStep, Intel Xeon, Intel, Itanium-based, Itanium, Pentium 4, Pentium M, Pentium, VTune, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.



# Foreword

The genesis for this book began in 1997 when, in response to increasing customer demand for performance information, I decided to write a white paper addressing real-world performance issues. The title of that document was *Fundamentals of Server Performance*. This document was so well received by customers, Business Partners and IBM® support personnel that IBM decided to use it as the basis for a new IBM Redbooks® publication addressing a multitude of real-world server performance issues. And in 1998, *Netfinity Performance Tuning with Windows NT 4.0* was published.

Now in its sixth edition, *Tuning IBM Systems x Servers for Performance* is by far the most comprehensive and easy-to-understand performance guide specifically developed for Industry Standard servers. Yes, Industry Standard servers, so if you deploy non-IBM servers you can also benefit greatly from this book. The explanations, tips and techniques can show you the way to better understanding server operation and solving even the most complex performance problems for any Windows®, Linux®, Intel®, or Optero-based server. In addition, this book will enlighten you about some of the special and unique performance optimizations that IBM engineers have introduced into IBM System x® server products.

Finally, I would like to sincerely thank the team that wrote this latest version. Thank you for keeping this vital work current, informative, and enjoyable to read. I am certain that the universe of server administrators and IT workers who benefit from the vast knowledge included in this volume also share my gratitude.

Respectfully,

Gregg McKnight

Vice President, System x Energy Efficiency and Emerging Technologies

IBM Distinguished Engineer

IBM Corporation

Research Triangle Park, North Carolina



# Preface

This IBM Redbooks publication describes what you can do to improve and maximize the performance of your business server applications running on IBM System x hardware and Windows, Linux, or VMware® ESX operating systems. It describes how to improve the performance of the System x hardware and operating system using the available performance monitoring tools.

The keys to improving performance are to understand what configuration options are available to you as well as the monitoring tools that you can use, and to analyze the results that the tools provide so that you can implement suitable changes that positively affect the server.

The book is divided into five parts. Part 1 introduces the concepts of performance tuning and benchmarking, and provides an overview of the way server hardware is used. Part 2 explains the technology implemented in the major subsystems in System x servers and shows what settings can be selected or adjusted to obtain the best performance. Each of the major subsystems covered in Part 2 are closely examined so that you can find specific bottlenecks. Options are also presented which explain what can be done to resolve these bottlenecks. A discussion is provided to enable you to anticipate future bottlenecks, as well.

Part 3 describes the performance aspects of the operating systems Microsoft® Windows Server® 2003, Windows Server 2008, Red Hat® Enterprise Linux, SUSE® Linux Enterprise Server, and VMware ESX.

Part 4 introduces the performance monitoring tools that are available to users of System x servers. We describe the tools specific to Windows, Linux, and VMware ESX. Detailed instructions are provided showing you how to use these tools.

Part 5 shows you how to analyze your system to find performance bottlenecks, and illustrates what to do to eliminate them. We describe an approach you can take to solve a performance bottleneck. We also provide details about what to look for and how to resolve problems. Part 5 also includes a sample analysis of real-life servers, showing how tools can be used to detect bottlenecks and explaining the recommendations for particular systems.

This book is targeted to people who configure Intel and AMD™ processor-based servers running Windows, Linux, or VMware ESX, and who seek to maximize performance. Some knowledge of servers is required. Skills in performance tuning are not assumed.

## The team who wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Raleigh Center.



**David Watts** is a Consulting IT Specialist at the IBM ITSO Center in Raleigh. He manages residencies and produces IBM Redbooks publications on hardware and software topics related to IBM System x and BladeCenter® servers and associated client platforms. He has authored more than 80 books, papers, and technotes. He holds a Bachelor of Engineering degree from the University of Queensland (Australia) and has worked for IBM both in the United States and Australia since 1989. David is an IBM Certified IT Specialist.



**Alexandre Chabrol** is an IT specialist for IBM in Montpellier, France. His areas of expertise include IBM System x and BladeCenter servers, IBM TotalStorage® subsystems, Windows and Linux operating systems and Energy Efficiency. He holds an MS® degree in Computer Science (Master II SITN) from Paris Dauphine University, France, and has worked in the IT industry for five years. Alexandre works on customer performance benchmarks in the Products and Solutions Support Center of Montpellier (PSSC), and is part of the EMEA benchmarks center and the Network Transformation Center teams.



**Phillip Dundas** is a manager with the Wintel Engineering Group at a global financial institution headquartered in Australia. He has almost 15 years of experience in platform engineering, delivery, support and management. He holds a Bachelor of Applied Science (Computing) degree and a Master of Commerce (Information Systems Management) degree. He is a co-author of four other IBM Redbooks publications. Phillip holds industry certifications from Microsoft, VMware, Cisco, Citrix, and Novell®.



**Dustin Fredrickson** is a Senior Engineer and Technical Team Lead for the System x Performance Development team in Research Triangle Park, North Carolina. He holds a BS degree in Computer Engineering from Florida Institute of Technology, and has a background in small business IT management. He has 10 years of experience in the System x Performance Labs, with responsibilities including competitive systems analysis, file server and high speed network analysis, and server systems tuning. Dustin currently leads a team of engineers responsible

for early hardware performance development activities across System x and BladeCenter servers and options.



**Marius Kalmantas** is a Brand Manager for System x and BladeCenter working for IBM CEEMEA, and he is based in Vienna. He has been with IBM for 13 years, mostly working in various product management positions with experience at the country, region, and geography levels. His areas of expertise include the IBM System x and BladeCenter product range, product positioning, lifecycle management, and solution optimization focused on finding the best product match for specific customer needs. He is actively participating in relationships with external vendors to develop business for the System x ecosystem. Marius co-authored the Redbooks publication *IBM eServer xSeries Clustering Planning Guide*, SG24-5845.



**Mario Marroquin** has worked for IBM since 2003 as an Advisory Technical Services Professional, and has received an award for the successful architecting, planning, solution, and implementation of the first VMware farm for an important customer. He has worked as an IT Architect, delivering virtualization solutions for different accounts within Strategic Outsourcing at IBM Global Services, supporting Enterprise class VMware farms. He has also been recognized as a subject matter expert for Intel-Windows and Intel-Linux class servers. Mario is an active member of the Virtualization Center of Competence, assigned to several accounts supporting engagements, transitions, implementations of VMware platforms and as a Security Focal overseeing Security Compliance of VMware solutions for Strategic Outsourcing customers. He is also a member of the VMware Governance Board. He holds a BS degree in Electronic Engineering and Communications as well as a BS degree in Information Technology from the University of Massachusetts. He is working towards his certification as a VMware Certified Design Expert.



**Rajeev Puri** has worked for IBM since 1994 and is a Senior Technical Staff Member. During his career with IBM, he has been recognized as an expert in Enterprise Infrastructure Management. He continues to architect integrated solutions covering a range of heterogeneous technologies from IBM, BMC, HP, Sun™, Microsoft and many others for IBM customers. He leads global strategy in the areas of Operating System Provisioning and Electronic Software Distribution for Strategic Outsourcing customers. He possesses IT experience in all phases of IT delivery, from engagement to transition, implementation, lifecycle management, and client-related critical situations. Rajeev received his MS degree in Computer

Science and BS degree in Electronic Engineering from the University of North Carolina at Charlotte.



**Jose Rodriguez Ruibal** is a Spanish IT Architect and Team Leader for Strategic Alliances for IBM in Montpellier, France, with the STG Industry Systems Next Generation Networks and OEM team. He has over 10 years of experience in IT, and has worked for IBM for more than seven years. His experience includes serving as Benchmark Manager in the IBM PSSC Benchmark Center in Montpellier, and working as an IT Architect for Nokia while living in Finland for three years. Prior to joining IBM, he worked for Red Hat and other consulting firms. He holds an MSC and a BSC in Computer Engineering and Computer Systems from Nebrija University, Madrid. His areas of expertise include Strategic Alliances and long-term IT projects in the Telecom industry, high-level IT architecture and complex solutions design, Linux and all x86 hardware. Jose has co-authored two other Redbooks, one on Linux solutions and another on IBM x86 servers.



**David Zheng** is a Technical Support Engineer in IBM China (ISC Shenzhen). He has two years of experience in the technical support of System x products, and previously worked on the development of embedded microcontrollers. He holds a Bachelor of Engineering degree from Xi'an Jiaotong University, Xi'an, China. His areas of expertise include System x servers and C programming. He has written several documents on System x for customers and for the field support team.

Chapter 10, “Memory subsystem” on page 183 includes the paper *Optimizing the Performance of IBM System x and BladeCenter Servers using Intel Xeon 5500 Series Processors*. The authors of this paper are:

- ▶ Ganesh Balakrishnan
- ▶ Ralph M. Begun
- ▶ Mark Chapman (editor)

Thanks to the authors of the previous editions of this IBM Redbooks publication.

- ▶ Authors of the fifth edition, *Tuning IBM System x Servers for Performance*, published in February 2007, were:

David Watts  
Erwan Auffret  
Phillip Dundas  
Mark Kapoor  
Daniel Koeck  
Charles Stephan

- ▶ Authors of the fourth edition, *Tuning IBM eServer xSeries Servers for Performance*, published in December 2004, were:

David Watts  
Gregg McKnight  
Marcelo Baptista  
Martha Centeno  
Eduardo Ciliendo  
Jean-Jacques Clar  
Phillip Dundas  
Brian Jeffery  
Frank Pahor  
Raymond Phillips  
Luciano Tomé
- ▶ Authors of the third edition, *Tuning IBM eServer xSeries Servers for Performance*, published in July 2002, were:

David Watts  
Gregg McKnight  
Jean-Jacques Clar  
Mauro Gatti  
Nils Heuer  
Karl Hohenauer  
Monty Wright
- ▶ Authors of the second edition, *Tuning Netfinity Servers for Performance—Getting the Most Out of Windows 2000 and Windows NT*, published by Prentice Hall in May 2000, were:

David Watts  
Gregg McKnight  
Peter Mitura  
Chris Neophytou  
Murat Gülder
- ▶ Authors of the first edition, *Netfinity Performance Tuning with Windows NT 4.0*, published in October 1998, were:

David Watts  
Gregg McKnight  
M.S. Krishna  
Leonardo Tupaz

Thanks to the following people for their contributions to this project:

IBM Redbooks

- ▶ Bert Dufrasne
- ▶ Linda Robinson
- ▶ Mary Comianos
- ▶ Tamikia Barrow

IBM Marketing

- ▶ Bob Zuber
- ▶ Andrew Bradley

System x Performance Lab, Raleigh NC:

- ▶ Zeydy Ortiz
- ▶ Darryl Gardner
- ▶ Joe Jakubowski
- ▶ Phil Horwitz
- ▶ Ray Engler
- ▶ Charles Stephan

SSG Performance lab:

- ▶ Xuelian Lin
- ▶ Joaquin Pacheco

IBM Development

- ▶ Donna Casteel Hardee
- ▶ Vinod Kamath
- ▶ Jeff Van Heuklon
- ▶ Nathan Skalsky
- ▶ Jim Marschausen

Other IBM employees

- ▶ Robert Wolford
- ▶ Don Roy
- ▶ David Feishammel
- ▶ Scott Piper
- ▶ Matthew Archibald
- ▶ Steve Pratt

Intel employees

- ▶ Bill Horan

LSI

- ▶ Scott Partington
- ▶ David Worley

## Become a published author

Join us for a two- to six-week residency program! Help write a book dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

[ibm.com/redbooks/residencies.html](http://ibm.com/redbooks/residencies.html)

## Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks in one of the following ways:

- ▶ Use the online **Contact us** review Redbooks form found at:

[ibm.com/redbooks](http://ibm.com/redbooks)

- ▶ Send your comments in an e-mail to:

[redbooks@us.ibm.com](mailto:redbooks@us.ibm.com)

- ▶ Mail your comments to:

IBM Corporation, International Technical Support Organization  
Dept. HYTD Mail Station P099  
2455 South Road  
Poughkeepsie, NY 12601-5400





# Part 1

# Introduction

In this part we describe how this book is organized. We introduce the different types of server applications that clients use System x servers for, and also introduce the concept of benchmarking server to measure performance. A brief description of the major server benchmarks available is also provided.





# Introduction to this book

The server is the heart of the entire network operation. The performance of the server is a critical factor in the efficiency of the overall network, and it affects all users. Although simply replacing the entire server with a newer and faster one might be an alternative, it is often more appropriate to replace or to add only to those components that need it and to leave the other components alone.

Often, poor performance is due to bottlenecks in individual hardware subsystems, an incorrectly configured operating system, or a poorly tuned application. The proper tools can help you to diagnose these bottlenecks and removing them can help improve performance.

For example, adding more memory or using the correct device driver can improve performance significantly. Sometimes, however, the hardware or software might not be the direct cause of the poor performance. Instead, the cause might be the way in which the server is configured. In this case, reconfiguring the server to suit the current needs might also lead to a considerable performance improvement.

This chapter provides an overall introduction to this book and discusses the following topics:

- ▶ 1.1, “Operating an efficient server - four phases” on page 4
- ▶ 1.2, “Performance tuning guidelines” on page 5
- ▶ 1.3, “The System x Performance Lab” on page 5
- ▶ 1.4, “IBM Center for Microsoft Technologies” on page 7

- ▶ 1.5, “Linux Technology Center” on page 7
- ▶ 1.7, “Understanding the organization of this book” on page 10

## 1.1 Operating an efficient server - four phases

To operate an efficient server, follow these four phases:

1. Have an overall understanding of the environment.

There are many components within the network environment that can impact server performance and that can present themselves as potential bottlenecks. It is important to understand the role that the server has to play in this environment and to understand where it is located in the network and in relation to other servers in the environment.

2. Pick the correct server for the job.

After you have established a need for a new server, it is important to have components that allow sufficient bandwidth through those critical subsystems. For example, a file server needs a disk subsystem and a network subsystem that provide sufficient bandwidth for client needs.

3. Configure the hardware appropriately and eliminate initial bottlenecks.

After you have selected the server hardware (and application software), you need to configure the subsystems (for example, stripe size on the RAID array and RAID levels) to maximize performance. To ensure that you are actually improving performance, you need to take initial performance readings (called *baseline* readings) and then compare those with readings taken after you have implemented your changes.

4. Capture and analyze ongoing performance data to ensure that bottlenecks do not occur.

When the server is in production, you need to continue to gather and process performance figures to ensure that your server is still at a near-optimal configuration. You might need to add specific hardware upgrades, such as memory, to achieve this optimal configuration.

As well as looking at the current situation, it is also appropriate that you perform trend analysis so that you can recognize future bottlenecks before they occur. Trend analysis allows you to plan for hardware upgrades before they are actually needed.

Performance monitoring and tuning is an ongoing task. It is not reasonable to simply tune a server once and then assume that it will remain tuned forever. Because the server workload mix changes, so do the location and appearance (and disappearance) of bottlenecks.

## 1.2 Performance tuning guidelines

Table 1-1 lists guidelines to assist you with server management and performance tuning. Although not directly applicable to tuning, following these guidelines should assist you in preventing bottlenecks and identifying bottlenecks.

*Table 1-1 Performance tuning guidelines*

Guideline	Reasoning
Centralize servers where possible	Assists with management and can isolate components such as WAN.
Minimize the number of server types	Enables you to focus on specific subsystems within specific server types.
Standardize configurations	Enables you to focus on specific subsystems within specific server types.
Use industry accepted protocols and standards	Prevents attempts to identify bottlenecks with obscure third-party products and tools.
Use appropriate tools	Fit-for-purpose tools assists with subsystem monitoring and bottleneck analysis.

## 1.3 The System x Performance Lab

IBM puts significant effort into ensuring that its servers have the highest performance level possible. Part of this effort involves the System x Performance Lab, a group in Research Triangle Park, North Carolina, where work is done on System x servers through the development phase and after the servers become publicly available.

During the development phase, the lab creates performance models using subsystem and system functional specifications, chip functional specifications, input from the IBM development engineering departments, as well as performance traces to accomplish the following:

- ▶ Optimize the performance of the individual server subsystems and overall system before the product is manufactured.
- ▶ Make design decision trade-offs.
- ▶ Select the optimum performance among various available chipsets that are intended to be used as part of the subsystem or system.
- ▶ Select optimum settings of the chipset parameters.

This information is used to provide subsystem and system design guidance to the development engineering departments.

As the system development phase nears completion, performance measurements are made with prototype subsystems and systems, as well as with ship-level systems, to do the following:

- ▶ Perform stress testing.
- ▶ Validate product functional specifications.
- ▶ Validate the subsystem and system performance models.
- ▶ Optimize the performance of the subsystem and system.
- ▶ Improve performance of third-party vendors tools, adapters, and software packages to perform well on System x servers.
- ▶ Develop performance white papers for marketing demonstrating the competitiveness of the System x systems.
- ▶ Develop performance tuning guides for customers using specified applications.

Marketing and sales departments and vendors use this information to sell the System x systems, and customers can use this information to select the appropriate system and to tune their systems for their applications.

To provide performance data, the System x Performance Lab uses the following benchmarks:

- ▶ TPC-C
- ▶ TPC-E
- ▶ TPC-H
- ▶ SPECweb2005
- ▶ SPECjbb2005
- ▶ SPEC CPU2006
- ▶ SPECpower\_ssj2008
- ▶ Linpack Benchmark
- ▶ VMmark
- ▶ vConsolidate
- ▶ Oracle® Applications Standard Benchmark
- ▶ SAP® Standard Application Benchmarks
- ▶ LS-DYNA
- ▶ Fluent™ Benchmark

A description of each of those Industry-standard benchmarks can be found in 3.2, “The main industry standard benchmarks” on page 33.

## 1.4 IBM Center for Microsoft Technologies

The IBM Center for Microsoft Technologies is part of the IBM Systems and Technology Group. The facility is located a few minutes from the Microsoft campus in Redmond, Washington, and acts as the technical primary interface to Microsoft. The center is staffed with highly trained IBM technical professionals who are dedicated to working with Windows operating systems and Microsoft Enterprise Server products to exploit the unique features in IBM System x, BladeCenter, and Storage products.

The Center for Microsoft Technologies has four functional groups:

- ▶ Development of device drivers, BIOS, Service Processor, Baseboard Management Controller, and Windows code for System x servers, including development of new technologies for the Windows platforms.
- ▶ Testing of IBM systems in the IBM Microsoft-Certified Hardware Compatibility Lab to meet Microsoft Logo requirements for all systems, devices and clusters. IBM applications being developed for Windows operating systems are also tested for Microsoft standards compliance here.
- ▶ Providing defect support with IBM Level 3 Support in high-severity situations when it is necessary to work directly with Microsoft Development personnel to resolve problems. The CMT also serves as a technical backup for the IBM Help Centers and as a worldwide center of IBM expertise in installation planning.
- ▶ Providing pre-sales support for enterprise large accounts and business partners through the Executive Briefing Center. The Windows Solutions Lab provides facilities for customers and independent software and hardware vendors to complete proofs of concept on their unique Windows workloads on System x servers.

## 1.5 Linux Technology Center

The Linux Technology Center (LTC) is the IBM Linux open source development team. The LTC serves as a center of technical competency for Linux both within IBM and externally. It provides technical guidance to internal software and hardware development teams and fulfills the role of an IBM extension to the open source Linux development community.

The LTC is a worldwide development team within IBM. Its goal is to use world-class programming resources and software technology from IBM to actively accelerate the growth of Linux as an enterprise operating system, while simultaneously helping IBM brands exploit Linux for market growth.

The LTC has programmers involved in numerous Linux projects, including scalability, serviceability, security, network security, networking, file systems, volume management, performance, directory services, standards, documentation, accessibility, test, security certification, systems management, cluster management, virtualization, high availability, storage and I/O, hardware architecture support, power management, reliability, and others required to make Linux an enterprise operating system ready for mission-critical workloads.

The LTC works closely with customers to deliver Linux solutions that meet their needs. In collaboration with these customers the LTC designs, delivers, and implements innovative solutions. Customers leverage the LTC's Linux operating system expertise, our hardware enablement, and our development of new technologies to satisfy complex IT requirements. The LTC's interactive collaboration process between our customers and developers accelerates those customers' return on investment.

The LTC works closely with brand teams across IBM to ensure IBM wins with Linux. Virtually everything that the LTC works on is at the request of one or more IBM brands whose customers need that technology.

The LTC works closely with industry teams (such as the The Linux Foundation workgroups) to accelerate the expansion of Linux into usage scenarios such as telecommunications, the data center, the desktop, the embedded space, and anywhere else that the world's most modular and flexible operating system can go.

Members of the LTC work directly in the open source community using standard open source development methodology. They work as peers within the shared vision of the Linux community leadership, and participate in setting Linux design and development direction.

More information on the LTC can be find at the following URL:

<http://www.ibm.com/linux/ltc/>

## 1.6 IBM Client Benchmark Centers

IBM worldwide benchmark centers perform application benchmarks customized to specific installations as well as generalized application benchmarks.

Those centers can configure IBM System x and BladeCenter technology with storage solutions to your specification so that you can stress, tune and test your application or database, measure performance and determine workload capacity.

IBM Client Benchmark Centers can help clients to:

- ▶ Resolve performance and scalability questions
- ▶ Run their applications on requested configurations
- ▶ Enable them to work with IBM experts who have world-class, certified skills

The centers provide access to equipment, facilities, consulting services, and project-management skills. They can help you to make the right decisions about the platforms.

Depending on the request, the centers can provide:

- ▶ Business Intelligence Center of Competency (BICoC) benchmarks
  - The BICoC provides technical sales support to IBM sales channels to close business for Business Intelligence (BI) opportunities. The primary service offered by the BICoC is large-scale solution validation tests.
- ▶ Commercial benchmarks
  - Commercial benchmarks are done in support of a business opportunity for IBM for all sectors except high performance computing.
- ▶ High-performance computing (HPC) benchmarks
  - HPC benchmarks refer to presales activities for the high performance technical computing market. These include customer and ISV opportunities.

IBM benchmark centers located around the world can provide proofs of concept and benchmarks.

- ▶ In the Americas
  - Poughkeepsie, New York
  - Dallas, Texas
  - Kirkland, Washington
- ▶ In EMEA
  - Montpellier, France
- ▶ In Asia Pacific
  - Beijing, China
  - Tokyo, Japan

In addition to those benchmarks centers, IBM has centers dedicated to specific software products such as the IBM SAP International Competence Center in Walldorf and the Oracle-IBM Joint Solutions Center in Montpellier.

- ▶ The IBM SAP International Competence Center (ISICC) has a variety of offerings to assist your company, including Executive Customer Briefings that are individually tailored demonstrations to help you profit from the knowledge of IBM and SAP experts.

- ▶ The Oracle-IBM Joint Solutions Center (JSC) resulted from the decision by Oracle and IBM to work closely together and offer their customers the optimal solutions to their needs.

There are also specific centers dedicated to an industry, such as the Network Transformation Center (NTC) for the telecommunication sector.

- ▶ The NTC is a world-class Next Generation Network (NGN) enablement laboratory specifically designed to offer the latest IBM NGN systems hardware and middleware offerings to assist IBM telecommunication partners and customers in enabling and testing telecommunications applications.

You can request a benchmark by completing the online form found at the following address:

[http://www.ibm.com/systems/services/benchmarkcenter/contact\\_bc\\_x.html](http://www.ibm.com/systems/services/benchmarkcenter/contact_bc_x.html)

## 1.7 Understanding the organization of this book

This book is organized as follows:

1. Understanding hardware subsystems
2. Understanding operating system performance
3. Working with performance monitoring tools
4. Detecting and removing performance bottlenecks

After the introductory chapters, the chapters are divided into parts to make it easier to find the information that you need:

- ▶ Part 2, “Server subsystems” on page 43 covers each of the major subsystems and their contributions to the overall performance of the server:
  - Energy Efficiency
  - CPU
  - Virtualization hardware assists
  - Chipsets
  - PCI bus architecture
  - Memory
  - Disk subsystem
  - Network adapter
  - Operating system
- ▶ Part 3, “Operating systems” on page 349 describes performance aspects of the operating systems that are covered in this book:
  - Windows Server 2003 and Windows Server 2008
  - Red Hat Enterprise Linux and SUSE Linux Enterprise Server
  - VMware ESX

- ▶ Part 4, “Monitoring tools” on page 531 covers the tools that are available to users of System x servers that run these operating systems. With these tools, you can identify and remove existing performance bottlenecks and avoid future ones.
- ▶ Part 5, “Working with bottlenecks” on page 661 describes how to use these tools. This part covers:
  - How to identify a performance problem and solve it quickly
  - A detailed explanation of the analysis of performance bottlenecks
  - Case studies that show real-life examples of performance analysis





# Understanding server types

To optimize server performance, it is important to first understand the intended use of the system and the performance constraints that you might encounter. When you have identified the critical subsystems, you can then focus your attention on these components when resolving performance issues.

This chapter describes the common server types and the subsystems that are most likely to be the source of a performance bottleneck. When defining the bottlenecks for server types, we list them in order of impact.

This chapter discusses the following topics:

- ▶ 2.1, “Server scalability” on page 14
- ▶ 2.2, “Authentication services” on page 15
- ▶ 2.3, “File servers” on page 17
- ▶ 2.4, “Print servers” on page 18
- ▶ 2.5, “Database servers” on page 18
- ▶ 2.6, “E-mail servers” on page 20
- ▶ 2.7, “Web servers” on page 21
- ▶ 2.8, “Groupware servers” on page 22
- ▶ 2.9, “Multimedia server” on page 23
- ▶ 2.10, “Communication server” on page 24
- ▶ 2.11, “Terminal server” on page 25
- ▶ 2.12, “Infrastructure servers” on page 26
- ▶ 2.13, “Virtualization servers” on page 28
- ▶ 2.14, “High Performance Computing” on page 28

## 2.1 Server scalability

Scalability is about increasing the capability of the server to allow the services that are provided to meet increased demands. Server scalability is generally achieved by adopting either *scale-out* or *scale-up* strategies, which are defined as follows:

- ▶ Scale-up is where server type subcomponents are increased in capacity to meet the increase in demand.

For example, in a server where the memory subsystem is a potential bottleneck, and the CPU is overloaded, the amount of memory in the server can be increased to accommodate demand and more CPUs can also be added. For enterprise customers with high performance demand on a single server, the IBM System x3950 M2 is a prime example of a server that is suited for scale-up.

- ▶ Scale-out is where multiple separate servers function together to perform a given task, often seen from the outside as a single system.

Scale-out is generally achieved through a form of load-balancing and task distribution. For example, Microsoft Network Load Balancing offers scalability by balancing incoming client requests across clusters of individual servers. Tools such as NLB require you to install and configure additional components on the operating system. Thus, analyzing bottlenecks will become more complex. Other scale-out approaches can be Grid or Cloud computing, or HPC task distribution engines.

For enterprise customers, the IBM BladeCenter family is a prime example of a server complex that is suited for scale-out.

There are server types which support applications that are capable of supporting their own scale-out options. Citrix and Weblogic are two examples. There are also hardware solutions that work at the network layer called *network load balancers*. These are different from Microsoft Network Load Balancing because a hardware device in the network controls incoming traffic and redirects network traffic to a number of individually grouped servers that provide a single service. For example, Radware Web Server Director is, in essence, a network device that will load balance incoming requests to a number of Web servers.

Determining which approach to adopt influences how performance tuning is done. For example, although it is important to be able to identify potential bottlenecks, it is also important to understand how to resolve them. Attempting to add additional capacity to subcomponents that are at their maximum threshold will not resolve a bottleneck, so the answer may be to scale out. Likewise, undertaking analysis of a server that is located inside a network load balanced cluster will be more complex than troubleshooting an individual server.

Table 2-1 lists server types and some of the scaling options that are available for medium to large customers.

*Table 2-1 Server types and typical scalability options*

Server type	Scale option	Scale method
File server	Scale-out	Windows load balance
Print servers	Scale-up	Hardware
Terminal servers	Scale-out	Native
Web servers	Scale-out	Network load balance
E-mail servers	Scale-up	Hardware
Database servers	Scale-up	Hardware
Computation servers	Scale-out	Native

## 2.2 Authentication services

Domain controllers provide authentication services and are central to the management of network resources including users, devices, and computers. They maintain and apply rules to provide secure and reliable working environments. Domain controllers communicate with each other continually to ensure that all rules are maintained consistently throughout the environment. For example, these servers communicate to ensure that user accounts, security settings, access control lists, and policies are synchronized.

Domain controllers perform the following functions:

- ▶ User authentication
- ▶ Resource access validation
- ▶ Security control

Common implementations are LDAP and Microsoft Active Directory®.

### 2.2.1 Windows Server 2008 Active Directory domain controllers

Active Directory stores domain-wide directory data such as user authentication data, system security policies, and network objects (such as user, computer and printer names) in its replica LDAP database. It also provides the required tools to manage user and domain interactions, such as the logon process and validation, resource allocation, and directory searches.

Windows Server 2008 Active Directory domain controllers are the latest generation of the Windows domain controllers, providing improved server management, access management, security and performance. Some of the new performance enhancing features are:

- ▶ More efficient WAN replication, most notably through improvements in the SMB 2.0 protocol.
- ▶ Read-only domain controller (RODC) functionality, thus reducing the surface area for attack on a domain controller, reducing risk for branch offices, and minimizing replication requirements.
- ▶ Active Directory now runs as a service that can be switched on and off more easily rather than as an integrated part of the underlying operating system.

You can configure Windows Server 2008 servers as domain controllers by using the Active Directory wizard. You need to have an existing Active Directory in place to create additional domains.

The Knowledge Consistency Checker (KCC) constructs and maintains the replication topology for Active Directory automatically, but administrators can also configure which domain members are synchronized with other domain members. Note that Windows 2008 RODCs can only replicate with Windows 2008 writable domain controllers; they cannot replicate directly with Windows 2003 or other Windows 2008 RODCs

Because all Active Directory objects use fully qualified domain names (FQDNs), DNS is an extremely important service. When Active Directory is installed on a Windows Server 2008 domain controller, the installation wizard requires DNS to be installed locally. This is most typically done using Active Directory integrated DNS zones.

To allow Active Directory to service requests quickly with Windows Server 2008 domain controllers servers requires adequate and reliable network bandwidth to perform replication, synchronization, logon validation, and other services.

On an Active Directory domain controller, there are two kind of activities:

- ▶ Server-to-server activities

These activities include the replication of the Active Directory partitions to the other Domain Controllers in your domain structure. There are five main partitions and a various number of application partitions; these are split into domain and forest-wide replication domains.

- ▶ Client-to-server activities

These activities include logon validation processes, security access validation and LDAP queries (for example, from clients directly, or via global catalog lookups from, for example, an Exchange server).

To adequately provide a highly responsive service without delays, the following hardware subsystems are should be analyzed on domain controllers to check for performance bottlenecks:

- ▶ Memory
- ▶ Processor
- ▶ Network

As a general rule, the larger the number of objects in the Active Directory database and the more complex the distribution of domain controllers in an enterprise, the harder the domain controller will need to work to service server-to-server and client-to-server requests. Accordingly, domain controllers should be appropriately specified to hardware and matched to the task.

Of particular note are domain controllers functioning as global catalogs in an enterprise - especially those using Microsoft Exchange. Domain controllers can be significantly impacted by busy Exchange servers; in some instances, domain controllers are isolated into separate Active Directory sites with Exchange servers to provide dedicated throughout and service for global catalog queries.

## 2.3 File servers

The role of the file server is to store, retrieve, and update data that is dependent on client requests. Therefore, the critical areas that impact performance are the speed of the data transfer and the networking subsystems. The amount of memory that is available to resources such as network buffers and disk I/O caching also influence performance greatly. Processor speed or quantity typically has little impact on file server performance.

In larger environments, also consider where the file servers are located within the networking environment. It is advisable to locate them on a high-speed backbone as close to the core switches as possible.

The subsystems that have the most impact on file server performance are:

- ▶ Network
- ▶ Memory
- ▶ Disk

**Tip:** A common misconception is that CPU capacity is important. CPU is rarely a source of performance bottlenecks for file servers.

The network subsystem, particularly the network interface card or the bandwidth of the LAN itself, might create a bottleneck due to heavy workload or latency.

Insufficient memory can limit the ability to cache files and thus cause more disk activity, which results in performance degradation.

When a client requests a file, the server must initially locate, then read and forward the requested data back to the client. The reverse of this applies when the client is updating a file. Therefore, the disk subsystem is potentially a bottleneck.

## 2.4 Print servers

Print servers remove the requirement to install printers on individual clients, and they are capable of supporting a large number of printer types and print queues. They manage client print requests by spooling the print job to disk. The printer device itself can influence performance, because having to support slower printers with limited memory capacity takes longer to produce output while using resources on the print server. Therefore, the critical areas that impact performance are the speed of the data transfer and memory configuration. By default, the spool directory is located on the same disk as the operating system files. However, it is better to redirect the directory to a physical drive other than the operating system disk, so that the pooling I/O operations will not influence the rest of the system performance.

The subsystems that have the most impact on print server performance are:

- ▶ Memory
- ▶ Disk
- ▶ Processor

Implementing printer pools and virtual printer configurations might help to reduce printing workload.

## 2.5 Database servers

The database server's primary function is to store, search, retrieve, and update data from disk. Examples of Database engines include IBM DB2®, Microsoft SQL Server®, and Oracle. Due to the high number of random I/O requests that database servers are required to do and the computation-intensive activities that occur, the potential areas that have the most impact on performance are:

- ▶ Memory
- ▶ Disk
- ▶ Processor
- ▶ Network

The subsystems that have the most impact on database server performance are:

- ▶ **Memory subsystem**

Buffer caches are among the most important components in the server, and both memory quantity and memory configuration are critical factors. If the server does not have sufficient memory then paging occurs, which results in excessive disk I/O, which in turn generates latencies. Memory is required for both the operating system and the database engine. You need to consider this when sizing database servers.

Refer to the following sections to determine how to better use the memory in your systems:

- Windows: 21.3, “Analyzing memory bottlenecks” on page 697
- Linux: 22.3, “Memory subsystem bottlenecks” on page 728

- ▶ **Disk subsystem**

Even with sufficient memory, most database servers will perform large amounts of disk I/O to bring data records into memory and flush modified data to disk. The disk subsystem needs to be well designed to ensure that it is not a potential bottleneck.

Therefore, it is important to configure a sufficient number of disk drives to match the CPU processing power that is used. With most database applications, more drives equals greater performance.

It is also important to keep your log files on disks that are different from your database.

Even when using SAN devices for storage, you must pay particular attention to Fibre channel network and SAN configuration to ensure that the storage environment does not place constraints on the server.

- ▶ **CPU subsystem**

Processing power is another important factor for database servers because database queries and update operations require intensive CPU time. The database replication process also requires a considerable number of CPU cycles.

Database servers are multi-threaded applications. Therefore, SMP-capable systems provide improved performance scaling to 16-way and beyond. L2 cache size is also important due to the high hit ratio, which is the proportion of memory requests that fill from the much faster cache instead of from memory. For example, the SQL server’s L2 cache hit ratio approaches 90%.

- ▶ **Network subsystem**

The networking subsystem tends to be the least important component on an application or database server because the amount of data returned to the

client is a small subset of the total database. The network can be important, however, if the application and the database are on separate servers.

A balanced system is especially important. For example, if you are adding additional CPUs, then consider upgrading other subsystems (such as increasing memory and ensuring that disk resources are adequate).

In database servers, the design of an application is critical (for example, database design and index design).

## 2.6 E-mail servers

E-mail servers act as repositories and routers of electronic mail, and they handle the transfer of e-mail to its destination. Because e-mail servers must communicate regularly to perform directory replication, mail synchronization, and interface to third-party servers, they generate network traffic. Because they also have to store and manage mail, the disk subsystem is becoming increasingly more important.

The important subsystems for e-mail servers are:

- ▶ Memory
- ▶ CPU
- ▶ Disk
- ▶ Network

E-mail servers use memory to support database buffers and e-mail server services. Ensuring that memory is sized appropriately and that the disk subsystems are effective is very important because these will impact server performance. For example, if memory size is sufficient, the server is capable of caching more data, which results in improved performance.

E-mail servers use log files to transfer modified data to an information store. These log files are written sequentially, which means that new transactions are appended to the end of the transaction files. Log files and database files have different usage patterns: log files perform better with separate physical disks, and database files perform better with striped disk arrays due to the random workload. Using several drives instead of a single drive can significantly increase e-mail throughput. Read-ahead disk-caching disk subsystems can also offer performance benefits.

User mailboxes can be stored on the server or on each user's local hard drive, or on both. In each case, you need high network performance because clients still retrieve their mail over the network. The larger the size of the e-mails, then the more bandwidth that is required. Also, server-to-server replication traffic can be a

significant load in the network and using multiple LAN adapters can help to improve network performance.

When an e-mail server receives a message, it determines where the appropriate server is to handle the e-mail. If the address is local, it is stored in the database of the e-mail server. If the address is not local, the e-mail is forwarded to the most appropriate server for processing. If the address is a distribution list, the server checks the addresses in the list and routes the message accordingly. These processes require CPU cycles, and sufficient memory must be allocated to ensure that these processes run efficiently.

If your server supports directory replication and connectors between sites, your server will experience high distribution list usage, and the CPU will be a more important factor in e-mail server performance.

Adequate network bandwidth between e-mail servers and their clients is essential. However, contrary to popular belief, this is not the most impacted subsystem. If IPsec is to be used to encrypt network traffic, using a specialized network card to offload the encryption process will reduce CPU utilization.

## 2.7 Web servers

Today, Web servers are responsible for hosting Web pages and running server-intensive Web applications. If Web site content is static, the following subsystems might be sources of bottlenecks:

- ▶ Network
- ▶ Memory
- ▶ CPU

If the Web server is computation-intensive (as with dynamically created pages), then the following subsystems might be sources of bottlenecks:

- ▶ Memory
- ▶ Network
- ▶ CPU
- ▶ Disk

The performance of Web servers depends on the site content. There are sites which use dynamic content that connect to databases for transactions and queries, and this requires additional CPU cycles. It is important that in this type of server there is adequate RAM for caching and for managing the processing of dynamic pages for a Web server. Also, additional RAM is required for the Web server service. The operating system automatically adjusts the size of cache, depending on requirements.

Because of a high hit ratio and the transfer of large amounts of dynamic data, the network can be another potential bottleneck.

### 2.7.1 Web 2.0 servers

The use of “Web 2.0” applications has exploded over recent years, and these applications are typically Web-based applications focused on a more interactive user experience and written using asynchronous technologies such as AJAX. Many of these applications are designed for millions of simultaneous users.

These applications are run on servers in large data centers with massively distributed networks. The hardware is often thousands or tens of thousands of compute nodes that are located near low-cost power sources. Redundancy is handled at the software level rather than at the hardware level, meaning that the compute nodes are often low-cost and easily-replaceable systems.

Compared to traditional Web servers, these Web 2.0 servers are built on a shared and pooled environment, supported by parallel programming developments in which an individual server’s performance is less important than the overall service or application availability. This is changing the paradigm of traditional Web computing, because the focus is now more on distributed applications, with low-cost hardware to take care of the service that the applications provide.

The following are key elements in Web 2.0 servers that are relevant when defining a Web 2.0 application or service:

- ▶ Price/performance per watt
- ▶ Fast, large scale-out deployment
- ▶ Compute density
- ▶ Customization
- ▶ Targeted workloads

Therefore, the performance of Web 2.0 servers should instead be evaluated for all the servers running an application, instead of as individual servers. The elements that are important to examine are:

- ▶ Network
- ▶ Memory

## 2.8 Groupware servers

Groupware servers (such as Lotus® Notes® and Microsoft Exchange, among others) are designed to allow user communities to share information. This

enhances the teamwork concept for company users, and is usually implemented in a client/server model.

Important subsystems include:

- ▶ Memory
- ▶ CPU
- ▶ Disk I/O

Groupware servers generally support public folder access, scheduling, calendaring, collaboration applications, and workflow applications. These systems require significant CPU power similar to e-mail servers. Routing and real-time collaboration require additional CPU cycles.

Memory is used for caching just as it is for e-mail servers, and groupware servers use a special memory cache design to increase the data access rate. Therefore, the server should be configured with enough memory to eliminate or reduce paging to disk.

Groupware servers are transactional-based client/server database applications. Similarly as with database servers, the disk subsystem is an important factor in performance.

When designing groupware systems, pay particular attention to the amount of server-to-server traffic anticipated. Slow LAN/WAN links must also be considered.

## 2.9 Multimedia server

Multimedia servers provide the tools and support to prepare and publish streaming multimedia presentations utilizing your intranet or the Internet. They require high-bandwidth networking and high-speed disk I/O because of the large data transfers.

If you are streaming audio, the most probable sources of bottlenecks are:

- ▶ Network
- ▶ Memory
- ▶ Disk

If you are streaming video, most important subsystems are:

- ▶ Network
- ▶ Disk I/O
- ▶ Memory

Disk is more important than memory for a video server due to the volume of data being transmitting and the large amount of data being read.

If the data is stored on the disk, disk speed is also an important factor in performance. If compressing/decompressing the streaming data is required, then CPU speed and amount of memory are important factors as well.

## 2.10 Communication server

Communication servers provide remote connection to your LAN, and the most popular communication server is the Windows 2003 remote access services (RAS) server.

A communication server's bottlenecks are usually related to the speed of the communication lines and cards themselves. Typically, these applications do not put a stress on the processor, disk, or memory subsystems, and the speed of the communication line will dictate the performance of the communication server. A high-speed T1 line, for example, causes less performance degradation than a 56 Kbps line.

The subsystems that are the most probable sources of bottlenecks are:

- ▶ Communication lines

These are the physical connections between the client and server. As previously mentioned, the most critical performance factor is the speed of these communication lines. Select faster communication lines to achieve better performance.

- ▶ Digital communications

Select digital lines if possible because they are more efficient at transferring data, and they transmit with fewer errors. Digital communications also benefit because fault detection and correction software and hardware might not have to be implemented.

- ▶ Port configuration

Port is the input/output source for the communication devices. For example, if you have modem devices, configure your port speed, flow control, and buffering to increase data flow performance.

Other features, such as multilink and pure digital communications, will help to improve performance. Correctly configuring the operating system's port status and using the correct device driver are other important tasks in maintaining high performance.

## 2.11 Terminal server

Windows Server Terminal Services enables a variety of desktops to access Windows applications through terminal emulation. In essence, the application is hosted and executed on the terminal server and only screen updates are forwarded to the client. It is important to first understand the factors in terminal server performance:

- ▶ Your application
  - Application memory requirements
  - Shareable application memory
  - Application screen refresh rate
  - Applications typing requirements
- ▶ Your users
  - Typing speed
  - Leave the applications open
  - Logon time
  - Logged on all day long or not
  - Whether or not most logins occur at a specific time of day
- ▶ Your network
  - Users' typing speed
  - Whether or not applications are graphic-intensive
  - Client workstations' display resolutions
  - Application network bandwidth requirements

The following subsystems are the most probable sources of bottlenecks:

- ▶ Memory
- ▶ CPU
- ▶ Network

As the terminal servers execute applications and send the results to the client workstation, all the processing load is on the server. Terminal servers require powerful CPUs and sufficient memory. Because these servers can support multiple concurrent clients, the network is another important subsystem.

Terminal servers do not benefit from large L2 cache sizes primarily because they have a very large working set. The *working set* is the number of instructions and data that are frequently accessed by the CPU. This working set is too large, and addresses generated by terminal server applications are more random across this large address space than most server applications. As a result, most terminal server configurations will obtain minimal benefits from large L2 processor caches.

Generally, double the number of users requires double the performance of the CPU and double the amount of memory. CPU and memory requirements increase linearly, so you should use SMP-capable servers.

The following factors also affect performance:

- ▶ Hard-disk throughput (for higher performance, use RAID devices)
- ▶ High-bandwidth network adapters
- ▶ Intelligent dial-up communications adapter (to reduce interrupt overhead and increase throughput)

## 2.12 Infrastructure servers

Infrastructure servers is the name given to DNS, DHCP, WINS, and other services that provide connectivity.

### 2.12.1 DNS server

Domain Name System (DNS) is a protocol for naming computers and network services. It is used to locate computers and services through user-friendly names. When a client uses a DNS name, DNS services can resolve the name to other information associated with that name, such as an IP address.

The number of requests that the DNS server is required to respond to will be determined by the size of the environment that it is supporting and the number of DNS servers that will be located within that environment. Consider these factors when sizing the server type.

Important subsystems include:

- ▶ Network
- ▶ Memory

The network subsystem, particularly the network interface card or the bandwidth of the LAN itself, can create a bottleneck due to heavy workload or latency. Insufficient memory might limit the ability to cache files and thus cause more disk and CPU activity, which results in performance degradation.

Because of the nature of DNS serving, the processor subsystem is the least likely to cause a bottleneck.

## 2.12.2 DHCP server

Dynamic Host Configuration Protocol (DHCP) is a protocol for using a server to manage and administer IP addresses and other related configuration items in the network. When a device starts, it might issue a request to obtain a IP address. The DHCP server responds and provides that device with a valid IP address that is valid for a predefined period of time. This protocol removes the requirement to assign individual IP addresses for each device.

The number of requests that the DHCP server is required to respond to and the size of IP address scope will be critical in determining the server size. Having multiple DHCP and splitting the scope might reduce overheads on individual servers.

Important subsystems include:

- ▶ Network
- ▶ Disk
- ▶ Memory

The network subsystem, particularly the network interface card or the bandwidth of the LAN itself, can create a bottleneck due to heavy workload or latency. High disk I/O requests require an appropriately designed disk subsystem. Insufficient memory might limit the ability to cache files and thus cause more disk and CPU activity, which results in performance degradation.

Because of the nature of DHCP serving, the processor subsystem is the least likely to cause a bottleneck.

## 2.12.3 WINS server

Windows Internet Name Service (WINS) is a system that resolves NetBIOS names to IP addresses. For example, when a client uses a NetBIOS reference, the WINS server can resolve the NetBIOS name to other information associated with that name, such as an IP address.

The number of requests that the WINS server is required to respond to will be determined by the size of the environment that it is supporting and the number of WINS servers that are located within that environment. Consider these factors when sizing the server type.

Important subsystems include:

- ▶ Network
- ▶ Disk
- ▶ Memory

The network subsystem, particularly the network interface card or the bandwidth of the LAN itself, might create a bottleneck due to heavy workload or latency. High disk I/O requests require an appropriately designed disk subsystem. Insufficient memory might limit the ability to cache files and thus cause more disk and CPU activity, which results in performance degradation.

Because of the nature of WINS serving, the processor subsystem is the least likely to cause a bottleneck.

## 2.13 Virtualization servers

Virtualization servers provide the ability to run multiple simultaneous servers (or *virtual machines*) on a single hardware platform. This is achieved by installing a product such as VMware ESX Server, which provides the capability to divide the hardware subsystems into smaller partitions that then appear as multiple individual servers.

These partitions can then be configured with an operating system and function as a traditional server type. For example, a server with two CPUs and 2 GB of RAM with 300 GB of disk can be partitioned into four servers, each with ½ CPU and 500 MB of RAM with 75 GB of disk. These servers could then be configured as different server types. For example, they can be configured as a Active Directory server, WINS server, DNS server, and DHCP server.

The benefit is that servers that have spare capacity can be reconfigured as multiple different servers, thereby reducing the number of physical servers that need to be supported in the environment.

The individual virtual server type will still have the same potential bottlenecks and performance issues as the physical server type, and there is still the added overhead of having to support the virtualization layer.

Potential bottlenecks on the virtual operating system are:

- ▶ Memory
- ▶ CPU
- ▶ Network

## 2.14 High Performance Computing

Computation servers provide floating-point and memory resources to compute-intensive applications such as those found in high-performance computing (HPC). These servers are often clustered together using extremely

high-speed interconnects, such as Myrinet or InfiniBand®, to provide significantly greater computational performance than would otherwise be available to a single server alone. Typical applications are characterized by their dependence on 32-bit or 64-bit floating-point operations.

A computation server's bottlenecks are generally related to the speed with which floating-point computations can be performed. Numerous factors can affect that, including the native vector or scalar performance of the processor and the size of processor cache. Vector operations are arithmetic operations that repeatedly perform the same operation on streams of related data. Scalar operations work on each element separately.

The speed at which data can be retrieved from memory is often one of the most important performance bottlenecks. Many HPC applications stride through large arrays in a uniform manner that brings data into the processor, uses it for a few operations, then writes a result back to memory. This characteristic is unfriendly to caches and pushes the performance bottleneck out to main memory.

Computation servers need high network latency or throughput performance. To accomplish this, they are connected through high speed interconnects such as Myrinet, InfiniBand, or Quad. Depending on the specific HPC workload types, every technology has its own advantages.

Potential bottlenecks on the virtual operating system are:

- ▶ Memory
- ▶ Network
- ▶ CPU





# Performance benchmarks for servers

A benchmark is a standardized problem or test used to measure system performance. The purpose is typically to make some sort of comparison between two offerings, whether they are software, hardware, or both.

Many types of benchmarks are undertaken, and they vary dramatically in purpose, size, and scope. A very simple benchmark may consist of a single program executed on a workstation that tests a specific component such as the CPU. On a larger scale, a system-wide benchmark may simulate a complete computing environment, designed to test the complex interaction of multiple servers, applications, and users. In each case the ultimate goal is to quantify system performance to take measurements.

This chapter introduces performance benchmarks for servers and discusses the following topics:

- ▶ 3.1, “IBM and benchmarks” on page 32
- ▶ 3.2, “The main industry standard benchmarks” on page 33
- ▶ 3.3, “Understanding IBM System x benchmarks” on page 41

## 3.1 IBM and benchmarks

IBM worldwide benchmark centers perform application benchmarks that are customized to specific installations, as well as generalized application benchmarks. In this chapter we illustrate the different kinds of benchmarks, and you will learn why IBM run benchmarks on System x servers.

### 3.1.1 The different kinds of benchmarks

There are three overall types of benchmarks:

- ▶ Industry standard benchmarks

Industry standard benchmarks consist of well-known benchmarks that have been developed, maintained, and regulated by independent organizations.

These benchmarks are designed to represent client workloads (for example, e-commerce or OLTP). They allow readers to make comparisons between systems when the workload matches their intended use. The configurations are based on “off-the-shelf” hardware and applications. We introduce industry benchmarks in 3.2, “The main industry standard benchmarks” on page 33.

- ▶ Client workload benchmarks

These benchmarks involve benchmarking with a client’s actual workload. This type yields the most relevant information, but is difficult to perform. Some IBM centers are dedicated to this kind of benchmark.

IBM Client Benchmark Centers provide customer-demanded benchmark capability for IBM server and storage technology, including proof of concept, scaling and performance services, as well as assistance in executing ISV application benchmarks.

We introduce those centers in 1.6, “IBM Client Benchmark Centers” on page 8.

- ▶ **Unregulated benchmarks**

Also very common are unregulated benchmarks that are application- or component-specific. You can sometimes purchase or download these benchmark suites to test how specific components perform.

However, exercise caution when using the results of these tools. By far the majority of testing suites available test workstation performance and are not relevant for testing *server* performance. Many benchmarking tools are designed to test workstations and to stress system components by executing a single task or a series of tasks. Servers and workstations are designed for very different purposes. A workstation performs a single task as quickly as possible. In contrast, servers are generally optimized to service multiple users

performing many tasks simultaneously. A server often performs poorly in such tests because this is not what they are optimized to do.

### 3.1.2 Why IBM runs benchmarks on System x servers

Benchmarking is one of the few ways in the computer industry to objectively compare offerings between different computer vendors. Leading technology from IBM is compared against other server vendors in a standardized way of producing and publishing results.

Benchmarks are published records, and IBM invests much time, effort, and money to achieve the best possible results. All of the #1 results that System x solutions win are proof points that IBM technology made a huge difference in pushing the industry forward.

IBM takes benchmarking very seriously. This is primarily because our clients also take it seriously and consider the results an important differentiator. Additionally, IBM undertakes benchmarks as an integral part of the development process of System x servers. The System x performance lab is actively involved in the process of bringing a new server to market to ensure that each system is properly tuned for its intended clients' use.

In the same way, client workload benchmarks allow the customer to compare server performance between different computer vendors on its own applications. IBM has dedicated centers which can help its customers to tune their applications on IBM System x servers. The purpose of those benchmarks is to obtain the best performance for a customer application in its production environment.

## 3.2 The main industry standard benchmarks

IBM uses a wide range of industry standard benchmarks to evaluate server performance.

Many models of the IBM System x family have maintained a leadership position for benchmark results for several years. These benchmarks help clients to position System x servers in the marketplace, but they also offer other benefits to clients including driving the industry forward as a whole by improving the performance of applications, drivers, operating systems, and firmware.

There is a common misconception that industry benchmark results are irrelevant because they do not reflect the reality of client configurations and the performance and transaction throughput that is actually possible in the “real world.” However, such benchmarks are useful and relevant to clients because the

results help them understand how one solution offering performs relative to another.

### 3.2.1 Types of information that benchmarks can provide

The IBM System x performance lab has produced many industry-leading benchmark results on System x hardware (for more information about this lab, refer to 1.3, “The System x Performance Lab” on page 5). These results demonstrate the capabilities of the server and validate the system designs. They are also a valuable source of information to use in decision-making processes.

It is important to understand, however, that the actual performance figures produced from these benchmarks are unlikely to be obtained in a real production environment.

The reasons for this disparity include the following:

- ▶ System utilization often runs at levels that are not reasonable in a production environment. In most commercial environments, a system operating at a utilization level of very close to 100% cannot handle peaks in demand or allow for future growth.
- ▶ The hardware configurations may not be representative of a real production server. For the TPC-C benchmark, hundreds of disks feed as much data as possible to memory and the CPUs.
- ▶ In addition to a large number of drives, each drive is configured with a “stroke” (a portion of the disk’s surface actually containing data) of less than 15%. The objective is to minimize latency by reducing the disk drive head movement as much as possible. This means that only a small portion of each disk is used. This is unrealistic in a production environment.
- ▶ Many benchmarks do not require RAID arrays to protect data. For these benchmarks, data is striped across the disks in an unprotected manner (that is, RAID-0) to maximize performance and minimize cost.
- ▶ Servers respond differently to different workloads. A server that is industry-leading in one benchmark may perform poorly in another. When using published benchmarks as a guide to selecting a server, the results of benchmarks that most closely resemble the intended workload should be examined.
- ▶ A vendor benchmark team typically has more time and greater access to highly skilled engineers (including the hardware and software designers) than a typical client benchmark team.

**Tip:** Use benchmark results as a guide to the potential or *theoretical* performance and throughput of a server under a specific workload type, and not as an indicator of *actual* performance.

### 3.2.2 System benchmarks

This section introduces the major industry system benchmarks.

#### TPC-C

The TPC Benchmark C (TPC-C), approved in July 1992, is an online transaction processing (OLTP) benchmark. TPC-C simulates a complete computing environment in which a population of users executes transactions against a database.

The benchmark is centered around the principal activities (transactions) of an order-entry environment. These transactions include entering and delivering orders, recording payments, checking the status of orders, and monitoring the level of stock at the warehouses. Although the benchmark portrays the activity of a wholesale supplier, TPC-C is not limited to the activity of any particular business segment. Instead, it represents any industry that must manage, sell, or distribute a product or service. However, it should be stressed that it is not the intent of TPC-C to specify how to best implement an order-entry system.

The performance metric reported by TPC-C measures the number of orders that can be fully processed per minute. It is expressed in transactions per minute (tpmC). Two other metrics are associated: \$/tpmC and the date of availability for the priced system. For more information about this benchmark, consult the following site:

[http://www\(tpc.org/tpcc/](http://www(tpc.org/tpcc/)

#### TPC-E

Since 1992, system performance increased significantly, and some modern applications are not based on the old transaction model. A new model that handles the current situation was introduced by the TPC committee, namely TPC Benchmark E (TPC-E).

TPC-E is not a new version of the TPC-C. Instead, because TPC-E uses current architecture and reduces the total benchmark cost (because less hardware is needed), it will replace the TPC-C Benchmark.

TPC-E is an OLTP workload. The TPC-E Benchmark simulates the OLTP workload of a brokerage firm. The focus of the benchmark is the central

database that executes transactions related to the firm's customer accounts. Although the underlying business model of TPC-E is a brokerage firm, the database schema, data population, transactions, and implementation rules have been designed to be broadly representative of modern OLTP systems.

The TPC-E metrics are tpsE (transactions per second E) and \$/tpsE. The tpsE metric is the number of trade-result transactions that the server can sustain over a period of time. The price/performance metric, \$/tpsE, is the total system cost for hardware, software, and maintenance, divided by the performance. For more information about this benchmark, consult the following site:

<http://www.tpc.org/tpce/>

## **TPC-H**

TPC Benchmark H (TPC-H) models a decision support system. Decision support systems are used to analyze OLTP information for use in business decisions. This benchmark illustrates decision support systems that examine large volumes of data, execute queries with a high degree of complexity, and give answers to critical business questions.

The performance metric reported by TPC-H is called the TPC-H Composite Query-per-Hour Performance Metric (QphH@Size). This metric reflects multiple aspects of the capability of the system to process queries. These aspects include the query processing power when queries are submitted by a single user, and the query throughput when queries are submitted by multiple concurrent users. Because of its impact on performance, the size of the database against which the queries are executed is also included in the TPC-H metric. For more information about this benchmark, consult the following site:

<http://www.tpc.org/tpch>

## **SPECweb2005**

SPECweb2005 emulates users sending browser requests over broadband Internet connections to a Web server. It provides three new workloads: a banking site (HTTPS), an e-commerce site (HTTP/HTTPS mix), and a support site (HTTP). Dynamic content is implemented in PHP and JSP™. For more information about this benchmark, consult the following site:

<http://www.spec.org/web2005/>

## **SPECjbb2005**

SPECjbb2005 (Java™ Server Benchmark) is the SPEC benchmark for evaluating the performance of server-side Java. Like its predecessor, SPECjbb2000, SPECjbb2005 evaluates the performance of server-side Java by emulating a three-tier client/server system (with emphasis on the middle tier).

The benchmark exercises the implementations of the Java Virtual Machine (JVM™), the Just-In-Time (JIT) compiler, garbage collection, threads and some aspects of the operating system. It also measures the performance of CPUs, caches, memory hierarchy and the scalability of shared memory processors (SMPs). SPECjbb2005 provides a new enhanced workload, implemented in a more object-oriented manner, to reflect how real world applications are designed. It introduces new features such as XML processing and BigDecimal computations to make the benchmark a more realistic reflection of today's applications. For more information about this benchmark, consult the following site:

<http://www.spec.org/jbb2005/>

## **SPECjAppServer2004**

SPECjAppServer2004, the Java Application Server, is a multi-tier benchmark for measuring the performance of Java 2 Enterprise Edition (J2EE™) technology-based application servers. SPECjAppServer2004 is an end-to-end application which exercises all major J2EE technologies implemented by compliant application servers as follows:

- ▶ The Web container, including servlets and JSPs
- ▶ The EJB™ container
- ▶ EJB2.0 Container Managed Persistence
- ▶ JMS and Message Driven Beans
- ▶ Transaction management
- ▶ Database connectivity

Moreover, SPECjAppServer2004 also heavily exercises all parts of the underlying infrastructure that make up the application environment, including hardware, JVM software, database software, JDBC™ drivers, and the system network. For more information about this benchmark, consult the following site:

<http://www.spec.org/jAppServer2004/>

## **SPEC CPU2006**

CPU2006 is the SPEC next-generation, industry-standardized, CPU-intensive benchmark suite which is designed to stress a system's processor, memory subsystem and compiler. SPEC designed CPU2006 to provide a comparative measure of compute-intensive performance across the widest practical range of hardware using workloads developed from real user applications. SPEC CPU2006 contains two benchmark suites: CINT2006 for measuring and comparing compute-intensive integer performance, and CFP2006 for measuring and comparing compute-intensive floating point performance. For more information about this benchmark, consult the following site:

<http://www.spec.org/cpu2006/>

## **SPECpower\_ssj2008**

SPECpower\_ssj2008 is the first industry-standard SPEC benchmark that evaluates the power and performance characteristics of volume server class computers. With SPECpower\_ssj2008, SPEC defines server power measurement standards in the same way as with performance. You can find more details at 5.3, “Rack-level solutions” on page 76.

For more information about this benchmark, consult the following site:

[http://www.spec.org/power\\_ssj2008/](http://www.spec.org/power_ssj2008/)

## **Linpack Benchmark**

The Linpack Benchmark is a measure of a computer’s floating-point rate of execution. It is determined by running a computer program that solves a dense system of linear equations. Over time, the characteristics of the benchmark have changed somewhat. In fact, there are three benchmarks included in the Linpack Benchmark report.

The Linpack Benchmark is derived from the Linpack software project. It was originally intended to help users of the package approximate how long it would take to solve certain matrix problems.

The Linpack Benchmark was chosen by the TOP500 committee because it is widely used and performance numbers are available for almost all relevant systems. For more information about this benchmark, consult the following site:

<http://www.top500.org/project/linpack>

## **VMmark**

VMmark is a tool that hardware vendors, virtualization software vendors, and other organizations use to measure the performance and scalability of applications running in virtualized environments. This virtualization benchmark software features a novel, tile-based scheme for measuring application performance. It provides a consistent methodology that captures both the overall scalability and individual application performance.

VMware developed VMmark as a standard methodology for comparing virtualized systems. The benchmark system in VMmark is comprised of a series of “sub-tests” that are derived from commonly used load-generation tools, as well as from benchmarks developed by the Standard Performance Evaluation Corporation (SPEC). For more information about this benchmark, consult the following site:

<http://www.vmware.com/products/vmmark/>

## **vConsolidate**

vConsolidate, launched in April 2007 at the Intel Developer Forum (IDF), is designed to simulate real world server performance in a typical environment, and to enable clients to compare the performance of multi-processor platforms in a virtualized environment. For more information about this benchmark, consult the following site:

<http://www.intel.com/pressroom/archive/releases/20070417gloc1.htm>

### **3.2.3 Product-specific benchmarks**

This section introduces some of the product-specific benchmarks currently available.

#### **Oracle Applications Standard Benchmark**

The Oracle Applications Standard Benchmark is focused on ERP applications. It represents a mixed workload intended to model the most common transactions operating on the most widely used enterprise application modules. Definitions of transactions that compose the benchmark load were obtained through collaboration with functional consultants, and are representative of typical customer workloads, with batch transactions representing 25% of the total workload. For more information about this benchmark, consult the following site:

[http://www.oracle.com/apps\\_benchmark](http://www.oracle.com/apps_benchmark)

#### **BaanERP**

Baan Enterprise Resource Planning (ERP) is a suite of client/server business solutions that integrates a company's business transactions into a single software solution. Baan ERP software provides applications which customers use to manage financial, accounting, sales and distribution, materials management, production planning, quality management, plant maintenance and human resource functions.

The Baan Benchmark Methodology is used to measure the performance of different computer system configurations running the standard BaanERP benchmark suite in two-tier client/server mode, which determines the exact number of Baan Reference Users (BRUs) that can be supported on a specific vendor's computer system.

#### **SAP Standard Application Benchmarks**

SAP Standard Application Benchmarks test and prove the scalability of mySAP™.com® solutions. The benchmark results provide basic sizing recommendations for customers by testing new hardware, system software components, and Relational Database Management Systems (RDBMS). They

also allow for comparison of different system configurations. The original SAP Standard Application Benchmarks have been available since R/3 Release 1.1H (April 1993), and are now available for many SAP components.

The benchmarking procedure is standardized and well defined. It is monitored by the SAP Benchmark Council, which is comprised of representatives of SAP and technology partners involved in benchmarking. Originally introduced to strengthen quality assurance, the SAP Standard Application Benchmarks can also be used to test and verify scalability, concurrency and multi-user behavior of system software components, RDBMS, and business applications. All performance data relevant to system, user, and business applications are monitored during a benchmark run, and can be used to compare platforms and as basic input for sizing recommendations. For more information about this benchmark, consult the following site:

<http://www.sap.com/solutions/benchmark>

### **LS-DYNA**

LS-DYNA, developed by the Livermore Software Technology Corporation, is a general purpose transient dynamic finite element program capable of simulating complex real world problems. It is optimized for shared and distributed memory Unix, Linux, and Microsoft Windows platforms. LS-DYNA is being used by automobile, aerospace, manufacturing and bioengineering companies. For more information about this benchmark, consult the following site:

[http://www.topcrunch.org/benchmark\\_results\\_search.sfe](http://www.topcrunch.org/benchmark_results_search.sfe)

### **Fluent Benchmark**

The Fluent Benchmarks can be used to compare performance of different hardware platforms running the FLUENT flow solver. The broad physical modeling capabilities of FLUENT have been applied to industrial applications ranging from air flow over an aircraft wing to combustion in a furnace, from bubble columns to glass production, from blood flow to semiconductor manufacturing, from clean room design to wastewater treatment plants. The ability of the software to model in-cylinder engines, aero-acoustics, turbo-machinery, and multiphase systems has served to broaden its reach. For more information about this benchmark, consult the following site:

<http://www.fluent.com/software/fluent/f15bench>

## **3.2.4 Industry standard benchmark results on IBM System x**

The latest industry standard benchmark results on the System x platform with Intel and AMD processors can be found at the following site:

<http://www.ibm.com/systems/x/resources/benchmarks/>

The latest industry-standard benchmarks results on the BladeCenter platform with Intel and AMD processors can be find at the following site:

<http://www.ibm.com/systems/bladecenter/resources/benchmarks/>

### 3.3 Understanding IBM System x benchmarks

Many models of the IBM System x family have maintained a leadership position for benchmark results for several years. These benchmarks help clients to position System x servers in the marketplace and understand how one solution offering performs relative to another.

For a deeper understanding of IBM System x benchmarks, refer to the IBM Redpaper publication, *Understanding IBM eServer xSeries Benchmarks*, REDP-3957, which is available at the following site:

<http://www.redbooks.ibm.com/abstracts/redp3957.html>

This paper is intended for clients, IBM Business Partners, and IBM employees.



# Server subsystems

In this part, we explain the technology that is implemented in the major subsystems in System x servers and show what settings you can make or adjust to obtain the best performance. We provide rules of thumb to guide you regarding what to expect from any changes that you consider. We examine closely each of the major subsystems so that you can find specific bottlenecks, and we present options explaining what you can do to resolve these bottlenecks. We also discuss how to anticipate future bottlenecks.





# Introduction to hardware technology

Servers are made up of a number of subsystems, and each subsystem plays an important role in how the server performs. Depending on the use of the server, some of these subsystems are more important and more critical to performance than others.

This chapter defines the server subsystems.

## 4.1 Server subsystems

The subsystems in a server are:

- ▶ Processor and cache

The processor is the heart of the server, and it is involved in most of the transactions that occur in the server. Although the CPU is an important subsystem, many people mistakenly believe that the CPU is often the source of a performance bottleneck so buying a server with the fastest CPU is best.

In reality, however, in most server installations the CPU is actually overpowered, and the other subsystems are underpowered. Only specific applications are truly CPU-intensive, taking advantage of the full power of today's multi-core and 64-bit processors.

The classic example of a server that does not need much CPU power is the file server (which is, coincidentally, the most common use of a server). Most file request traffic uses direct memory access (DMA) techniques to bypass the CPU and rely in the network, memory, and disk subsystems for throughput capacity.

There are a number of processors available from Intel and AMD that are used in System x servers today. It is important to understand their differences and their strengths.

Cache, while strictly part of the memory subsystem, is physically packaged with the processor these days. The CPU and cache are coupled together tightly and run at full or half the speed of the processor. In this book, we have grouped the cache and processor together.

- ▶ PCI bus

The PCI bus is the “pipe” along which all data traverses into and out of the server. All System x servers use the PCI bus (PCI-X and PCI Express) for their critical adapter resources, SCSI and disk for example. High-end servers now have multiple PCI buses and many more PCI slots than they used to.

Advances in the PCI bus include the PCI Express (PCI-E) 1X to 16X technologies, which provide greater throughput and connectivity options.

Connecting to the CPU and cache is the PCI chipset. This set of components governs the connections between the PCI bus and the processor and memory subsystems. The PCI chipset is carefully matched and tuned to the processors and memory to ensure the maximum performance of the system.

- ▶ Memory

Memory is critical to a server's performance. Without enough memory installed, the system will perform poorly because the operating system will swap data to disk when it needs to make room for other data in memory.

A feature in the Enterprise X-Architecture® System x servers is memory mirroring for increased fault tolerance. The feature, part of IBM Active Memory™, is roughly equivalent to RAID-1 in disk arrays, in that memory is divided in two ports and one port is mirrored to the other. All mirroring activities are handled by the hardware without any additional support by the operating system.

New memory technologies include the DDR3 and MetaSDRAM, providing higher capacity, bandwidth, and improved flexibility.

► Disk

Perhaps the most configurable subsystem from an administrator's perspective, the disk subsystem is often critical to a server's performance. In the pyramid of online storage devices (cache, memory, and disk), disk drives are by far the slowest and also the biggest, mainly because they are mechanical components. For many server applications, most of the data that is accessed will be stored on disk, so a fast disk subsystem is very important.

To maximize capacity, RAID is commonly employed in server configurations. However, the configuration of the RAID arrays can make a significant difference in the performance characteristics.

First, the choice of RAID level for the defined logical drives will affect performance as well as capacity and fault tolerance, which you would normally expect. There are many RAID levels available using IBM ServeRAID and IBM Fibre Channel adapters, and each has its place in specific server configurations. Equally important for performance reasons is the number of hard disks you configure in each array: the more disks, the better the throughput. An understanding of how RAID handles I/O requests is critical to maximizing performance.

Serial technologies are on all System x servers to improve price-performance and scalability. These include Serial ATA (SATA) and Serial-attached SCSI (SAS). Near-Line SAS HDD are now available to take advantage of both SAS and SATA technology.

Solid State Drive (SSD) technology is beginning to emerge in System x servers as a new category of hard disk drive. SSD consumes less power, and can offer faster data access and higher reliability due to its inherent memory-based characteristics.

► Network

The network adapter card is the server's interface to the rest of the world. If the amount of data through this portal is significant, then an underpowered network subsystem will have a serious impact on server performance.

Beyond the server, the design of your network is equally important. The use of switches to segment the network or the use of such technologies as ATM should be considered.

1 Gbps network adapters are now standard in servers, and 10 Gbps network are readily available to provide the necessary bandwidth for high-throughput applications. Moreover, new technologies such as IOAT and TCP Chimney Offload are introduced to help improve performance.

- ▶ **Video**

The video subsystem in a server is relatively insignificant. The only use of it is when an administrator is working on the server's console. Production users will never make use of the video, so emphasis is rarely placed on this subsystem. We do not cover video in this book for this reason.

- ▶ **Operating system**

We consider the operating system to be a subsystem that can be the source of bottlenecks similar to other hardware subsystems. Windows, Linux, and ESX Server have settings that you can change to improve performance of the server.

This part of the book describes each of the major subsystems in detail. It explains how they are important and what you can do to tune them to your requirements. The subsystems that are critical to performance depend on what you are using the server for. The bottlenecks that occur can be resolved by gathering and analyzing performance data; however, this task is not a one-time job. Bottlenecks can vary depending on the workload coming into the server, and can change from day to day and from week to week.



# Energy efficiency

Energy efficiency (defined here as using less energy to provide the same level of service) in data centers is a critical priority for IT managers. As energy and power costs become a significant portion of IT cost, understanding and investing in energy management has never been more important.

For example, looking at today's power and cooling costs, the cost for power and cooling a server for three years is 1.5 times the cost of purchasing the server hardware. Projections for the year 2012 raise the factor from three times to 22 times, depending on the assumptions used<sup>1</sup>.

Some companies have outgrown their current data centers because of exceeding available power, cooling resources, or space and have been forced to relocate or to build a new data center. If the new data center is properly planned, however, this expense may actually result in a financial return.

Thinking globally, estimated data center power demands are growing at unsustainable rates:

- ▶ 1.2% of global electrical output is used by servers and cooling.<sup>2</sup>
- ▶ USD\$7.2 billion was spent on data centers worldwide in 2005.<sup>2</sup>

<sup>1</sup> Brill, Kenneth G., *Data Center Energy Efficiency and Productivity*, The Uptime Institute, Inc., 2007, <http://www.uptimeinstitute.org/whitepapers>

<sup>2</sup> Koomey, Jonathan, *Estimating Total Power Consumption by Servers in the U.S and the World*, 2007, <http://enterprise.amd.com/Downloads/svrpwrusecompletesfinal.pdf>

- ▶ Only about half the power entering the data center is used by the IT equipment.<sup>3</sup>

IBM recognizes the importance of going green in the data center for both environmental and financial reasons, and has initiated *Project Big Green*. This is the broadest initiative ever at IBM with the intention to reallocate \$1 billion a year to achieve the following goals:

- ▶ Guarantee the research and development funding for IT energy efficiency technology
- ▶ Create a worldwide IBM “Green Team” of energy efficiency specialists
- ▶ Plan, build, or prepare its facilities to be green data centers based on IBM best practices and innovative technologies in power and cooling
- ▶ Use virtualization as the technology accelerator for our green data centers to drive up utilization and drive down annual power cost per square foot

IBM also intends to make its technologies, practices, and experience available to assist clients in making their data centers more efficient.

There are several metrics for energy efficiency of servers. One of the most commonly used measures is “performance per watt.” It usually refers to the maximum performance that the system can achieve for every watt consumed.

However, there are other metrics that incorporate space (SWaP metric from Sun), average performance per watt (like the score from SPECpower), and so on.

This chapter discusses the following topics:

- ▶ 5.1, “Importance of finding an energy efficiency balance” on page 51
- ▶ 5.2, “Server-level solutions” on page 53
- ▶ 5.3, “Rack-level solutions” on page 76
- ▶ 5.4, “Data center-level solutions” on page 84
- ▶ 5.5, “Power and performance benchmarks” on page 88

---

<sup>3</sup> U.S. Environmental Protection Agency, *Report to Congress on Server and Data Center Energy Efficiency* [http://www.energystar.gov/index.cfm?c=prod\\_development.server\\_efficiency#epa](http://www.energystar.gov/index.cfm?c=prod_development.server_efficiency#epa)

## 5.1 Importance of finding an energy efficiency balance

Being energy efficient in a data center is a complex challenge. On the one hand, organizations continue to deploy more servers and storage devices to keep pace with their exploding computing needs. Every day the demand grows for more speed and greater capacity. To address this demand, IBM has developed a wide range of products providing the latest powerful technologies. Generation after generation, those products continue to offer improved computation and response time, and to increase the number of concurrent users on a given platform.

On the other hand, faster processors, larger memory configurations, more disk drives, and more I/O adapters add up to more power needed and therefore increased electrical cost. There is also the additional cost to remove all the heat produced by the hardware. Heat translates into a shorter lifespan for hardware. The more excess heat is present, the more damage can be incurred by hardware. Thus you either pay to keep the hardware cool, or you pay to replace heat-damaged components. And rising utility rates make the solutions ever more costly.

At this point, the cost of electricity to run and cool computer systems exceeds the cost of the initial purchase. The goal of IT managers is also to reduce the demand for electricity and keep servers and the data center cool. Doing so reduces costs and increases hardware reliability.

This analysis illustrates that a balance must be found between performance and power consumption in a data center. To understand how to optimize energy efficiency, you also need to understand where and how energy is used, and learn how to optimize it.

Figure 5-1 on page 52 shows how energy is used in several components of a typical non-optimized data center. Each component is divided into two portions:

- ▶ IT equipment (servers, storage, and network) uses about 45% of the energy.
- ▶ The infrastructure that supports this equipment—such as chillers, humidifiers, computer room air conditioners (CRAC), power distribution units (PDU), uninterruptible power supplies (UPS), lights, and power distribution—uses the other 55% of the energy.

Companies must consider the energy consumption of the components at the IT equipment level. For example, in a typical server, the processor uses only 30% of the energy and the remainder of the system uses 70%. Therefore, efficient hardware design is very important, as well.

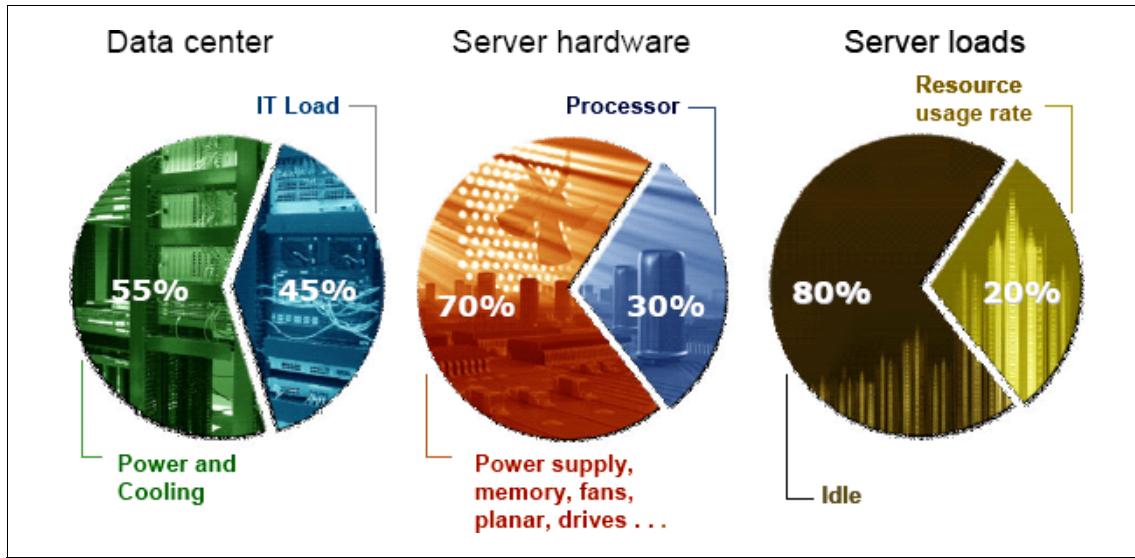


Figure 5-1 How energy is used in a typical data center

Finally, companies should consider the use of IT resources in data centers. Commonly, servers are underutilized, yet they consume the same amount of energy as if they were running at 100%. A typical server utilization rate is 20%. Underutilized systems can be a major issue because a significant amount of energy is expended on non-business purposes, thus wasting a major investment. Virtualization and consolidation help utilize the entire capacity of your IT equipment.

In this book, we discuss energy savings on IT resources. To learn more about optimizing non-IT equipment, read the IBM Redpaper publication *The Green Data Center: Steps for the Journey*, REDP-4413, which is available at the following site:

<http://www.redbooks.ibm.com/abstracts/redp4413.html>

Energy optimizations on IT resources can be undertaken at three different levels, as addressed in the following sections:

- ▶ 5.2, “Server-level solutions” on page 53
- ▶ 5.3, “Rack-level solutions” on page 76
- ▶ 5.4, “Data center-level solutions” on page 84

## 5.2 Server-level solutions

Optimizing energy efficiency can be done at the server level. You have to find the power/performance balance appropriate to your needs.

Figure 5-2 illustrates the power budgeted for a typical server. Notice that nearly a third of the total power, 31%, is allocated to the CPUs. Next, 26% is allocated for the memory and 21% for the redundant fans. The 6 hard disk drives represent only 7% of the total power allocated behind the 4 PCI slots of 9%. Finally, the motherboard is allocated approximately 6%.

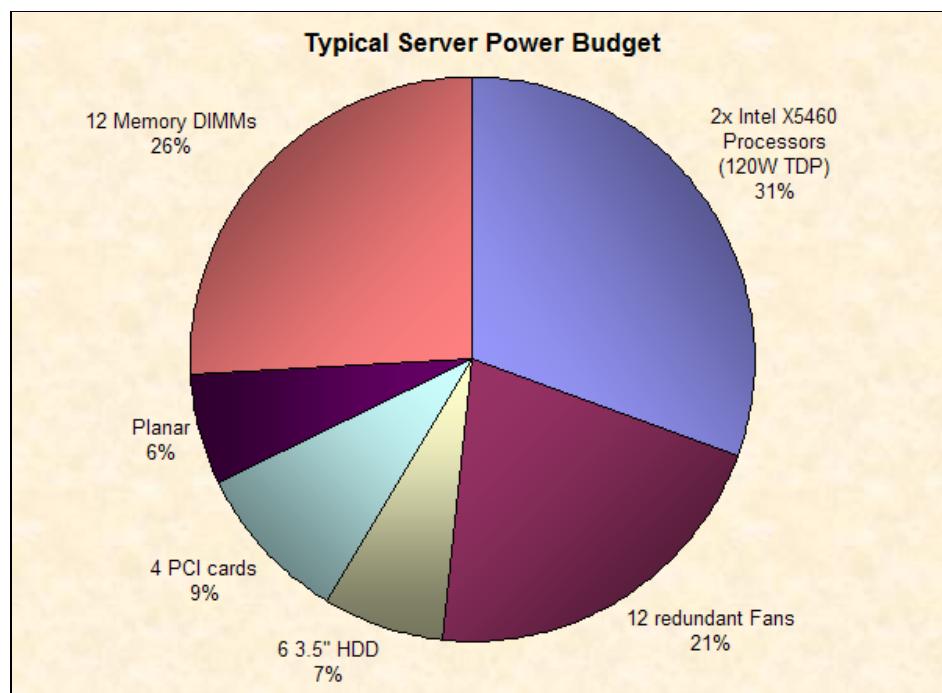


Figure 5-2 Power budgeted for a typical server

To better understand how energy efficiency is important, Table 5-1 on page 54 illustrates how a small power savings at the low level in a server, namely 1 extra watt saved on a CPU, cascades to a large power savings at the higher data center level.

In Table 5-1 on page 54, you can see that 1 extra watt consumed on a CPU:

- = 1.25W at the input of an 80% efficient VRM
- = 1.47W at the input to an 80% efficient power supply
- = 1.52W at the input of an 97% power distribution infrastructure

- =1.58W at the input to a 96% efficient 480VAC-to-208VAC transformer
- =1.75W at the input to a 92% efficient UPS
- =1.79W at the input to a 98% efficient feeder transformer

Table 5-1 Data Center Power Distribution Systems and Efficiency

Data Center Power Distribution Systems and Efficiency							
	Feeder Transformer	UPS System	Raised Floor Transformer	Distribution (all combined)	Server Bulk PS	VRM	Load
Requires this much power at the facility level							
One extra watt of chip power							
Std raised floor	11.2KV(AC to 480VAC	480VAC to 400VDC to 480VAC	480VAC to 208VAC		208VAC to 400VDC to 12VDC	12VDC to 1VDC	1VDC to heat
Std efficiencies	98%	92%	96.2%	96.7%	85%	80%	
Input watts for each 1 W load	<b>1.75</b>	<b>1.72</b>	<b>1.58</b>	<b>1.52</b>	<b>1.47</b>	<b>1.25</b>	<b>1</b>
Not shown: Electrical power needed for extra cooling capacity required to cool 1 watt = 3.41 BTUs							

Small power improvements at the component level have a large cascading effect at the data center level. Therefore, optimizing within the server is very important. The following sections describe some of the major components.

### 5.2.1 Processors

Processors are the main focus of an energy efficiency effort at the server level because they have a major influence on performance. Processors are also the component allocated the highest portion of the power budget.

Intel and AMD provide features on their latest CPUs to reduce their power consumption. In this chapter, we describe those technologies and how to use them.

#### Current technologies

CPU power management has advanced every year, starting with mobile optimized CPUs.

Today, both Intel and AMD offer low-voltage versions of some of their processors. They run at the same clock rates as their higher-voltage cousins, but consume less power.

Figure 5-3 shows the power allocation by component of a server with 3.16 GHz, quad core processors, and the same system with 2.5 GHz quad core processors. This figure illustrates how the choice of CPU can affect the overall power allocation of a server. This CPU change reduces by 58% the power allocated by the processors. It results in an 18% global power savings for the system.

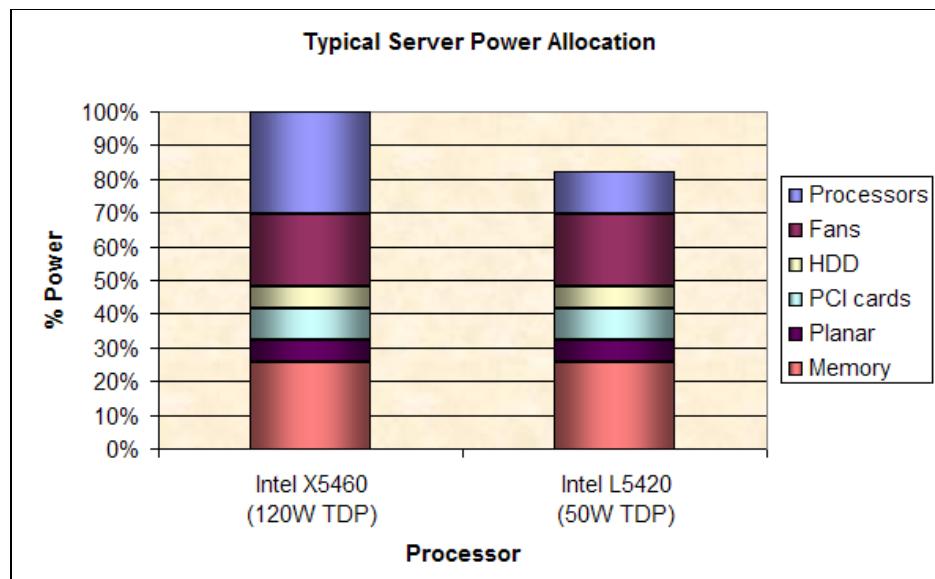


Figure 5-3 Power budgeted for a typical server with different CPU frequencies

Figure 5-3 also illustrates that the challenge of energy efficiency cannot be solved only by the choice of processor, but the system as a whole.

The processors have maintained the same thermal design power (TDP) while increasing the number of processor cores per socket. TDP represents the maximum amount of power the cooling system in a computer is required to dissipate. This increases energy efficiency by increasing performance while keeping power essentially the same.

Dual-core processors improve performance at the same power consumption as a single-core version of the same processor. IBM servers that use low-voltage

dual-core processors include many blade servers and System x servers. All of the following servers offer models that consume fewer than 75W per processor:

- ▶ Some System x and blade models employ 35W (17.5W per core) or 40W (20W per core) dual-core Xeon processors. They use 63% and 58% less power, respectively, than standard 95W Xeon processors.
- ▶ Some System x and blade models use 68W dual-core AMD Opteron™ processors (34W per core), instead of the 95W variety (a 28% reduction).

System x systems are also using power-efficient quad-core processors.

Quad-core processors improve the performance of dual-core processors, at a lower per-core power usage. Many IBM System x and blade servers currently offer quad-core processors.

- ▶ The 80W quad-core Xeon processors offered in many System x and blade servers consume only 20W of power per core. Not only does one 80W quad-core processor potentially offer up to 90% more performance than a 95W dual-core Xeon processor (at the same clock rate), but it also uses 16% less power in doing so.
- ▶ Some System x and blade servers use 50W quad-core Xeon processors. These consume only 12.5W per core, for an even better performance-per-watt ratio.

Here is an example of two Intel Xeon® processors (5400-series) that have the same front-side bus speed, but have a different TDP (lower is better):

- ▶ Intel E5420 2.50 GHz with a front-side bus at 1333 MHz with TDP= 80 W
- ▶ Intel L5420 2.50 GHz with a front-side bus at 1333 MHz with TDP= 50 W

Increasing the number of cores per CPU socket is beneficial because it helps to amortize the support logic in the CPU socket among the CPU cores. CPU support logic can include things such as cache, memory controllers, PLLs, bus interfaces, and miscellaneous control logic. Dividing the power for the support logic among a larger number of CPU cores boosts the efficiency per core. It also allows the CPU socket to achieve a higher performance level while running each core at a slower speed compared to a CPU socket that contains fewer cores.

There is also a relationship between CPU frequency and power consumption, but not a direct correlation. Intel and AMD sometimes offer processors with the same frequency but different TDPS. In this case, the performance remains the same (as long as other features like cache size have not changed) but power consumption is different.

## Manage power consumption on CPUs

Managing power consumption on CPUs implies that you need to find a balance between power and performance on your system. Based on Advanced

Configuration and Power Interface (ACPI) specifications, there are three states that can be used to reduce power consumption on CPUs<sup>4</sup>:

- ▶ T-states - throttling states

T-states will further throttle down a CPU, but not the actual clock rate, by inserting STPCLK (stop clock) signals and thus omitting duty cycles.

The T-state mechanism works by “gating-off” clock ticks to the processor core. The processor clock always ticks at a fixed rate, but one or more tick in every group of 8 ticks may be masked from the processor. If a clock tick is masked, the internal gates in the processor do not transition on that tick.

Because it is gate transitions that consume the most power, this has the effect of making the processor consume less power.

However, performance is reduced at the same time however. If 1 out of every 8 clock ticks is gated-off, then performance is reduced by roughly 1/8 or 12.5%. If 7 out of 8 ticks are gated-off (the maximum possible throttling), performance is reduced by roughly 7/8 or 87.5%. When the processor is not being throttled, no clock ticks are gated off and the processor runs at full speed. Roughly speaking, both the decrease in power and the decrease in performance are linearly proportional to the amount of throttling.

- ▶ P-states - performance states

P-states allow clock rate reduction. The P-state mechanism involves actually slowing the clock down so that it ticks less frequently (rather than just masking off ticks). When the processor is run at a slower speed, the voltage to the processor can be reduced as well.

The reduction in power is linearly proportional to the reduction in clock speed, but more than linearly proportional to the reduction in voltage; power is actually proportional to the square of the voltage. So throttling using P-states gives a more than linear reduction in power, while causing only a linear reduction in performance. The power/performance trade-off is therefore better with P-states than with T-states.

On the other hand, it takes longer for a processor to change P-states than T-states, so you have to be careful about how often you change the P-state to avoid degrading performance (while the P-state is changing, the processor essentially stops). The switch between P-states is controlled by the operating system.

Intel Xeon processors incorporate this feature and call it Demand Based Switching (DBS) with Enhanced Intel Speedstep Technology. DBS, which is operating system-dependent, is included in all dual, quad, and 6-core Xeon processor-based System x and BladeCenter servers.

---

<sup>4</sup> For more information, you could download ACPI Specs at <http://www.acpi.info/spec.htm>

There are often situations where a processor is not fully utilized. This provides the opportunity, if the workload allows it, to run the processor at reduced speed and consequently consume less power. Originally developed by Intel for Xeon processors, and then further optimized by IBM, the DBS feature utilizes the operating system to monitor processor utilization based on the applications running on the server. The system can automatically change to a lower power state (frequency and voltage) when less processing power is needed. For example, an e-mail server may run at capacity during business hours, yet be idle during the evenings and on weekends. Or a server might have surplus capacity that has not yet been tapped. DBS provides the ability to dynamically change from peak performance to cost-saving mode automatically.

To use this feature on System x with Intel processors, you need to be sure that the operating system will support it and then you have to activate it in the BIOS. As an example for the System x3850 M2, enter Setup at boot time, and go to **Advanced Setup > CPU Options** and set the Processor Performance States option to Enable.

AMD provides a similar capability (called PowerNow! technology with Optimized Power Management) in its Opteron processors. PowerNow! technology can reduce CPU power at idle by as much as 75% when the server is running an operating system that supports this feature.

To use this feature on System x with AMD processors, you need to be sure that the operating system will support it and then you have to activate it in the BIOS. As an example for the System x3755, enter Setup and go to **Advanced Setup > CPU Options** and set the Processor power management option to Enable.

- ▶ C-states - Sleep states

C-states allow the clock to be halted.

The C-states mechanism works via different levels of sleep states. When the CPU is active, it always runs at C0, meaning that the CPU is 100% turned on.

Conversely, the higher the C number, the deeper the CPU sleep mode will be. That is, more parts of the CPU are turned off and it will take longer for the CPU to return to C0 mode to be active again. The operating system tries to maintain a balance between the amount of power it can save, and the overhead of entering and exiting to and from that sleep state.

Enhanced C1 state (C1E) is an option present in System x BIOS with Intel processors. C1E is an automatic voltage/frequency reduction that occurs when an operating system places the processor in a C1 state. This option can be activated in the BIOS. As an example for the System x3850 M2, go to **Advanced Setup > CPU Options** and set the C1E option to Enable.

IBM Systems Director Active Energy Manager™ uses the T-state mechanism for power capping. The duty cycle of the CPU is lowered, based on what power capping is needed. The P-state mechanism is available to the operating system to be used for dynamic power savings.

There is a distinction between power capping and power saving, as explained here:

**Power capping** This refers to setting a specific power limit in terms of a fixed number of watts that the system should not exceed. Active Energy Manager will throttle the system to keep power from going higher than this limit. The goal is to set the cap high enough that it will rarely, if ever, be reached in practice. Knowing the maximum power that can be consumed by the system allows you to size the infrastructure (cooling) correctly.

We generally say that the goal of power capping is reduce *allocated* power, where allocated power means the amount of power and cooling capacity that has to be allocated in the data center to run a given system.

**Power savings** This refers to trying to reduce *actual* power, as opposed to allocated power. The idea is that if processors are not performing useful work, they can be throttled to use less power without hurting performance. If the workload increases and more performance is needed, the processor throttling can be removed, thereby increasing performance through the use of more power.

Understand that the goal is not to have a specific power cap that is never exceeded; instead, it is to use as little power as possible without impacting performance.

There can be a trade-off between CPU power savings and achievable performance. However, most servers rarely operate at their peak utilization level and moving to a lower power, slower CPU will often not affect the overall performance of the customer's workload. For more information about this topic, refer to 5.2.6, "Operating systems" on page 68.

## Next generation technologies from Intel

The new Nehalem platform from Intel includes a Power Control Unit (PCU) dedicated to optimize the efficiency of the chip. Nehalem's PCU breaks from the past and uses an on-chip micro-controller (one million transistors) with dynamic power sensors to actively manage entire multi-core chip power and performance.

In conjunction with PCU, Nehalem has new Integrated Power Gates that enable idle cores to be completely shut off from the power supply, thus reducing the leakage to near zero for sleeping cores. These Power Gates remove the multi-core penalty of leakage when running single or few threaded workloads.

With its new Turbo Mode capability, the PCU can dynamically alter the voltage and frequency of CPU cores to provide a significant performance boost when most cores are idle or running lower power workloads. This active power management by an on-chip micro-controller offers considerable benefit and sets the direction for power management going forward.

## 5.2.2 Memory

Although CPU is the most important factor affecting energy efficiency, memory is still an important component for power efficiency.

### Main memory technologies

Today, there are two main memory technologies used in System x servers:

- ▶ DDR2 (refer to “DDR2” on page 189 for more information)
- ▶ Fully Buffered DIMM (refer to 10.2.6, “Fully-buffered DIMMs” on page 191, for more information)

**Note:** FB-DIMM is not the next generation of DRAM. Instead, it is a new way of accessing the same DDR2 DRAMs from a new memory controller.

FB-DIMM can provide very good performance with greater throughput, but have some latency to memory accesses due to its serial architecture. For more information about this topic, refer “FB-DIMM performance” on page 195.

In terms of power consumption, FB-DIMM is very stable. Its power consumption starts fairly high at idle, and remains more or less constant regardless of the load on the memory subsystem. On the other hand, the DDR2 DIMM does not consume much power when idle, and power consumption increases proportionally as the load on the DIMM increases.

Fully Buffered DIMM is based on DDR2 for its implementation. The extra power consumption of FB-DIMMs is due to the advanced memory buffer (AMB), which consumes almost 5W. The AMB usage causes the DIMMs to get hotter, and a heat spreader on the DIMMs helps the internal fan to keep them cool. Some improvements have been made to the latest FB-DIMMs as described in “Green FB-DIMMs” on page 61.

eX4 systems also use Buffer on Board technology, which together can save up to 37% over systems that use FB-DIMMs. The cost savings can be substantial, considering that memory consumes 25 to 40% of data center power.

Recently, a vConsolidate performance benchmark was run by the independent test company Principled Technologies. This test was run on:

- ▶ A competitor server with four 2.93 GHz Intel Xeon x7350 processors and 32 2 GB DIMMs using FB-DIMMs
- ▶ A System x3850 M2 with four 2.93 GHz Intel Xeon x7350 processors and 32 2 GB DIMMs using DDR2 DIMMs
- ▶ A System x3950 M2 with eight 2.93 GHz Intel Xeon x7350 processors and 64 2 GB DIMMs using DDR2 DIMMs

Not only did the two IBM servers deliver higher performance compared to the competitor server, but power consumption was also measured and revealed that the x3850 M2 and x3950 M2 equipped with DDR2 DIMMs consumed less power than the competitor server equipped with FB-DIMMs: 13.6% less for the System x3850 M2 than the competitor server, and 19.6% for the x3950 M2 than the competitor server.

Logically, the x3850 M2 and x3950 M2 have higher performance per watt compared to the competitor server.

For more information about this benchmark, refer to the following URL:

<http://www.principaledtechnologies.com/Clients/Reports/IBM/IBMvCon1p0808.pdf>

## **Green FB-DIMMs**

To improve FB-DIMM power consumption versus DDR2 memory, certain mechanisms have been used on the latest FB-DIMMs. These new, low power FB-DIMMs for System x optimize the AMB and the DRAM. The AMB suppliers have released new, reduced power AMBs that can be used with low voltage DDR2 memory.

As an example, instead of having, on a standard FB-DIMM, the DRAM I/O running at 1.8V and the AMB core running at 1.5V—on the low power DIMM, the DRAM I/O runs at 1.5 V and the AMB core runs at 1.5 V, as shown in Figure 5-4.

On the DRAM side, the AMB drives and receives all DRAM signals with VDDQ (Voltage Supplied in memory) = 1.5V rather than at VDDQ = 1.8V. In some cases the voltage used on those new DIMMs can be different than 1.5V, but lower than the standard ones.

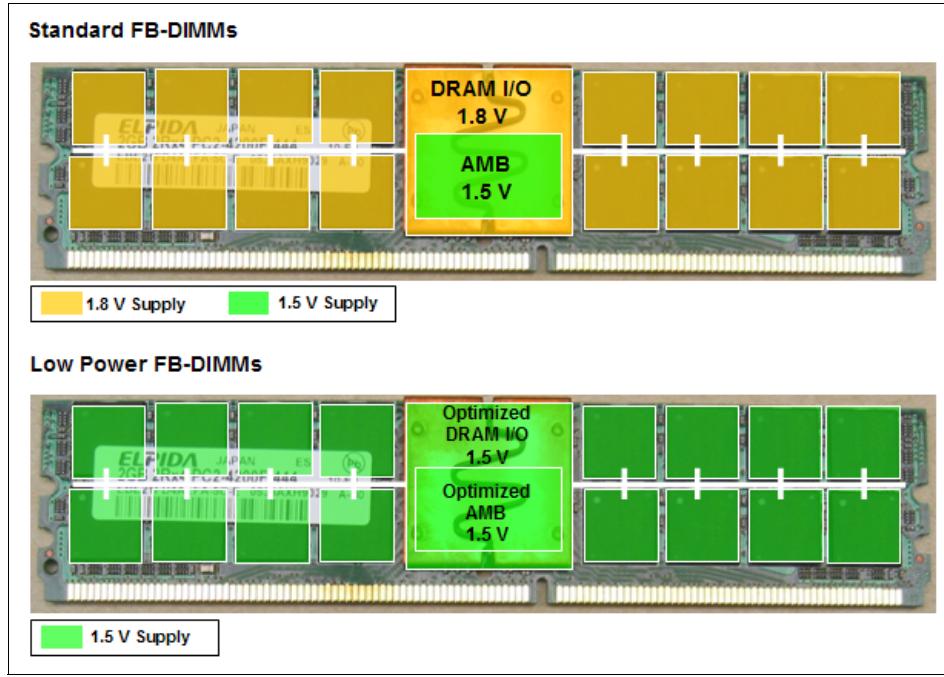


Figure 5-4 Standard FB-DIMM and low power FB-DIMM voltage comparison

The performance, timing, and other operating requirements for the low power FB-DIMMs are identical to the standard FB-DIMMs. The high-speed buses between the memory controller and the FBDIMM modules populated in the channel are exactly the same, so you do not have to replace your server.

The use of green FB-DIMMs is transparent to the system board, and you can mix them with normal FB-DIMMs. In such cases, however, you will not achieve the most energy-efficient solution but will realize some energy efficiency gains on the slots with low power DIMMs.

These improvements help to lower the power costs of the servers by reducing the power draw of the memory and, thus, the overall server power draws. Because these low-power FB-DIMMs use less power, they generate less heat, which means that the cooling system consumes less electrical power.

## Memory layout

Memory layout influences power efficiency. The following parameters can affect power consumption on memory:

- ▶ Number of DIMMS presents in the server
- ▶ Size of the DIMMS

- ▶ Number of bits per chip on each memory
- ▶ Memory clock
- ▶ Rank numbers

Table 5-2 shows a system with DDR2 memory. The measurement was performed on an LS22 blade. In this test, we varied the memory configuration; that is, the number of DIMMs, DIMM size (GB), memory speed (MHz), and ranks. For each configuration, we measured the performance of the system and monitored the blade power with Active Energy Manager (for more information about this topic, refer to “Active Energy Manager” on page 78).

Here are the results:

- ▶ The lowest power was obtained with four DIMMs (4 x 1GB at 667 MHz with 1 rank).
- ▶ The best performance was obtained with eight DIMMs (8 x 4 GB at 800 MHz with 2 ranks)
- ▶ The best performance/watt (983) was obtained with eight DIMMs (8 x 1 GB at 667 MHz with 1 rank). This solution is relevant with a configuration that does not need a significant amount of memory.

This best performance/watt did not achieve the highest performance or the lowest power, but instead it balanced both.

*Table 5-2 Memory test on LS22 blade*

# DIMMs	DIMM		Mem			Performance	Power	Perf/Watt
	size	Speed	Details					
8	4	800	2Rx4			234,293	267	878
4	4	800	2Rx4			231,010	238	971
8	4	667	2Rx4			229,586	265	866
4	4	667	2Rx4			225,822	237	953
8	2	667	1Rx4			228,616	245	933
4	2	667	1Rx4			222,388	230	967
8	1	667	1Rx8			228,030	232	983
4	1	667	1Rx8			208,037	220	946

For FB-DIMMs, the Advanced Memory Buffer (AMB) consumes 5W of power. Therefore, using 2x 4GB DIMMs will consume less power than 4x 2GB DIMMs. However, the trade-off is that only half of the memory channels will be used (assuming that all FB-DIMM systems have 4 FB-DIMM channels), so perform an analysis to determine the most appropriate configuration for your needs.

**Note:** Lowest power does not necessarily equate to the best performance/watt, and in most cases best performance/watt would probably occur with 1 DIMM per channel, or 4 DIMMs minimum on an FB-DIMM system.

### Memory bus speed

Some of the latest System x servers support the ability to change their memory bus speed. This tuning can be done in the BIOS. Depending on the target applications you are running on your system, slowing down the memory bus will reduce power consumption.

**Note:** Changing the memory bus speed could reduce performance for most applications. Performance analysis needs to be done to be sure that there is no impact on your server applications.

### 5.2.3 Drives

Solid® state drives (SSDs) are available on System x and BladeCenter, and they offer power savings over conventional hard disk drives. An added benefit is that SSDs are three times more reliable because they have no moving parts.

Having local disks on every server consumes a tremendous amount of power. Conventional 2.5-inch SAS/SATA drives use as much as a 16W per drive. Because solid state drives do not have moving mechanical parts, the latest solid State Drives are more power efficient. The power saving of SSDs over other drives that IBM has measured was approximately 6 W.

Other ways to save power on traditional HDDs (SAS/SATA drives) are to use slower spinning drives and to spin down the HDD to save power when it is not in use. For the spin down method to work properly, the operating system has to be configured to spin the drive down after a certain time of inactivity.

### 5.2.4 Fans

Because it represents an important part of the total power allocated in a server, IBM has developed algorithms on IBM System x Server and BladeCenter that optimize the systems for low power consumption.

IBM System x servers use fans and blowers that are variable in their rotational speeds. There are sensors distributed within the chassis that determine the need of the components to be kept within specified operating temperatures for reliable

operation. Some servers use zonal cooling in cases where the fans responsible for the zone alone respond to a cooling need.

Sensors available within our chassis detect the inlet air temperature and altitude of operation in addition to component temperatures to provide adequate system cooling.

Fan speed control algorithms detect the ambient conditions and the component thermal limits and provide just the optimized airflow to cool the servers as needed. With the ability to know component thermal profiles and ambient conditions, the servers do not waste fan power providing more cooling than is necessary.

### 5.2.5 Power supplies

Power supplies are a significant source of efficiency (and inefficiency) when it comes to power consumption. Power supply efficiency can vary considerably with input voltage and load (see Figure 5-5 on page 66), but power supplies are generally more efficient the higher the load. Efficiency is 2% to 3% better on average with input voltage in the 200V range when compared to the 100V range.

Compounding the situation is the requirement for redundant power supplies to ensure that a server is operational even after a power supply fails. The trade-off to this uptime benefit is energy efficiency. Redundant supplies will run at or below 50% and will not be at the top of their efficiency curve. Two power supplies running at around 50% capacity will always draw more than one that is running at 90% capacity as you can see in Figure 5-5 on page 66.

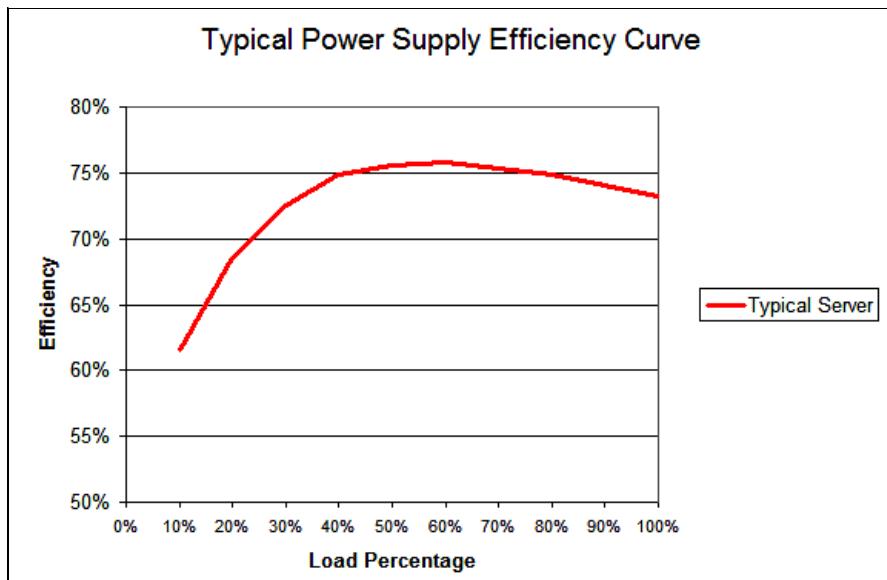


Figure 5-5 Typical power supply efficiency curve on a server

Some systems are designed with non-redundant power, some are designed with redundant power, and some are designed to work either way. In the non-redundant case, if the system allows a choice in power supply output, the choice should be made such that the system power using your feature set is in the “sweet spot” of the efficiency curve.

There are two redundant cases; N+N and N+1. It is important to understand how redundancy impacts efficiency:

- ▶ For N+N redundant systems, power supplies are typically loaded to 15% to 35% of maximum load, which is usually not the best part of their efficiency curve (see Figure 5-5).
- ▶ For N+1 redundant systems, the load on each power supply is also dependent on what N is. If N is 1, then the configuration is considered to be N+N. The larger N is, the higher the percentage of load on the power supply, which is desirable from an efficiency perspective.

It is also important for power supplies to have relatively flat efficiency curves so that they run efficiently when they are full load, light load, or in redundant mode. Several new industry standards enforce high efficiency at 20%, 50%, and 100% load levels.

Wherever it can, IBM designs power supplies to maximize efficiency for a given configuration. For example, if the system is to be non-redundant, the power

supply is designed for maximum efficiency at the higher end of the load range. Conversely, if a system is to be redundant, then the power supply is designed for maximum efficiency at the lower end of the load range.

Most of our servers have AC-to-DC power supplies. Some have DC-to-DC power supplies, like the BladeCenter T and some 1U and 2U server offerings.

**Tip:** Use the power supply that supports the load for the configuration you have, rather than one that supports the maximum configuration.

There are two kinds of power presents in the data center:

- |           |  |
|-----------|--|
| <b>AC</b> | Distributed from the wall in most data centers |
| <b>DC</b> | Used directly by each component in servers     |

A typical power supply used in the server industry is approximately 65% to 75%<sup>5</sup> efficient at converting AC voltage to DC power used inside the server. This means that for every 1000W consumed by the server, perhaps only 700W are used productively and 300W does nothing more than generate waste heat, as shown in Figure 5-6. 300W equals 1023 BTUs of hot air that needs to be cooled for no benefit.

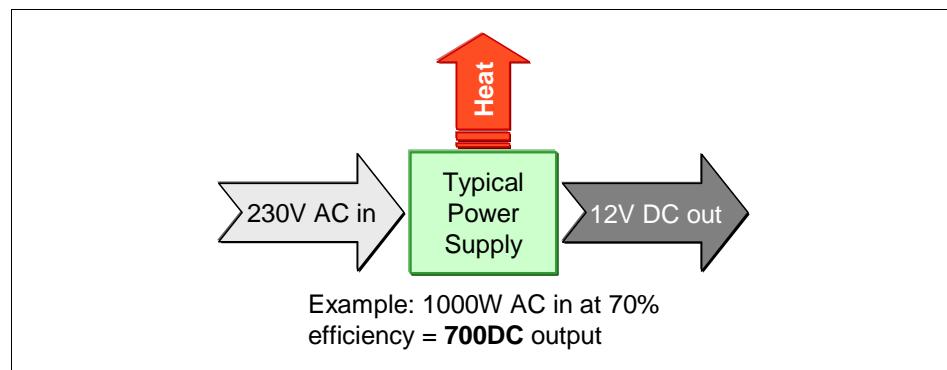


Figure 5-6 Standard power supply

By contrast, the power supplies that IBM uses in System x servers and BladeCenter chassis are significantly better with a peak of 91% efficiency in the case of BladeCenter<sup>6</sup>, as shown in Figure 5-7 on page 68. This means that for every 1000W of power consumed by the server, you would use 910W for

<sup>5</sup> See chapter 2 “Existing power Supply Efficiency”, in report from High Performance Buildings [http://hightech.lbl.gov/documents/PS/Final\\_PS\\_Report.pdf](http://hightech.lbl.gov/documents/PS/Final_PS_Report.pdf)

<sup>6</sup> See IBM press release at the following URL:  
<http://www-03.ibm.com/press/us/en/pressrelease/20633.wss>

processing and waste only 90W generating heat. This will save money both on power consumption up front and on cooling at the back end.

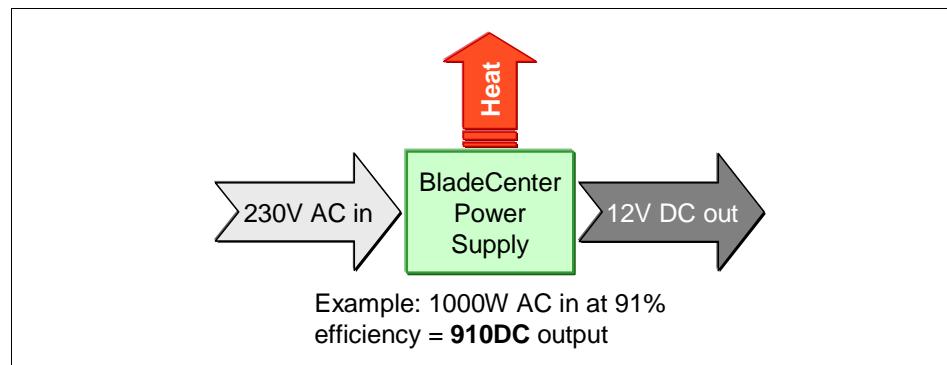


Figure 5-7 *BladeCenter power supply*

There are other energy-related aspects of power supply design that are included in power supplies, depending on the model:

- ▶ A near unity power factor at light, medium and heavy loads.
- ▶ An optimal fan speed algorithm if the power supply includes a cooling fan.
- ▶ A power supply can support high input voltages.
- ▶ The location of the power supply in the server.
- ▶ The monitoring of the input and output power on the power supply.

### 5.2.6 Operating systems

We have seen that energy efficiency can be done at the hardware level, but some parameters can be optimized under the operating system. This section addresses this topic for Microsoft Windows Server 2008 and the Linux operating systems.

#### Microsoft Windows Server 2008

Microsoft designed its new system Windows Server 2008 with the idea of being more power efficient. The latest ACPI processor power management (PPM) features have been included, such as support for processor performance states (P-states) and processor idle sleep states on multiprocessor systems.

Microsoft has conducted power consumption tests<sup>7</sup> comparing Windows Server 2003 and Windows Server 2008:

- ▶ One test was performed without any tuning ("out of the box" test); it was simply a standard installation with the same platform. The result of this test

showed that at comparable levels of charge, Windows Server 2008 can reach a power saving of up to 10% over Windows 2003.

- ▶ Another test was conducted on Internet Information Services 7 (IIS 7) on those two Microsoft Operating Systems (Windows Server 2003 and 2008 with “out-of-the-box” settings). Power consumption measurements on IIS 7 were performed in idle mode (no users), and also with 20 active users. In both cases, Windows Server 2008 delivered better results than Windows Server 2003. In idle mode, 2008 achieved power savings of up to 2.3%. With 20 users, 2008 achieved power savings of up to 6.8%.

Those improvements in energy efficiency can be attributed in part to the default activation of the processor power management (PPM) features under Windows Server 2008.

By default, in Windows Server 2003, the CPU always runs at P0 (the highest-performance processor state). In the case of Windows Server 2008, it takes advantage of the processor performance state adapting the CPUs on the workload. For more information about P-states and C-states, or to learn how to activate those features on System x servers, refer to “Manage power consumption on CPUs” on page 56.

**Note:** You will need to confirm the impact these changes have on the performance of your applications running on the server. It is always a question of balance between performance and power consumption.

Microsoft provides a document that can help you to optimize the balance of your Windows 2008 Servers between performance and power savings. This document is available at the following URL:

<http://www.microsoft.com/whdc/system/pnppwr/powermgmt/ProcPowerMgmt.mspx>

## Linux

With kernel 2.6.18 or later, the process scheduler for multi-core systems provides the ability to be tuned. When the number of running tasks is less important than the logical CPUs available, the system will minimize the number of CPU cores carrying the process load. The system attempts to keep the remaining idle cores idle longer and then save power<sup>8</sup>.

---

<sup>7</sup> See the Out-of-the-Box Power Savings chapter in the following document:  
[http://download.microsoft.com/download/4/5/9/459033a1-6ee2-45b3-ae76-a2dd1da3e81b/Windo ws\\_Server\\_2008\\_Power\\_Savingr.docx](http://download.microsoft.com/download/4/5/9/459033a1-6ee2-45b3-ae76-a2dd1da3e81b/Windo ws_Server_2008_Power_Savingr.docx)

<sup>8</sup> Refer to the Intel Web Site: <http://software.intel.com/sites/oss/pdf/mclinux.pdf>

This tuning can be activated in the /sys/devices/system/cpu/sched\_mc\_power\_savings file.

By default, the system is set for optimal performance with a value set to zero (0). To activate this tuning, change this value to 1 as follows:

```
echo 1 > /sys/devices/system/cpu/sched_mc_power_savings
```

Activating the C1E option in the BIOS enables a maximum power saving on the processor when idle. Refer to “Manage power consumption on CPUs” on page 56 for more details about how to activate this feature in the BIOS.

Figure 5-8 and Figure 5-9 on page 71 show the CPU utilization on the same System x3850 M2 with four quad-core CPUs at 2.93 GHz under RHEL 5.2. From an operating system perspective, the system has 16 processors.

Figure 5-8 shows the utilization of this server with the sched\_mc\_power\_savings set to 0. In this figure, all the cores present in the system are used. One CPU has a utilization around 53%, another has 24%, and all the others have between 5% and 15%.

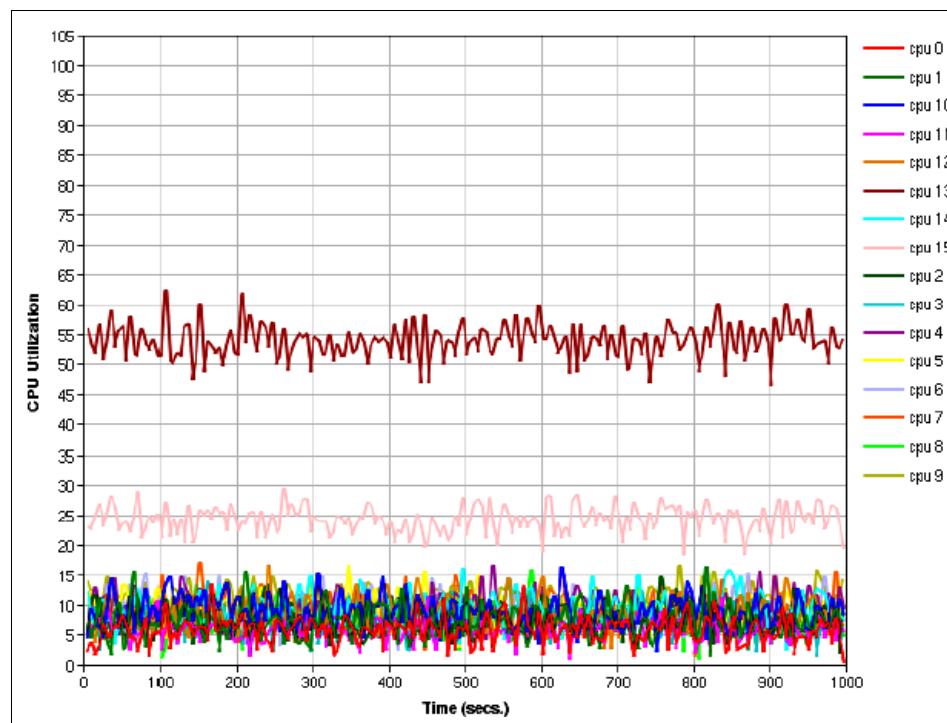


Figure 5-8 CPU utilization - x3850 M2 with sched\_mc\_power\_save deactivated

Figure 5-9 shows the utilization of this server with the parameter `sched_mc_power_savings` set to 1, meaning that the scheduler multi-core power saving option is active. In this figure, only some cores are used. Some of those cores are not really used to compare to the same core on Figure 5-8 on page 70. This means that before distributing the process load to cores in other processor packages, the scheduler distributes the process load such that all the cores in a processor package are busy.

Figure 5-9 confirms this point. It shows one CPU used at 53%, four CPUs used at 25%, and three CPUs used at 10% (refer to number 1 in Figure 5-9). Compare to Figure 5-8 on page 70, which shows fewer CPUs working, but at a higher level of utilization.

As shown in Figure 5-9, the eight other CPUs are not really used and stay in idle mode longer as part of other processor packages (refer to number 2 in Figure 5-9).

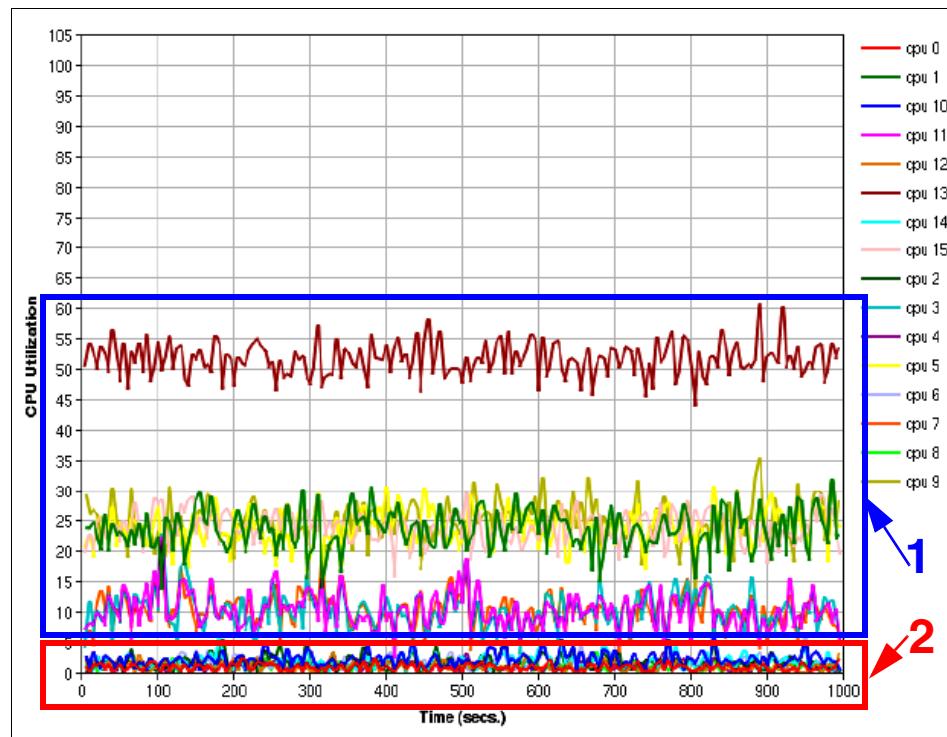


Figure 5-9 CPUs utilization x3850 M2 with `sched_mc_power_save` activated

This tuning (setting the parameter value to 1) can save a significant amount of power in cases where there is significant idle time in your system.

**Note:** You will need to confirm the impact this change has on the performance of your applications running on the server. Again, it is always a question of balance between performance and power consumption.

If there is some impact, you may be able to create a power-saving policy (using scripts) to use this setting during non-working hours.

The cpufreq infrastructure, included in the latest Linux kernel version, allows users to change the frequency policy using some “governor policies”. Those profiles are based on different criteria such as CPU usage.

You can choose between different governors:

- ▶ Ondemand governor

The CPUfreq governor “ondemand” sets the CPU, depending on the current usage. To do this, the CPU must have the capability to switch the frequency very quickly. There are a number of sysfs file-accessible parameters available, such as sampling\_rate, show\_sampling\_rate\_(min|max), up\_threshold, sampling\_down\_factor and ignore\_nice\_load.

- ▶ Conservative governor

The CPUfreq governor “conservative”, much like the ondemand governor, sets the CPU depending on the current usage. It differs in behavior in that it gracefully increases and decreases the CPU speed rather than jumping to maximum speed the moment there is any load on the CPU. This behavior more suitable in a battery-powered environment.

- ▶ Userspace governor

The CPUfreq governor “userspace” allows the user, or any userspace program running with UID root, to set the CPU to a specific frequency by making a sysfs file “scaling\_setspeed” available in the CPU-device directory.

- ▶ Performance governor

The CPUfreq governor “performance” sets the CPU statically to the highest frequency within the borders of scaling\_min\_freq and scaling\_max\_freq.

- ▶ Powersave governor

The CPUfreq governor “powersave” sets the CPU statically to the lowest frequency within the borders of scaling\_min\_freq and scaling\_max\_freq.

### 5.2.7 Virtualization

Virtualization is an alluring solution for rationalizing the current management practice of dedicating a single workload to a server. To improve upon that low

level of utilization of resources, many IT departments are looking at virtualization as a way to rationalize the RAS benefits of isolated workloads with potentially higher server utilization.

Typical software is unable to keep a processor busy most of the time<sup>9</sup>. In fact, the average utilization of x86 processors (Intel and AMD) is on the order of only 10% to 40%. This means that much of the energy used by the processor is wasted while the processor is idling.

Figure 5-10 from an IBM Consolidation study confirms that the “typical” x86 server utilization is around 10% of CPU in a given standard enterprise data center.

Consolidation Parameters for Source Workloads					
Legacy 2-P Workloads	Typical Processor	Avg CPU Utilization	Peak CPU Utilization	Avg Memory Used	Peak Memory Used
	Xeon 2.0GHz	8%	48%	568	768
	Xeon 1.8GHz	5%	47%	592	768
	Xeon 1.8GHz	8%	52%	611	768
	Xeon 1.8GHz	9%	60%	1,199	1,536
	PIII 1.4GHz	9%	70%	603	1,024
	Xeon 2.0GHz	6%	50%	994	1,280
Legacy 4-P Workloads	Typical Processor	Avg CPU Utilization	Peak CPU Utilization	Avg Memory Used	Peak Memory Used
	Xeon MP 2.5GHz	6%	35%	841	1,024
	Xeon MP 2.5GHz	4%	24%	737	1,024
	Xeon MP 2.7GHz	4%	34%	935	1,280
	Xeon MP 2.5GHz	5%	37%	1,553	1,792
	Xeon MP 2.7GHz	6%	45%	882	1,536
	Xeon MP 2.8GHz	4%	34%	1,295	1,536

**Source: IBM Consolidation Study**

Figure 5-10 “Typical” x86 server utilization - Source: IBM Consolidation Study

IBM supports tools such as VMware ESX, Microsoft Hyper-V™ and Xen, which allow you to partition processors and other server resources (local and remote) so that multiple operating systems and multiple application sets can be running concurrently and independently on the same server.

<sup>9</sup> EPA Energy Star, Final Server Energy Measurement Protocol, November 3, 2006

If four different software stacks, for example, are each assigned 20% of the processor resources, you can achieve up to 80% utilization of the processor cycles (instead of 15% to 40%), with headroom to spare. If one or more of those stacks later requires additional resources, or if you need to add another stack, you have the available cycles. Using quad-core processors gives you even more flexibility in this regard. This is an excellent way to make efficient use of your processing power, even with single-threaded applications.

IBM eX4 technology provides the x3850 M2 and x3950 M2 servers with advanced capabilities designed to offer higher throughput, exceptional reliability, and the ideal platform for virtualization. For more information about the eX4, refer to 9.3.3, "IBM XA-64e fourth-generation chipset" on page 175.

On a Quad sockets IBM System x server, a test was conducted internally at the IBM System Performance Lab (refer to 1.3, "The System x Performance Lab" on page 5 for more information). Figure 5-11 on page 75 shows the result of this test. Efficiency has been measured based on the number of virtual machines (VMs) running on the system and on the power consumption's server. It is the power consumption's server divided by the number of VMs. The power consumption was measured on the server.

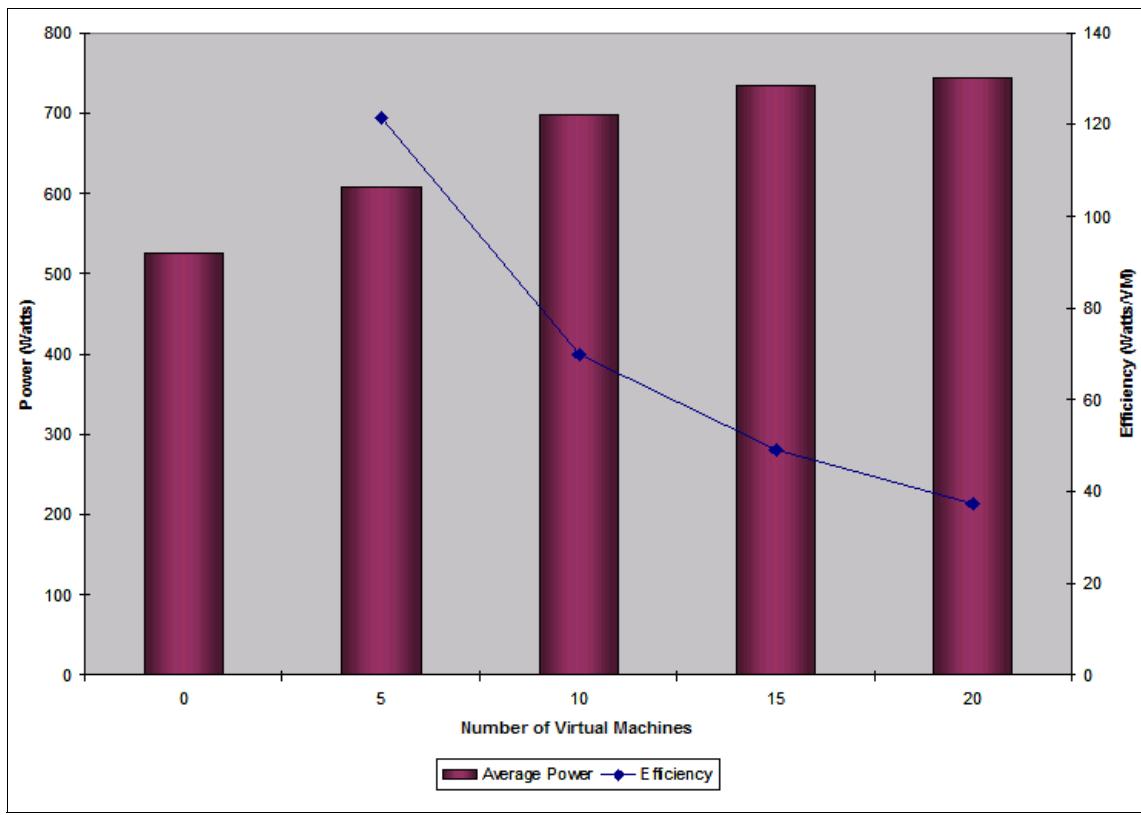


Figure 5-11 Virtualization and Energy efficiency test performed on Quad sockets IBM System x server

In idle mode, without a virtual machine, the server consumed 525W. When 5 VMs were added, it consumed 608W. With 10 VMs, it consumed 697W. When another 5 VMs were added, the server consumed 735Ws. Finally, with 20 VMs, it needed 744W. Between 10 and 20 VMs, when the number of VMs was doubled, the server only consumed an additional 7% of power.

These results are very interesting. They show that you can really increase the efficiency (power consumed per virtual machine) of your server by adding VMs to your server (the lower is the better). With 5 VMs, the efficiency is at 122; with 10 VMs, the efficiency is at 70; with 15 VMs, the efficiency is at 49; and finally with 20 VMs, the efficiency is at 37. Using this kind of configuration, you can replace 20 servers and use only 37W per virtual machine.

Introducing virtualization and potentially garnering the power savings that may result in the consolidation of existing server and storage hardware can be a first step in helping to reduce power consumption.

Another, or complementary, way to further cut consumption is to use and optimize your efficiency at the rack level, as explained in the following section.

## 5.3 Rack-level solutions

In addition to addressing power and thermal efficiency at the server level, focusing on the issue from a rack perspective leads you to additional solutions for handling heat and power problems.

### 5.3.1 IBM products to manage energy efficiency

IBM has developed tools, such as Power Configurator and Active Energy Manager, that can be used to manage energy efficiency on System x and BladeCenter products. This section describes those tools.

To read actual power, be aware that there is a difference between using the label rating power versus using the Power Configurator power, and versus using Active Energy Manager. The least accurate is the label rating. The most accurate guide for cooling infrastructure planning is Active Energy Manager's actual power reading.

#### IBM System x Power Configurator

IBM System x Power Configurator is a Windows tool that can be used to obtain power estimates for a given configuration on IBM System x or BladeCenter servers.

This tool can be downloaded from the IBM Web Site at the following URL:

<http://www.ibm.com/systems/bladecenter/resources/powerconfig/>

After the software is installed, you simply choose the configuration you want to estimate. In the following order select your country, and then AC or DC power. Next, select the rack, server, CPUs, memory, and so on. After you enter all of those parameters, the application will estimate the power consumption. If necessary, the configuration can be exported to a spreadsheet.

Figure 5-12 on page 77 shows a screenshot of this application. It displays the power consumption of an x3850 M2 with Max Configuration.

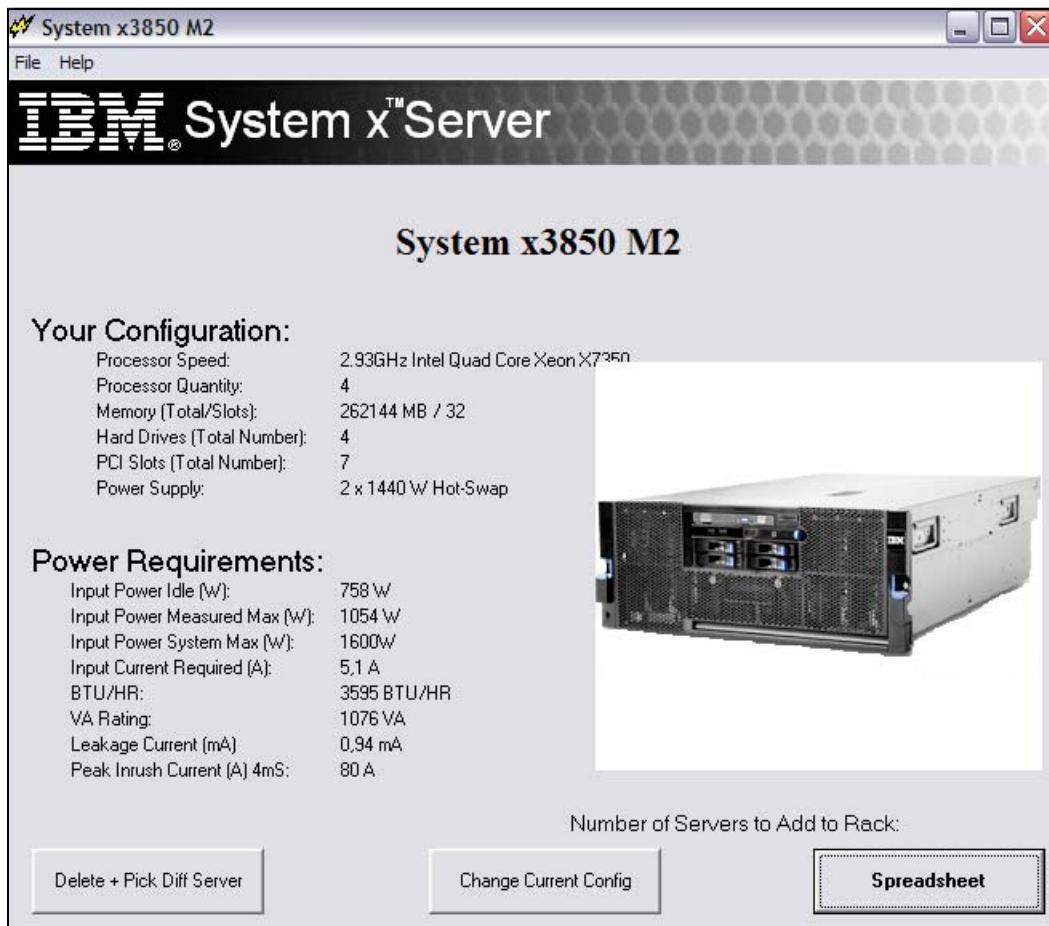


Figure 5-12 IBM System x power configuration used for a System x3850 M2 server

This tool estimates:

- ▶ Idle power

This refers to the power drawn with the machines logged into the operating system, and with no other applications running. Memory, HDDs, optical drives, and tape drives are not being utilized.

- ▶ Max measured power

This refers to the power drawn by the machine with the CPUs 100% utilized and with memory utilized 10% more than base utilization. The exerciser used is Prime95 (torture test), which can be downloaded from this site:

<http://www.mersenne.org>

**Note:** Max measured power differs from the SPECpower\_ssj2008 benchmark, which gives an average performance per watt measurement.

- ▶ System rating

This refers to the maximum power able to be drawn by the machine according to the system label rating.

Figure 5-13 shows the power estimations on the same system x3850 M2 with Max Configuration that is displayed in Figure 5-12 on page 77.

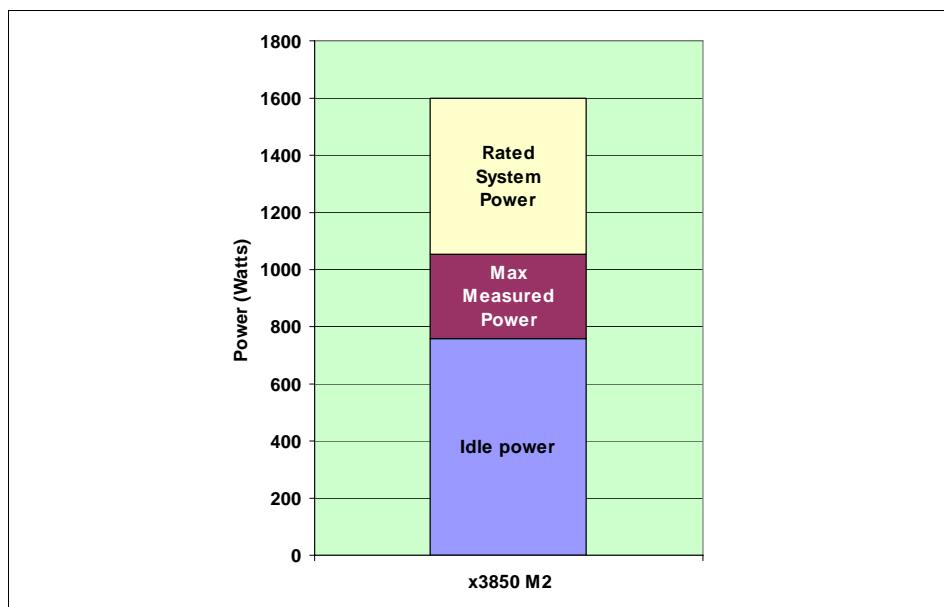


Figure 5-13 Power estimations for x3850 M2 with Max Configuration option

The Max measured power differs from the Rated system power. The max measured power is the point that can be reached in a data center environment.

**Note:** The Power configurator tool provides guidance about server consumption, but does not replace real measurements.

## Active Energy Manager

IBM Systems Director Active Energy Manager (AEM) measures, monitors, and manages the energy components built into IBM systems, thus enabling a cross-platform management solution. AEM extends the scope of energy

management to include facility providers to enable a more complete view of energy consumption within the data center.

Starting with V4.1, Active Energy Manager is integrated into the Web-based interface of IBM Systems Director. AEM supports the following endpoints: IBM BladeCenter, System x, POWER®, and System z® servers. IBM storage systems and non-IBM platforms can be monitored through PDU+ support. In addition, Active Energy Manager can collect information from select facility providers including Eaton, Emerson Liebert, Raritan and SynapSense.

Figure 5-14 shows you the interface of the latest version of AEM.

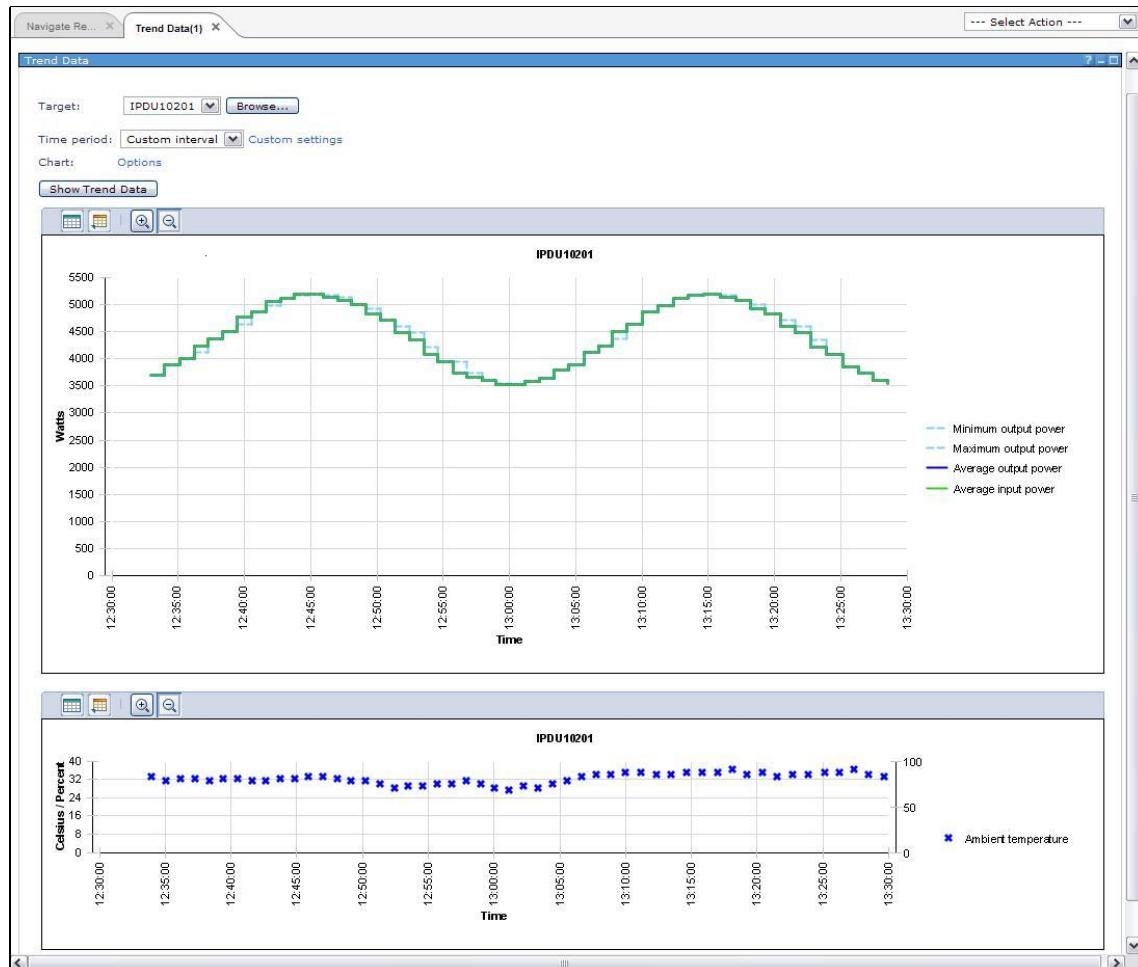


Figure 5-14 Example of power consumption and temperature trends for an IPDU on AEM

The Active Energy Manager server can run on the following platforms: Windows on System x, Linux on System x, Linux on System p®, and Linux on System z. Active Energy Manager uses agent-less technology and therefore no agents are required on the endpoints.

Active Energy Manager can provide a single view of the actual power usage across multiple platforms, as opposed to the benchmarked or rated power consumption. It can effectively monitor and control power in the data center at the system, chassis, or rack level. By enabling these power management technologies, data center managers can more effectively power manage their systems while lowering the cost of computing.

The following power management functions are available with Active Energy Manager:

- ▶ Power trending

With power trending, you can monitor the consumption of power by a supported power-managed object in real time. You can use this data not only to track the actual power consumption of monitored devices, but also to determine the maximum value over time. The data can be presented either graphically or in tabular form.

- ▶ Thermal trending

With thermal trending, you can monitor the heat output and ambient temperature of a supported power managed object in real time. You can use this data to help avoid situations where overheating may cause damage to computing assets, and to study how the thermal signature of various monitored devices varies with power consumption. The data can be presented either graphically or in tabular form.

- ▶ CPU trending

With CPU trending, you can determine the actual CPU speed of processors for which either the power saver or power cap function is active. The data can be presented either graphically or in tabular form.

- ▶ Power saver

With power saver, you can save energy by throttling back the processor voltage and clocking rate. You can use the power saver function to match computing power to workload, while at the same time reducing your energy costs. Power saver can be scheduled using the IBM Systems Director scheduler. You can also write a script to turn power saver on or off based on the CPU utilization.

- ▶ Power cap

With power cap, you can allocate less energy for a system by setting a cap on the number of watts that the power managed system can consume. If the power consumption of the server approaches the cap, Active Energy Manager throttles back the processor voltage and clocking rate in the same way as for the power saver function. In this way you can guarantee that the power cap value is not exceeded.

The advantage of power cap is that you can limit the energy consumption of supported systems to a known value and thereby allow data center managers to better match power requirements to power availability. Power cap can be scheduled using the IBM Systems Director scheduler.

The latest version of Active Energy Manager allows the user to define some power policies. A *power policy* is either a power cap or power savings setting that can be defined and applied to any number of individual systems or groups of systems. A *group power capping policy* specifies an overall power cap that the systems in the group collectively may not exceed, and it can be applied to any number of groups. These policies are continually enforced by Active Energy Manager on the systems or groups to which the policies are applied.

Active Energy Manager also provides a source of energy management data that can be exploited by IBM Tivoli® enterprise solutions such as IBM Tivoli Monitoring and IBM Tivoli Usage and Accounting Manager.

Active Energy Manager offers a solution that helps to determine the proper power allocation for each server in the data center. It can assist customers in determining how to allocate power more efficiently to existing servers so that additional servers can be accommodated without the need for additional power and cooling. When power is constrained, chargeable optional features of Active Energy Manager allow power to be rationed on a server-by-server basis, enabling available processing power to match current workload more closely.

To conclude, Active Energy Manager enables you to optimize the usage of your computing resources by measuring, monitoring, and controlling energy consumption, thereby helping you to reduce IT costs and provide more efficient planning for your data center. Active Energy Manager's open design and support for industry standards enable heterogeneous physical management with support for multiple platforms and operating systems, helping to protect your IT investment.

### 5.3.2 IBM intelligent power distribution units

For systems that do not have embedded or manageable instruments on board, intelligent Power Distribution Units (iPDUs) are available. The IBM DPI® C13

PDU+ and IBM DPI C19 PDU+ iPDU's contain versatile sensors that provide power consumption information of the attached devices, and environmental information such as temperature and humidity.

The iPDU's serial and LAN interfaces allow for remote monitoring and management through a Web browser, any SNMP based Network Management System, Telnet, or a console over a serial line. Events can be notified by SNMP traps or e-mail, and it is possible to send out daily history reports, also by e-mail.

Active Energy Manager is also capable of managing the iPDU's.

### 5.3.3 Hardware consolidation

Typical racks contain many power-consuming devices in addition to servers and storage. There may be KVM switches, Ethernet switches, Fibre Channel switches, Myrinet and other high-speed communication switches, plus hundreds of power, KVM and communications cables to link everything together. In addition, each rack-optimized server contains components that consume power, including floppy and CD-ROM drives, systems management adapters, power supplies, fans, and so on as explained in a previous chapter. Some hardware solutions can help to consolidate those elements. Those solutions are more energy efficient, and are described in this chapter.

#### BladeCenter technology

By consolidating up to four communications switch modules, four power supply modules, two blower modules, two management modules, a CD-ROM drive and a floppy drive into one BladeCenter chassis (containing 14 blade servers for BladeCenter E and H), IBM is able to remove more than a hundred components from the individual servers and racks and replace them with a few centralized components per BladeCenter chassis. This offers a number of advantages: lower overall power usage, lower heat output, fewer potential points of failure, simplified server management, and extensive redundancy and "hot-swappability".

According to IBM internal testing, the BladeCenter design can reduce power utilization by as much as 20% to 40% compared to an equivalent number of 1U servers, and greatly improve air flow behind the rack. Consolidation of hardware along with using smaller, lighter hardware such as 2.5-inch drives also addresses the nontrivial issue of how much weight a data center's floor can support.

#### iDataPlex Technology

IBM System x iDataPlex™, an Internet-scale solution, helps clients face constraints in power, cooling, or physical space. The innovative design of the iDataPlex solution integrates Intel processor-based processing into the node,

rack, and data center for power and cooling efficiencies, and required compute density.

The iDataPlex solution help to pack more processors into the same power and cooling envelope, better utilizing floor space, and creating the right-size data center design.

Figure 5-15 shows the difference between a typical enterprise rack and the new iDataPlex optimized rack.

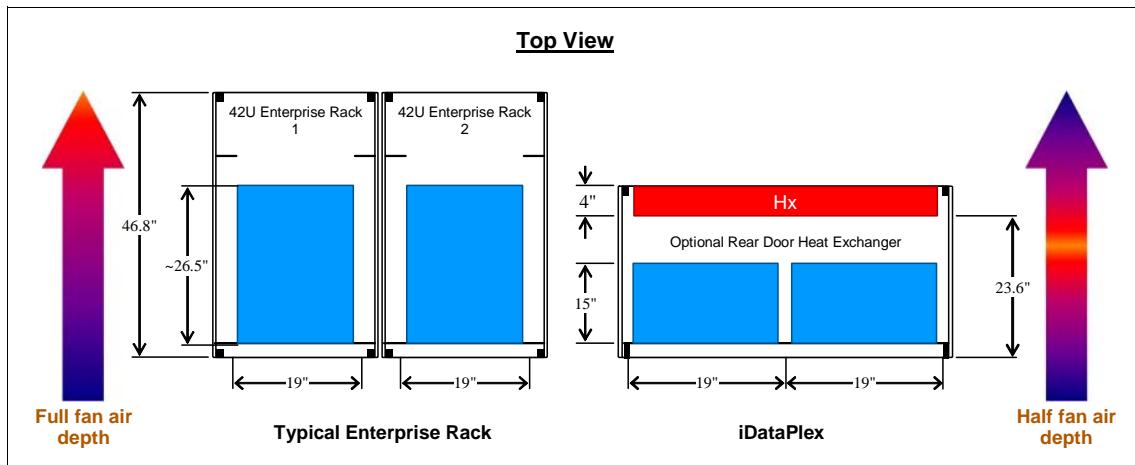


Figure 5-15 iDataPlex: Rack innovated power and space saving

In this iDataPlex rack, server density has been doubled. The direct impact is increased air flow efficiency. This rack reduces the amount of air needed to cool the servers by half. It cuts cooling costs 20% as compared to equivalent compute power in an enterprise rack. An optional Rear Door Heat eXchanger for the iDataPlex rack is available. Instead of blowing hot air from rack, which gets diluted with cooler air before reaching the CRAC units, the rear door directly removes heat from air where it is the warmest (most efficient heat transfer) without the air mover cost.

Energy efficiency attributes (some of which are addressed in 5.2, “Server-level solutions” on page 53) are present in the iDataPlex chassis:

- ▶ Optimized CPU heat sinks using heat pipes wrapped that allow reducing airflow per node.
- ▶ Multiple low powered Intel processors are available (45W-60W).
- ▶ DIMM cooling has been improved; they have been placed at the front of the iDataPlex nodes to get fresh air.

- ▶ “Green DIMMs” are available for the iDataPlex racks solutions; refer to “Green FB-DIMMs” on page 61.
- ▶ BIOS parameters can be configured to optimize power saving by reducing server component power consumption.
- ▶ The shared power supply is very efficient, with up to 90% efficiency.
- ▶ Shared fans maximize airflow utilization. A speed control algorithm has been developed and server fans adapt to workload demand to allow for less airflow during periods of lower server utilization.
- ▶ The fans are also very efficient and their power even at maximum airflow is low.
- ▶ A low impedance node flow can minimize air moving device pumping power.
- ▶ A node can be removed without having flow bypass around active nodes.

## 5.4 Data center-level solutions

After you have minimized power and heat issues at the server and the rack levels, the next step is to remove the excess heat from the data center as efficiently as possible.

The basic rule in the data center is that “power in equals heat out.” As servers require more power, they also require more cooling. So the rule is essentially stating the obvious, without accounting for any efficiency in the power and cooling systems themselves.

British thermal unit (BTU) is a unit of energy used in the power, steam generation, and heating and air conditioning industries. In the IT industry, it is used to measure the heat output from the servers or racks: BTU/hour. The BTU/hour is equal to the AC wattage x 3.41 where 3.41 is a constant value.

For example, the x3850 M2 in Figure 5-12 on page 77 has an AC Power Measured Max at 1054 Watts. The BTU/HR = 1054 x 3.41= 3594.14.

When discussing power in the data center, values for Power Usage Effectiveness (PUE) and Data Center Infrastructure Efficiency (DCiE) are often used to indicate how efficient the data center is. One part of the equation is the IT equipment power. These values are calculated as shown in Figure 5-16 on page 85. A data center with high power efficiency is one with a DCiE of more than 60%.

**PUE:** Power Usage Effectiveness  
**DCiE:** Data Center Infrastructure Efficiency

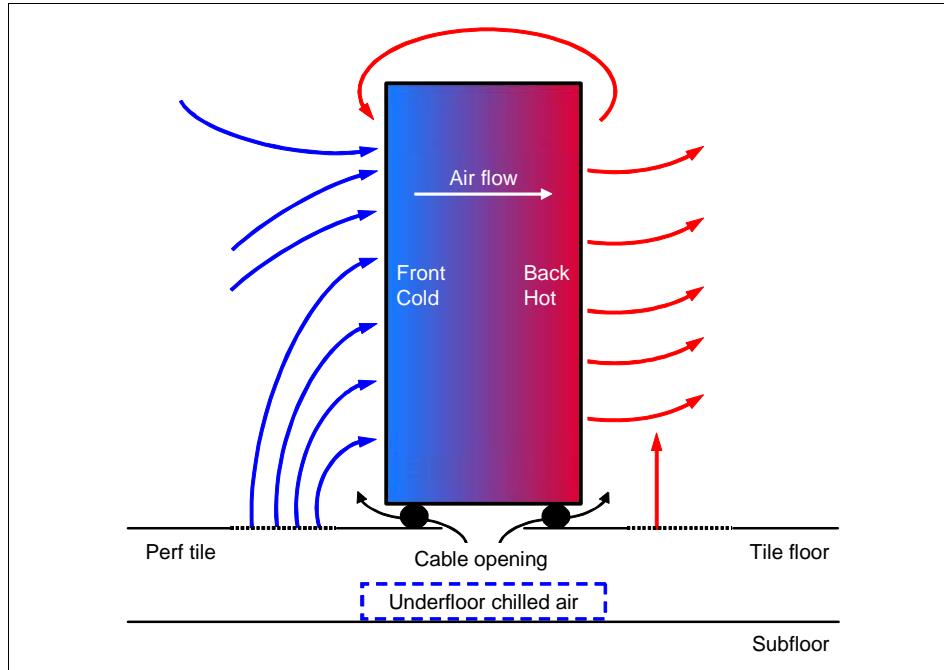
$$\text{PUE} = \frac{\text{Total Facility Power}}{\text{IT Equipment Power}}$$

$$\text{DCiE} = \frac{1}{\text{PUE}} = \frac{\text{IT Equipment Power}}{\text{Total Facility Power}} \times 100\%$$

Figure 5-16 Power Usage Effectiveness and Data Center Infrastructure Efficiency

**Important:** IT equipment that uses less power requires less energy for cooling. This frees up energy for further IT equipment and can also free up floor space used by cooling equipment.

Typically, rack servers pull in cool air through the front of the rack and blow hot air out the back. Heated air rises and is captured by the computer room air conditioning (CRAC) units and cooled before being returned to the room to begin the process again. Unfortunately, this system is rarely as effective as you would like it to be. Inevitably, some of the hot air circulates toward the front of the rack where it is sucked back into the upper servers in the rack. As a result, the cooling effect is diminished, as illustrated in Figure 5-17 on page 86.



*Figure 5-17 Typical data center rack*

To address the difficulty of managing airflow characteristics and the thermal load issues in many data centers, IBM developed the IBM Rear Door Heat eXchanger. It attaches to the back of a 42U IBM enterprise rack cabinet to provide high-efficiency water cooling right at the rack. This helps to reduce the strain on existing computer room air conditioning (CRAC).

The Rear Door Heat eXchanger is ideal for “hot spot” management and super-high density applications where the data center cooling solution may be inadequate. It can remove up to 50,000 BTUs of heat per rack.

The IBM Rear Door Heat eXchanger allows you to have a high-density data center environment that will not increase cooling demands and may actually lower them. It eliminates the heat coming off the rack even if the rack is fully populated.

With this exchanger in place, when hot air exits the rear of the servers, the heat is absorbed by the chilled water circulating within the door. The door requires no electricity and can be opened easily to allow servicing of the devices in the rack. The water carries the heat out of the data center through the plumbing in the floor, using existing AC chilled water lines. Then the water is chilled again and

returns to the Rear Door Heat eXchanger to continue the cycle, as illustrated in Figure 5-18.

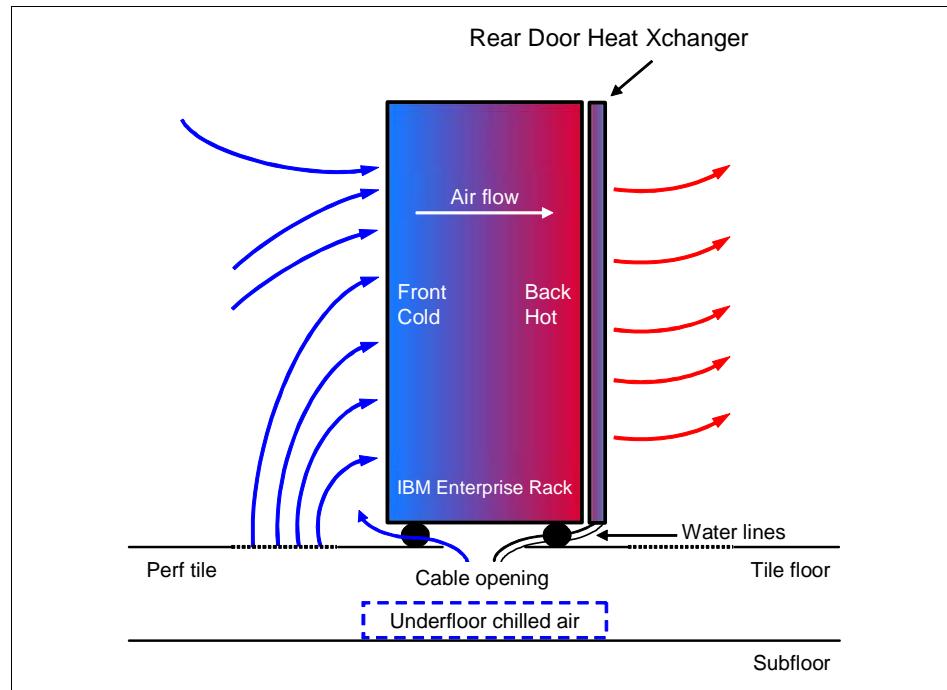


Figure 5-18 Data center rack using an IBM Rear Door Heat eXchanger

With further adjustments, this exchanger can be used to actually cool the room, thus helping to reduce or even eliminate the need for air conditioning in the data center.

As an example, in most cases with the IBM Rear Door Heat eXchanger installed in an iDataPlex rack, outlet temperatures from the back of this iDataPlex rack are up to 10 degrees lower than inlet air (room) temperatures, and it eliminates heat generated by servers using up to 33 KW of power in this rack. Figure 5-19 on page 88 shows the rear door of the iDataPlex and some of its features, including its patented hex airflow design and industry standard hose fittings.

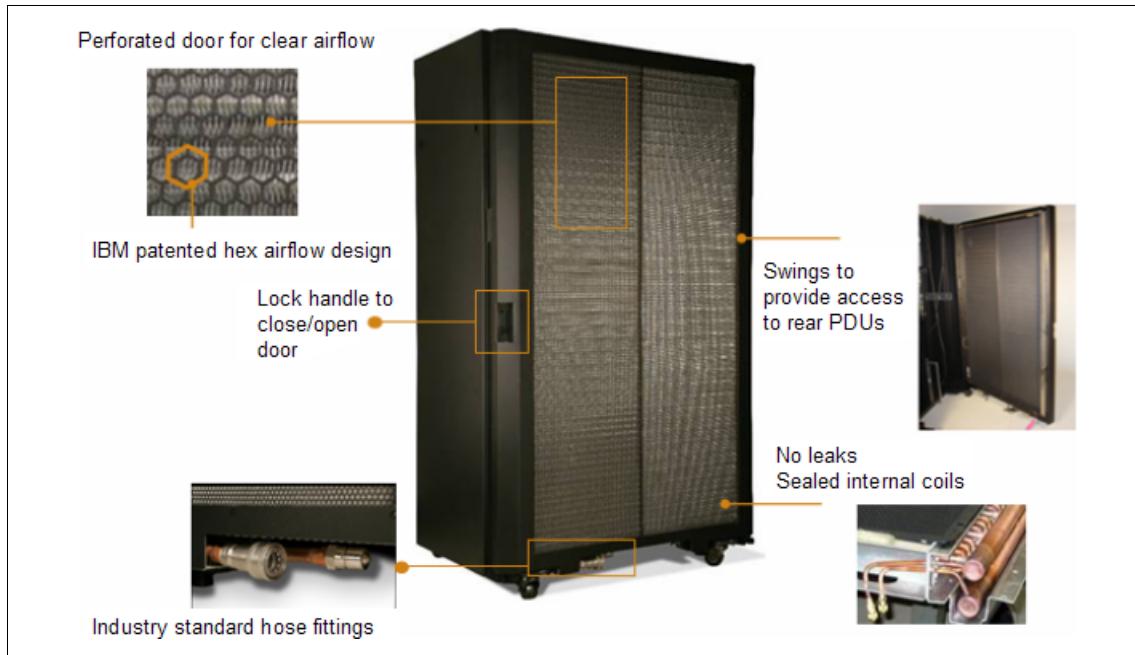


Figure 5-19 iDataPlex Rear Door Heat eXchanger for System x iDataPlex Rack

You can view a brief video that shows a customer using the IBM Rear Door at the following URL:

[http://www-07.ibm.com/systems/includes/content/x/about/media/GA\\_Tech\\_video.mov](http://www-07.ibm.com/systems/includes/content/x/about/media/GA_Tech_video.mov)

## 5.5 Power and performance benchmarks

To assess the relative energy efficiency of servers, a benchmark defined by SPEC is used as an industry standard. In this section, we describe this benchmark and explain the methodology used for an IBM System x server.

### 5.5.1 The benchmark description

SPECpower\_ssj2008 is the first industry-standard SPEC benchmark that evaluates the power and performance characteristics of volume server class computers. With SPECpower\_ssj2008, SPEC defines server power measurement standards in the same way as done for performance.

The drive to create a power and performance benchmark is based on the recognition that the IT industry, computer manufacturers, and governments are increasingly concerned with the energy use of servers. Currently, many vendors report some energy efficiency figures, but these are often not directly comparable due to differences in workload, configuration, test environment, and so on. The development of this benchmark provides a means to measure power (at the AC input) in conjunction with a performance metric. This should help IT managers to consider power characteristics along with other selection criteria to increase the efficiency of data centers.

The initial benchmark addresses only one subset of server workloads: the performance of server side Java. It exercises the CPUs, caches, memory hierarchy and the scalability of shared memory processors (SMPs), as well as the implementations of the Java Virtual Machine (JVM), Just-in-Time (JIT) compiler, garbage collection, threads and some aspects of the operating system. Additional workloads are planned.

The benchmark runs on a wide variety of operating systems and hardware architectures. It should not require extensive client or storage infrastructure; see:

[http://www.spec.org/power\\_ssj2008/](http://www.spec.org/power_ssj2008/)

### 5.5.2 The benchmark methodology

When IBM runs that benchmark, it uses the methodology, the workload, and the rules defined by the Standard Performance Evaluation Corp. (SPEC) Power performance committee. This benchmark is CPU- and memory-intensive. It does not exercise disk or network I/O.

After the runs complete, IBM calculates the score by taking the sum of the performance from 100% - 10% and dividing it by the sum of the power consumption from 100% - 10%, as well as idle.

Figure 5-20 on page 90 shows a SPECpower\_ssj2008 benchmark results summary on a IBM x3200 M2.

After all runs have been completed, you have to calculate the score (shown in the Performance to Power Ratio column in Figure 5-20 on page 90) for each target load.

This computation must be performed for 100%, 90% and so on, to active Idle. You divide the performance result (shown in the ssj\_ops column in Figure 5-20 on page 90) by the average power (shown in the Average Power (W) column in Figure 5-20 on page 90). As an example, for a target load at 100%, the performance-to-power ratio is  $185,456/115=1,610$

The final score for this benchmark is the sum of all the ssj\_ops results divided by the sum of all the Average power results.

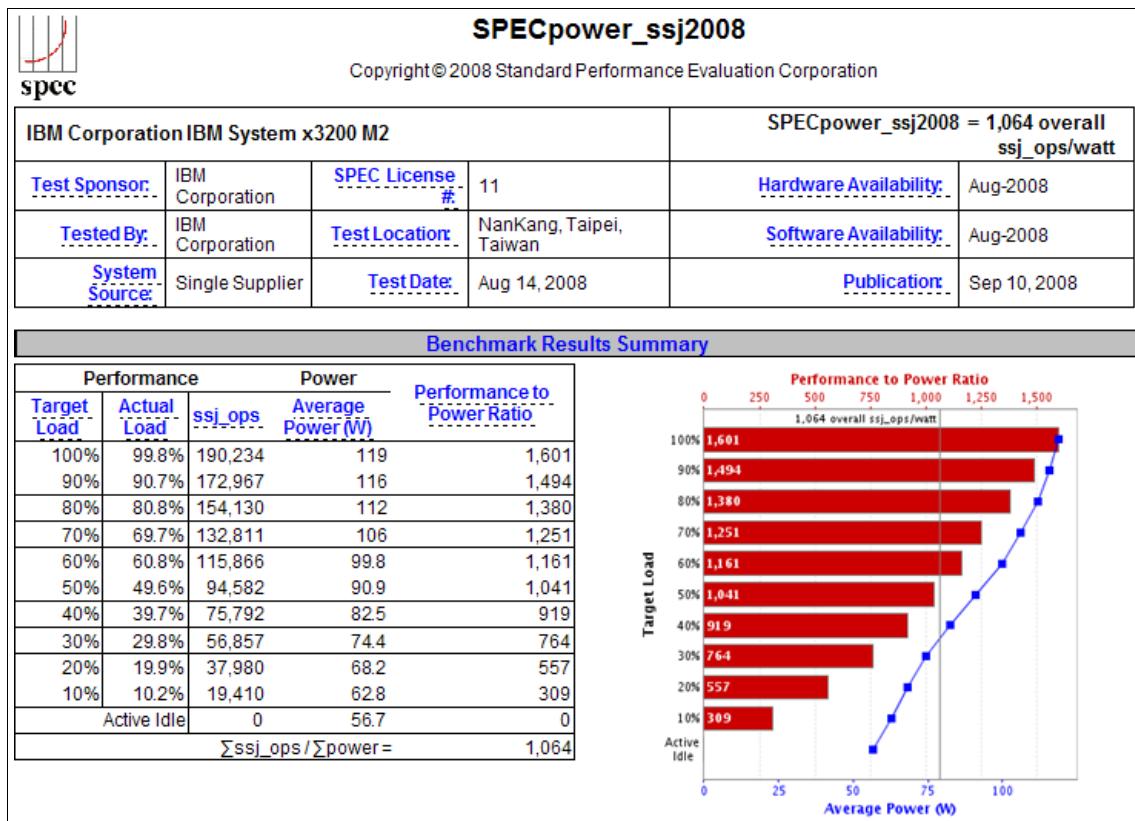


Figure 5-20 SPECpower\_ssj2008 benchmark result on the IBM x3200 M2

Detailed data is reported in a tabular format as well as a graphical representation.

The complete benchmark result for this server is available at the following URL:

[http://www.spec.org/power\\_ssj2008/results/res2008q2/power\\_ssj2008-20080506-00050.html](http://www.spec.org/power_ssj2008/results/res2008q2/power_ssj2008-20080506-00050.html)

## 5.6 Resources

Refer to the following Web resources for guidance regarding energy efficiency:

- ▶ IBM Green:  
[http://ibm.com/systems/optimizeit/cost\\_efficiency/energy\\_efficiency/](http://ibm.com/systems/optimizeit/cost_efficiency/energy_efficiency/)
- ▶ Energy Efficiency Self-Assessment:  
[http://ibm.com/systems/optimizeit/cost\\_efficiency/energy\\_efficiency/services.html](http://ibm.com/systems/optimizeit/cost_efficiency/energy_efficiency/services.html)
- ▶ PDU and UPS offerings:  
[http://www.powerware.com/ibm/US/Products/UPS\\_systemx.asp](http://www.powerware.com/ibm/US/Products/UPS_systemx.asp)
- ▶ Rack and Power Solutions:  
<http://www-304.ibm.com/jct03004c/servers/eserver/xseries/storage/rack.html>
- ▶ Raised floor blog (datacenter efficiency):  
<http://theraisedfloor.typepad.com/blog/>
- ▶ Cool Blue™ Technology Videos:  
<http://www.backhomeproductions.net/ibm/ftp/coolblue/flash/>
- ▶ Power Configurator:  
<http://www.ibm.com/systems/bladecenter/powerconfig/>
- ▶ Active Energy Manager:  
<http://ibm.com/systems/management/director/extensions/powerexec.html>





# Processors and cache subsystem

The central processing unit (CPU or processor) is the key component of any computer system. In this chapter, we cover several different CPU architectures from Intel (IA32, Intel 64 Technology, and IA64) and AMD<sup>1</sup> (AMD64), and outline their main performance characteristics.

This chapter discusses the following topics:

- ▶ 6.1, “Processor technology” on page 94
- ▶ 6.2, “Intel Xeon processors” on page 94
- ▶ 6.3, “AMD Opteron processors” on page 109
- ▶ 6.4, “64-bit computing” on page 116
- ▶ 6.5, “Processor performance” on page 122

**Note:** In this book, we collectively refer to the processors from Intel (IA32, Intel 64 Technology) and AMD (AMD64) as Intel-compatible processors.

Intel 64 Technology is the new name for Extended Memory 64 Technology (EM64T).

<sup>1</sup> Content in this chapter is reprinted by permission of Advanced Micro Devices™, Inc.

## 6.1 Processor technology

The central processing unit (CPU) has outperformed all other computer subsystems in its evolution. Thus, most of the time, other subsystems such as disk or memory will impose a bottleneck upon your application (unless pure number crunching or complex application processing is the desired task). Understanding the functioning of a processor in itself is already quite a difficult task, but today IT professionals are faced with multiple and often very different CPU architectures.

Comparing different processors is no longer a matter of looking at the CPU clock rate, but is instead a matter of understanding which CPU architecture is best suited for handling which kind of workload. Also, 64-bit computing has finally moved from high-end UNIX® and mainframe systems to the Intel-compatible arena, and has become yet another new technology to be understood.

The Intel-compatible microprocessor has evolved from the first 8004 4-bit CPU, produced in 1971, to the current line of Xeon and Core processors. AMD offers the world's first IA32 compatible 64-bit processor. Our overview of processors begins with the current line of Intel Xeon CPUs, followed by the AMD Opteron and the Intel Core™ Architecture. For the sake of simplicity, we do not explore earlier processors.

## 6.2 Intel Xeon processors

Moore's Law states that the number of transistors on a chip doubles about every two years. Similarly, as transistors have become smaller, the frequency of the processors have increased, which is generally equated with performance.

However, around 2003, physics started to limit advances in obtainable clock frequency. Transistor sizes have become so small that electron leakage through transistors has begun to occur. Those electron leaks result in large power consumption and substantial extra heat, and could even result in data loss. In addition, cooling processors at higher frequencies by using traditional air cooling methods has become too expensive.

This is why the material that comprises the dielectric in the transistors has become a major limiting factor in the frequencies that are obtainable. Manufacturing advances have continued to enable a higher per-die transistor count, but have only been able to obtain about a 10% frequency improvement per year. For that reason, processor vendors are now placing more processors on the die to offset the inability to increase frequency. Multi-core processors provide the ability to increase performance with lower power consumption.

### 6.2.1 Dual-core processors

Intel released its first dual core Xeon processors in October 2005. Dual core processors are two separate physical processors combined onto a single processor socket. Dual core processors consist of twice as many cores, but each core is run at lower clock speeds as an equivalent single core chip to lower the waste heat usage. *Waste heat* is the heat that is produced from electron leaks in the transistors.

Recent dual-core Xeon processor models that are available in IBM System x servers:

- ▶ Xeon 7100 Series MP processor (Tulsa)

Tulsa follows the Xeon MP line and is the follow-on to the Paxville MP processor. Tulsa is similar in architecture to Paxville with the main exception that it includes a shared L3 cache and is built on a 65 nm technology. Previous dual core Intel processors did not include an L3 cache.

The key benefits of the Tulsa processor include:

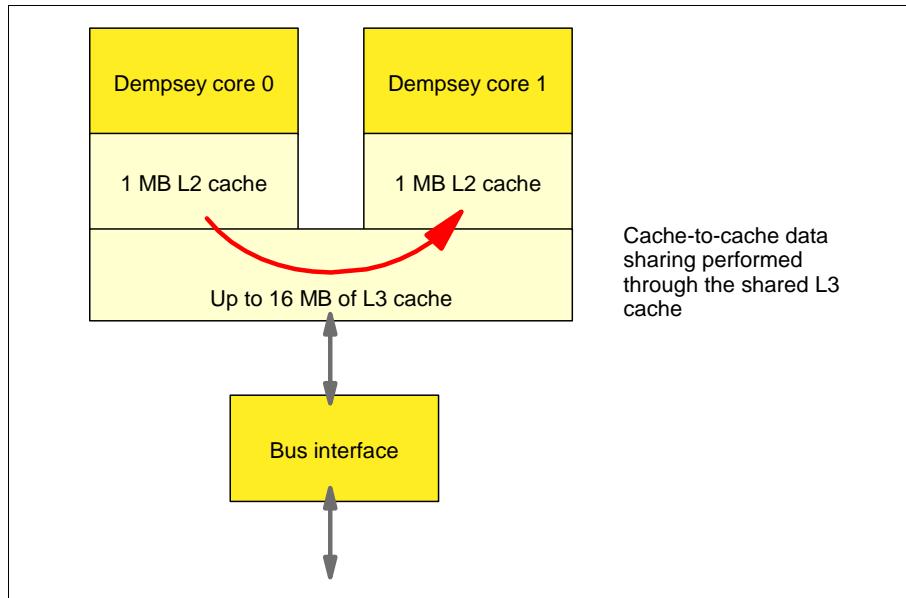
- Frequencies of 2.5 - 3.4 GHz, which provides a greater selection of processor frequencies than the Paxville MP processor.
- 1 MB L2 cache in each core and a shared L3 cache ranging in size from 4 MB to 16 MB. The L3 cache is shared, instead of having a separate one in each core.
- Front-side buses of 677 or 800 MHz.

Figure 6-1 on page 96 illustrates the overall structure of the Tulsa processor.

The Tulsa processor includes two Dempsey cores, each with a 1 MB L2 cache. L3 cache is much faster than the previous Potomac processor's L3 cache. The Potomac processor's L3 cache experienced high latencies in determination of a cache miss, which is improved greatly in the Tulsa L3 cache.

One major issue that occurs with early Intel dual core processors is that the cores within the same processor socket are unable to communicate directly to each other internally. Instead, cores within the same processor use the external front-side bus to transmit data between their individual caches.

Tulsa processors incorporate a shared L3 cache between cores. Because both cores share the same L3 cache, core-to-core data communication can occur internally within the processor instead of externally on the front-side bus. Traffic that occurred previously on the front-side bus is now moved internally into the processor, which frees up front-side bus traffic.



*Figure 6-1 Tulsa processor architecture*

- ▶ Xeon 5100 Series DP processor (Woodcrest)

The Woodcrest processor is the first Xeon DP processor that uses the Intel Core microarchitecture instead of the Netburst microarchitecture. See 6.2.4, “Intel Core microarchitecture” on page 103 for details.

Frequencies of 1.6 to 3.0 GHz are supported with an L2 cache of 4 MB. The front-side bus runs at a frequency of either 1066 or 1333 MHz, as shown in Table 6-1. None of these processors support Hyper-Threading.

Woodcrest uses a low power model incorporated in the Core microarchitecture. This is an improvement of the 95W to 130W power consumption of its predecessor, Dempsey. In addition to the substantial performance-per-watt increases, the Core microarchitecture of the Woodcrest processor provides substantial improvements in random memory throughput applications.

*Table 6-1 Woodcrest processor models*

Processor model	Speed	L2 cache	Front-side bus	Power (TDP)
Xeon 5110	1.6 GHz	4 MB	1066 MHz	65 W
Xeon 5120	1.86 GHz	4 MB	1066 MHz	65 W
Xeon 5130	2.00 GHz	4 MB	1333 MHz	65 W

Processor model	Speed	L2 cache	Front-side bus	Power (TDP)
Xeon 5140	2.33 GHz	4 MB	1333 MHz	65 W
Xeon 5148 LV	2.33 GHz	4 MB	1333 MHz	40 W
Xeon 5150	2.66 GHz	4 MB	1333 MHz	65 W
Xeon 5160	3.0 GHz	4 MB	1333 MHz	80 W

- Xeon 5200 Series DP processor (Wolfdale)

The Wolfdale dual-core processor is based on the new 45 nm manufacturing process. It features SSE4, which provides expanded computing capabilities over its predecessor. With the Intel Core microarchitecture and Intel EM64T, the Wolfdale delivers superior performance and energy efficiency to a broad range of 32-bit and 64-bit applications.

For specific models, see Table 6-2.

*Table 6-2 Wolfdale processor models*

Processor model	Speed	L2 cache	Front-side bus	Power (TDP)
E5205	1.86 GHz	6 MB	1066 MHz	65 W
L5238	2.66 GHz	6 MB	1333 MHz	35 W
X5260	3.33 GHz	6 MB	1333 MHz	80 W
X5272	3.40 GHz	6 MB	1600 MHz	80 W
X5270	3.50 GHz	6 MB	1333 MHz	80 W

- Xeon 7200 Series MP processor (Tigerton)

Tigerton comes with 7200 series 2-core and 7300 series 4-core options. For detailed information about this topic, refer to Xeon 7300 Tigerton in 6.2.2, “Quad-core processors” on page 98.

- Xeon 5500 (Gainestown)

The Intel 5500 series, with Intel Nehalem Microarchitecture, brings together a number of advanced technologies for energy efficiency, virtualization, and intelligent performance. Interaged with Intel QuickPath Technology, Intelligent Power Technology and Intel Virtualization Technology, the Gainestown is available with a range of features for different computing demands. Most of the models for Gainestown are quad-core. Refer to Table 6-6 on page 101 for specific models.

## 6.2.2 Quad-core processors

Quad-core processors differ from single-core and dual-core processors by providing four independent execution cores. Although some execution resources are shared, each logical processor has its own architecture state with its own set of general purpose registers and control registers to provide increased system responsiveness. Each core runs at the same clock speed.

Intel quad-core and six-core processors include the following:

- ▶ Xeon 5300 Series DP processor (Clovertown)

The Clovertown processor is a quad-core design that is actually made up of two Woodcrest dies in a single package. Each Woodcrest die has 4 MB of L2 cache, so the total L2 cache in Clovertown is 8 MB.

The Clovertown processors are also based on the Intel Core microarchitecture as described in 6.2.4, “Intel Core microarchitecture” on page 103.

Processor models available include the E5310, E5320, E5335, E5345, and E5355. The processor front-side bus operates at either 1066 MHz (processor models ending in 0) or 1333 MHz (processor models ending in 5). For specifics, see Table 6-3. None of these processors support Hyper-Threading.

In addition to the features of the Intel Core microarchitecture, the features of the Clovertown processor include:

- Intel Virtualization Technology - processor hardware enhancements that support software-based virtualization.
- Intel 64 Architecture (EM64T) - support for both 64-bit and 32-bit applications.
- Demand-Based Switching (DBS) - technology that enables hardware and software power management features to lower average power consumption of the processor while maintaining application performance.
- Intel I/O Acceleration Technology (I/OAT) - reduces processor bottlenecks by offloading network-related work from the processor.

*Table 6-3 Clovertown processor models*

Processor model	Speed	L2 cache	Front-side bus	Power (TDP)	Demand-based switching
E5310	1.60 GHz	8 MB	1066 MHz	80 W	No
E5320	1.86 GHz	8 MB	1066 MHz	80 W	Yes
E5335	2.00 GHz	8 MB	1333 MHz	80 W	No

Processor model	Speed	L2 cache	Front-side bus	Power (TDP)	Demand-based switching
E5345	2.33 GHz	8 MB	1333 MHz	80 W	Yes
E5355	2.66 GHz	8 MB	1333 MHz	120 W	Yes

- ▶ Xeon 7300 Series MP processor (Tigerton)

The Xeon 7300 Tigerton processor is the first quad-core processor that Intel offers for the multi-processor server (7xxx series) platform. It is based on the Intel Core microarchitecture described in 6.2.4, “Intel Core microarchitecture” on page 103.

The Tigerton comes with 2-core and 4-core options. Xeon 7200 Tigerton dual-core processors are a concept similar to a two-way SMP system except that the two processors, or *cores*, are integrated into one silicon die. This brings the benefits of two-way SMP with lower software licensing costs for application software that licenses per CPU socket. It also brings the additional benefit of less processor power consumption and faster data throughput between the two cores. To keep power consumption down, the resulting core frequency is lower, but the additional processing capacity means an overall gain in performance.

Each Tigerton core has separate L1 instruction and data caches, as well as separate execution units (integer, floating point, and so on), registers, issue ports, and pipelines for each core. A multi-core processor achieves more parallelism than Hyper-Threading technology because these resources are not shared between the two cores.

The Tigerton processor series is available in a range of features to match different computing demands. Up to 8 MB L2 Cache and 1066 MHz front-side bus frequency are supported, as listed in Table 6-4. All processors integrate the APIC Task Programmable Register, which is a new Intel VT extension that improves interrupt handling and further optimizes virtualization software efficiency.

For specific processor SKUs, see Table 6-4.

Table 6-4 Tigerton processor models

Processor model	Speed	L2 Cache	Front-side bus	Power (TDP)
E7310	1.60 GHz	4 MB	1066 MHz	80 W
L7345	1.86 GHz	8 MB	1066 MHz	80 W
E7320	2.13 GHz	4 MB	1066 MHz	80 W

Processor model	Speed	L2 Cache	Front-side bus	Power (TDP)
E7330	2.40 GHz	6 MB	1066 MHz	80 W
X7350	2.93 GHz	8 MB	1066 MHz	130 W

- ▶ Xeon 5400 Series DP processor (Harpertown)

The Harpertown is based on the new 45 nm manufacturing process and features up to a 1600 MHz front-side bus clock rate and 12 MB L2 cache. Harpertown includes new Intel Streaming SIMD Extensions 4 (SSE4) instructions, thus providing building blocks for delivering expanded capabilities for many applications. The new 45nm enhanced Intel Core microarchitecture delivers more performance per watt in the same platforms.

For specific processor SKUs, see Table 6-5.

*Table 6-5 Harpertown processor models*

Processor model	Speed	L2 Cache	Front-side bus	Power (TDP)
E5405	2.00 GHz	12 MB	1333 MHz	80 W
E5420	2.50 GHz	12 MB	1333 MHz	80 W
L5430	2.66 GHz	12 MB	1333 MHz	50 W
E5440	2.83 GHz	12 MB	1333 MHz	80 W
X5450	3.00 GHz	12 MB	1333 MHz	120 W
X5460	3.16 GHz	12 MB	1333 MHz	120 W
X5482	3.20 GHz	12 MB	1600 MHz	150 W

- ▶ Xeon 7400 Series MP processor (Dunnington)

Dunnington comes with 4-core and 6-core options. For more detailed information about this topic, refer to 6.2.3, “Six-core processors” on page 101.

- ▶ Xeon 5500 Series DP processor (Gainestown)

The Intel 5500 series is based on the Nehalem microarchitecture. As the follow-on to the 5200/5400 (Wolfdale/ Harpertown), the Gainestown offers the models listed in Table 6-6 on page 101.

Table 6-6 Gainestown processor models

Processor model	Speed	L3 cache	QPI link speed (giga transfer/s)	Power (TDP)
W5580	3.20 GHz	8 MB	6.4 GT/s	130 W
X5570	2.93 GHz	8 MB	6.4 GT/s	95 W
X5560	2.80 GHz	8 MB	6.4 GT/s	95 W
X5550	2.66 GHz	8 MB	6.4 GT/s	95 W
E5540	2.53 GHz	8 MB	5.86 GT/s	80 W
E5530	2.40 GHz	8 MB	5.86 GT/s	80 W
L5520	2.26 GHz	8 MB	5.86 GT/s	60 W
E5520	2.26 GHz	8 MB	5.86 GT/s	80 W
L5506	2.13 GHz	4 MB	4.8 GT/s	60 W
E5506	2.13 GHz	4 MB	4.8 GT/s	80 W
E5504	2.00 GHz	4 MB	4.8 GT/s	80 W
E5502	1.86 GHz	4 MB	4.8 GT/s	80 W

### 6.2.3 Six-core processors

Six-core processors extend the quad-core paradigm by providing six independent execution cores.

- ▶ Xeon 7400 Series MP processor (Dunnington)

With enhanced 45 nm process technology, the Xeon 7400 series Dunnington processor features a single-die 6-core design with 16 MB of L3 cache. Both 4-core and 6-core models of Dunnington have shared L2 cache between each pair of cores. They also have a shared L3 cache across all cores of the processor. A larger L3 cache increases efficiency of cache-to-core data transfers and maximizes main memory-to-processor bandwidth. Specifically built for virtualization, this processor comes with enhanced Intel VT, which greatly optimizes virtualization software efficiency.

As Figure 6-2 on page 102 illustrates, Dunnington processors have three levels of cache on the processor die:

- L1 cache

The L1 execution, 32 KB instruction and 32 KB data for data trace cache in each core is used to store micro-operations (decoded executable machine

instructions). It serves those to the processor at rated speed. This additional level of cache saves decode time on cache hits.

- L2 cache

Each pair of cores in the processor has 3 MB of shared L2 cache, for a total of 6 MB, or 9 MB of L2 cache. The L2 cache implements the Advanced Transfer Cache technology.

- L3 cache

The Dunnington processors have 12 MB (4-core), or 16 MB (6-core) shared L3 cache.

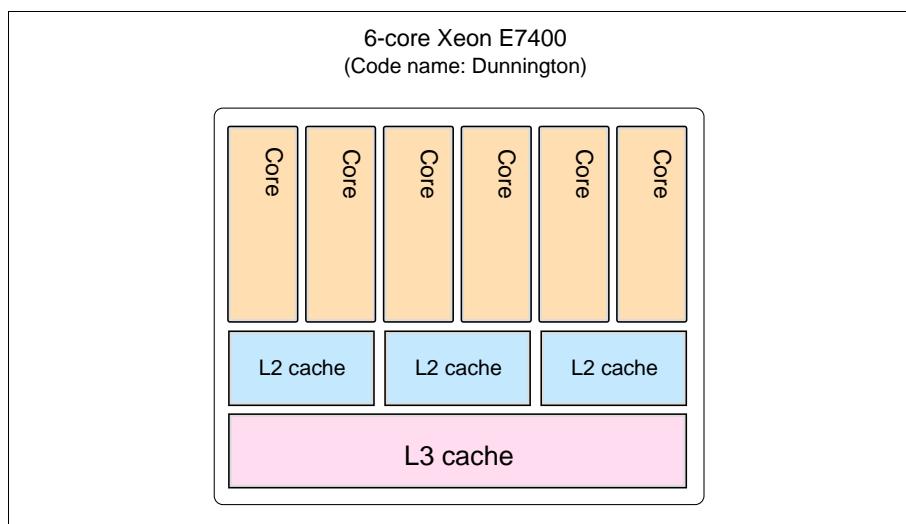


Figure 6-2 Block diagram of Dunnington 6-core processor package

Dunnington comes with 4-core and 6-core options; for specific models see Table 6-7.

Table 6-7 Dunnington processor models (quad-core and 6-core)

Processor model	Cores per processor	Speed	L3 Cache	Front-side bus	Power (TDP)
L7445	4	2.13 GHz	12 MB	1066 MHz	50 W
E7420	4	2.13 GHz	8 MB	1066 MHz	90 W
E7430	4	2.13 GHz	12 MB	1066 MHz	90 W
E7440	4	2.40 GHz	16 MB	1066 MHz	90 W
L7455	6	2.13 GHz	12 MB	1066 MHz	65 W

Processor model	Cores per processor	Speed	L3 Cache	Front-side bus	Power (TDP)
E7450	6	2.40 GHz	12 MB	1066 MHz	90 W
X7460	6	2.66 GHz	16 MB	1066 MHz	130 W

### 6.2.4 Intel Core microarchitecture

The Intel Core microarchitecture is based on a combination of the energy-efficient Pentium® M microarchitecture found in mobile computers, and the current Netburst microarchitecture that is the basis for the majority of the Xeon server processors. The Woodcrest processor is the first processor to implement the Core microarchitecture.

The key features of the Core microarchitecture include:

- ▶ Intel Wide Dynamic Execution

The Core microarchitecture is able to fetch, decode, queue, execute, and retire up to four instructions simultaneously in the pipeline. The previous Netburst Microarchitecture was only able to run three instructions simultaneously in the pipeline. The throughput is improved effectively by processing more instructions in the same amount of time.

In addition, certain individual instructions are able to be combined into a single instruction in a technique known as *macrofusion*. By being combined, more instructions can fit within the pipeline. To use the greater pipeline throughput, the processors have more accurate branch prediction technologies and larger buffers, thus providing less possibility of pipeline stalls and more efficient use of the processor.

- ▶ Intel Intelligent Power Capability

The advantage of the Pentium M™ microarchitecture is the ability to use less power and enable longer battery life in mobile computers. Similar technology has been improved and modified and added into the Core microarchitecture for high-end computer servers.

The Intelligent Power Capability provides fine-grained power control that enables sections of the processor that are not in use to be powered down. Additional logic is included in components such as the ALU unit, FP unit, cache logic, and bus logic that improves power consumption on almost an instruction-by-instruction basis. Processor components can then be powered on the instant they are needed to process an instruction, with minimal lag time so that performance is not jeopardized. Most importantly, the actual power utilization is substantially lower with the Core microarchitecture due to this additional power control capability.

► Intel Advanced Smart Cache

The L2 cache in the Core microarchitecture is shared between cores instead of each core using a separate L2. Figure 6-3 illustrates the difference in the cache between the traditional Xeon with Netburst microarchitecture and the Intel Core microarchitecture.

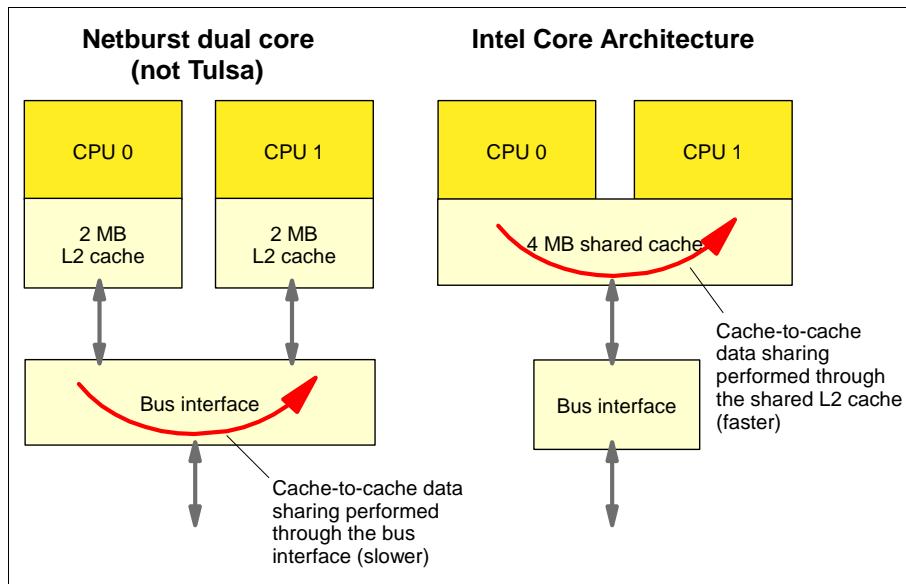


Figure 6-3 Intel Xeon versus Intel Core Architecture

The front-side bus utilization would be lower, similar to the Tulsa L3 shared cache as discussed in “Xeon 7100 Series MP processor (Tulsa)” on page 95. With a dual-core processor, the Core microarchitecture allows for one, single core to use the entire shared L2 cache if the second core is powered down for power-saving purposes. As the second core begins to ramp up and use memory, it will allocate the L2 memory away from the first CPU until it reaches a balanced state where there is equal use between cores.

Single core performance benchmarks, such as SPEC Int and SPEC FP, benefit from this architecture because single core applications are able to allocate and use the entire L2 cache. SPEC Rate benchmarks balance the traffic between the cores and more effectively balance the L2 caches.

► Intel Smart Memory Access

Intel Smart Memory Access allows for additional technology in the processor to prefetch more often, which can assist performance. Previously, with the NetBurst architecture, when a write operation appeared in the pipeline, all subsequent read operations would stall until the write operation completed. In

that case, prefetching would halt, waiting for the write operation to complete, and the pipeline would fill with **nop** or **stall** commands instead of productive commands. Instead of making forward progress, the processor would make no progress until the write operation completed.

Using Intel's memory disambiguation technologies, additional load-to-memory operations are executed prior to a store operation completing. If the load instruction turns out to be incorrect, the processor is able to back out the load instruction and all dependent instructions that might have executed based on that load. However, if the load instruction turns out to be valid, the processor has spent less time waiting and more time executing, which improves the overall instruction level parallelism.

- ▶ Intel Advanced Digital Media Boost

Advanced Digital Media Boost increased the execution of SSE instructions from 64 bits to 128 bits. SSE instructions are Streaming Single Instruction Multiple Data Instructions that characterize large blocks of graphics or high bandwidth data applications. Previously, these Instructions would need to be broken into two 64-bit chunks in the execution stage of the processor, but now support has been included to have those 128-bit instructions executed at one per clock cycle.

For more information about the Core microarchitecture, go to:

<http://www.intel.com/technology/architecture/coremicro>

### 6.2.5 Intel Nehalem microarchitecture

The new Nehalem is built on the Core microarchitecture and incorporates significant system architecture and memory hierarchy changes. Being scalable from 2 to 8 cores, the Nehalem includes the following features:

- ▶ QuickPath Interconnect (QPI)

QPI, previously named Common System Interface or CSI, acts as a high-speed interconnection between the processor and the rest of the system components, including system memory, various I/O devices, and other processors. A significant benefit of the QPI is that it is point-to-point. As a result, no longer is there a single bus that all processors must connect to and compete for in order to reach memory and I/O. This improves scalability and eliminates the competition between processors for bus bandwidth.

Each QPI link is a point-to-point, bi-directional interconnection that supports up to 6.4 GTps (giga transfers per second). Each link is 20 bits wide using differential signaling, thus providing bandwidth of 16 GBps. The QPI link carries QPI packages that are 80 bits wide, with 64 bits for data and the remainder used for communication overhead. This gives a 64/80 rate for valid data transfer; thus, each QPI link essentially provides bandwidth of

12.8 GBps, and equates to a total of 25.6 GBps for bi-directional interconnection.

- ▶ **Integrated Memory Controller**

The Nehalem replaces front-side bus memory access with an integrated memory controller and QPI. Unlike the older front-side bus memory access scheme, this is the Intel approach to implementing the scalable shared memory architecture known as non-uniform memory architecture (NUMA) that we introduced with the IBM eX4 “Hurricane 4” memory controller on the System x3950 M2.

As a part of Nehalem’s scalable shared memory architecture, Intel integrated the memory controller into each processor’s silicon die. With that, the system can provide independent high bandwidth, low latency local memory access, as well as scalable memory bandwidth as the number of processors is increased. In addition, by using Intel QPI, the memory controllers can also enjoy fast efficient access to remote memory controllers.

- ▶ **Three-level cache hierarchy**

Comparing to its predecessor, Nehalem’s cache hierarchy extends to three levels; see Figure 6-4. The first two levels are dedicated to individual cores and stay relatively small. The third level cache is much larger and is shared among all cores.

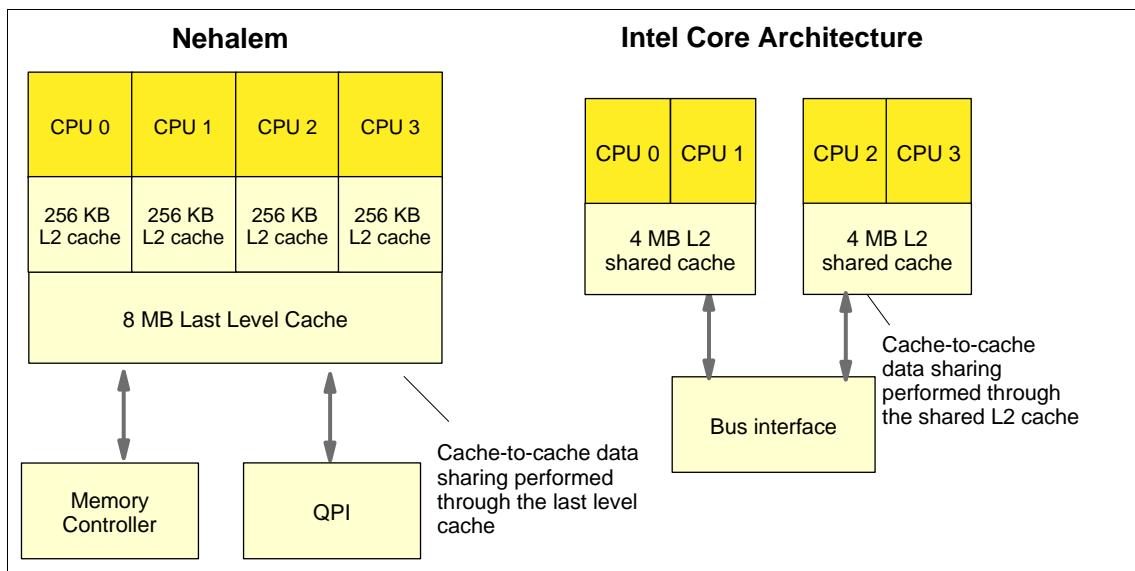


Figure 6-4 Intel Nehalem versus Intel Core

Note the following points:

- Individual 256 KB L2 memory caches for each core
  - Each core of Nehalem has a dedicated 256 KB L2 cache with 8-way associativity, providing very low latency access to data and instructions.
  - Nehalem uses individual L2 caches for each core. It adds the third level cache (Last Level) to be shared and to manage communications between cores.
- New large 8 MB fully-shared Last Level Cache
  - Nehalem's 8 MB, 16-way associative Last Level Cache is shared between all cores and is inclusive of all L1 and L2 cache data, as shown in Figure 6-4 on page 106. The benefit offered by an inclusive cache is that almost all coherency traffic can be handled at the Last Level Cache, and it acts like a snoop filter for all cache hits. Although this has the disadvantage of wasting cache space with the duplication, much more bandwidth is left for actual data in the caches.
- ▶ Second translation look-aside buffer (TLB) and branch target buffer (BTB)
  - The TLB is a CPU cache that is used to improve the speed of virtual address translation. Nehalem adds a new low-latency unified second level TLB that can store up to 512 entries, thereby increasing the performance of applications that use large sets of data.
  - The branch predictor is responsible for predicting the next instructions of a program to be fetched and loaded to CPU. This process, to some extent, avoids wasting time having the CPU load instructions from memory. BTB is a part of the branch predictor that stores the results of the branches as execution progresses. Nehalem improves this process over its predecessor by using two levels of BTB, improving predict accuracy for applications with large code size (such as a database application). Through more accurate prediction, higher performance and lower power consumption are achieved.
- ▶ Simultaneous Multi-Threading (SMT)
  - For the most part, SMT functions the same as Intel Hyper-Threading Technology which was introduced in the Intel NetBurst® microarchitecture. Both are a means of executing two separate code stream (threads) concurrently. Nevertheless, SMT is more efficient as implemented in Nehalem due to a shorter pipeline, larger and lower latency caches, and wider execution engine to enhance parallelism.
  - There are also more resources available to aid SMT than with its predecessor. For example, to accommodate SMT in Nehalem, the reservation station that dispatches operations to execution units has increased from 32 to 36 entries. The load buffer and the store buffer have increased to 48 and 32

entries, respectively. All of these combine to make SMT in Nehalem more efficient and higher performing than Hyper-Threading.

- ▶ Macrofusion and Loop Stream Detector

- Macrofusion

As introduced in 6.2.4, “Intel Core microarchitecture” on page 103, Macrofusion is a feature that improves performance and lowers CPU power consumption by combining certain individual instructions into just one instruction. Nehalem improves this ability in two ways to make higher performance and greater power efficiency possible. First, it expands by providing additional support for several branching instructions that could not be combined on its predecessor. And second, it can now be applied in both 32-bit and 64-bit mode.

- Loop Stream Detector (LSD)

Loops are very common in most software when the same instruction is executed a given number of times. In a CPU pipeline, each instruction would normally go through routine processes like predicting, fetching, decoding, and so forth. In a loop, as many of the same instructions feed the pipeline, correspondingly the same predicting, fetching and decoding processes occur again and again.

However, those processes do not need to be executed over and over again to generate the same result repeatedly in such loop scenario. LSD identifies the loop and idles these processes during the loop, thereby saving power and boosting performance. Previous implementations of LSD in Core architecture disable fetching and predicting during a loop so that many of the same instructions in the loop only have fetching and predicting performed once. In Nehalem, LSD disables another unneeded process, which is decoding during the loop. This provides more power savings and performance gain.

## Naming convention

Intel processors use this naming convention to group the processor types:

- ▶ EN: entry-level products
- ▶ EP: energy efficient, high performance server products (similar to the previous DP family of processors)
- ▶ EX: high-end expandable server products (similar to the previous MP family)
- ▶ MC: mission-critical server products based on the Itanium® architecture

## 6.3 AMD Opteron processors

The first Opteron processor was introduced in 2003 after a major effort by AMD to deliver the first 64-bit processor capable of running in 64-bit mode and running existing 32-bit software as fast as or faster than current 32-bit Intel processors. Opteron was designed from the start to be an Intel-compatible processor, but AMD also wanted to introduce advanced technology that would set Opteron apart from processors developed by Intel.

The Opteron processor has a physical address limit of 40-bit addressing (meaning that up to 1 TB of memory can be addressed) and the ability to be configured in 2-socket and 4-socket multi-processor configurations.

Apart from improved scalability, there was another significant feature introduced with the Opteron CPU, namely the 64-bit addressing capability of the processor. With the Itanium processor family, Intel introduced a completely new 64-bit architecture that was incompatible with existing software. AMD decided there was significant market opportunity for running existing software and upgraded the existing x86 architecture to 64-bit.

As with the transitions that the x86 architecture underwent in the years before both the move to 16-bit with the 8086 and the move to 32-bit with the Pentium architecture, the new AMD64 architecture simply expanded the IA32 architecture by adding support for new 64-bit addressing, registers, and instructions.

The advantage of this approach is the ability to move to 64-bit without having to perform a major rewrite of all operating system and application software. The Opteron CPU has three distinct operation modes that enable the CPU to run in either a 32-bit mode or a 64-bit mode, or a mixture of both. This feature gives users the ability to transition smoothly to 64-bit computing. The AMD64 architecture is discussed in more detail in “64-bit extensions: AMD64 and Intel 64 Technology” on page 117.

### 6.3.1 AMD Revision F (1207 socket) Opteron

The current AMD processors, Revision F, have the following features:

- ▶ All Revision F processors are multi-core.
- ▶ Support for DDR2 memory is added. The following number of memory DIMM slots are incorporated into the system:
  - Eight DIMMs at DDR2 speeds of 400 or 533 MHz instead of 266/333 MHz DDR1.
  - Four DIMMs at 667 MHz DDR2 instead of 400 MHz DDR1.

- Two DIMMs eventually will be supported at 800 MHz DDR2 instead of 400 MHz DDR1.
- ▶ The current 1 GHz HyperTransport™ technology is incorporated as the interface to initial Rev F processors.
- ▶ PCI Express support is added.
- ▶ AMD-V™ virtualization technology and power management technologies are incorporated.

Figure 6-5 shows the architecture of the Opteron Revision F processor.

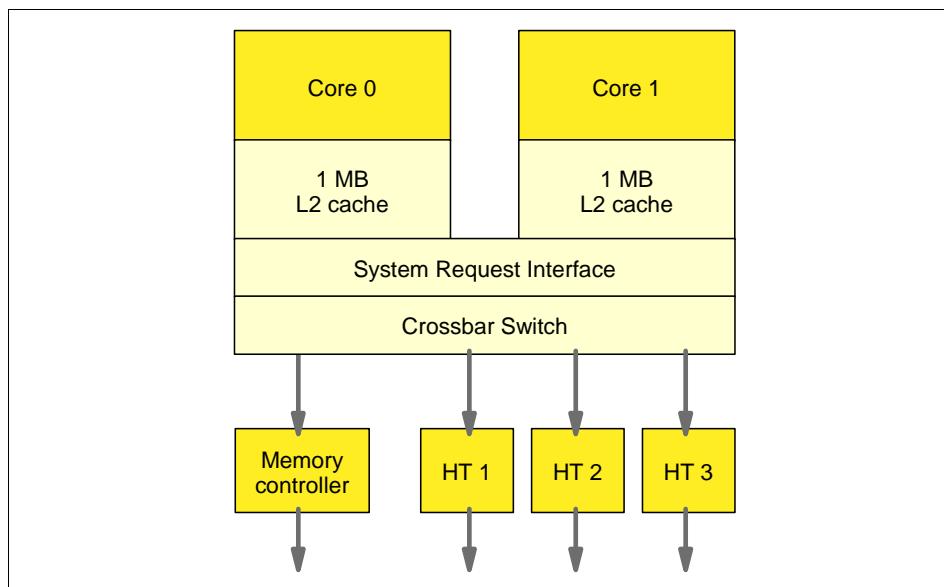


Figure 6-5 Architecture of dual core Rev F processor

Internally, each core of the processor is connected directly to a 1 MB L2 cache. Memory transfers between the two caches on the processors occur directly through a crossbar switch, which is on the chip. By direct communication between the L2 caches on different cores, the HyperTransport is not used for processor core-to-core communication. The bandwidth of the HyperTransport can then be used to communicate with other devices or processors.

The chipset that interfaces with the AMD Opteron processors as well as the HyperTransport links are discussed in more detail in 9.4, “PCI bridge-based chipsets” on page 177.

### 6.3.2 AMD quad-core Barcelona

Barcelona is the third-generation AMD Opteron processor, and it extends the design of the Revision F processor. It features a native multi-core design where all four cores are on one piece of silicon, as opposed to packaging two dual-core die together into one single processor. With several significant enhancements over previous generations, Barcelona provides increased performance and energy efficiency.

Features of the Barcelona processor include:

- ▶ AMD Memory Optimizer Technology increases memory throughput greatly compared to previous generations of AMD Opteron processors.
- ▶ AMD Wide Floating Point Accelerator provides 128-bit SSE floating point capabilities, which enables each core to simultaneously execute up to four floating point operations per clock for significantly performance improvement.
- ▶ AMD Balanced Smart Cache represents significant cache enhancements with 512 KB L2 cache per core and 2 MB shared L3 cache across all four cores.
- ▶ AMD CoolCore technology can reduce energy consumption by turning off unused parts of the processor.
- ▶ Independent Dynamic Core Technology enables variable clock frequency for each core, depending on the specific performance requirement of the applications it is supporting, thereby helping to reduce power consumption.
- ▶ Dual Dynamic Power Management (DDPM) technology provides an independent power supply to the cores and to the memory controller, allowing the cores and memory controller to operate on different voltages, depending on usage.

The AMD Opteron processors are identified by a four-digit model number in the form ZYXX, with the following meaning:

- ▶ Z indicates the maximum scalability of the processor.
  - 1000 Series = 1-way servers and workstations
  - 2000 Series = Up to 2-way servers and workstations
  - 8000 Series = Up to 8-way servers and workstations
- ▶ Y indicates socket generation. This digit is always 3 for third-generation AMD Opteron processors for Socket F (1207) and Socket AM2.
  - 1000 Series in Socket AM2 = Models 13XX
  - 2000 Series in Socket F (1207) = Model 23XX
  - 8000 Series in Socket F (1207) = Model 83XX

- ▶ XX indicates relative performance within the series.  
XX values above 40 indicate a third-generation quad-core AMD Opteron processor.

### 6.3.3 AMD quad-core Shanghai

Shanghai is next generation AMD processor that will be produced on 45nm process. Shanghai will have a third iteration of HyperTransport and likely to have three times more cache than Barcelona, with improved performance and power characteristics.

Key features of Shanghai will include:

- ▶ Four-cores per CPU, 512K L2/core cache, 6 MB shared L3 cache
- ▶ HyperTransport Technology 3.0, at up to 16.0 GBps (4 GT/s)
- ▶ 128-bit floating-point unit per core, 4 FLOPS/clk peak per core
- ▶ Two memory channels using registered DDR2-800
- ▶ Offered in the following thermal bands: 55 W for HE, 75 W for standard, 105 W for SE

### 6.3.4 Opteron split-plane

Split-plane, also referred to as Dual Dynamic Power Management (DDPM) technology, was introduced in the Barcelona processor as described in 6.3.2, “AMD quad-core Barcelona” on page 111. Figure 6-6 on page 113 illustrates that in a split plane implementation, power delivery is separated between the cores and the integrated memory controller, as opposed to a unified power plane design. The new and flexible split-plane design allows power to be dynamically allocated as needed, thus making more efficient use of power and boosting performance.

For instance, when the core is executing a CPU-intensive computation, the memory controller can adjust the power requirement to lower the power consumption for more efficiency. Conversely, when needed to boost memory performance, the memory controller can request increased power supply to obtain higher frequency and achieve lower memory latency.

Because the power is coming from the system board, refreshing the system board design is required to support a split-plane power scheme implementation. For the best level of compatibility, however, a split-plane CPU will work well with the feature disabled while installed in a system board that does not support the split-plane implementation.

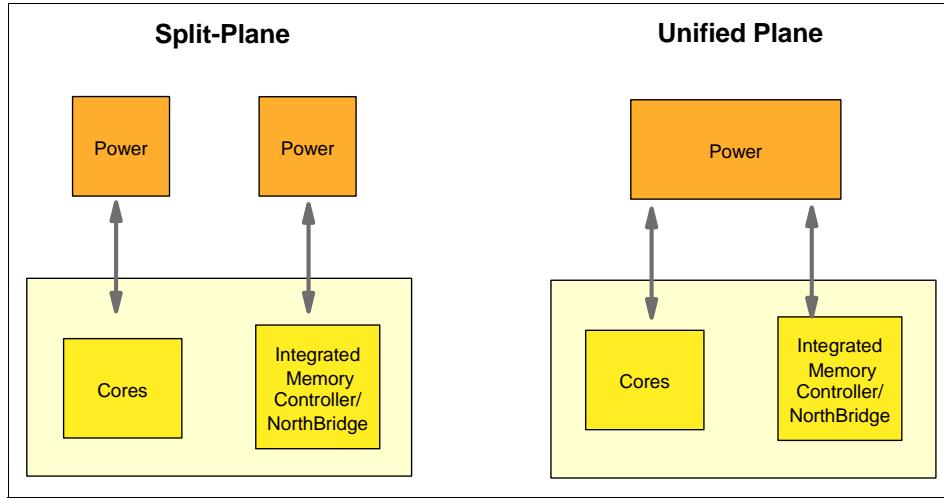


Figure 6-6 *Split-plane versus unified plane*

### 6.3.5 IBM CPU passthru card

AMD Opteron-based systems such as the System x3755 support up to four processors and their associated memory. In a four-way configuration, the x3755 has the configuration that is illustrated in Figure 6-7 on page 114.

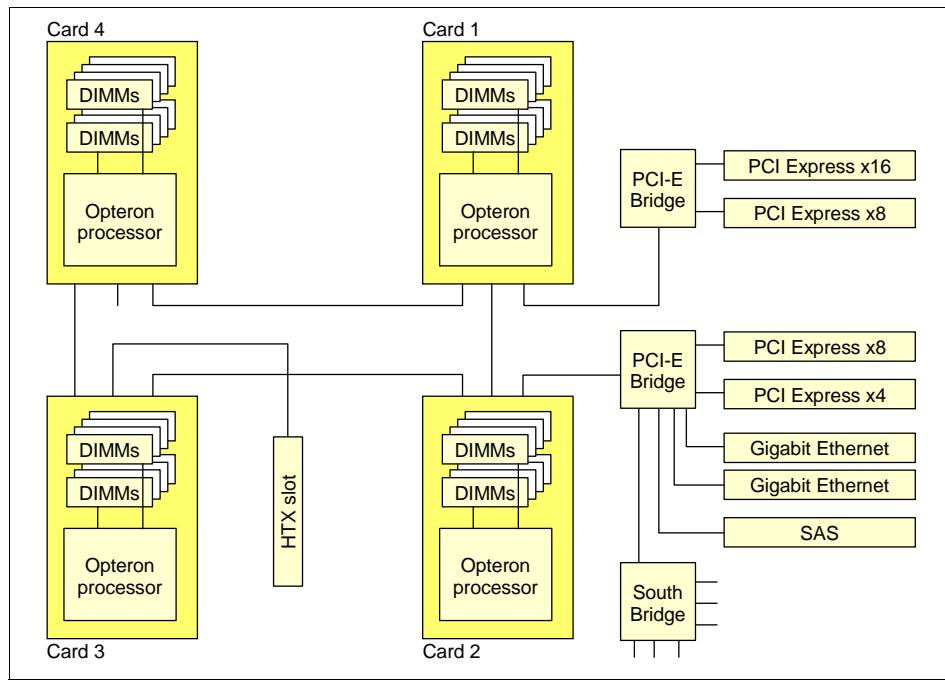


Figure 6-7 Block diagram of the x3755 with four processors installed

Note that each adjacent processor is connected through HyperTransport links, forming a square. The third HT link on each processor card connects to an I/O device, or is unused.

The x3755 also supports a 3-way configuration (by removing the processor card in slot 4 in the upper-left quadrant) as shown in Figure 6-7. This configuration results in processor connections as shown in the top half of Figure 6-8.

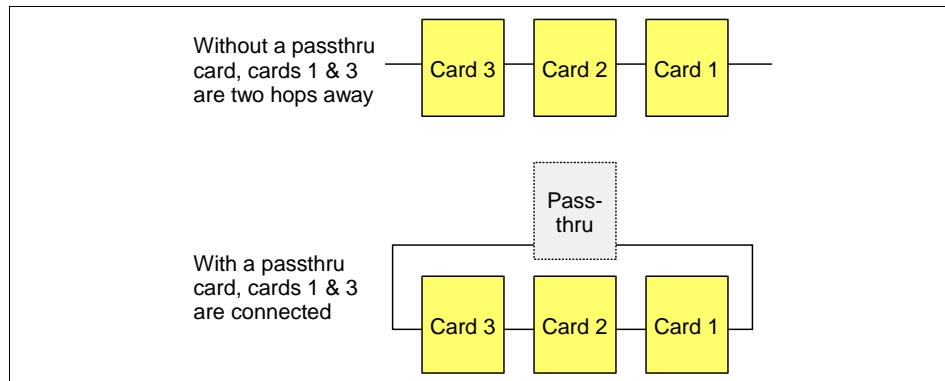


Figure 6-8 The benefit of the passthru card for three-way configurations

However, with the addition of the IBM CPU Passthru card, part number 40K7547, the processors on cards 1 and 3 are directly connected together, as shown in the bottom half of Figure 6-8 on page 114.

The passthru card basically connects two of the HyperTransport connectors together and provides a seamless connection between the processors on either side. The resulting block diagram of the x3755 is shown in Figure 6-9.

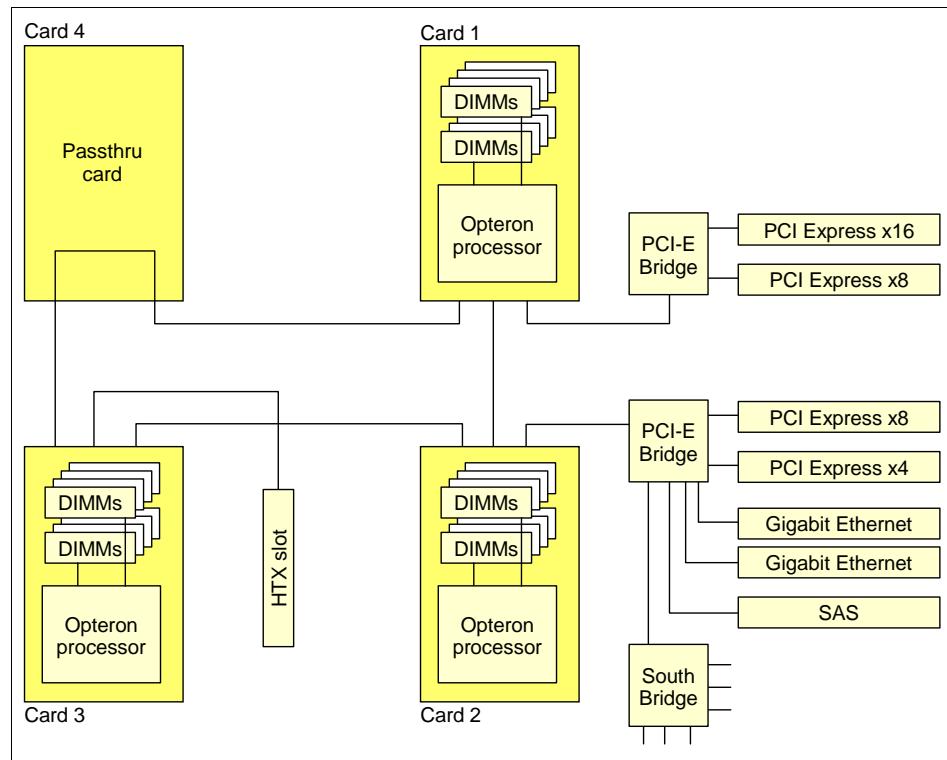


Figure 6-9 Block diagram of a three-way x3755 with a passthru card installed

There are performance benefits which can be realized by adding the passthru card in a three-way configuration. Without the passthru card, the configuration requires that snoop requests and responses originating from one of the two end processors (see Figure 6-8 on page 114), and certain non-local references, travel over two hops. With the passthru card, this now becomes only one hop.

The benefit in decreased latency and increased memory throughput is shown in Figure 6-10 on page 116.

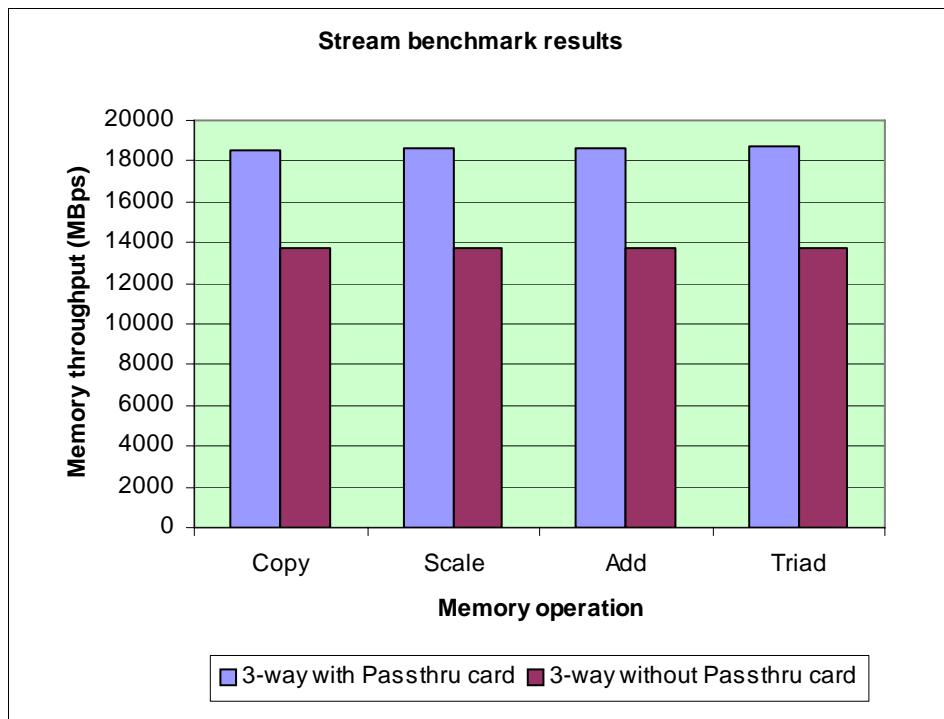


Figure 6-10 Memory throughput benefit of the passthru card

For more information see the white paper *Performance of the IBM System x 3755* by Douglas M Pase and Matthew A Eckl, which is available from:

<http://www.ibm.com/servers/eserver/xseries/benchmarks/related.html>

## 6.4 64-bit computing

As discussed in 6.1, “Processor technology” on page 94, there are three 64-bit implementations in the Intel-compatible processor marketplace:

- ▶ Intel IA64, as implemented on the Itanium 2 processor
- ▶ Intel 64 Technology, as implemented on the 64-bit Xeon DP and Xeon MP processors
- ▶ AMD AMD64, as implemented on the Opteron processor

There exists some uncertainty as to the definition of a 64-bit processor and, even more importantly, the benefit of 64-bit computing.

**Definition of 64-bit:** A 64-bit processor is a processor that is able to address 64 bits of virtual address space. A 64-bit processor can store data in 64-bit format and perform arithmetic operations on 64-bit operands. In addition, a 64-bit processor has general purpose registers (GPRs) and arithmetic logical units (ALUs) that are 64 bits wide.

The Itanium 2 has both 64-bit addressability and GPRs and 64-bit ALUs. So, it is by definition a 64-bit processor.

Intel 64 Technology extends the IA32 instruction set to support 64-bit instructions and addressing, but are Intel 64 Technology and AMD64 processors *real* 64-bit chips? The answer is yes. Where these processors operate in 64-bit mode, the addresses are 64-bit, the GPRs are 64 bits wide, and the ALUs are able to process data in 64-bit chunks. Therefore, these processors are full-fledged, 64-bit processors in this mode.

Note that while IA64, Intel 64 Technology, and AMD64 are all 64-bit, they are not compatible for the following reasons:

- ▶ Intel 64 Technology and AMD64 are, with exception of a few instructions such as 3DNOW, binary compatible with each other. Applications written and compiled for one will usually run at full speed on the other.
- ▶ IA64 uses a completely different instruction set to the other two. 64-bit applications written for the Itanium 2 will not run on the Intel 64 Technology or AMD64 processors, and vice versa.

## 64-bit extensions: AMD64 and Intel 64 Technology

Both the AMD AMD64 and Intel 64 Technology (formerly known as EM64T) architectures extend the well-established IA32 instruction set with:

- ▶ A set of new 64-bit general purpose registers (GPR)
- ▶ 64-bit instruction pointers
- ▶ The ability to process data in 64-bit chunks
- ▶ Up to 1 TB of address space that physical memory is able to access
- ▶ 64-bit integer support and 64-bit flat virtual address space

Even though the names of these extensions suggest that the improvements are simply in memory addressability, both the AMD64 and the Intel 64 Technology are fully functional 64-bit processors.

There are three distinct operation modes available in AMD64 and Intel 64 Technology:

- ▶ 32-bit legacy mode

The first and, in the near future, probably most widely used mode is the 32-bit legacy mode. In this mode, both AMD64 and Intel 64 Technology processors will act just like any other IA32 compatible processor. You can install your 32-bit operating system on such a system and run 32-bit applications, but you will not be able to make use of the new features such as the flat memory addressing above 4 GB or the additional General Purpose Registers (GPRs). 32-bit applications will run just as fast as they would on any current 32-bit processor.

Most of the time, IA32 applications will run even faster because there are numerous other improvements that boost performance regardless of the maximum address size. For applications that share large amounts of data, there might be performance impacts related to the NUMA-like architecture of multi-processor Opteron configurations because remote memory access might slow down your application.

- ▶ Compatibility mode

The second mode supported by the AMD64 and Intel 64 Technology is compatibility mode, which is an intermediate mode of the full 64-bit mode described next. To run in compatibility mode, you will need to install a 64-bit operating system and 64-bit drivers. If a 64-bit operating system and drivers are installed, then both Opteron and Xeon processors will be enabled to support a 64-bit operating system with both 32-bit applications or 64-bit applications.

Compatibility mode gives you the ability to run a 64-bit operating system while still being able to run unmodified 32-bit applications. Each 32-bit application will still be limited to a maximum of 4 GB of physical memory. However, the 4 GB limit is now imposed on a per-process level, not at a system-wide level. This means that every 32-bit process on this system gets its very own 4 GB of physical memory space (assuming sufficient physical memory is installed). This is already a huge improvement compared to IA32, where the operating system kernel and the application had to share 4 GB of physical memory.

Additionally, compatibility mode does not support virtual 8086 mode, so real-mode legacy applications are not supported. 16-bit protected mode applications are, however, supported.

- ▶ Full 64-bit mode (*long mode*)

The final mode is the full 64-bit mode. AMD refers to this as *long mode*. Intel refers to it as *IA-32e mode*. This mode is when a 64-bit operating system and 64-bit application are used. In the full 64-bit operating mode, an application can have a virtual address space of up to 40-bits (which equates to 1 TB of

addressable memory). The amount of physical memory will be determined by how many DIMM slots the server has and the maximum DIMM capacity supported and available at the time.

Applications that run in full 64-bit mode will get access to the full physical memory range (depending on the operating system) and will also get access to the new GPRs, as well as to the expanded GPRs. However, it is important to understand that this mode of operation requires not only a 64-bit operating system (and, of course, 64-bit drivers), but also requires a 64-bit application that has been recompiled to take full advantage of the various enhancements of the 64-bit addressing architecture.

For more information about the AMD64 architecture, see:

<http://www.x86-64.org/>

For more information about Intel 64 Technology, see:

<http://www.intel.com/technology/64bitextensions/>

## **Benefits of 64-bit (AMT64, Intel 64 Technology) computing**

In the same way that 16-bit processors and 16-bit applications are no longer used in this space, it is likely that at some point in the future, 64-bit processors and applications will fully replace their 32-bit counterparts.

Processors using the Intel 64 Technology and AMD64 architectures are making this transition very smooth by offering 32-bit and 64-bit modes. This means that the hardware support for 64-bit will be in place before you upgrade or replace your software applications with 64-bit versions. IBM System x already has many models available with the Intel 64 Technology-based Xeon and AMD64 Opteron processors.

The question you should be asking is whether the benefit of 64-bit processing is worth the effort of upgrading or replacing your 32-bit software applications. The answer is that it depends on the application. Here are examples of applications that will benefit from 64-bit computing:

- ▶ Encryption applications

Most encryption algorithms are based on very large integers and would benefit greatly from the use of 64-bit GPRs and ALUs. Although modern high-level languages allow you to specify integers above the  $2^{32}$  limit, in a 32-bit system, this is achieved by using two 32-bit operands, thereby causing significant overhead when moving those operands through the CPU pipelines. A 64-bit processor will allow you to perform 64-bit integer operation with one instruction.

- ▶ Scientific applications

Scientific applications are another example of workloads that need 64-bit data operations. Floating-point operations do not benefit from the larger integer size because floating-point registers are already 80 or 128 bits wide even in 32-bit processors.

- ▶ Software applications requiring more than 4 GB of memory

The biggest advantage of 64-bit computing for commercial applications is the flat, potentially massive, address space.

32-bit enterprise applications such as databases are currently implementing Page Addressing Extensions (PAE) and Addressing Windows Extensions (AWE) addressing schemes to access memory above the 4 GB limit imposed by 32-bit address limited processors. With Intel 64 Technology and AMD64, these 32-bit addressing extension schemes support access to memory up to 128 GB in size.

One constraint with PAE and AWE, however, is that memory above 4 GB can only be used to store data. It cannot be used to store or execute code. So, these addressing schemes only make sense for applications such as databases, where large data caches are needed.

In contrast, a 64-bit virtual address space provides for direct access to up to 2 Exabytes (EB), and even though we call these processors 64-bit, none of the current 64-bit processors actually supports full 64 bits of physical memory addressing, simply because this is such an enormous amount of memory.

In addition, 32-bit applications might also get a performance boost from a 64-bit Intel 64 Technology or AMD64 system running a 64-bit operating system. When the processor runs in Compatibility mode, every process has its own 4 GB memory space, not the 2 GB or 3 GB memory space each gets on a 32-bit platform. This is already a huge improvement compared to IA32, where the operating system and the application had to share those 4 GB of memory.

When the application is designed to take advantage of more memory, the availability of the additional 1 or 2 GB of physical memory can create a significant performance improvement. Not all applications take advantage of the global memory available. APIs in code need to be used to recognize the availability of more than 2 GB of memory.

Furthermore, some applications will not benefit at all from 64-bit computing and might even experience degraded performance. If an application does not require greater memory capacity or does not perform high-precision integer or floating-point operations, then 64-bit will not provide any improvement.

In fact, because 64-bit computing generally requires instructions and some data to be stored as 64-bit objects, these objects consume more physical memory

than the same object in a 32-bit operating environment. The memory capacity inflation of 64-bit can only be offset by an application taking advantage of the capabilities of 64-bit (greater addressing or increased calculation performance for high-precision operations), but when an application does not make use of the 64-bit operating environment features, it often experiences the overhead without the benefit.

In this case, the overhead is increased memory consumption, leaving less physical memory for operating system buffers and caches. The resulting reduction in effective memory can decrease performance.

Software driver support in general is lacking for 64-bit operating systems compared to the 32-bit counterparts. General software drivers such as disk controllers or network adapters or application tools might not have 64-bit code in place for x64 operating systems. Prior to moving to an x64 environment it might be wise to ensure that all third-party vendors and software tools support drivers for the specific 64-bit operating system that you are planning to use.

## 64-bit memory addressing

The width of a memory address dictates how much memory the processor can address. A 32-bit processor can address up to  $2^{32}$  bytes or 4 GB. A 64-bit processor can theoretically address up to  $2^{64}$  bytes or 16 Exabytes (or 16777216 Terabytes), although current implementations address a smaller limit, as shown in Table 6-8.

**Note:** These values are the limits imposed by the processors. Memory addressing can be limited further by the chipset implemented in the server. For example, the XA-64e chipset used in the x3950 M2 Xeon-based server addresses up to 1 TB of memory.

Table 6-8 Memory supported by processors

Processor	Flat addressing	Addressing with PAE <sup>a</sup>
Intel 32-bit Xeon MP (32-bit) processors including Foster MP and Gallatin	4 GB (32-bit)	128 GB
Intel 64-bit Xeon DP Nocona (64-bit)	64 GB (36-bit)	128 GB in compatibility mode
Intel 64-bit Xeon MP Cranford (64-bit)	64 GB (36-bit)	128 GB in compatibility mode
Intel 64-bit Xeon MP Potomac (64-bit)	1 TB (40-bit)	128 GB in compatibility mode
Intel 64-bit dual core MP including Paxville, Woodcrest, and Tulsa	1 TB (40-bit)	128 GB in compatibility mode

Processor	Flat addressing	Addressing with PAE <sup>a</sup>
Intel 64-bit quad core MP including Clovertown, Tigerton, and Harpertown	1 TB (40-bit)	128 GB in compatibility mode
Intel 64-bit (64-bit) six core MP Dunnington	1 TB (40-bit)	128 GB in compatibility mode
AMD Opteron Barcelona	256 TB (48-bit)	128 GB in compatibility mode

a. These values may be further limited by the operating system. See “Windows PAE and Address Windowing Extensions” on page 209 for more information about this topic.

The 64-bit extensions in the processor architectures Intel 64 Technology and AMD64 provide better performance for both 32-bit and 64-bit applications on the same system. These architectures are based on 64-bit extensions to the industry-standard x86 instruction set, and provide support for existing 32-bit applications.

## 6.5 Processor performance

Processor performance is a complex topic because the effective CPU performance is affected by system architecture, operating system, application, and workload. This is even more so with the choice of three different CPU architectures, IA32, Itanium 2, and AMD64/Intel 64.

An improvement in system performance gained by a CPU upgrade can be achieved only when all other components in the server are capable of working harder. Compared to all other server components, the Intel processor has experienced the largest performance improvement over time and in general, the CPU is much faster than every other component. This makes the CPU the least likely server component to cause a bottleneck. Often, upgrading the CPU with the same number of cores simply means the system runs with lower CPU utilization, while other bottlenecked components become even more of a bottleneck.

In general, server CPUs execute workloads that have very random address characteristics. This is expected because most servers perform many unrelated functions for many different users. So, core clock speed and L1 cache attributes have a lesser effect on processor performance compared to desktop environments. This is because with many concurrently executing threads that cannot fit into the L1 and L2 caches, the processor core is constantly waiting for L3 cache or memory for data and instructions to execute.

### 6.5.1 Comparing CPU architectures

Every CPU we have discussed so far had similar attributes. Every CPU has two or more pipelines, an internal clock speed, L1 cache, L2 cache (some also L3 cache). The various caches are organized in different ways; some of them are 2-way associative, whereas others go up to 16-way associativity. Some have an 800 MHz FSB; others have no FSB at all (Opteron).

Which is fastest? Is the Xeon DP the fastest CPU of them all because it is clocked at up to 3.8 GHz? Or is the Itanium Montvale the fastest because it features up to 18 MB of L3 cache? Or is it perhaps the third generation Opteron because its L3 cache features a 32-way associativity?

As is so often the case, there is never one simple answer. When comparing processors, clock frequency is only comparable when comparing processors of the same architectural family. You should never compare isolated processor subsystems across different CPU architectures and think you can make a simple performance statement. Comparing different CPU architectures is therefore a very difficult task and has to take into account available application and operating system support.

As a result, we do not compare different CPU architectures in this section, but we do compare the features of the different models of one CPU architecture.

### 6.5.2 Cache associativity

Cache associativity is necessary to reduce the lookup time to find any memory address stored in the cache. The purpose of the cache is to provide fast lookup for the CPU, because if the cache controller had to search the entire memory for each address, the lookup would be slow and performance would suffer.

To provide fast lookup, some compromises must be made with respect to how data can be stored in the cache. Obviously, the entire amount of memory would be unable to fit into the cache because the cache size is only a small fraction of the overall memory size (see 6.5.3, “Cache size” on page 127). The methodology of how the physical memory is mapped to the smaller cache is known as *set associativity* (or just *associativity*).

First, we must define some terminology. Referring to Figure 6-11 on page 124, main memory is divided into *pages*. Cache is also divided into pages, and a memory page is the same size as a cache page. Pages are divided up into lines or *cache lines*. Generally, cache lines are 64 bytes wide.

For each page in memory or in cache, the first line is labeled *cache line 1*, the second line is labeled *cache line 2*, and so on. When data in memory is to be

copied to cache, the line that this data is in is copied to the equivalent slot in cache.

Looking at Figure 6-11, when copying cache line 1 from memory page 0 to cache, it is stored in cache line 1 in the cache. This is the only slot where it can be stored in cache. This is a one-way associative cache, because for any given cache line in memory, there is only one position in cache where it can be stored. This is also known as *direct mapped*, because the data can only go into one place in the cache.

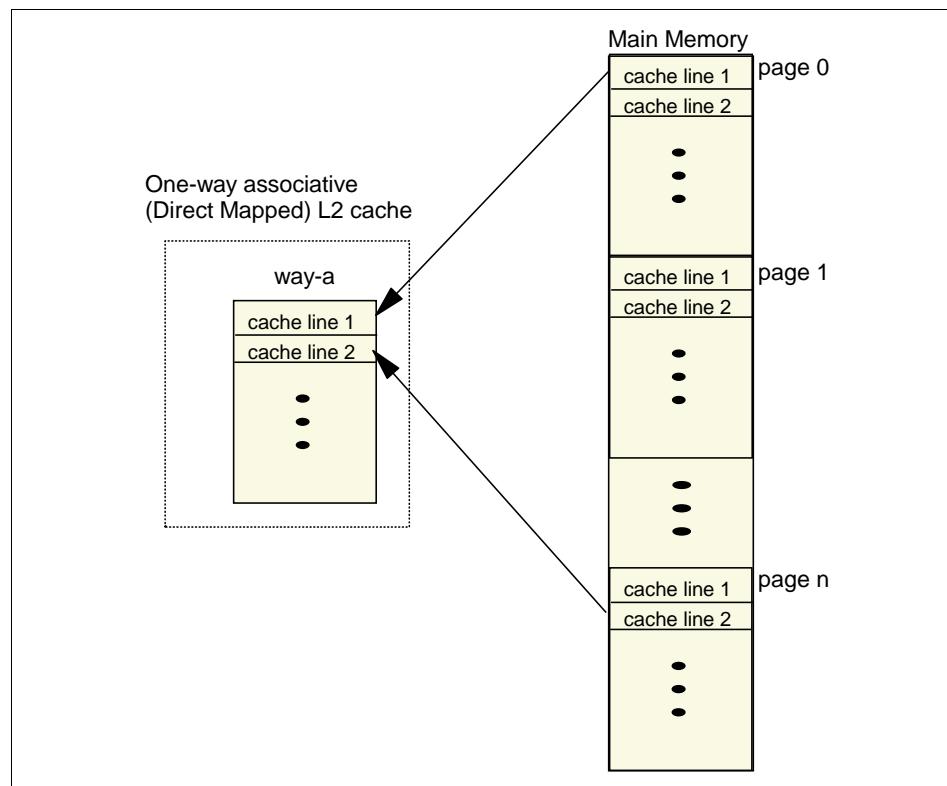


Figure 6-11 One-way associative (direct mapped) cache

With a one-way associative cache, if cache line 1 in another memory page needs to be copied to cache, it too can only be stored in cache line 1 in cache. You can see from this that you would get a greater cache hit rate if you use greater associativity.

Figure 6-12 on page 125 shows the 2-way set associative cache implementation. Here there are two locations in which to store the first cache line for any memory page. As the figure illustrates, main memory on the right side will be able to store

up to two *cache line 1* entries concurrently. Cache line 1 for page 0 of main memory could be located in *way-a* of the cache; cache line 1 for page n of main memory could be located in *way-b* of the cache simultaneously.

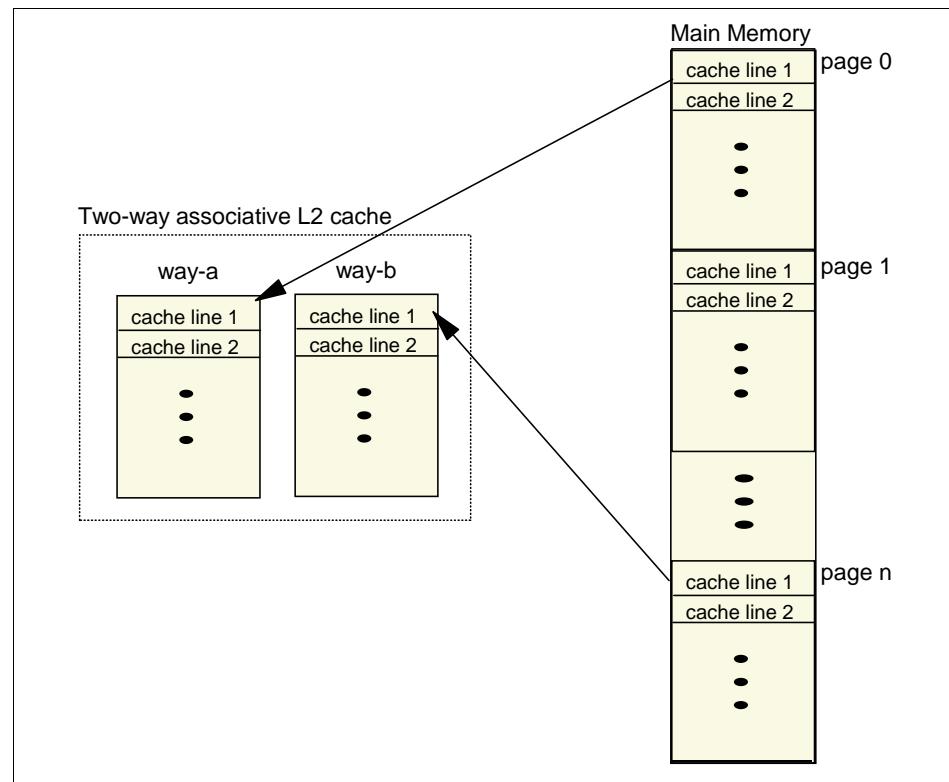


Figure 6-12 A 2-way set associative cache

Expanding on a one-way and two-way set associative cache, a 3-way set associative cache, as shown in Figure 6-13, provides three locations. A 4-way set associative cache provides four locations. An 8-way set associative cache provides eight possible locations in which to store the first cache line from up to eight different memory pages.

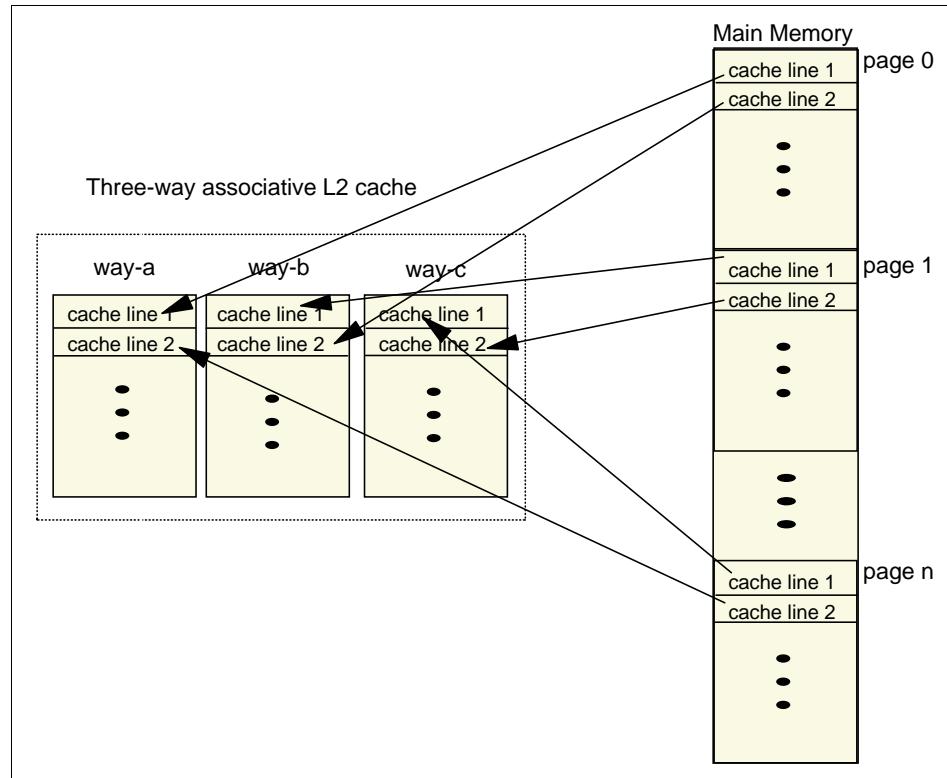


Figure 6-13 3-way set associative cache

Set associativity greatly minimizes the cache address decode logic necessary to locate a memory address in the cache. The cache controller simply uses the requested address to generate a pointer into the correct cache page. A hit occurs when the requested address matches the address stored in one of the fixed number of cache locations associated with that address. If the particular address is not there, a cache miss occurs.

Notice that as the associativity increases, the lookup time to find an address within the cache could also increase because more pages of cache must be searched. To avoid longer cache lookup times as associativity increases, the lookups are performed in parallel. However, as the associativity increases, so does the complexity and cost of the cache controller.

For high performance X3 Architecture systems such as the System x3950, lab measurements determined that the most optimal configuration for cache was 9-way set associativity, taking into account performance, complexity, and cost.

A *fully associative cache* in which any memory cache line could be stored in any cache location could be implemented, but this is almost never done because of the expense (in both cost and die areas) in parallel lookup circuits required.

Large servers generally have random memory access patterns, as opposed to sequential memory access patterns. Higher associativity favors random memory workloads due to its ability to cache more distributed locations of memory.

### 6.5.3 Cache size

Faster, larger caches usually result in improved processor performance for server workloads. Performance gains obtained from larger caches increase as the number of processors within the server increase. When a single CPU is installed in a four-socket SMP server, there is little competition for memory access. Consequently, when a CPU has a cache miss, memory can respond, and with the deep pipeline architecture of modern processors, the memory subsystem usually responds before the CPU stalls. This allows one processor to run fast almost independently of the cache hit rate.

On the other hand, if there are four processors installed in the same server, each queuing multiple requests for memory access, the time to access memory is greatly increased, thus increasing the potential for one or more CPUs to stall. In this case, a fast L2 hit saves a significant amount of time and greatly improves processor performance.

As a rule, the greater the number of processors in a server, the more gain from a large L2 cache. In general:

- ▶ With two CPUs, expect 4% to 6% improvement when you double the cache size
- ▶ With four CPUs, expect 8% to 10% improvement when you double the cache size
- ▶ With eight or more CPUs, you might expect as much as 10% to 15% performance gain when you double processor cache size

Of course, there are diminishing returns as the size of the cache improves; these are simply rules of thumb for the maximum expected performance gain.

#### **6.5.4 Shared cache**

Shared cache is introduced in 6.2.4, “Intel Core microarchitecture” on page 103 as shared L2 cache to improve resource utilization and boost performance. In Nehalem, sharing cache moves to the third level cache for overall system performance considerations. In both cases, the last level of cache is shared among different cores and provides significant benefits in multi-core environments as oppose to dedicated cache implementation.

Benefits of shared cache include:

- ▶ Improved resource utilization, which makes efficient usage of the cache. When one core idles, the other core can take all the shared cache.
- ▶ Shared cache, which offers faster data transfer between cores than system memory, thus improving system performance and reducing traffic to memory.
- ▶ Reduced cache coherency complexity, because a coherency protocol does not need to be set for the shared level cache because data is shared to be consistent rather than distributed.
- ▶ More flexible design of the code relating to communication of threads and cores because programmers can leverage this hardware characteristic.
- ▶ Reduced data storage redundancy, because the same data in the shared cache needs to be stored only once.

As the trend moves to the multi-core processor for a computing performance leap, the last-level shared cache mechanism and relevant software design techniques become more and more important and prevalent. AMD also implements shared L3 cache on the Barcelona architecture.

#### **6.5.5 CPU clock speed**

Processor clock speed affects CPU performance because it is the speed at which the CPU executes instructions. Measured system performance improvements because of an increase in clock speed are usually not directly proportional to the clock speed increase. For example, when comparing a 3.0 GHz CPU to an older 1.6 GHz CPU, you should not expect to see 87% improvement. In most cases, performance improvement from a clock speed increase will be about 30% to 50% of the percentage increase in clock speed. So for this example, you could expect about 26% to 44% system performance improvement when upgrading a 1.6 GHz CPU to a 3.0 GHz CPU.

### 6.5.6 Scaling versus the number of processor cores

In general, the performance gains shown in Figure 6-14 can be obtained by adding CPUs when the server application is capable of efficiently utilizing additional processors—and there are no other bottlenecks occurring in the system.

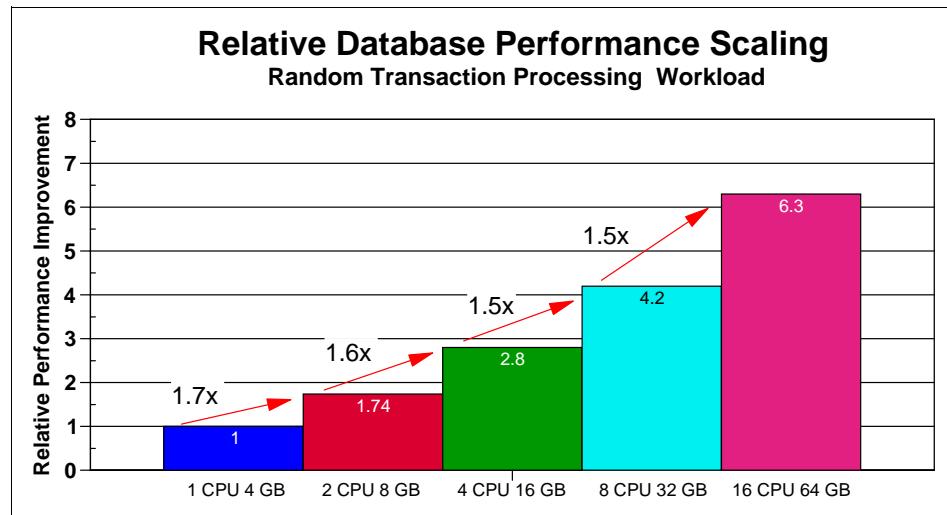


Figure 6-14 Typical performance gains when adding processors

These scaling factors can be used to approximate the achievable performance gains that can be obtained when adding CPUs and memory to a scalable Intel IA-32 server.

For example, begin with a 1-way 3.0 GHz Xeon MP processor and add another Xeon MP processor. Server throughput performance will improve up to about 1.7 times. If you increase the number of Xeon processors to four, server performance can improve to almost three times greater throughput than the single processor configuration.

At eight processors, the system has slightly more than four times greater throughput than the single processor configuration. At 16 processors, the performance increases to over six-fold greater throughput than the single CPU configuration.

High performing chipsets such as the XA-64e generally are designed to provide higher scalability than the average chipset. Figure 6-15 on page 130 shows the performance gain of a high performing chipset such as the X4 Hurricane chipset in the x3850 M2 as processors are added, assuming no other bottlenecks occur in the system.

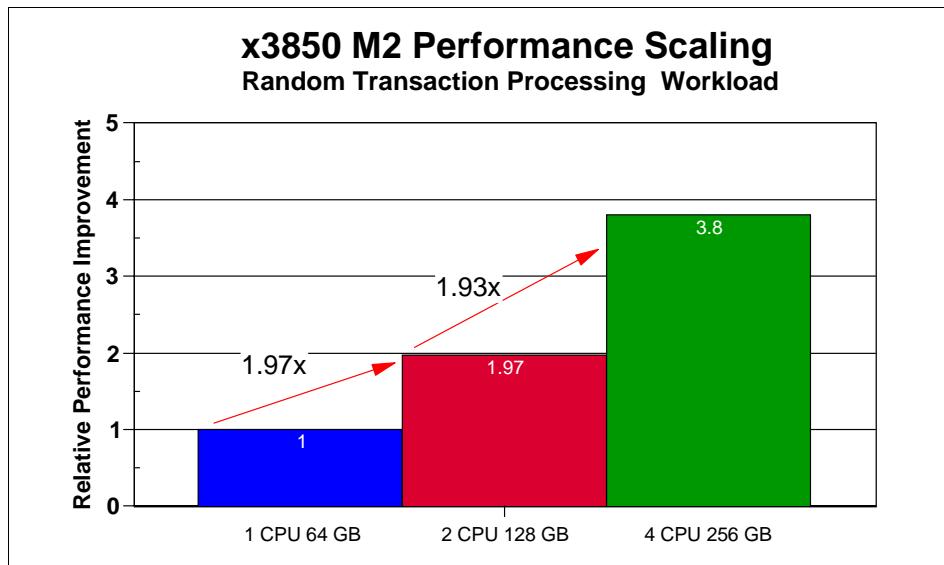


Figure 6-15 System x3850M2 performance scaling when adding processors

Database applications such as IBM DB2, Oracle, and Microsoft SQL Server usually provide the greatest performance improvement with increasing numbers of CPUs. These applications have been painstakingly optimized to take advantage of multiple CPUs. This effort has been driven by the database vendors' desire to post #1 transaction processing benchmark scores. High-profile industry-standard benchmarks do not exist for many applications, so the motivation to obtain optimal scalability has not been as great. As a result, most non-database applications have significantly lower scalability. In fact, many do not scale beyond two to four CPUs.

### 6.5.7 Processor features in BIOS

BIOS levels permit various settings for performance in certain IBM System x servers.

- ▶ Processor Adjacent Sector Prefetch

When this setting is enabled (and enabled is the default for most systems), the processor retrieves both sectors of a cache line when it requires data that is not currently in its cache.

When this setting is disabled, the processor will only fetch the sector of the cache line that includes the data requested. For instance, only one 64-byte line from the 128-byte sector will be prefetched with this setting disabled.

This setting can affect performance, depending on the application running on the server and memory bandwidth utilization. Typically, it affects certain benchmarks by a few percent, although in most real applications it will be negligible. This control is provided for benchmark users who want to fine-tune configurations and settings.

- ▶ Processor Hardware Prefetcher

When this setting is enabled (disabled is the default for most systems), the processor is able to prefetch extra cache lines for every memory request. Recent tests in the performance lab have shown that you will get the best performance for most commercial application types if you disable this feature. The performance gain can be as much as 20%, depending on the application.

For high-performance computing (HPC) applications, we recommend that you turn HW Prefetch enabled. For database workloads, we recommend that you leave the HW Prefetch disabled.

- ▶ IP Prefetcher

Each core has one IP prefetcher. When this setting is enabled, the prefetcher scrutinizes historical reading in the L1 cache, in order to have an overall diagram and to try to load foreseeable data.

- ▶ DCU Prefetcher

Each core has one DCU prefetcher. When this setting is enabled, the prefetcher detects multiple reading from a single cache line for a determined period of time and decides to load the following line in the L1 cache.

Basically, hardware prefetch and adjacent sector prefetch are L2 prefetchers. The IP prefetcher and DCU prefetcher are L1 prefetchers.

All prefetch settings decrease the miss rate for the L1/L2/L3 cache when they are enabled. However, they consume bandwidth on the front-side bus, which can reach capacity under heavy load. By disabling all prefetch settings, multi-core setups achieve generally higher performance and scalability. Most benchmarks turned in so far show that they obtain the best results when all prefetchers are disabled.

Table 6-9 on page 132 lists the recommended settings based on benchmark experience in IBM labs. Customers should always use these settings as recommendations and evaluate their performance based on actual workloads. When in doubt, always vary to determine which combination delivers maximum performance.

*Table 6-9 Recommended prefetching settings*

<b>Turning options</b>	<b>Default settings</b>	<b>Recommended for most environments</b>
Adjacent Sector	Off	Off
Hardware	Off	Off
IP	Off	On
DCU	Off	Off



# Virtualization hardware assists

Initially all virtualization on x86 architecture was implemented in software. However, Intel and AMD have developed hardware virtualization technology that is designed to:

- ▶ Allow guest operating systems, VMMs, and applications to run at their standard privilege levels
- ▶ Eliminate the need for binary translation and paravirtualization
- ▶ Provide more reliability and security

The first phase of this iterative development endeavor was implemented in the processors. Intel named its hardware virtualization technology Intel VT. AMD named its hardware virtualization technology AMD-V. With each new generation of CPU technology, both vendors are adding additional phases of hardware virtualization technology for I/O and memory.

This chapter discusses the following topics:

- ▶ 7.1, “Introduction to virtualization technology” on page 134
- ▶ 7.2, “Virtualization hardware assists” on page 138
- ▶ 7.3, “Support for virtualization hardware assists” on page 143
- ▶ 7.4, “Resources” on page 144

## 7.1 Introduction to virtualization technology

**Note:** This chapter provides an introduction to virtualization hardware assists. It touches on some virtualization concepts, but is not intended to provide a lengthy review of virtualization concepts. For more information regarding virtualization, see the IBM Redbooks publication, *An Overview of Virtualization on IBM System x Servers*, REDP-4480.

Conventionally, a server is loaded with a single operating system that controls access to the server's hardware resources such as the processors, memory, and I/O-related technology. Virtualization enables multiple operating systems, called *guest* operating systems, to run on a single server and to share access to the server's hardware resources. To share resources between multiple guest operating systems, a software virtualization layer is required to manage the utilization of the resources by each guest operating system.

Figure 7-1 illustrates a guest operating system that is running on a server with a software virtualization layer between the key hardware resources and the guest operating systems.

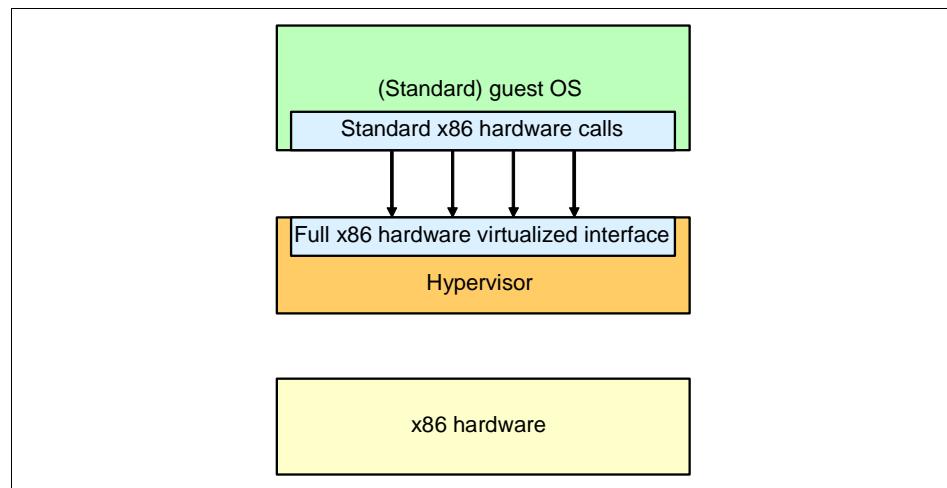


Figure 7-1 Full virtualization architecture

### 7.1.1 Privilege levels

There are challenges with the virtualization model shown in Figure 7-1 on an x86 architecture. On an x86 architecture without virtualization, the operating system is designed to run at privilege level 0, which is the most powerful privilege level.

Privilege level 0 gives the operating system access to the hardware resources of a server, so that it can execute instructions to obtain the state of the hardware resources and control those resources. Applications tend to run at privilege level 3 and do not have direct access to the hardware.

When the software virtualization layer is introduced, the guest operating system is bumped to privilege level 1, and the virtualization layer runs at privilege level 0. This shift from a more powerful privilege to a less powerful privilege level is called *ring deprivileging*. Ring deprivileging can introduce faults, because many Intel architecture instructions which control CPU functionality were designed to be executed from privilege level 0, not privilege level 1. Therefore, the software virtualization layer must *trap* certain instructions and hardware accesses and then emulate those instructions back to the guest operating system. These additional steps introduce more complexity and, therefore, the chance of more faults.

If a guest operating system needs to access memory, the virtualization layer must intercept, interpret, execute, and then return the result back to the guest operating system. In addition, the virtualization layer must handle all interrupts to and from the guest operating system. A guest operating system can decide whether to block or to permit interrupts, depending on the operation in progress. The virtualization layer must be able to track what the guest operating system decides to do with an interrupt, which adds overhead to system resources. The overhead produces a performance penalty.

### 7.1.2 Binary translation and paravirtualization

*Binary translation* is a software method used to address the challenges of ring deprivileging. Both VMware and Microsoft virtualization products use binary translation. The hypervisor traps certain privileged guest operating system instructions, and then translates them into instructions that the virtualized environment can execute.

When a privileged instruction is received by the hypervisor, it takes control of the required hardware resource, resolves any conflicts, and returns control to the guest operating system. If the hypervisor functions correctly, the guest operating system does not know that the hypervisor is emulating the hardware resource. However, binary translation requires specific builds of operating systems. A new release of the operating system requires testing and perhaps changes to the hypervisor code.

*Paravirtualization* is another software solution to the challenges of ring deprivileging. It is used by Xen, an open source virtualization project. In this case, the operating system source code is altered so that it can call the hypervisor directly to perform low-level functions. Figure 7-2 illustrates the paravirtualization architecture.

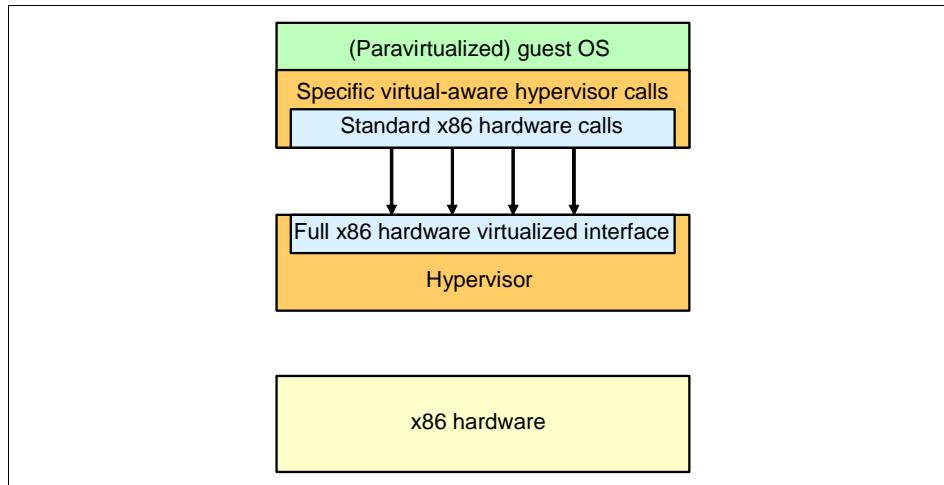


Figure 7-2 Paravirtualization architecture

Paravirtualization reduces the complexity of the virtualization software layer, and it can improve performance. In this case, the modified guest operating system is *virtual-aware* and shares the virtualization load with the hypervisor. This reduces the complexity of the hypervisor, which can be implemented more efficiently. However, off-the-shelf operating systems cannot be used with paravirtualization.

**Tip:** See 15.11, “Xen virtualization” on page 496, for more information about Xen virtualization.

### 7.1.3 Memory-intensive workload

All virtualization workloads are dependent on memory subsystems and are really memory intensive. The AMD Opteron has an integrated memory controller, and the Direct Connect™ Architecture was designed for dedicated memory access and better scalability with an increasing number of CPU cores.

Intel has changed from a front-side bus architecture to an integrated memory controller with its new generation of server processors starting with the Intel Xeon Processor 5500 series. This allows significantly reduced latency of

transition from one virtual machine task to another virtual machine when physical hardware resources should be moved to serve different guest operating systems.

#### 7.1.4 Nested paging

Nested paging (also known as Hardware Assisted Paging) is a technology that allows significantly improved handling of multiple guest operating systems.

Processors supporting the x86 architectures translate linear addresses, generated by operating system and applications, into physical addresses, which are used to access memory. The translation process is called *paging*.

Processors apply paging to all memory accesses that use linear addresses.

When a processor is in paged mode, paging is involved in every data and instruction access. x86 processors utilize various hardware facilities to reduce overheads associated with paging. However, under virtualization, where the guest's view of physical memory is different from system's view of physical memory, a second level of address translation is required to convert guest physical addresses to machine addresses.

To enable this additional level of translation, the hypervisor must virtualize processor paging. Current software-based paging virtualization techniques such as shadow-paging incur significant overheads, which result in reduced virtualized performance, increased CPU utilization, and increased memory consumption.

CPU vendors have developed technologies to move this translation to the CPU level controlled by VMM. Enhancements in paging circuitry and buffers to support dual layer translations is called *nested paging*. Nested paging provides a 5% to 30% performance improvement, depending on the type of workload for applications running in virtual machines. Intel nested paging technology is known as Extended Page Tables (EPT). AMD nested paging technology is known as Rapid Virtualization Indexing (RVI).

When translation is performed, it is stored for future use in a Translation Look-aside Buffer (TLB). In support of nested paging, CPU hardware assist technologies also had to address TLB improvements. Intel has added VPID. AMD has a similar technology known as Tagged TLBs.

These specific technologies are described in 7.2.2, “Intel VT” on page 139 and 7.2.3, “AMD-V” on page 141.

## 7.2 Virtualization hardware assists

Virtualization hardware assists have been developed and will continue to be developed to overcome the challenges and complexities of software virtualized solutions discussed in the preceding sections of this chapter.

The following sections discuss hardware technologies that help to reduce software complexity and overhead when working with virtual machines.

### 7.2.1 CPU command interface enhancements

Intel's virtualization hardware assist is called Intel Virtualization Technology (Intel VT). Intel VT-x provides hardware support for IA32 and 64-bit Xeon processor virtualization.

**Note:** VT-i provides hardware support for Itanium processor virtualization; however, a discussion of VT-i is beyond the scope of this book.

VT-x introduces two new CPU operations:

- ▶ VMX root operation - VMM functions
- ▶ VMX non-root operation - guest operating system functions

Both new operations support all four privilege levels. This support allows a guest operating system to run at privilege level 0 where an operating system is designed to run on an x86 platform.

VT-x also introduces two transitions:

- ▶ VM entry is defined as a transition from VMX root operation to VMX non-root operation, or VMM to guest operating system.
- ▶ VM exit is defined as a transition from VMX non-root operation to VMX root operation, or from guest operating system to VMM.

VT-x also introduces a new data structure called the virtual machine control structure (VMCS). The VMCS tracks VM entries and VM exits, as well as the processor state of the guest operating system and VMM in VMX non-root operations.

AMD's virtualization hardware, AMD-V, introduces a new processor mode called Guest Mode. Guest Mode is similar to the VMX non-root mode in Intel VT. A new data structure called the virtual machine control block (VMCB) tracks the CPU state for a guest operating system.

If a VMM wants to transfer processor control to a guest operating system, it executes a VMRUN command. A VMRUN entry is analogous to the Intel VT VM entry transition. A VMRUN exit is analogous to the Intel VT VM exit transition.

Studies and measurements indicate that the first pass of virtualization hardware assists that were implemented in the processor as described in this section did not significantly improve performance. It appears as though the software implementations of binary translation and paravirtualization are still the best performing methods to implement privileging.

This is not a complete surprise, because processor utilization does not tend to be the largest area of concern with respect to virtualization performance. In the future, look for hardware-based page tables implemented in processors that will replace shadow page tables currently implemented in software, which should decrease hypervisor overhead. In addition, look for I/O hardware assists, which tend to be one of the largest areas of concern with respect to virtualization performance. These two areas will be addressed in the next passes of virtualization hardware assists.

### 7.2.2 Intel VT

As mentioned, Intel introduced a nested paging feature with the Xeon 5500 processor series (Nehalem) in 2009. There are other improvements for Intel VT, including VPID, EPT, and a significant virtual machine transition latency reduction.

The features of Intel VT include the following:

- ▶ VPID, or Virtual Processor Identifier

This feature allows a virtual machine manager to assign a different non-zero VPID to each virtual processor (the zero VPID is reserved for the VMM). The CPU can use VPIDs to tag translations in the Translation Look-aside Buffers (TLB). This feature eliminates the need for TLB flushes on every VM entry and VM exit, and eliminates the adverse impact of those flushes on performance.

- ▶ EPT, or Extended Page Table

EPT is the Intel implementation of nested paging for virtualization. When this feature is active, the ordinary IA-32 page tables (referenced by control register CR3) translate from linear addresses to guest-physical addresses. A separate set of page tables (the EPT tables) translate from guest physical addresses to the host physical addresses that are used to access memory. As a result, guest software can be allowed to modify its own IA-32 page tables and directly handle page faults. This allows a VMM to avoid the VM

exits associated with page-table virtualization, which are a major source of virtualization overhead without EPT.

- Improved virtualization latency

There are significant improvements with each Intel CPU generation for virtual machine transition latency reduction. Figure 7-3 shows how the latency was reduced on different CPU implementations. It is interesting to note that the VPID feature on Nehalem removed TLB flush delay.

The CPU implementation codenames are listed here:

- PSC - Pentium 4™ (Prescott)
- CDM - Cedar Mill
- YNH - Yonah
- MRM - Merom
- PNR - Penryn
- NHM - Nehalem

To estimate the impact on virtualization overhead reduction, you can use this rule of thumb: every 100-cycle reduction in event cost reduces virtualization overhead by ~0.5%-1.0%.

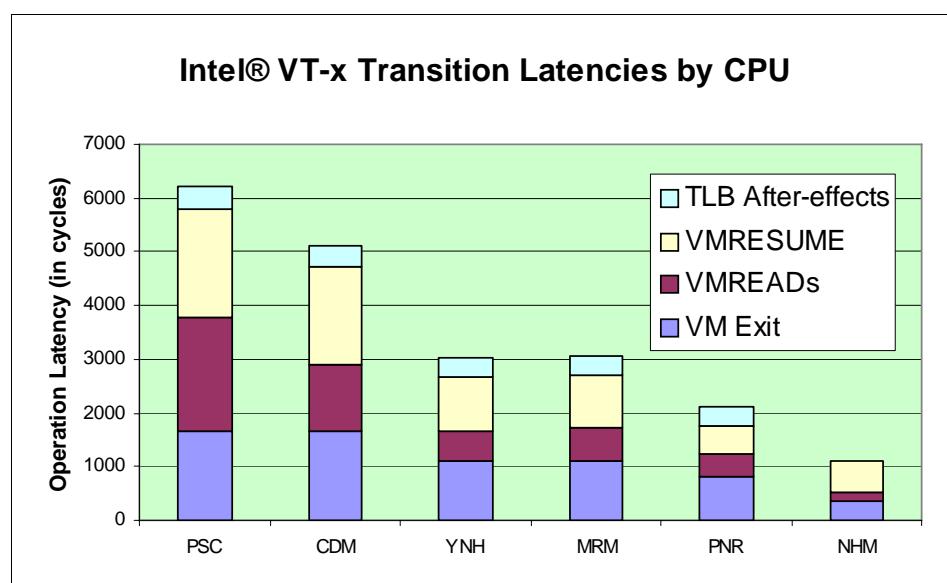


Figure 7-3 Intel VT-x latency reductions by CPU implementation

- Chipset features

There are other developments in the Intel Virtualization Technology for Directed I/O (Intel VT for Directed I/O); specifically, it focuses on the components supporting I/O virtualization as it applies to platforms that use

Intel processors and core logic chipsets complying with Intel platform specifications.

For more information, see the Intel document, Intel Virtualization Technology for Directed I/O - Architecture Specification, which is available from:

[http://download.intel.com/technology/computing/vptech/Intel\(r\)\\_VT\\_for\\_Direct\\_Io.pdf](http://download.intel.com/technology/computing/vptech/Intel(r)_VT_for_Direct_Io.pdf)

The document covers the following topics:

- An overview of I/O subsystem hardware functions for virtualization support
- A brief overview of expected usages of the generalized hardware functions
- The theory of operation of hardware, including the programming interface

Figure 7-4 shows Intel developments for I/O virtualization improvement which include Intel I/O Acceleration Technology (IOAT), Virtual Machine Device Queues (VMDq), and Single Root I/O Virtualization (SR-IOV) Implementation.

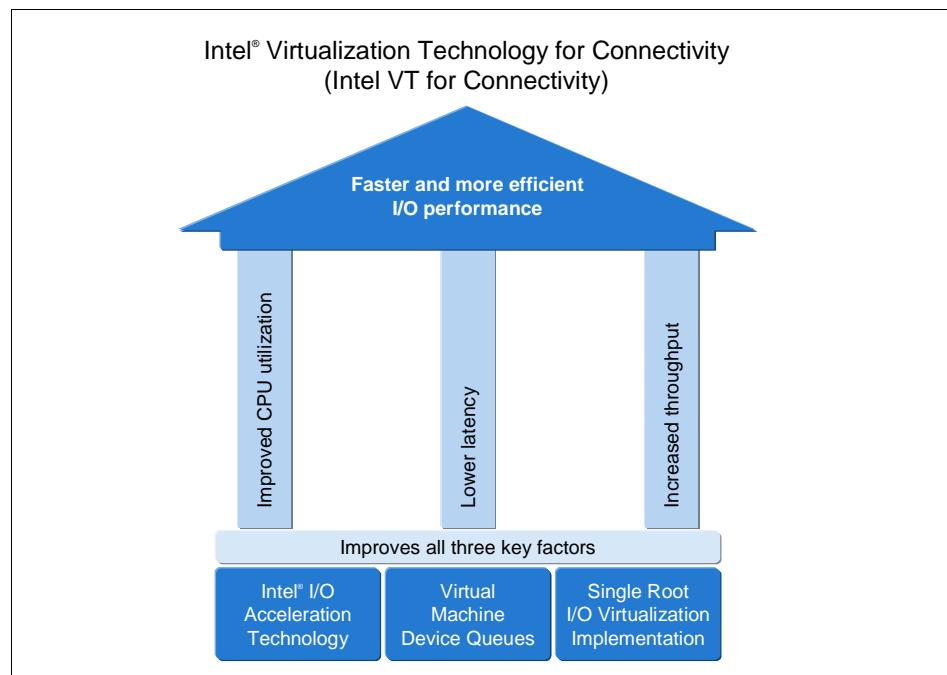


Figure 7-4 Intel VT for Connectivity

### 7.2.3 AMD-V

AMD has provided multiple additional processor extensions with 64-bit AMD Opteron Rev-F processors to increase virtualization performance. The processors already support a second level of memory address translation

(nested paging) as implemented by the components of AMD-V such as Rapid Virtualization Indexing (RVI) and Tagged TLBs.

The following list describes the latest technologies from AMD to improve virtualization performance:

- ▶ RVI, or Rapid Virtualization Indexing

RVI allows virtual machines to more directly manage memory, thus helping to improve performance on many virtualized applications. Utilizing on-die silicon resources rather than software, RVI can minimize the number of hypervisor cycles needed, as well as the associated performance penalty that is commonly associated with virtualization.

Rapid Virtualization Indexing is also designed to minimize the “world-switch time”, which is the time spent switching from one virtual machine to another, for faster application responsiveness.

RVI is supported currently by:

- Citrix XenServer 4.1
- Novell SUSE Linux 10 SP1
- Red Hat Enterprise Linux 5.2
- VMware ESX 3.5u1
- Xen 3.1

VMware refers to this technology as Nested Page Tables (NPT) or hardware virtual memory management unit (hardware virtual MMU). VMware added support beginning with ESX 3.5 Update 1.

- ▶ Integrated memory controller for memory-intensive workload

AMD has designed CPUs where the memory controller is an integrated part of CPU. This provides fast access to memory with high bandwidth throughput, low latency, and scalable access. Because utilization of resources including memory is much higher on virtualized machines where multiple guest operating systems access memory simultaneously, an integrated memory controller provides a clear benefit for performance.

- ▶ Tagged Translation Look-aside Buffer (TLB)

The Tagged Translation Look-aside Buffer (TLB), which is unique to AMD Opteron processors, allows for faster switching times between virtual machines by maintaining a mapping to the individual memory spaces used by the VMs. Distinguishing between the memory spaces used by each VM helps reduce memory management overhead and enhances responsiveness when switching between virtual machines.

- ▶ Live migration

For high availability and maintenance, it is common to use *live migration* in virtualized environments. Such function allows virtual machines, that is, guest

operating systems, to be moved live to another host. To satisfy compatibility requirements for guest operating systems to match the CPU id, VM should believe it is running on the same CPU. AMD-V Extended Migration (the AMD implementation of live migration) is designed to enable virtualization software solutions to achieve live migration of virtual machines across the entire current range of AMD Opteron processors. It is supported by multiple VMMs and allows VM to be moved from one type of hardware to another (newer or older) hardware without even stopping.

## 7.3 Support for virtualization hardware assists

To use virtualization hardware assists, you must have all of the following:

- ▶ System BIOS enablement
- ▶ Hardware technology (for example, technology within the processors for first-generation hardware assists)
- ▶ Hypervisor support

Privileged level hardware assist of processor virtualization is included in the following processors:

- ▶ Intel Xeon 5000 (Dempsey), 5100 (Woodcrest), 5300 (Clovertown) 5200 (Wolfdale), and 5400 (Harpertown) series processors.
- ▶ Intel Xeon 7000 (Paxville), 7100 (Tulsa), 7200, 7300 (Tigerton), and 7400 (Dunnington) series processors.
- ▶ New Intel VT features as VPID and EPT are included with Intel Xeon processor 5500 series.
- ▶ Processor virtualization hardware assists and new features as RVI and Tagged TLB are included in Opteron Rev F processors from AMD.

Keep in mind that the end result of these virtualization hardware assists is to reduce the complexity of the hypervisor, which can reduce overhead and improve performance significantly. There are new developments by CPU vendors to move the hardware assist for virtualization deeper. The focus is on I/O virtualization.

## 7.4 Resources

For more information on Intel technology, see the following pages:

- ▶ Intel Virtualization Technology: Hardware support for efficient processor virtualization  
<http://www.intel.com/technology/itj/2006/v10i3/1-hardware/8-virtualization-future.htm>
- ▶ Intel technology brief, *Intel Virtualization Technology for Connectivity*  
[http://softwarecommunity.intel.com/isn/downloads/virtualization/pdfs/20137\\_lad\\_vtc\\_tech\\_brief\\_r04.pdf](http://softwarecommunity.intel.com/isn/downloads/virtualization/pdfs/20137_lad_vtc_tech_brief_r04.pdf)

For more information on AMD technology, see the following pages:

- ▶ AMD Virtualization™ resource Web page:  
[http://www.amd.com/us-en/0,,3715\\_15781,00.html](http://www.amd.com/us-en/0,,3715_15781,00.html)
- ▶ AMD white paper, *Virtualizing Server Workloads*  
[http://www.amd.com/us-en/assets/content\\_type/DownloadableAssets/AMD\\_WP\\_Virtualizing\\_Server\\_Workloads-PID.pdf](http://www.amd.com/us-en/assets/content_type/DownloadableAssets/AMD_WP_Virtualizing_Server_Workloads-PID.pdf)



# PCI bus subsystem

The Peripheral Component Interconnect (PCI) bus is the predominant bus technology that is used in most Intel architecture servers. The PCI bus is designed to allow peripheral devices, such as LAN adapters and disk array controllers, independent access to main memory. PCI adapters that have the ability to gain direct access to system memory are called *bus master devices*. Bus master devices are also called *direct memory access* (DMA) devices.

To simplify the chapter, we have combined the discussion of PCI and PCI-X into one section and have outlined any differences between the two standards.

This chapter discusses the following topics:

- ▶ 8.1, “PCI and PCI-X” on page 146
- ▶ 8.2, “PCI Express” on page 149
- ▶ 8.3, “Bridges and buses” on page 153

## 8.1 PCI and PCI-X

The PCI bus is designed as a synchronous bus, meaning that every event must occur at a particular clock tick or edge. The standard PCI bus uses a 33 MHz or 66 MHz clock that operates at either 32-bit or 64-bit. With the introduction of PCI-X, the speeds have been increased to include 66 MHz, 133 MHz, 133 MHz DDR, and 133 MHz QDR. This increase has raised the maximum transfer rate in burst mode from 276 MBps to 4.2 GBps.

PCI uses a multi-drop parallel bus that is a *multiplexed address and data bus*, meaning that the address and data lines are physically the same wires. Thus, fewer signal wires are required, resulting in a simpler, smaller connector. The downside to this design is that PCI transactions must include a *turnaround phase* to allow the address lines to be switched from address mode to data mode. The PCI bus also has a data-pacing mechanism that enables fast devices to communicate with slower devices that are unable to respond to a data transfer request on each clock edge. The generic name for any PCI device is the *agent*.

A basic data transfer operation on the PCI bus is called a *PCI transaction*, which usually involves request, arbitration, grant, address, turnaround, and data transfer phases. PCI agents that initiate a bus transfer are called *initiators*, while the responding agents are called *targets*. All PCI operations are referenced from memory. For example, a PCI read operation is a PCI agent reading from system memory. A PCI write operation is a PCI agent writing to system memory. PCI transactions do not use any CPU cycles to perform the transfer.

The language of PCI defines the initiator as the PCI bus master adapter that initiates the data transfer (for example, a LAN adapter or SCSI adapter) and the target as the PCI device that is being accessed. The target is usually the PCI bridge device or memory controller.

PCI-X 1.0 and later PCI-X 2.0 are built upon the same architecture, protocols, signals, and connectors as traditional PCI. This architecture has resulted in maintaining hardware and software compatibility with the previous generations of PCI. This design means that devices and adapters that are compliant with PCI-X 1.0 are fully supported in PCI-X 2.0.

When supporting previous PCI devices, it is important to note that the clock must scale to a frequency that is acceptable to the lowest speed device on the bus. This results in all devices on that bus being restricted to operating at that slower speed.

PCI-X was developed to satisfy the increased requirements of I/O adapters such as Gigabit Ethernet, Fibre Channel, and Ultra320 SCSI. PCI-X is fully compatible with standard PCI devices. It is an enhancement to the conventional PCI

specification V2.2 and enables a data throughput of over 4 GBps at 533 MHz/64-bits in burst mode.

Adapters with high I/O traffic, such as Fibre Channel and storage adapters, benefit significantly from PCI-X. These adapters provide a huge amount of data to the PCI bus and, therefore, need PCI-X to move the data to main memory.

Peak throughput has increased from PCI to PCI-X and this results in increased throughput. Other changes made in PCI-X that provide higher efficiency and, therefore, a performance benefit when compared to standard PCI include:

- ▶ Attribute phase

The attribute phase takes one clock cycle and provides further information about the transaction. PCI-X sends new information with each transaction performed within the attribute phase, which enables more efficient buffer management.

- ▶ Split transactions

Delayed transactions in conventional PCI are replaced by split transactions in PCI-X. All transactions except memory-write transactions are allowed to be executed as split transactions. If a target on the PCI bus cannot complete a transaction within the target initial latency limit, the target must complete the transaction as a split transaction. Thus, the target sends a split response message to the initiator telling it that the data will be delivered later on. This frees the bus for other communications.

- ▶ Allowable disconnect boundary

When a burst transaction is initiated to prevent a single process from monopolizing the bus with a single large transfer (bursts can be up to 4096 bytes), PCI-X gives initiators and targets the chance to place interruptions. The interruptions are not placed randomly (which might compromise the efficiency of the buffers and cache operations), but are fixed on 128-byte boundaries, which is a figure that is big enough to facilitate complete cache line transmissions.

The benefit of adopting the PCI-X standard is the increase in supported throughputs, which is evident with the 533 MHz implementation. When running at higher frequencies (133 MHz and higher), only one device can be on a PCI-X bus, making PCI-X a high-bandwidth point-to-point I/O channel. At lower speeds (less than 133 MHz), multiple devices can be connected on a single bus.

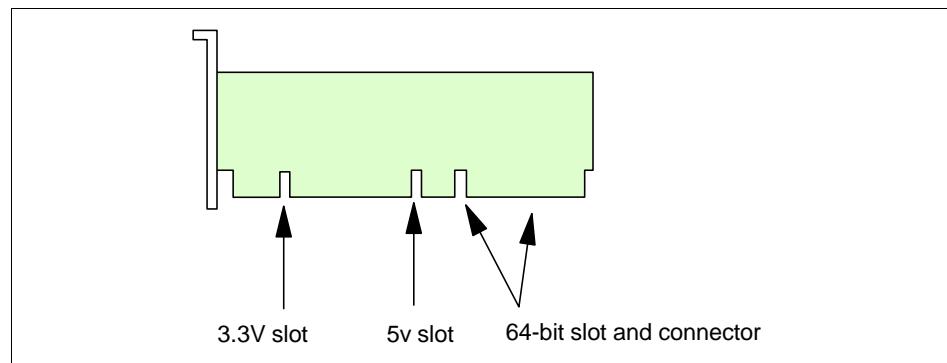
Note that the 66 MHz implementation of PCI-X doubles the number of slots supported on a current PCI 2.2 66 MHz bus. Table 8-1 shows the possible combinations of PCI modes and speeds.

*Table 8-1 PCI and PCI-X modes*

Mode	PCI Voltage (V)	64-bit		32-bit		16-bit
		Max slots	MBps	Max slots	MBps	MBps
PCI 33	5 or 3.3	4	266	4	133	Not applicable
PCI 66	3.3	2	533	2	266	Not applicable
PCI-X 66	3.3	4	533	4	266	Not applicable
PCI-X 133 <sup>a</sup>	3.3	2	800	2	400	Not applicable
PCI-X 133	3.3	1	1066	1	533	Not applicable
PCI-X 266	3.3 or 1.5	1	2133	1	1066	533 MBps
PCI-X 533	3.3 or 1.5	1	4266	1	2133	1066 MBps

a. Operating at 100 MHz

PCI-X devices use 3.3V I/O signalling when operating in PCI-X mode. They also support the 5V I/O signalling levels when operating in 33 MHz conventional mode, which results in cards either designed specifically for 3.3V PCI-X or universally keyed. Figure 8-1 illustrates adapter keying.



*Figure 8-1 Adapter keying*

PCI-X cards are designed to run at either 66 MHz or 133 MHz. PCI-X cards are not designed usually to run at 100 MHz. However, the number of loads on the bus can force a 133 MHz adapter to operate at 100 MHz.

All but a few servers in the System x line now use PCI Express instead of PCI-X. We discuss PCI Express in the next section.

## 8.2 PCI Express

PCI Express is the latest development in PCI to support adapters and devices. The technology is aimed at multiple market segments, meaning that it can be used to provide for connectivity for chip-to-chips, board-to-boards, and adapters.

PCI Express uses a serial interface and allows for point-to-point interconnections between devices using directly wired interfaces between these connection points. This design differs from previous PCI bus architectures which used a shared, parallel bus architecture.

A single PCI Express serial link is a dual-simplex connection that uses two pairs of wires (one pair for transmit and one pair for receive), and that transmits only one bit per cycle. Although this design sounds limiting, it can transmit at the extremely high speed of 2.5 Gbps, which equates to a burst mode of 320 MBps on a single connection. This connection of two pairs of wires is called a *lane*.

A PCI Express *link* is comprised of one or more lanes. In such configurations, the connection is labeled as x1, x2, x4, x12, x16, or x32, where the number is effectively the number of lanes. So, where PCI Express x1 would require four wires to connect, an x16 implementation would require 16 times that amount (64 wires). This implementation results in physically different-sized slots.

**Tip:** When referring to lane nomenclature, the word *by* is used, as in *by 8* for *x8*.

Figure 8-2 on page 150 shows the slots for a 32-bit PCI 2.0, PCI Express x1, and a PCI Express x16. From this figure, it is clear that the PCI Express x16 adapter will not fit physically in the PCI x1 slot.

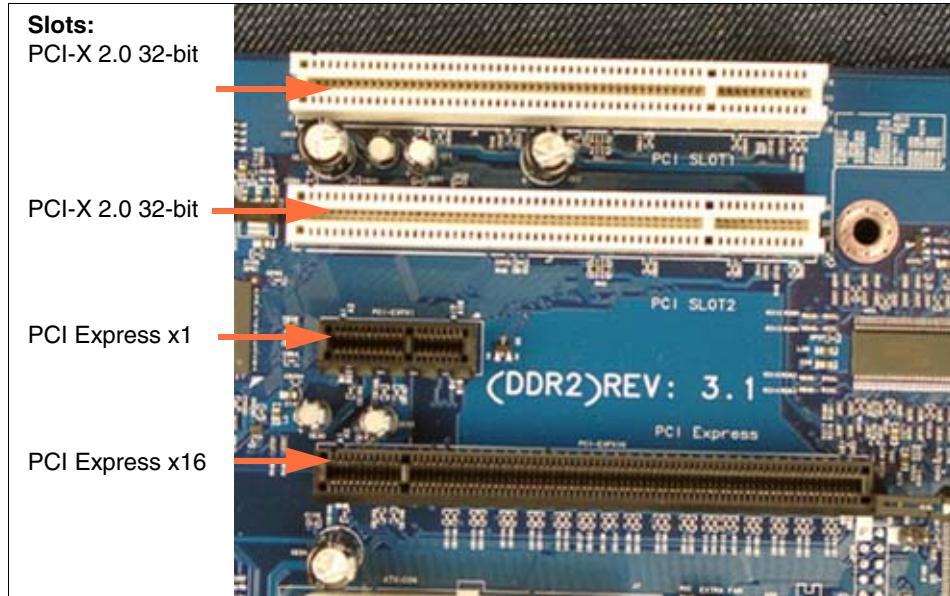


Figure 8-2 PCI 2.0 and PCI Express edge connectors

You can install PCI Express adapters in larger slots but not in smaller ones. For example, you can install a PCI Express x8 adapter into an x16 slot (although it will still operate at the x8 speed), but you cannot insert an x8 adapter into an x4 slot. Table 8-2 shows this compatibility.

Table 8-2 PCI Express slot compatibility

	x1 slot	x4 slot	x8 slot	x16 slot
x1 card	Supported	Supported	Supported	Supported
x4 card	No	Supported	Supported	Supported
x8 card	No	No	Supported	Supported
x16 card	No	No	No	Supported

Typically, the size of a slot matches the number of lanes it has. For example, a x4 slot typically is a x4 link (that is, it has 4 lanes). However, this is not always the case. The PCI Express specification allows for the situation where the physical connector is larger than the number of lanes of data connectivity. The only requirement on manufacturers is that the connector must still provide the full complement of power and ground connections as required for the connector size.

For example, in the System x3650, there are two pairs of slots:

- ▶ Two slots labelled “PCI Express x8 (x8 lanes)”
- ▶ Two slots labelled “PCI Express x8 (x4 lanes)”

The first pair are PCI Express with x8 physical connectors (in other words, they will physically accept x8 cards, as well as x4, x2 and x1 cards), and they have the bandwidth of a x8 link (8x 2.5 Gbps or 20 Gbps). The second pair are also PCI Express with x8 physical connectors, but only have the bandwidth of a x4 link (4x 2.5 Gbps or 10 Gbps).

If you have a need for x8 bandwidth (such as for an InfiniBand or Myrinet adapter), then ensure you select one of the correct slots (the ones with x8 lanes). It is important to understand this naming convention because it will have a direct impact on performance if you select a slot that is slower than the maximum supported by the adapter.

**Tip:** The physical size of a PCI Express slot is not the sole indicator of the possible bandwidth of the slot. You must determine from slot descriptions on the system board or the service label of the server what the bandwidth capacity is of each slot.

While the underlying hardware technology is different between PCI-X and PCI Express, they remain compatible at the software layer. PCI Express supports existing operating systems, drivers, and BIOS without changes. Because they are compatible at the level of the device driver model and software stacks, PCI Express devices look just like PCI devices to software.

A benefit of PCI Express is that it is not limited for use as a connector for adapters. Due to its high speed and scalable bus widths, you can also use it as a high speed interface to connect many different devices. You can use PCI Express to connect multiple onboard devices and to provide a fabric that is capable of supporting USB 2, InfiniBand, Gigabit Ethernet, and others.

### 8.2.1 PCI Express 2.0

PCI Express 2.0 specification was made available in 2007 and has been implemented in some of today's server chipsets, including the Intel 5400, as a high speed serial I/O connection. A number of improvements have been made to the protocol, software layers and the signal of the PCI Express architecture in the PCI Express 2.0 specification. Key features include:

- ▶ Doubling the interconnect bit rate of lane from 2.5 GT/s to 5 GT/s to offer twice the bandwidth of PC Express 1.1

- ▶ Redefining power limits to enable slot power limit values to accommodate devices that consume higher power, meaning more power can be supplied to PCI adapters
- ▶ Maintaining compatibility with previous standards, with support of PCI-Express 1.x

### 8.2.2 PCI Express performance

PCI Express 1.x runs at 2.5 Gbps or 200 MBps per lane in *each* direction, providing a total bandwidth of 80 Gbps in a 32-lane configuration and up to 160 Gbps in a full duplex x32 configuration. PCI Express 2.0 doubles the bit rate of lanes and achieves up to 320 Gbps in a full duplex x32 configuration.

Future frequency increases will scale up total bandwidth to the limits of copper (which is 12.5 Gbps per wire) and significantly beyond that through other media without impacting any layers above the physical layer in the protocol stack.

Table 8-3 shows the throughput of PCI Express at different lane widths.

*Table 8-3 PCI Express maximum transfer rates*

PCI-E 1	PCI-E 2.0	Throughput (duplex, bits)	Throughput (duplex, bytes)	Initial expected uses
x1	None	5 Gbps	400 MBps	Slots, Gigabit Ethernet
x2	x1	10 Gbps	800 MBps	None
x4	x2	20 Gbps	1.6 GBps	Slots, 10 Gigabit Ethernet, SCSI, SAS
x8	x4	40 Gbps	3.2 GBps	Slots, InfiniBand adapters, Myrinet adapters
x16	x8	80 Gbps	6.4 GBps	Graphics adapters
None	x16	160 Gbps	12.8 GBps	Graphics adapters

PCI Express uses an embedded clocking technique that uses 8b/10b encoding. The clock information is encoded directly into the data stream, rather than having the clock as a separate signal. The 8b/10b encoding essentially requires 10 bits per character or about 20% channel overhead. This encoding explains differences in the published specification speeds of 250 MBps (with the embedded clock overhead) and 200 MBps (data only, without the overhead). For ease of comparison, Table 8-3 shows throughput in both bps and Bps.

When compared to the current version of a PCI-X 2.0 adapter running at 133 MHz QDR (quad data rate, effectively 533 MHz), the potential sustained

throughput of PCI Express 2.0 x8 is more than double the throughput, as shown in Figure 8-3.

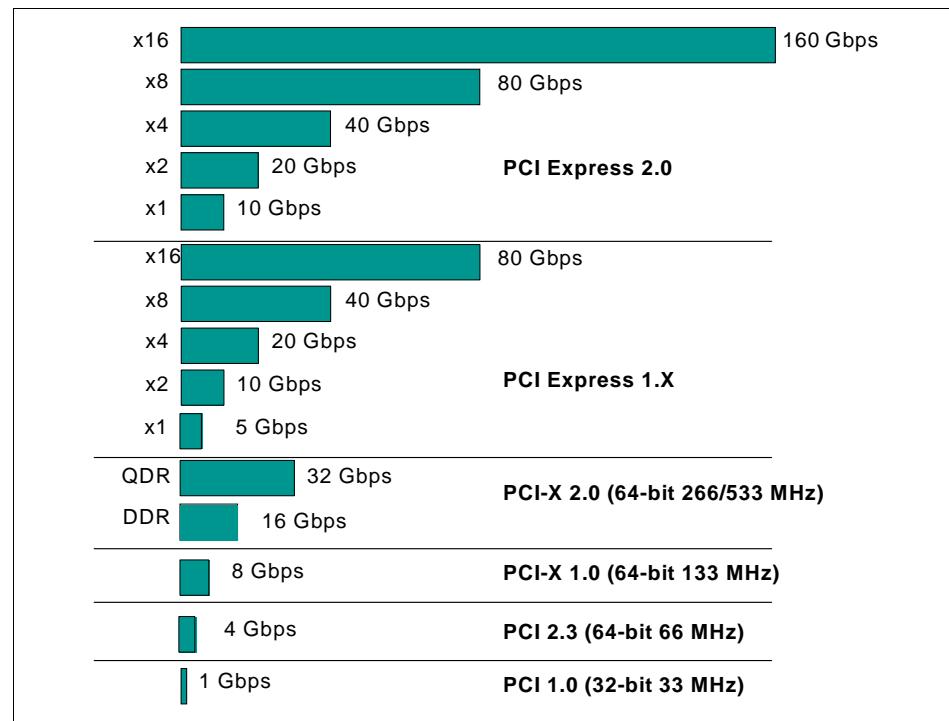


Figure 8-3 PCI Express and PCI-X comparison (in Gbps)

## 8.3 Bridges and buses

When PCI first appeared on the market, systems were limited to two or three PCI slots. This limitation was due to the signal limitations of the PCI bus. To overcome this limitation, the concept of the PCI-to-PCI (PtP) bridge was developed. Early implementations of the PtP bridge involved a primary bus and a secondary bus. Access to devices on the secondary bus was typically slower as the I/O requests negotiated the bridge.

Because PCI Express is point-to-point, as opposed to multiplexed parallel, the requirement to interface with multiple edge connectors through a bridge does not exist. In essence, the PCI Express slot interfaces directly with its controllers, which are integrated in the memory controller or I/O hub through a series of channels. This type of interface means that bandwidth to the edge connectors does not need to be managed in the same way.

With PCI-X, the aggregate speed of the edge connectors cannot exceed the allocated bandwidth between the memory controller and the PCI bridge. This places a limitation on the number and combinations of speeds of PCI slots that can be supported on a single bridge. Removing the requirement to connect PCI cards through a bridge also reduces latency because the data has one less hop to travel.

**Note:** It is important to remember that the primary function is to transfer data from the adapter into memory (through DMA) as quickly as possible so that it can be processed by the CPU. PCI Express transfers data faster by reducing the number of hops to memory and increasing throughput.

Figure 8-4 illustrates how the PCI Express slots connect directly to the memory controller while the PCI-X edge connectors connect to the memory controller through a PCI bridge. The x3650 implements these slots on replaceable riser cards.

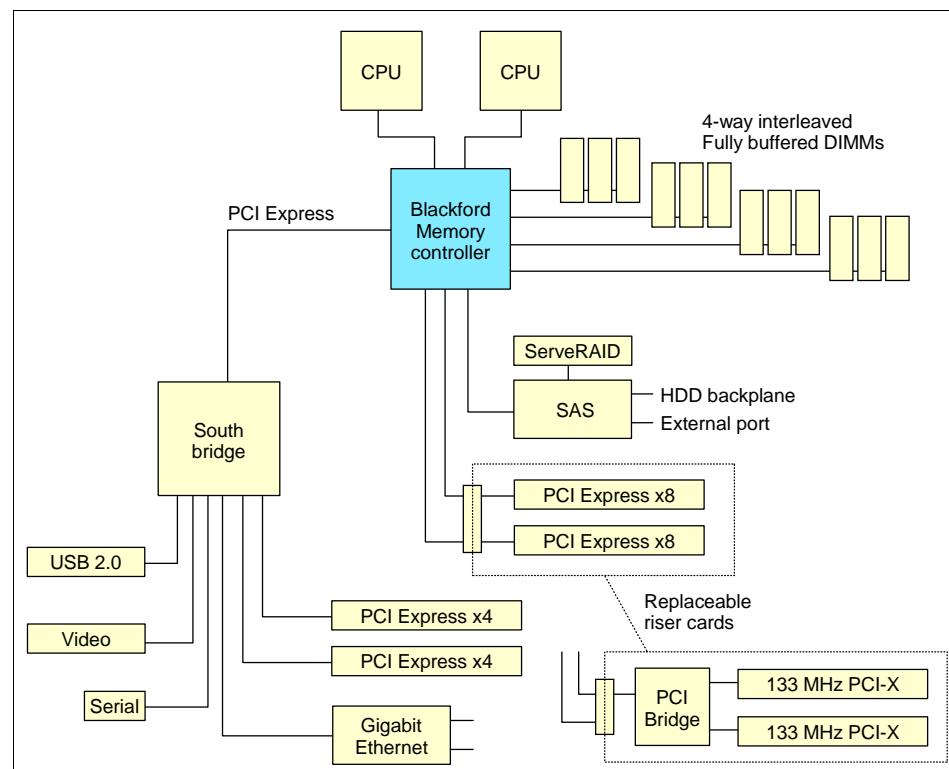


Figure 8-4 System x3650 block diagram with PCI Express or PCI-X riser cards

The IBM System x3850 M2 and x3950 M2 use a PCI Express design that employs two I/O bridges (Cal IOC2) to provide direct PCI Express connections, as shown in Figure 8-5.

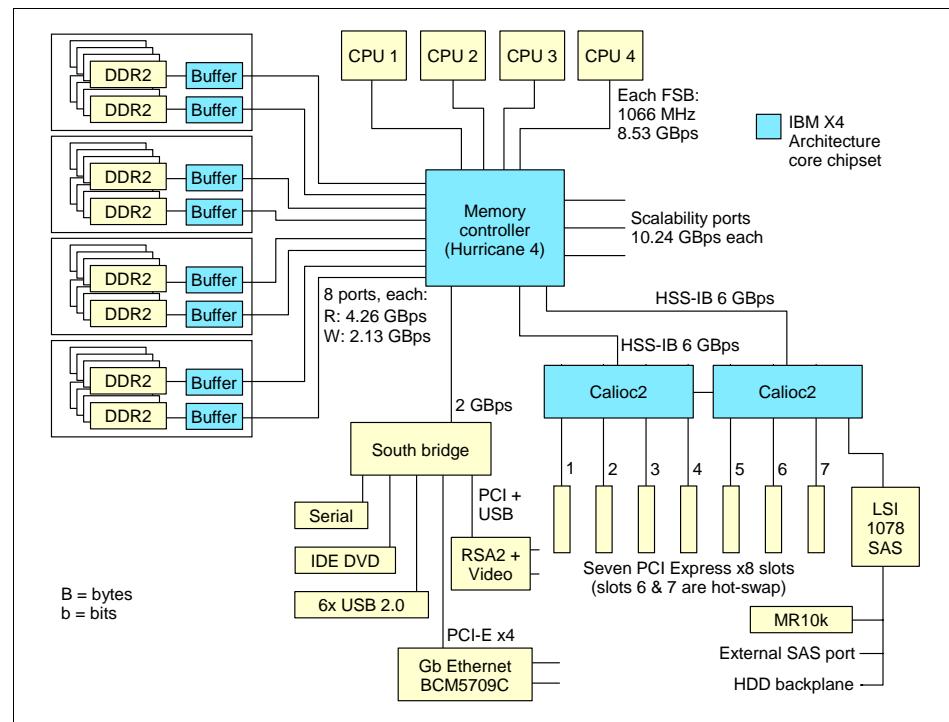


Figure 8-5 x3850 M2 block diagram





# Chipset architecture

The chipset architecture implements the control and data flow between the processor, memory, PCI devices, and system buses. Chipsets are varied in functionality and performance bottlenecks. Other functions such as video, keyboard, interrupt, diskette, and clock are provided by support chips.

This chapter discusses the following topics:

- ▶ 9.1, “Overview of chipsets” on page 158
- ▶ 9.2, “System architecture design and performance” on page 159
- ▶ 9.3, “Memory controller-based chipset” on page 171
- ▶ 9.4, “PCI bridge-based chipsets” on page 177

## 9.1 Overview of chipsets

Although processor performance has been increasing rapidly, improvements to memory have not been as dramatic. Increases in the working set sizes for software have caused larger memory footprints, which in turn have necessitated larger caches and main memory. A by-product of the increased cache size is a higher latency to access main memory. The chipset controls the flow of data between the processors, external caches, and memory, which makes the chipset an integral part in controlling system-wide latency.

System and chip designers generally use a key metric known as Cycles Per Instruction (CPI) to measure the number of processor clocks that a system uses to execute an instruction. Although the number of instructions to execute an operation is held constant, a decrease in the number of cycles combined with a higher clock rate provides a measurable increase in performance. Many workloads, particularly the random nature of workloads that are prevalent in servers, have frequent cache misses. As a result, a greater component of the CPI for server class workloads are dependent on the chipset and memory subsystem rather than on the core processor. Thus, the chipset is a major contributor to the overall performance to a system.

Table 9-1 lists the chipsets that current IBM System x and BladeCenter servers use. We discuss the chipsets that are listed in **bold** font in detail in this chapter.

*Table 9-1 Chipsets that System x and BladeCenter servers use*

Server	Chipset
System x servers	
x3400	<b>Intel 5000P chipset (page 171)</b>
x3450	<b>Intel 5400 chipset (page 173)</b>
x3455	Broadcom HT2100 PCI-E bridge chip
x3500	<b>Intel 5000P chipset (page 171)</b>
x3550	<b>Intel 5000X chipset (page 171)</b>
x3650	<b>Intel 5000P chipset (page 171)</b>
x3655	ServerWorks HT2100 PCI-E bridge chip
x3650T	Intel E7520
x3755	ServerWorks HT2100 PCI-E bridge chip
x3800	IBM XA-64e third-generation chipset

Server	Chipset
x3850	IBM XA-64e third-generation chipset
x3850 M2	<b>IBM XA-64e fourth-generation chipset (page 175)</b>
x3950	IBM XA-64e third-generation chipset
x3950 M2	<b>IBM XA-64e fourth-generation chipset (page 175)</b>
BladeCenter servers	
HS12	Intel 5100 chipset
HS21	<b>Intel 5000P chipset (page 171)</b>
JS21	BCM5780 PCI-E, HyperTransport Tunnel
LS21 / LS41	ServerWorks HT-2000 HT
LS22 / LS42	ServerWorks HT-2100 HT

**Tip:** The System x Reference (xREF) is a set of one-page specification sheets for each of the System x server models. It includes details of the chipsets used. xREF is available from:

<http://www.redbooks.ibm.com/xref>

## 9.2 System architecture design and performance

Today's server workloads tend to grow larger and larger and require more and more CPU and memory resources. Depending on the application, you have two options to meet these increasing demands:

- ▶ Scaling out (many smaller servers)
- ▶ Scaling up (one larger server)

Several applications allow you to scale out. A typical example is Web site hosting. Large Web sites are not hosted on a single large SMP server. Instead, they are hosted by a number of one-socket or two-socket servers using a distributed workload model. In general, this method is efficient because much of the data is read-only and not shared across many concurrent sessions. This fact enables scale-out computing to experience improved performance as nodes are added because each server can provide read-only data to unique concurrent users, independent of any other machine state.

There are also a number of applications, such as virtualization and database environments, that scale up. Scale-up refers to the idea of increasing processing

capacity by adding additional processors, memory, and I/O bandwidth to a single server, thus making it more powerful. A typical scale-up server for such an application is a multi-processor system such as the System x3950.

However, hardware scalability is only one aspect of building scalable multiprocessor solutions. It is paramount that the operating system, driver, and application scale just as well as the hardware. In this section, we first explore the main concepts of multi-processing and then examine software scalability.

### 9.2.1 Hardware scalability

When considering CPUs, cache, and memory in an SMP configuration, memory is the most frequently used and also has the greatest latency of the shared hardware resources. Hardware scalability is usually defined by how efficiently CPUs share memory, because the fast CPUs must frequently access the slower memory subsystem.

High-speed caches are used to accelerate access to memory objects that the CPU uses most frequently, but performance gains that are obtained by high-speed caches introduce problems that can often limit multi-processor hardware scalability.

There are two architectures available in System x servers with multiple processors: SMP and NUMA.

### 9.2.2 SMP

Most Intel-compatible systems are designed using an SMP, or symmetric multiprocessing, architecture. Designing a system to use multiple concurrently-executing CPUs is a complex task. The most popular method is to design the system so that all CPUs have symmetric access to all hardware resources such as memory, I/O bus, and interrupts, thus the name symmetric multiprocessing.

SMP is most popular because it simplifies the development of the operating system and applications. Using SMP, each CPU “sees” the same hardware resources, so no special software techniques are needed to access any resource. Therefore, SMP hardware scalability is directly related to how efficiently the many CPUs use the shared hardware resources.

One disadvantage of SMP is the limited scalability of this architecture. As processors are added to the system, the shared resources are frequently accessed by an increasingly greater number of processors. More processors using the same resources creates queuing delays similar to many people trying to pay in a market that has only one cashier. Although the single cashier is

symmetric, meaning that everyone has to pay in the same location thus making it easy to locate, the disadvantage is that everyone must wait in the same queue.

### 9.2.3 NUMA

The non-uniform memory access (NUMA) architecture is a way of building very large multi-processor systems without jeopardizing hardware scalability. The name NUMA is not completely correct because not only memory can be accessed in a non-uniform manner, but also I/O resources.

NUMA effectively means that every processor or group of processors has a certain amount of memory local to it. Multiple processors or multiple groups of processors are then connected together using special bus systems (for example, the HyperTransport links in the AMD-based System x3755 or the scalability ports of the Xeon-based System x3950 M2) to provide processor data coherency. The essence of the NUMA architecture is the existence of multiple memory subsystems, as opposed to a single one on an SMP system.

So-called *local* or *near* memory has the same characteristics as the memory subsystem in an SMP system. However, by limiting the number of processors that access that memory directly, performance is improved because of the much shorter queue of requests. Because each group of processors has its local memory, memory on another group of processors is considered remote to the local processor. This remote memory can be accessed but at a longer latency than local memory. All requests between local and remote memory flow over the inter-processor connection (HyperTransport or scalability ports).

Consider a two-node System x3950 M2 configuration with a total of eight processors and 128 GB of memory. Each x3950 M2 has four CPUs and 64 GB of RAM, as shown in Figure 9-1 on page 162. These two systems are connected together using two 10 GBps scalability connections.

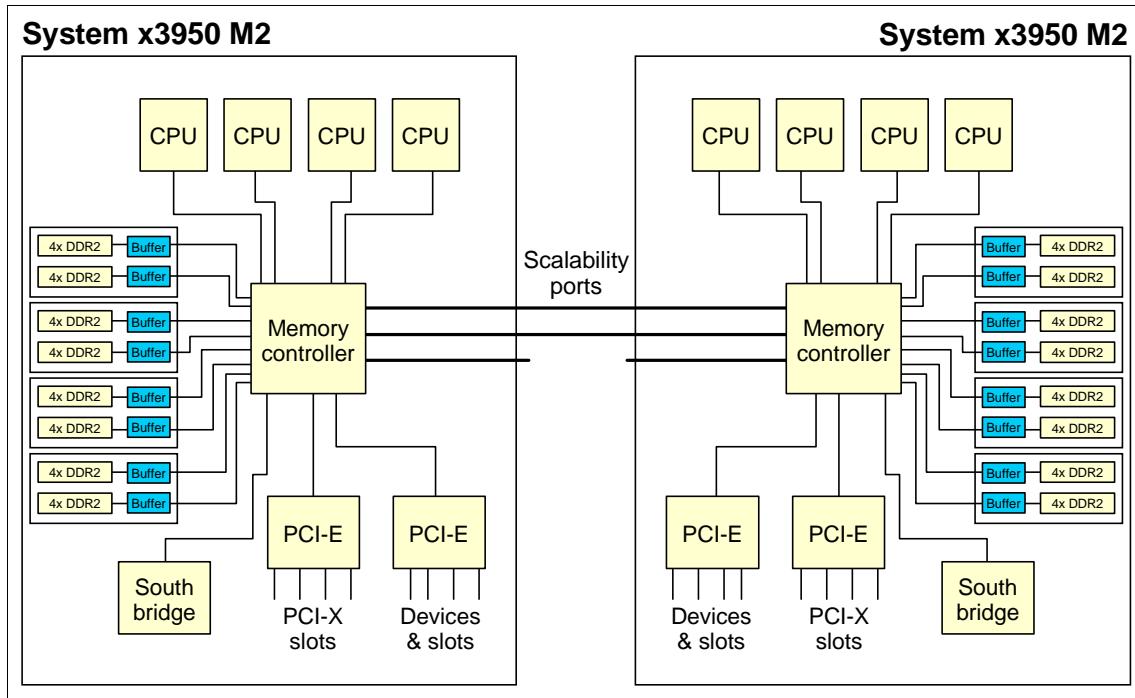


Figure 9-1 A two-node x3950M2 configuration

An application that is running on CPUs in one server node can access memory that is located physically in the other node (a *remote access*). This access incurs longer latency because the travel time to access remote memory on another expansion module is greater.

Many people think access latency is a problem with NUMA. However, this focus on latency misses the actual issue that NUMA is attempting to solve.

Another way to think about it is to imagine the following scenario. You are paying for your groceries in your favorite grocery store. Directly in front of you is a cashier with 20 customers standing in line, but 50 feet to your left is another cashier with only two customers standing in line. Which cashier would you choose? The cashier closest to your position has the lowest latency because you do not have far to travel. However, the cashier 50 feet away has much greater latency because you have to walk 50 feet.

Generally, most people would walk the 50 feet and suffer the latency to arrive at a cashier with only two customers instead of 20. We think this way because our experience tells us that the time waiting to check out with 20 people ahead is far longer than the time needed to walk to the “remote” cashier and wait for only two people ahead.

This analogy communicates the performance effects of queuing time versus latency. In a computer server, with many concurrent outstanding memory requests, we would gladly incur some additional latency (walking) to spread memory transactions (paying for our groceries) across multiple memory controllers (cashiers) because this improves performance greatly by reducing the queuing time.

We do not want to walk 50 feet to a cashier that has 20 customers paying when one is directly in front of us with only two customers. So, to reduce unnecessary remote access, NUMA systems such as the System x3950 M2 maintain a table of data in the firmware call the Static Resource Allocation Table (SRAT). The data in this table is accessible by operating systems such as Windows Server 2003 and 2008 (Windows 2000 Server does not support it) and current Linux kernels.

These modern operating systems attempt to allocate resources that are local to the processors that are used by each process. So, when a process and its threads start on node 0, all execution and memory access are local to node 0. As more processes are added to the system, the operating system balances them across the nodes. In this case, most memory accesses are evenly distributed across the multiple memory controllers, thus reducing remote access, greatly reducing queuing delays, and improving performance.

The AMD Opteron implementation is called Sufficiently Uniform Memory Organization (SUMO), and it is a NUMA architecture. In the case of the Opteron, each processor has its own *local* memory with low latency. Every CPU can also access the memory of any other CPU in the system, but with some latency.

## **NUMA optimization for Windows Server**

The Enterprise and Datacenter editions of Windows Server 2003 and 2008 are optimized for NUMA, as listed in Table 9-2 on page 164. Windows Server obtains the NUMA information from the Static Resource Affinity Table (SRAT) in the system BIOS while booting. That is, NUMA architecture servers must have the SRAT to use this function. Windows Server cannot recognize system topology without the SRAT.

Table 9-2 Versions of Windows Server optimized for NUMA

	x86 (32-bit)	x64 (64-bit)	IA64 (64-bit)
Windows 2003 Web Edition	No	Not applicable	Not applicable
Windows Server 2003 Standard Edition	No	No	Not applicable
Windows Server 2003 Enterprise Edition	Yes	Yes	Yes
Windows Server 2003 Datacenter Edition	Yes	Yes	Yes
Windows Server 2008 Standard Edition	No	No	Not applicable
Windows Server 2008 Enterprise Edition	Yes	Yes	Not applicable
Windows Server 2008 Datacenter Edition	Yes	Yes	Not applicable
Windows Server 2008 Web Edition	No	No	Not applicable

## NUMA optimization in Linux

The 2.6 kernel features NUMA awareness in the scheduler (the part of the operating system that assigns system resources to processes) so that the vast majority of processes execute in local memory. This information is passed to the operating system through the ACPI interface and the SRAT, similar to the Windows Operating System.

## Static Resource Affinity Table

The Static Resource Affinity Table (SRAT) includes topology information for all the processors and memory in a system. The topology information includes the number of nodes in the system and which memory is local to each processor. By using this function, the NUMA topology recognized by the operating system. The SRAT also includes hot-add memory information. Hot-add memory is the memory that can be hot-added while the system is running, without requiring a reboot.

The Advanced Configuration and Power Interface (ACPI) 2.0 specification introduces the concept of *proximity domains* in a system. Resources, including processors, memory, and PCI adapters in a system, are tightly coupled, and the operating system can use this information to determine the best resource allocation and the scheduling of threads throughout the system. The SRAT is based on this ACPI specification.

You can learn more about the SRAT table at:

<http://www.microsoft.com/whdc/system/CEC/SRAT.mspx>

The SRAT is automatically configured in systems such as the x3950M2 in firmware. For other systems, you should enable the SRAT information in the

system BIOS (if this is configurable) and run a NUMA-aware operating system. Keep in mind that many applications require at least two to four processors to reach maximum performance.

In this case, even with NUMA-aware operating systems, there can be a high percentage of remote memory accesses in an Opteron system because each processor is the only processor on a node. The frequency of NUMA access depends upon the application type and how users apply that application; it cannot be estimated without extensive analysis.

**Tip:** In the IBM eServer™ 326 system BIOS, enable the ACPI SRAT, disable Node Interleave, and set the DRAM Interleave to Auto to achieve the best performance in conjunction with a NUMA-aware operating system.

#### 9.2.4 The MESI protocol

A complicating factor for the design of any SMP system is the need to keep all CPU and cache data coherent. Because each CPU has access to the same data that is stored in memory, two or more CPUs should not modify the same data at the same time. This concurrent modification of the same data can cause unpredictable results. Furthermore, problems can also occur when one CPU is modifying data that another CPU is reading.

A protocol called MESI is employed on all Intel multi-processor configurations to ensure that each CPU is guaranteed to get the most recent copy of data even when other CPUs are currently using that data. MESI stands for *modified*, *exclusive*, *shared*, and *invalid*—the four possible data states that data can have when stored in a processor cache. One of these four states is assigned to every data element stored in each CPU cache.

To support the MESI protocol, regular communication must occur between every CPU whenever data is loaded into a cache. On each processor data load into cache, the processor must broadcast to all other processors in the system to check their caches to see if they have the requested data. These broadcasts are called *snoop* cycles, and they must occur during every memory read or write operation.

During the snoop cycle phase, each CPU in the SMP server checks its cache to see if it has the data that is being addressed by the requesting CPU. If the data is present in another CPU cache and the data has been modified, then that CPU must provide the data to the other requesting CPU. If the data is in some other CPU cache but it has not been modified, then that CPU must mark its data as shared or invalid, depending upon the type of operation that is requested on the front-side bus.

If the operation is a write request, the CPU that possesses the unmodified data must mark its data as invalid, indicating that the data can no longer be used. If the front-side bus request is a read operation, the data is marked as shared, indicating that its copy is read-only and cannot be modified without notifying the other CPUs. In this case, the CPU that generated the front-side bus read request will also mark its copy of the data in its cache as shared. If either CPU should then execute an instruction to modify the data (a write instruction), another front-side bus cycle occurs to inform the other CPUs to invalidate the data in any of their caches. At the completion of the snoop cycle for the write operation, the CPU updating the data marks the data as modified.

The exclusive state is used to indicate that data is stored in only one cache. Data that is marked exclusive can be updated in the cache without a snoop broadcast to the other CPUs. This is possible because at the time the data was read from memory, no other CPUs indicated that they had ownership of the same data in their caches. Therefore, the MESI state of the data was set to exclusive. Unless another CPU generated a front-side bus request for the data (in which case the data would be marked as shared), it would stay exclusive. If the data were modified, the write operation that performed the update would cause the state of the data to be set to modified. Any subsequent requests for the modified data would be satisfied by the cache providing the modified data to the requesting CPU.

The complete MESI protocol is quite complex and you do not need to understand all of its details to appreciate its impact on SMP scalability. We simply introduce the protocol here so that you can be aware of the overhead that is required each time CPUs are added to an SMP server.

## The MOESI protocol

The AMD Opteron uses a slightly different version of the MESI protocol: MOESI. The MOESI protocol expands the MESI protocol with yet another cache line status flag, namely the *owner* status (thus the O in MOESI).

After the update of a cache line, the line is not written back to system memory but instead is flagged as an owner. When another CPU issues a read, it gets its data from the owner's cache rather than from slower memory, thus improving memory performance in a multiprocessor system.

For more information about MOESI, see the *Chip Architect* article about the AMD64 processor at:

[http://chip-architect.com/news/2003\\_09\\_21\\_Detailed\\_Architecture\\_of\\_AMDs\\_64bit\\_Core.html#3.18](http://chip-architect.com/news/2003_09_21_Detailed_Architecture_of_AMDs_64bit_Core.html#3.18)

## **Large SMP configurations**

In general, snoop overhead increases as the number of CPUs increases. Snoop overhead can also increase as cache size increases. The larger the L2 cache, the greater the probability that snoop requests will hit data in another processor cache. These cycles delay the execution rate of the CPUs, because they must wait for the processor with ownership of the data to provide the data to the requesting CPU.

Dramatic increases in front-side bus speed plus Hyper-Threading and multiple processor cores sharing the same front-side bus have caused issues that are related to snoop latency. When the number of CPU cores increases to eight or more, unless special optimization such as in the XA-64e chipset are made, CPU performance can slow down. Many of the early Pentium Pro eight-socket SMP systems performed more slowly with eight CPUs than with four processors. This was due primarily to the long snoop latencies that resulted from each CPU juggling shared and modified data.

The snoop latency issue is exacerbated for many Intel SMP eight-socket configurations because these systems actually consist of 2 four-socket systems connected by a specially designed dual-ported memory controller. This architecture is necessary because the Intel front-side bus protocol is limited to four processor sockets.

To increase the number of CPUs beyond four, two or more independent front-side buses must somehow be connected together. The front-side bus-to-front-side bus connection had the potential to introduce overhead, which often meant the time to snoop the CPU caches on a remote front-side bus was much longer than the time to snoop the CPU caches on the local front-side bus. In general, this explains why there is a discontinuity in performance improvement from the fifth CPU compared to the gains that are obtained from the first to the fourth CPU.

The increases in the number of processor threads which include the additional cores plus the Hyper-Threading has changed the traditional methodology for designing processors. New architectures are starting to incorporate multiple front-side buses per memory controller. The penalty to snoop a processor on a second front-side bus heavily increases the front-side bus utilization.

This penalty is the reason why many architectures are beginning to incorporate a snoop filter similar to the XA-64e chipset in their designs. High front-side bus utilization explains why performance is optimal when splitting CPUs across two front-side busses, instead of fully populating a single front-side bus. Solutions to the performance problem are the use of the cache coherency filter or directory and “higher” levels of cache.

## **Cache coherency filter**

One significant hardware optimization to enhance the performance of high-end systems is the *cache coherency filter*. Typically, one filter is used for each group of four processors. To think of this another way, each filter is used to track all the operations that occur on the front-side bus. The filter provides information about the addresses of all data that is stored in the caches of the CPUs on the respective front-side bus.

These filters are used to store bits that indicate the presence of each cache line that is stored in all the caches of the processors. Whenever an address is snooped by a CPU, the memory controller looks up the address in the filter for the remote front-side bus (without an actual cycle to the remote front-side bus). If the remote filter responds with a hit, only then is the snoop cycle propagated to the remote front-side bus. If an address that is snooped is not present in the filter, then a snoop miss occurs, and the snoop completes quickly because it does not propagate to the remote bus.

Remember, the CPU that requests the data that caused the snoop cycle might be waiting for the snoop cycle to complete. Furthermore, the front-side bus cannot be used by other CPUs during the snoop cycle, so snoop cycles must execute quickly to obtain CPU scalability.

### **9.2.5 Software scalability**

Adding processors improves server performance because software instruction execution can be shared among the additional processors. However, the addition of processors requires software to detect the additional CPUs and generates additional work in the form of threads or processes that execute on the additional processors. The operating system provides a platform that enables the capability of multiprocessing, but it is up to the application to generate the additional threads and processes to execute on all processors. This ability is referred to as *application scalability*.

Faster server hardware means more parallelism (more processors, larger memory, larger disk arrays, additional PCI buses, and so on). The obvious case of software that does not scale is DOS. If you run DOS on a server with 8 CPUs and 64 GB of memory that is equipped with 250 15 K RPM disk arrays, you get about the same performance as though you have one CPU, one disk, and 640 KB of memory. Obviously, the server is not slow. The problem is that the software (in this case, DOS) does not scale. This example is extreme, but it makes it easier to understand how software must actually evolve to take advantage of more powerful server hardware.

Software scalability is a complex subject, and one that most people do not consider until it is too late. Often people purchase new high-performance servers

expecting huge performance gains with old applications, only to learn that the bottleneck is in the server application. In this case, there is little that they can do to efficiently use the new server until the application is modified.

A scalable application makes use of greater amounts of memory, generates scalable I/O requests as the number of disks in a disk array increases, and will use multiple LAN adapters when a single LAN adapter limits bandwidth. In addition, a scalable application has to detect the number of installed processors and spawn additional threads as the number of processors increases to keep all processors busy.

Hyper-Threading increases the number of logical processors, and demands that the software spawn additional threads to run at maximum efficiency. However, some applications do not yet do this. This is why, in general, Hyper-Threading performs quite well with two-socket and four-socket, single core SMP systems, because many applications already generate sufficient threads to keep four physical/logical CPUs busy.

However, at four-socket dual core, eight-socket, and 16-socket, the applications have to spawn even more threads to efficiently utilize Hyper-Threading or the additional cores. All of these things must be engineered into the server application and operating system. In general, the only applications that scale past four-socket are middle-tier applications, database applications, and virtualization applications.

## **Multi-processing and server types**

Multi-processing has a direct relationship with the type of application server that is used. If the server is used as a file server, adding a processor does not improve performance significantly. For a server used as an application server, however, adding a processor can result in a very significant performance gain.

Multi-processing will not provide a linear improvement in processing power as additional processors are added. You might achieve a 70% to 80% performance increase from the second processor, but each additional processor will provide less and less performance increase as other system bottlenecks come into play, as illustrated in Figure 9-2 on page 170.

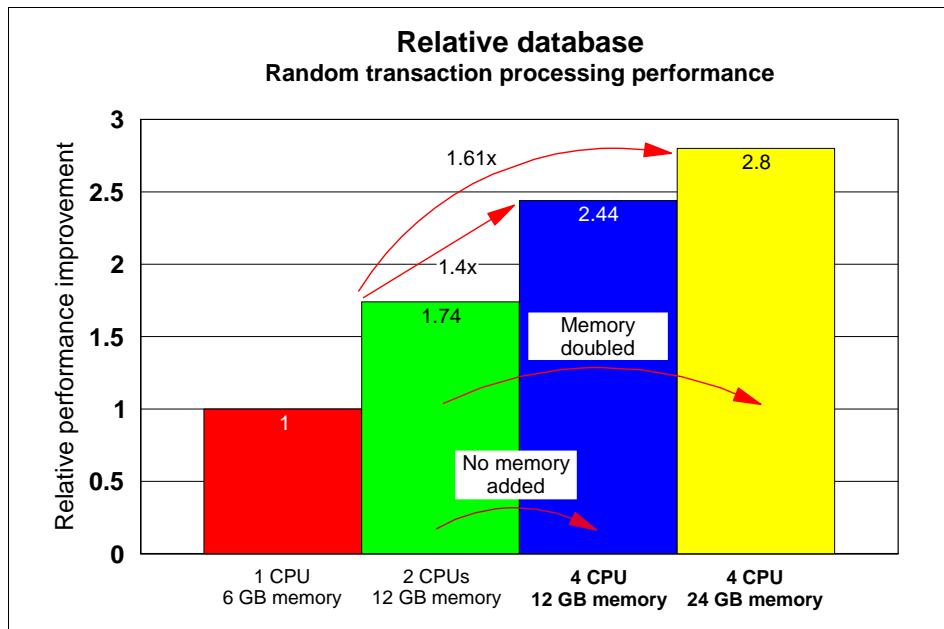


Figure 9-2 Relative performance scaling from adding 3.0 GHz processors

Using this chart, another point should be made: adding memory is critical to getting the most out of any processor upgrade. In this case, adding the third and fourth processors improves relative performance by 2.44, but if you also add memory, then the relative improvement is 2.8.

The scalability of multi-processor systems also varies greatly depending on whether an SMP or a NUMA design is used. Classic SMP designs scale very poorly beyond four-socket. However, NUMA systems show good scalability up to 32-socket (and even further on some very special NUMA systems).

### 9.2.6 Unified Extensible Firmware Interface

The Unified Extensible Firmware Interface (UEFI) is a replacement for the Basic Input Output System (BIOS) that has served the IBM x86 server products for nearly twenty years, and prior to that, the IBM Personal Computer. UEFI defines a standard interface between operating system, platform firmware, and external devices. It offers capabilities far exceeding that of legacy BIOS, as well as improved effectiveness of product development.

UEFI has evolved from EFI, which was primarily intended for the next generation of Intel Itanium-based™ computers. In its current form, UEFI 2.1, it is becoming

widely accepted throughout the industry. Many systems are making the transition to UEFI-compliant firmware, including the IBM x3450 server.

The IBM System x firmware contains many features that go well beyond the basic requirements of a UEFI-compliant system. Key features include Active Energy Manager, Memory Predictive Failure Alerts, enhanced Light Path Diagnostics, simplified POST Codes and out-of-band/lights-out configuration and deployment capabilities.

Compatibility is a cornerstone of IBM server design. As such, IBM-UEFI is designed to support the best of both worlds: UEFI capabilities and features, with legacy BIOS compatibility. This means that UEFI-based System x Servers are capable of booting UEFI operating systems as well as traditional, BIOS-booted operating systems, and they can make use of and boot from legacy adapters as well as from UEFI-compliant adapters.

## 9.3 Memory controller-based chipset

Before Nehalem, Intel-based chipsets use an architecture where all processors reside connected to a front-side bus. This front-side bus allows for direct processor-to-processor communication, as well as processor-to-memory communication. The memory controller handles communications that occur between the CPU, RAM, PCI Express controller, and the south bridge (I/O controller hub). The memory controller is the main aspect of a chipset that determines the number, speed, and type of CPU, as well as the number, speed, and type of main memory (RAM). This section describes some of the differences between different chipsets that are incorporated in servers in current server architectures.

### 9.3.1 Intel 5000 chipset family

The Intel 5000 chipset family exists as the follow-on to the E7520 chipset family. It includes the following features:

- ▶ New generation of dual-core Intel Xeon 5000 series processors, including Woodcrest and Dempsey (for a detailed discussion, see 6.2, “Intel Xeon processors” on page 94).
- ▶ Intel Virtualization Technology (for a detailed discussion, see 7.2, “Virtualization hardware assists” on page 138).
- ▶ I/O Acceleration Technology (for a detailed discussion, see 12.3.2, “I/O Accelerator Technology” on page 324).
- ▶ Support for Fully Buffered DIMMs (for a detailed discussion, see 10.2.6, “Fully-buffered DIMMs” on page 191).

- ▶ Hyper-Threading and Enhanced Intel SpeedStep® Technology.
- ▶ Intel 64 Technology (EM64T) (for a detailed discussion, see “64-bit computing” on page 116).

The Intel 5000P Blackford is implemented in the IBM System x3400, x3500, and x3650. The Intel 5000X chipset Greencreek is implemented in the IBM System x3550 model. The Intel 5000X Greencreek chipset differs from the Blackford chipset by its inclusion of a first-generation snoop filter.

The 5000 class chipset supports two processor sockets on dual, independent front-side buses that each operate at 266 MHz. The front-side buses are 64-bit, quad pumped, which allows for a total peak bandwidth of 17 GBps total or 8.5 GBps per front-side bus. Front-side bus speeds range from 1066 MHz to 1333 MHz.

Figure 9-3 on page 173 shows the block diagram for the System x3650, which includes the Intel 5000P Blackford chipset. Two processors and up to 12 Fully Buffered DIMMs are supported with this chipset. The PCI-Express adapters are connected directly to the Blackford memory controller.

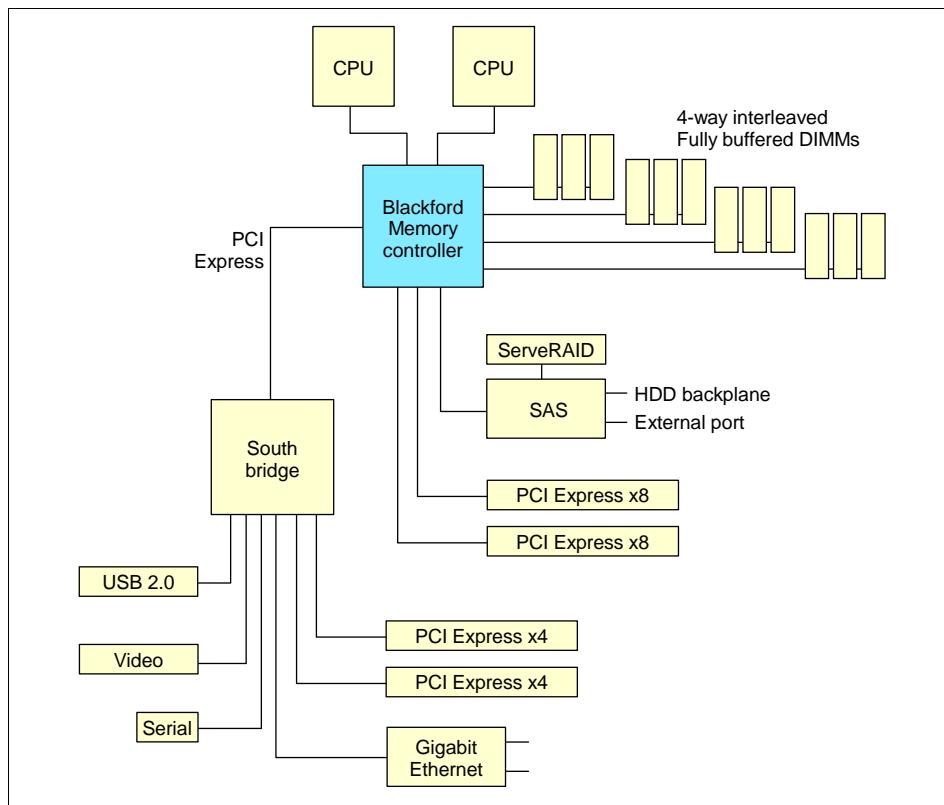


Figure 9-3 System x3650 block diagram

For more information, see:

<http://www.intel.com/products/chipsets/5000P/>  
<http://www.intel.com/products/chipsets/5000X/>  
<http://www.intel.com/products/chipsets/5000V/>

### 9.3.2 Intel 5400 chipset family

The Intel 5400 Stoakley chipset exists as a follow-on enhancement to the Intel 5000 family. It provides support for faster processor, bus, and I/O speed to boost system performance. Some key features include:

- ▶ Supports up to 1600 MT/s dual independent front-side bus
- ▶ Up to 800MHz 16 FB-DIMM memory slot interface support
- ▶ PCI Express 2.0(Gen2), offering higher bandwidth and lower latency connection between the MCH chipset and PCI Express adapter
- ▶ Enhanced Intel VT technology

The Intel 5400 chipset is implemented in the IBM System x3450, as shown in Figure 9-4.

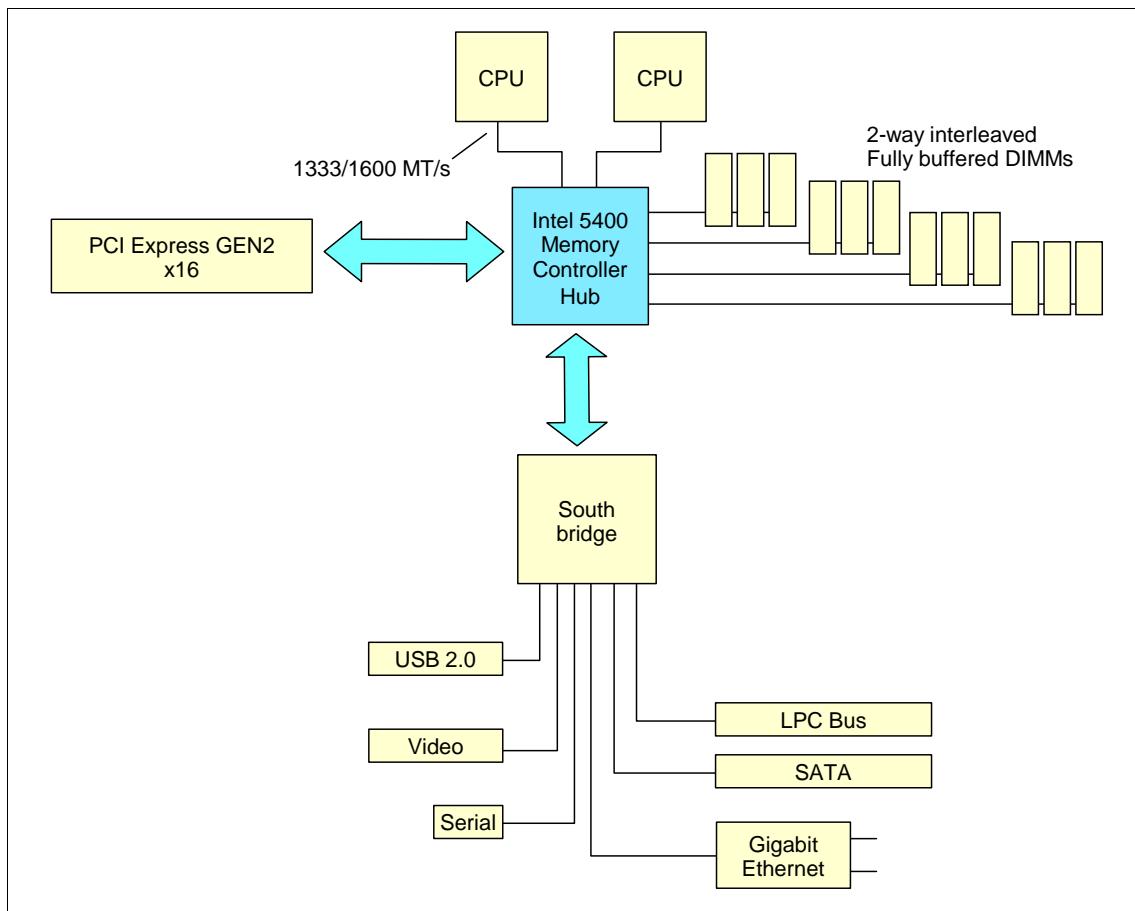


Figure 9-4 System x3450 block diagram

For more information, see:

<http://www.intel.com/products/server/chipsets/5400/5400-overview.htm>

### 9.3.3 IBM XA-64e fourth-generation chipset

IBM eX4 is the fourth-generation of the IBM XA-64e or eX4 chipset. The chipset is designed for the Xeon MP processor family from Intel. The IBM system x3850 M2 and x3950 M2 are based on this chipset.

Compared to its predecessor X3, the eX4 provides these significant improvements:

- ▶ Quad FSB delivers an increase in system bandwidth
- ▶ Eight channels to memory provide optimal memory read and write bandwidth and more DIMM slots per node
- ▶ Enhancements to scalability port protocol, placement, and design have led to a higher scalable bandwidth over X3

Figure 9-5 on page 176 shows a block diagram of the x3850 M2 and x3950 M2. As the diagram illustrates, eX4 architecture consists of the following components:

- ▶ One to four Xeon dual-core or quad-core processors
- ▶ Hurricane 4 Memory and I/O Controller (MIOC)
- ▶ Eight high-speed memory buffers
- ▶ Two PCI Express bridges
- ▶ One Enterprise South bridge Interface

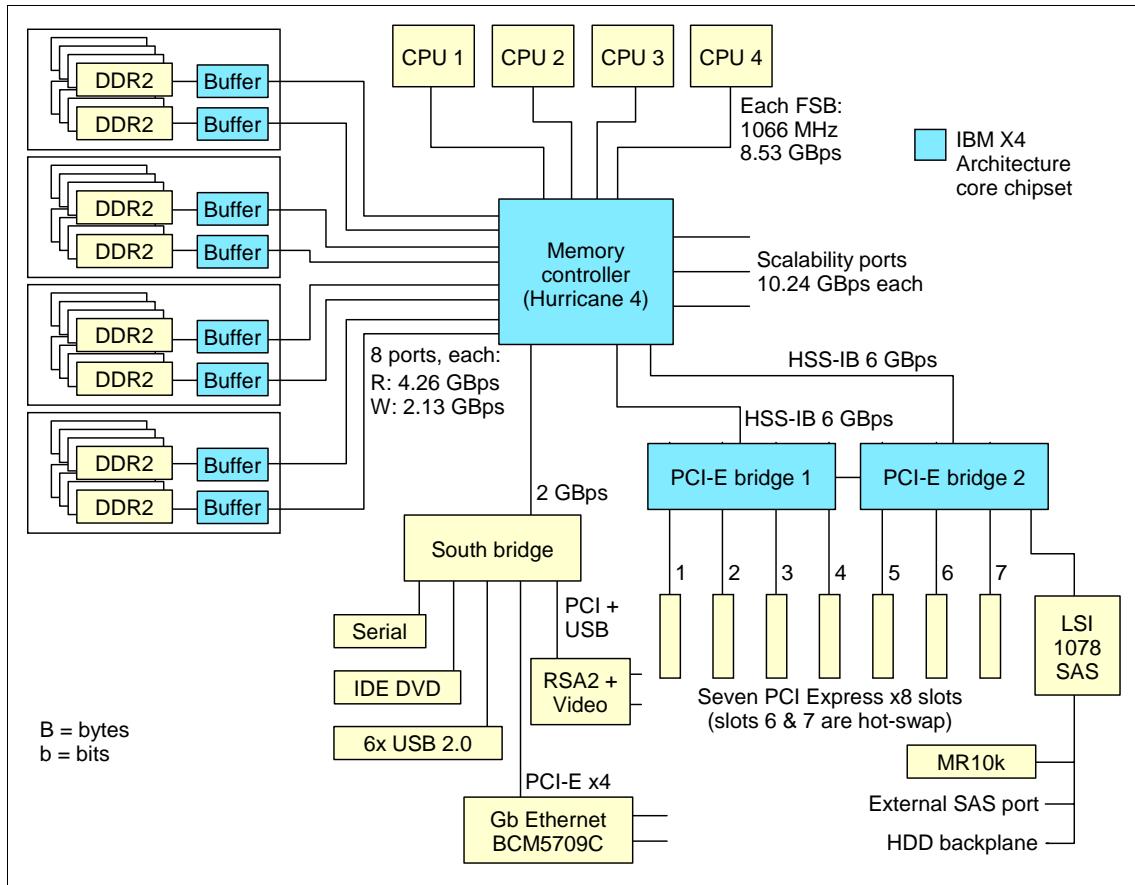


Figure 9-5 The eX4 Architecture system diagram

Each memory port out of the memory controller has a peak read throughput of 4.26 GBps and a peak write throughput of 2.13 GBps. DIMMs are installed in matched pairs, two-way interleaving, to ensure the memory port is fully utilized. Peak throughput for each PC2-5300 DDR2 DIMM is 4.26 GBps.

There are eight memory ports, and spreading installed DIMMs across all ports can improve performance. The eight independent memory ports provide simultaneous access to memory. With four memory cards installed and eight DIMMs in each card, peak read memory bandwidth is 34.1 GBps and peak write bandwidth is 17.1 GBps. The memory controller routes all traffic from the eight memory ports, four microprocessor ports, and the three PCI-E bridge ports.

The memory controller also has embedded DRAM which, in the x3850 M2 and x3950 M2, holds a snoop filter lookup table. This filter ensures that snoop

requests for cache lines go to the appropriate microprocessor bus and not to all four of them, thereby improving performance.

The three scalability ports are each connected to the memory controller via individual scalability links with a maximum theoretical bidirectional data rate of 10.24 GBps per port.

IBM eX4 has two PCI-E bridges. Each bridge is connected to a HSS-IB port of the memory controller with a maximum theoretical bidirectional data rate of 6 GBps. As shown in Figure 9-5 on page 176, PCI-E bridge 1 supplies four of the seven PCI Express x8 slots on four independent PCI Express buses. PCI-E bridge 2 supplies the other three PCI Express x8 slots plus the onboard SAS devices, including the optional ServeRAID-MR10k and a 4x external onboard SAS port.

A separate South bridge is connected to the Enterprise South bridge Interface (ESI) port of the memory controller via a PCI-E x4 link with a maximum theoretical bidirectional data rate of 2 GBps. The South bridge supplies all the other onboard PCI devices, such as the USB ports, onboard Ethernet, and the standard RSA II.

## 9.4 PCI bridge-based chipsets

The following sections detail PCI bridge-based chipsets.

### 9.4.1 AMD HyperTransport

AMD Opteron processors do not use the typical shared front-side bus that is connected to a memory controller used in most of Intel-based servers. Each Opteron processor has its own integrated memory controller and pins on the processor chip to directly connect to a memory bus. So, in Opteron, processor and memory controller logic are integrated into the same piece of silicon, thereby eliminating the need for a separate memory controller part. Hardware vendors simply add a memory bus and memory DIMMs and they have the core CPU and memory interface.

To keep data coherent between multiple Opteron processors, AMD introduced a new system bus architecture called HyperTransport. Three HyperTransport links

are available on each Opteron processor; two are used for CPU-CPU connectivity and one is used for I/O.

The two HyperTransport links used for CPUs enable the direct connection of two processors and the indirect connection of four or more processors. IBM System x and BladeCenter servers that have this type of architecture include the following:

- ▶ System x3455
- ▶ System x3755
- ▶ BladeCenter JS20/JS21
- ▶ BladeCenter LS20/LS21/LS22

With a four-processor configuration, the processors are placed at the corners of a square, with each line that makes up the square representing a HyperTransport connection between the processors, as shown in Figure 9-6.

With this design, whenever two processors that are on the same side of the square share data, the information passes directly over the HyperTransport interconnect to the other processor. When this remote access occurs, it is called *single-hop remote memory access*, and it is slower than a local memory access.

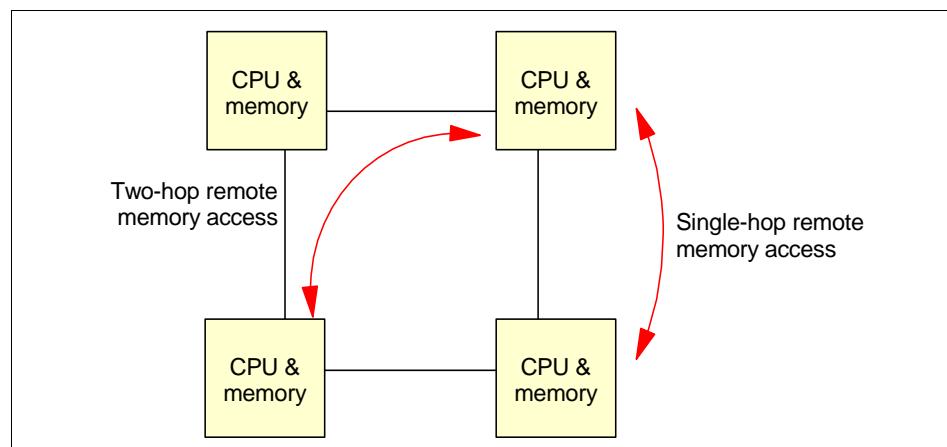


Figure 9-6 Remote memory access

However, when two processors on diagonal corners of the square share data or instructions, the information must travel through an additional processor connection before arriving at the diagonal processor. This extra hop adds some additional overhead and is referred to as *two-hop remote access*.

In systems such as the System x3755, when the server is configured with just three processors, a passthru card can be installed in place of the fourth processor to reduce the two-hop access to just a single hop, as shown in Figure 9-7. For more information, see 6.3.5, “IBM CPU passthru card” on page 113.

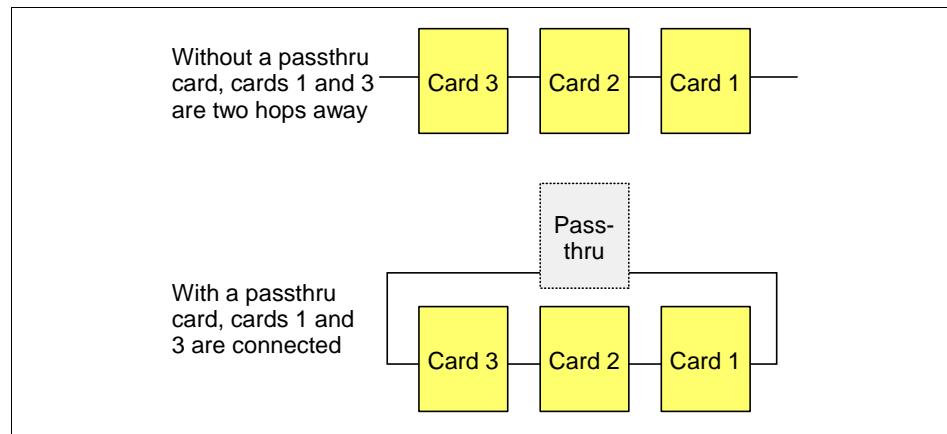


Figure 9-7 The benefit of the passthru card for three-way configurations

The third port of the HyperTransport link is not used to interconnect the diagonal processors because it must be used for connection to a PCI I/O bridge, which connects such devices as PCI slots, network, disk storage, mouse, keyboard, video, and so forth. Officially, Opteron processors support up to eight CPUs within a single system. However, the latency to include all eight sockets and the additional hops in a single architecture would add little or no performance gains over a four-socket system.

The remote memory access latency of a processor accessing another processor's memory space makes the Opteron configuration a NUMA design (refer to 9.2.3, “NUMA” on page 161). NUMA means that every processor has both memory that is *closer* and thus more rapidly accessible, and also memory that is *remote* and slower, which must be accessed through another Opteron processor.

AMD refers to its Opteron architecture as Sufficiently Uniform memory Organization (SUMO) rather than NUMA. From an architectural standpoint, it still is a NUMA architecture but the HyperTransport link is fast enough to run software written for SMP systems without very significant performance penalties. Current operating systems such as the latest versions of Linux and Windows Server 2008 support NUMA and make attempts to minimize remote memory transactions. However, in practice, the percentage of remote memory accesses

is largely determined by application behavior and by how data is manipulated by users of the application.

Figure 9-8 shows the Opteron architecture with the integrated memory controller and the HyperTransport connectors.

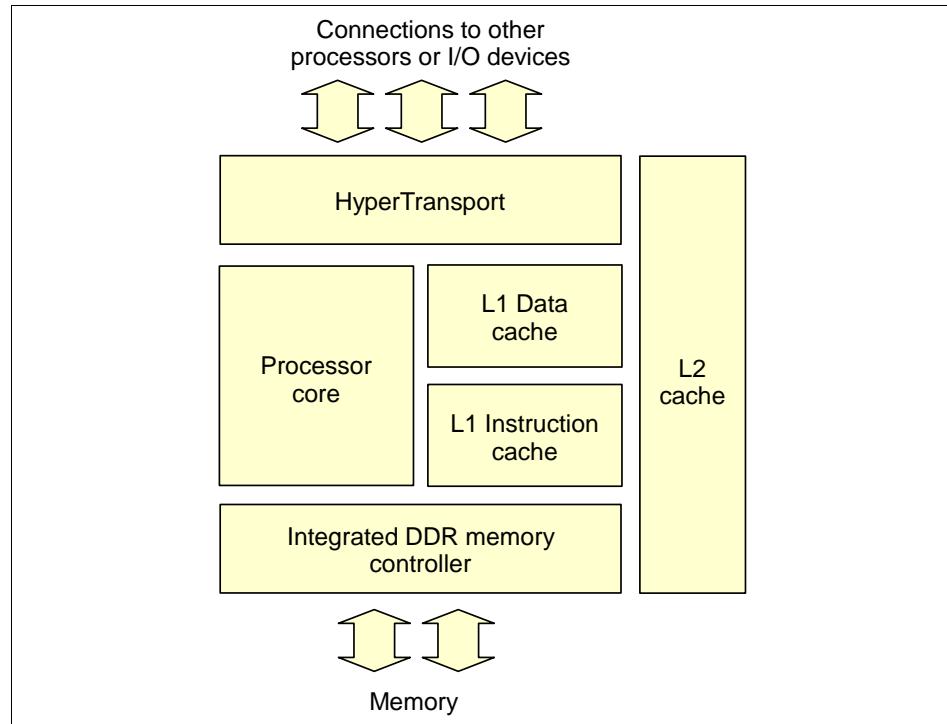


Figure 9-8 CPU architecture of the Opteron CPU with an integrated memory controller

## HyperTransport

The HyperTransport architecture was initially developed by AMD but is now managed by an open consortium of several large IT companies such as AMD, Apple, Cisco, Broadcom, ATI™, IBM, and many others. HyperTransport is an open standard for a high-speed, point-to-point link system that can be used for connecting a variety of chips.

HyperTransport technology is used in devices such as network devices and graphics cards or, as in the case of the AMD Opteron, as a high-speed interconnect for processors. The HyperTransport technology used for interconnecting Opteron processors is currently implemented at a speed of 1000 MHz with a bidirectional bandwidth of 4.0 GBps each way, which leads to a peak full-duplex capacity of 8.0 GB per second per link. Current Opteron

processors incorporate three HyperTransport links, which enables a peak bandwidth of 24 GBps per processor.

You can find more information about the HyperTransport and the HyperTransport Consortium at:

<http://www.hypertransport.org/>

### 9.4.2 Intel QuickPath Architecture

As introduced in 6.2.5, “Intel Nehalem microarchitecture” on page 105, starting with Nehalem, the memory controller is integrated into each processor’s silicon die, replacing the front-side bus and separate memory controller. This design forms the basis of the Intel approach to implementing a scalable shared-memory architecture. The new Intel system architecture is called Intel QuickPath Interconnect (QPI) Architecture.

As shown in Figure 9-9, in QPI each processor has its own memory controller and dedicated memory to access directly. QPI provides high-speed connections between processors and remote memory, and between processors and the I/O hubs.

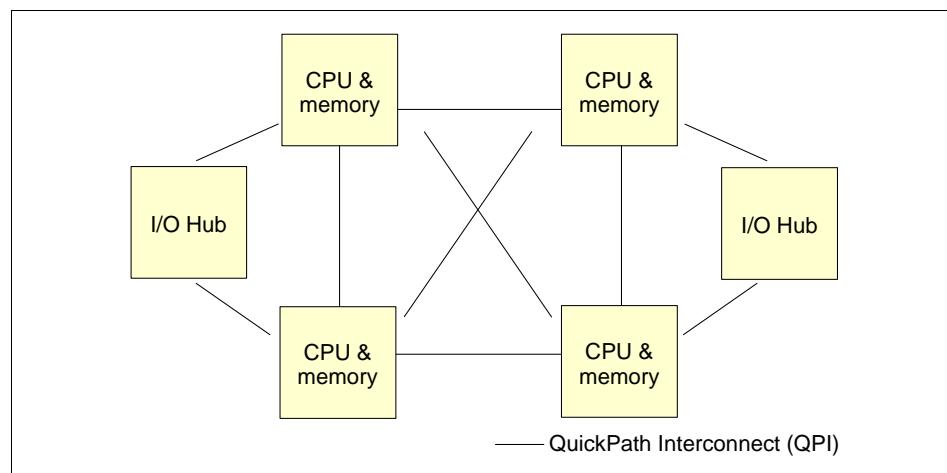


Figure 9-9 QuickPath Architecture

QPI is similar to HyperTransport in AMD Opteron processors. Both QPI and HyperTransport act as system interconnections, providing a high-speed, reliable data communication channel between system components. For a comparison of these two technologies, see Table 9-3 on page 182.

*Table 9-3 Comparing QPI and HyperTransport*

	<b>Link transfer rate</b>	<b>Bandwidth</b>	<b>Link width</b>	<b>Reliability features</b>
QPI	6.4 GT/s	25.6 GBps	20-bit	CRC, hot plug, self-healing, clock failover
HyperTransport V1.0	2.0 GT/s	8 GBps	17-bit	CRC

You can find more information about the QPI and the QuickPath Architecture at:

<http://www.intel.com/technology/quickpath>



# Memory subsystem

Insufficient memory is often the reason behind poor server performance. As the amount of memory that running programs and their data uses approaches the total available physical memory that is installed on the machine, a server's virtual memory handler increases the amount of data that is paged in and out of memory, to and from the paging area on disk, with a disastrous effect on performance. Fortunately, the memory subsystem is usually one of the easiest areas of the entire system to upgrade.

This chapter discusses the following topics:

- ▶ 10.1, “Introduction to the memory subsystem” on page 184
- ▶ 10.2, “Memory technology” on page 185
- ▶ 10.3, “Specifying memory performance” on page 199
- ▶ 10.4, “SMP and NUMA architectures” on page 203
- ▶ 10.5, “The 32-bit 4 GB memory limit” on page 207
- ▶ 10.6, “64-bit memory addressing” on page 210
- ▶ 10.7, “Advanced ECC memory (Chipkill)” on page 212
- ▶ 10.8, “Memory mirroring” on page 213
- ▶ 10.9, “Intel Xeon 5500 Series Processors” on page 215
- ▶ 10.10, “eX4 architecture servers” on page 231
- ▶ 10.11, “IBM Xcelerated Memory Technology” on page 234
- ▶ 10.12, “Memory rules of thumb” on page 234

## 10.1 Introduction to the memory subsystem

Over the years, memory capacity demands have increased because server operating systems and server application code have grown. Additionally, user content has evolved from simple character-based data to more expansive rich media such as audio and video. Applications and data that have previously existed on large high-end computers have migrated to x86 class servers, placing additional demands on memory capacity and capability on these rapidly evolving servers. The trend for increasing server memory demand is expected to continue well into the future.

Multiple levels of memory exist in any server architecture, with the sole purpose of providing necessary data to the processor cores. The fastest, most expensive, and thus smallest memories are the L1, L2, and L3 caches within a processor, while the slowest, cheapest, and largest memory is the physical disk storage.

In the middle of these, in terms of cost, speed, and capacity, is the server's main memory. Main memory bridges the performance gap between the fast processor caches and relatively slow disk subsystem. The processor caches bridge the performance gap between the processors and main memory. Together, the multiple levels of memory work to get the necessary pieces of data to the processor cores in the most efficient means possible, providing the performance levels expected from users.

Each of these levels of memory is used to hold both program data and the executable code required by each user. Network-attached users often access unique data objects on the server. Each of these data objects requires memory for storage. Furthermore, each task that a network user requests the server to perform requires many thousands of instructions to be executed. Network users do not usually require the same tasks to be performed by the server at the same time. Thus, as the number of users of a system grows, the memory requirement for program data and executable code required by each user also grows.

A server that does not have sufficient memory to meet the requirements of all active users will attempt to expand its memory to the next larger memory space, the disk drive. This activity, known as *memory paging*, tends to cripple server performance, because disk accesses are significantly slower than memory accesses.

The remainder of this chapter focuses on the primary memory types used in current servers.

## 10.2 Memory technology

This section introduces key terminology and technology that are related to memory; the topics discussed are:

- ▶ 10.2.1, “DIMMs and DRAMs” on page 185
- ▶ 10.2.2, “Ranks” on page 187
- ▶ 10.2.3, “SDRAM” on page 188
- ▶ 10.2.4, “Registered and unbuffered DIMMs” on page 188
- ▶ 10.2.5, “Double Data Rate memory” on page 189
- ▶ 10.2.6, “Fully-buffered DIMMs” on page 191
- ▶ 10.2.7, “MetaSDRAM” on page 195
- ▶ 10.2.8, “DIMM nomenclature” on page 196
- ▶ 10.2.9, “DIMMs layout” on page 198
- ▶ 10.2.10, “Memory interleaving” on page 199

### 10.2.1 DIMMs and DRAMs

Memory in servers is implemented in the form of Dual Inline Memory Modules (DIMMs). DIMMs contain a number of chips, known as Synchronous Dynamic RAM (SDRAM, or simply DRAM) chips. The number of chips implemented on the DIMM depends on the total capacity of the DIMM and whether the DIMM has error checking and correcting (ECC) functions. Without ECC, a DIMM typically has 8 or 16 SDRAM chips. With ECC, there are typically 9 or 18 chips. The largest DIMMs often have up to 36 chips.

Figure 10-1 is a photo of an ECC DIMM, with 9 SDRAM chips on each side.

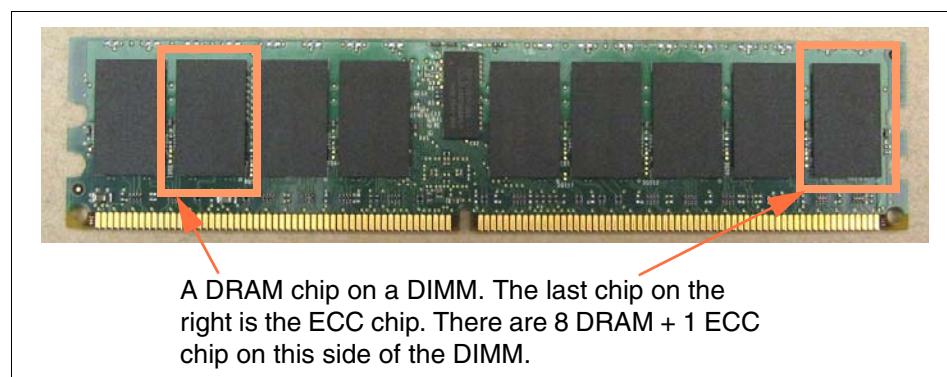


Figure 10-1 DRAM chips on a DIMM

The capacity of each DRAM is a number of “words” where each word can be 4 bits (“x4”), 8 bits (“x8”) and, though rarely used in the servers, 16 bits in length

("x16"). The number of words in the DRAM is sometimes written on the label of the DIMM, such as 128M, meaning that each DRAM has 128 million (actually  $128 \times 1024^3$ ) words. Figure 10-2 shows an example.

**Note:** The word length (x4 or x8) is normally not printed on the label. However, the DIMM manufacturer's Web site might list such specifications. It can also be calculated:

$$(\text{DIMM capacity in MB}) / (\text{Number of non-ECC DRAMs}) * 8 / (\text{M value})$$

So for the 1 GB DIMM in Figure 10-2,  $1024 \text{ MB} / 8 * 8 / 128 = 8\text{-bit word length.}$

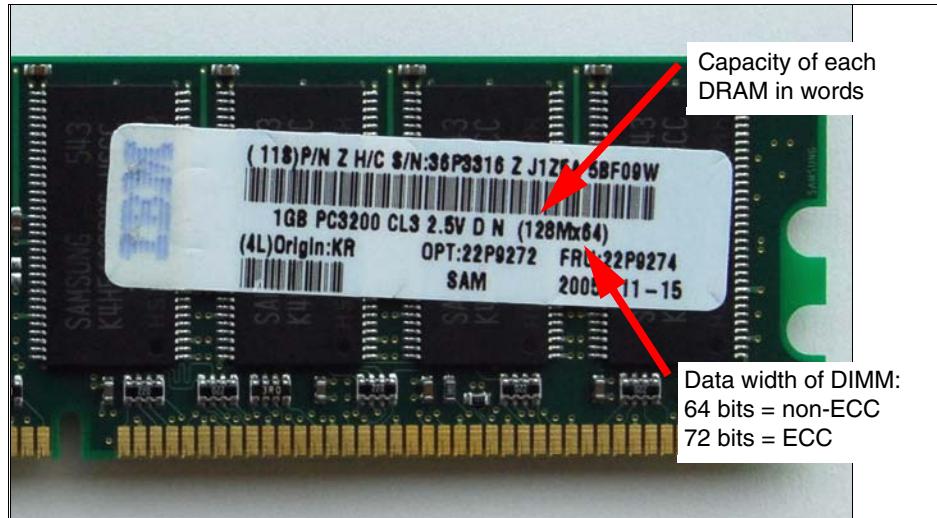


Figure 10-2 DRAM capacity as printed on a PC3200 (400 MHz) DDR DIMM

The sum of the capacities of the DRAM chips (minus any used for ECC functions if any), equals the capacity of the DIMM. Using the previous example, the DRAMs in Figure 10-2 are 8 bits wide, so:

$$8 \times 128\text{M} = 1024 \text{ Mbits} = 128 \text{ MB per DRAM}$$

$$128 \text{ MB} \times 8 \text{ DRAM chips} = 1024 \text{ MB or } 1 \text{ GB of memory}$$

## 10.2.2 Ranks

A *rank* is a set of DRAM chips on a DIMM that provides eight bytes (64 bits) of data. DIMMs are typically configured as either single-rank (1R) or double-rank (2R) devices but quad-rank devices (4R) are becoming more prevalent.

Using x4 DRAM devices, and not including DRAMs for ECC, a rank of memory is composed of  $64 / 4 = 16$  DRAMs. Similarly, using x8 DRAM devices, a rank is composed of only  $64 / 8 = 8$  DRAMs.

It is common, but less accurate, to refer to memory ranking in terms of “sides”. For example, single-rank DIMMs can often be referred to as single-sided DIMMs, and double-ranked DIMMs can often be referred to as double-sided DIMMs. However, single ranked DIMMs, especially those using x4 DRAMs often have DRAMs mounted on both sides of the DIMMs, and quad-rank DIMMs will also have DRAMs mounted on two sides. For these reasons, it is best to standardize on the true DIMM ranking when describing the DIMMs.

**Note:** Some servers do not allow the mixing of DIMMs with different numbers of ranks. Other systems do support mixing, but require DIMMs be placed in a certain order. Yet other systems allow combinations of DIMMs with different ranks. Regardless, most systems will perform best when DIMMs of the same size and ranking are populated evenly across all the memory channels in the server.

DIMMs may have many possible DRAM layouts, depending on word size, number of ranks, and manufacturer design. Common layouts for single and dual-rank DIMMs are identified here:

- ▶ x8SR = x8 single-ranked modules

These have five DRAMs on the front and four DRAMs on the back with empty spots in between the DRAMs, or they can have all 9 DRAMs on one side of the DIMM only.

- ▶ x8DR = x8 double-ranked modules

These have nine DRAMs on each side for a total of 18 (no empty slots).

- ▶ x4SR = x4 single-ranked modules

These have nine DRAMs on each side for a total of 18, and they look similar to x8 double-ranked modules.

- ▶ x4DR = x4 double-ranked modules

These have 18 DRAMs on each side, for a total of 36.

The rank of a DIMM also impacts how many failures a DIMM can tolerate using redundant bit steering. See “Memory ProteXion: redundant bit steering” on page 233 for details.

### 10.2.3 SDRAM

Synchronous Dynamic Random Access Memory (SDRAM) is used commonly in servers today, and this memory type continues to evolve to keep pace with modern processors. SDRAM enables fast, continuous bursting of sequential memory addresses. After the first address is supplied, the SDRAM itself increments an address pointer and readies the next memory location that is accessed. The SDRAM continues bursting until the predetermined length of data has been accessed. The SDRAM supplies and uses a synchronous clock to clock out data from the SDRAM chips. The address generator logic of the SDRAM module also uses the system-supplied clock to increment the address counter to point to the next address.

### 10.2.4 Registered and unbuffered DIMMs

There are two types of SDRAMs currently on the market: *registered* and *unbuffered*. Only registered SDRAM are now used in System x servers, however. Registered and unbuffered cannot be mixed together in a server.

With unbuffered DIMMs, the memory controller communicates directly with the DRAMs, giving them a slight performance advantage over registered DIMMs. The disadvantage of unbuffered DIMMs is that they have a limited drive capability, which means that the number of DIMMs that can be connected together on the same bus remains small, due to electrical loading.

In contrast, registered DIMMs use registers to isolate the memory controller from the DRAMs, which leads to a lighter electrical load. Therefore, more DIMMs can be interconnected and larger memory capacity is possible. The register does, however, typically impose a clock or more of delay, meaning that registered DIMMs often have slightly longer access times than their unbuffered counterparts.

These differences mean that fewer unbuffered DIMMs are typically supported in a system than for a design using registered DIMMs. While this might not be a problem for desktop systems and very low-end servers, mainstream servers need to support larger amounts of memory and therefore use registered DIMMs.

## 10.2.5 Double Data Rate memory

Data transfers made to and from an SDRAM DIMM use a synchronous clock signal to establish timing. For example, SDRAM memory transfers data whenever the clock signal makes a transition from a logic low level to a logic high level. Faster clock speeds mean faster data transfer from the DIMM into the memory controller (and finally to the processor) or PCI adapters. However, electromagnetic effects induce noise, which limits how fast signals can be cycled across the memory bus, and have prevented memory speeds from increasing as fast as processor speeds.

All Double Data Rate (DDR) memories, including DDR, DDR2, and DDR3, increase their effective data rate by transferring data on both the rising edge and the falling edge of the clock signal. DDR DIMMs use a *2-bit* prefetch scheme such that two sets of data are effectively referenced simultaneously. Logic on the DIMM multiplexes the two 64-bit results (plus ECC bits) to appear on subsequent data transfers. Thus, two data transfers can be performed during one memory bus clock cycle, enabling double the data transfer rate over non-DDR technologies.

### DDR2

DDR2 is the technology follow-on to DDR, with the primary benefits being the potential for faster throughput and lower power. In DDR2, the memory bus is clocked at two times the frequency of the memory core. Stated alternatively, for a given memory bus speed, DDR2 allows the memory core to operate at half the frequency, thereby enabling a potential power savings.

Although DDR memory topped out at a memory bus clock speed of 200 MHz, DDR2 memory increases the memory bus speed to as much as 400 MHz. Note that even higher DDR2 speeds are available, but these are primarily used in desktop PCs. Figure 10-3 shows standard and small form factor DDR2 DIMMs.



Figure 10-3 A standard DDR2 DIMM (top) and small form-factor DDR2 DIMM (bottom)

DDR2 also enables additional power savings through use of a lower operating voltage. DDR uses a range of 2.5 V to 2.8 V. DDR2 only requires 1.8 V.

Because only the highest speed DDR2 parts have similar memory core speeds as compared to DDR due to being clocked at half the bus rate, DDR2 employs a number of mechanisms to reduce the potential for performance loss. DDR2 increases the number of bits prefetched into I/O buffers from the memory core to 4 per clock, thus enabling the sequential memory throughput for DDR and DDR2 memory to be equal when the memory bus speeds are equal.

However, because DDR2 still has a slower memory core clock when the memory bus speeds are equal, the memory latencies of DDR2 are typically higher. Fortunately, the latest generation and most common DDR2 memory speeds typically also operate at significantly higher memory frequency.

The combination of prefetch buffer increases, frequency increases, and process updates allow DDR2 throughput to be as much as two times the highest standards of DDR, as memory latencies have improved to be about the same or somewhat better than the DDR technology it replaced.

DDR2 Memory is commonly found in recent AMD Opteron systems, as well as in the HS12 server blade and the x3850M2 and x3950M2 platforms.

Table 10-2 on page 197 lists the common DDR2 memory implementations.

### DDR3

DDR3 is the next evolution of DDR memory technology. Like DDR2, it promises to deliver ever-increasing memory bandwidth and power savings. DDR3 DIMMs are used on all 2-socket-capable servers using the Intel 5500-series processors and newer.

DDR3 achieves its power efficiency and throughput advantages over DDR2 using many of the same fundamental mechanisms employed by DDR2.

DDR3 improvements include:

- ▶ Lower supply voltages, down to 1.5 V in DDR3 from 1.8 V in DDR2
- ▶ Memory bus clock speed increases to four times the core clock
- ▶ Prefetch depth increases from 4-bit in DDR2 to 8-bit in DDR3
- ▶ Continued silicon process improvements

**Note:** Although DDR3 may appear to be simply a faster version of DDR2, this is not the case. DDR3 DIMMs are not pin- or backwards-compatible with DDR2 DIMMS, and cannot be inserted into DDR2 DIMM sockets.

DDR3 also implements a number of unique features that set this technology distinctly apart from previous DDR DIMMs. Most of these are electrical signalling-related and beyond the scope of this book. However, one feature in particular, *on-DIMM thermal sensors*, is of particular interest in modern systems. This finer-grained and more accurate thermal monitoring mechanism enables better control of DIMM temperature, partially enabling DDR3-based servers to support greater numbers of DIMMs while staying within a server's thermal limitations.

While bus clocking and prefetch buffer optimizations have enabled DDR3 DIMMs to increase their effective bandwidth, DDR3 still suffers from the same primary limitations of shared memory bus architectures. Due to electrical loading and bus noise on these shared buses, only a limited number of DDR3 DIMMs can be placed on each memory controller while maintaining the top rated speeds. Due to the very high clock rates of DDR3, it is possible to see a drop in speed for each DIMM added to a memory channel. Because multi-ranked DIMMs appear on the memory bus as multiple electrical loads, this can also factor into the effective speed of the memory subsystem.

For these reasons, maximum memory performance on DDR3-based systems will be achieved with just one or two DIMMs per channel. Systems based on the Intel 5500 series processors use three memory channels per processor, or six total memory channels in a dual-socket system configuration, with each channel supporting up to three DIMMs. Because DDR2-based systems typically only have two memory channels, this new configuration represents a significant boost in aggregate memory bandwidth, even if the channels clock down 1 or 2 speed levels to accommodate large DIMM configurations.

**Tip:** Maximum performance on a DDR3-based system is typically achieved when the fewest possible DIMMs are used per memory channel, while still providing sufficient memory for the server application workload.

Table 10-3 on page 197 lists the common DDR3 memory implementations.

### 10.2.6 Fully-buffered DIMMs

As CPU speeds increase, memory access must keep up so as to reduce the potential for bottlenecks in the memory subsystem. As the replacement for the parallel memory bus design of DDR2, FB-DIMMs enabled additional bandwidth capability by utilizing a larger number of serial channels to the memory DIMMs. This increase in memory channels, from two in DDR2 to four with FB-DIMMs, allowed a significant increase in memory bandwidth without a significant increase in circuit board complexity or the numbers of wires on the board.

Although fully-buffered DIMM (FB-DIMM) technology replaced DDR2 memory in most systems, it too has been replaced by DDR3 memory in the latest 2-socket systems utilizing the Intel 5500-series processors. Non-IBM systems using the Intel 7400-series processors or earlier also still utilize FB-DIMMs, though the x3850 M2 and x3950 M2 systems utilize IBM eX4 chipset technology to allow memory bandwidth improvements while maintaining usage of standard DDR2 DIMMs.

FB-DIMMs use a serial connection to each DIMM on the channel. As shown in Figure 10-4, the first DIMM in the channel is connected to the memory controller. Subsequent DIMMs on the channel connect to the one before it. The interface at each DIMM is a buffer known as the Advanced Memory Buffer (AMB).

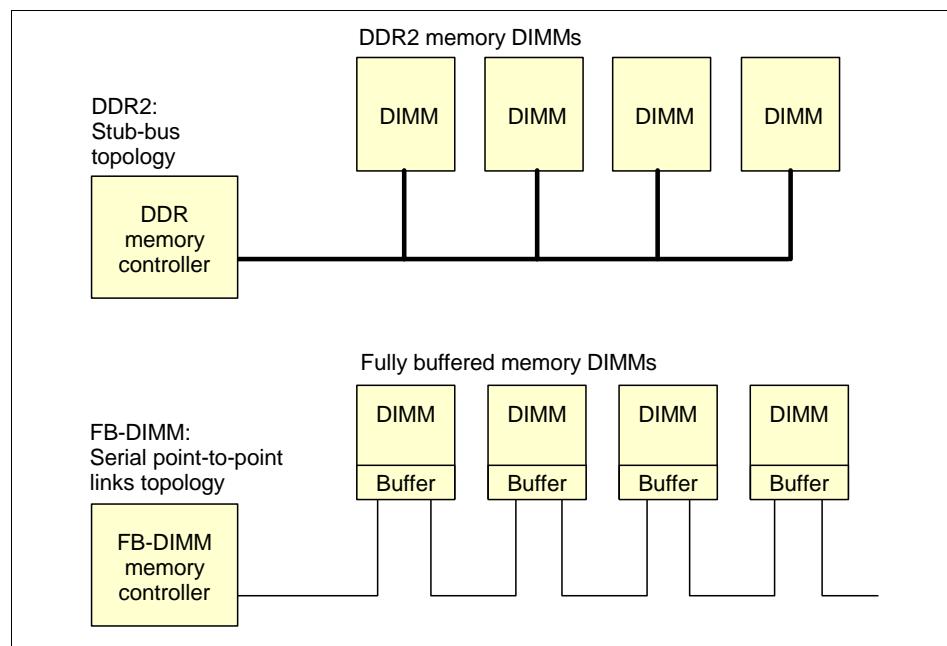


Figure 10-4 Comparing the DDR stub-bus topology with FB-DIMM serial topology

This serial interface results in fewer connections to the DIMMs (approximately 69 per channel) and less complex wiring. These links are relatively similar to other high speed serial technologies, including PCI Express, SATA, or SAS. The interface between the buffer and DRAM chips is the same as with DDR2 DIMMs. The DRAM chips are also the same as DDR2 DIMMs.

With this serial point-to-point connectivity, there is a built-in latency associated with any memory request that varies with the number of DIMMs used. The protocol of FB-DIMM mandates that even if a memory request is fulfilled by the first DIMM nearest to the memory controller, the address request must still travel

to the last DIMM in the serial chain before propagating back through each DIMM to the memory controller. As a consequence, the more DIMMs per channel that are used, the longer the latency.

Figure 10-5 illustrates the architecture of an FB-DIMM implementation, showing connections and the placement of the buffer and DRAM chips.

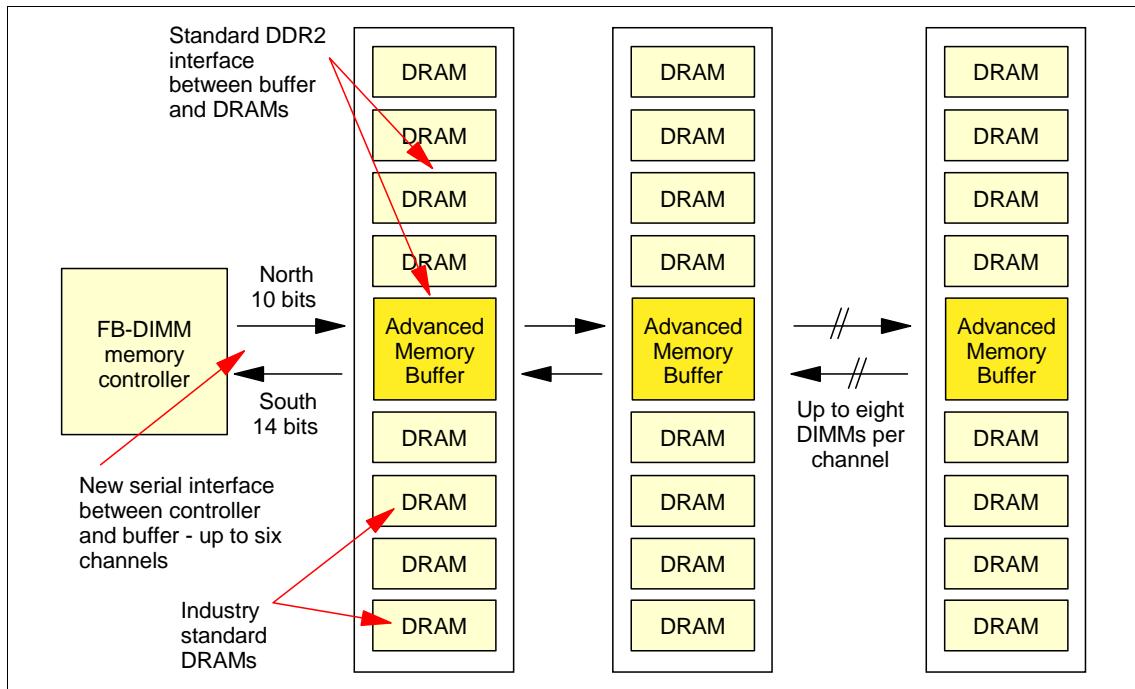


Figure 10-5 Fully buffered DIMMs architecture

**Note:** FB-DIMMs are *not* the next generation of DRAM. They represent a new way of accessing the same DDR2 DRAMs from a new memory controller, thereby enabling greater bandwidths without a significant increase in memory controller or board complexity.

Standard FB-DIMM memory controllers support up to 4 channels and up to 8 DIMMs per channel, allowing up to 32 DIMMs in large system designs.

### Advanced Memory Buffer

An FB-DIMM uses a buffer known as the Advanced Memory Buffer (AMB), as circled in Figure 10-6 on page 194. The AMB is a memory interface that connects an array of DRAM chips to the serial memory channel. The AMB is

responsible for handling FB-DIMM channel and memory requests to and from the local FB-DIMM, and for forwarding requests to other AMBs in other FB-DIMMs on the channel.



Figure 10-6 Advanced Memory Buffer on an FB-DIMM

The functions that the AMB performs include the following:

- ▶ Channel initialization to align the clocks and to verify channel connectivity. It is a synchronization of all DIMMs on a channel so that they are all communicating at the same time.
- ▶ Support the forwarding of southbound frames (writing to memory) and northbound frames (reading from memory), servicing requests directed to a specific FB-DIMMs AMB and merging the return data into the northbound frames.
- ▶ Detect errors on the channel and report them to the memory controller.
- ▶ Act as a DRAM memory buffer for all read, write, and configuration accesses addressed to a specific FB-DIMMs AMB.
- ▶ Provide a read and write buffer FIFO.
- ▶ Support an SMBus protocol interface for access to the AMB configuration registers.
- ▶ Provide a register interface for the thermal sensor and status indicator.
- ▶ Function as a repeater to extend the maximum length of FB-DIMM Links.

### Low Power FB-DIMMs

While FB-DIMMs commonly consume up to twice the power of standard DDR2 DIMMs due to the usage of the AMB, significant developments have been made to improve this in the latest generation of FB-DIMMs. While the performance aspects of these DIMMs are fundamentally unchanged, the power consumption of the DIMMs can be cut by up to 50%.

The following are some of the mechanisms used to reduce FB-DIMM power:

- ▶ AMB and DRAM process improvements
- ▶ Lower DRAM and AMB voltages
- ▶ Potential elimination of a level of voltage regulation, which saps efficiency

### **FB-DIMM performance**

By using serial memory bus technology requiring fewer board traces per channel, FB-DIMMs enabled memory controller designers to increase the number of memory channels, and thus aggregate bandwidth, without a significant increase in memory controller complexity. This daisy-chained, serial architecture adds latency to memory accesses, however, and when multiplied over many DIMMs can add up to a significant performance impact.

While all applications will have slightly different behavior, Figure 10-7 illustrates the performance loss for a moderately memory-intensive database application as FB-DIMMs are added to the memory channels of a server. Note that for this application, the performance loss was measured at ~2% overall per additional DIMM added across the memory channels.

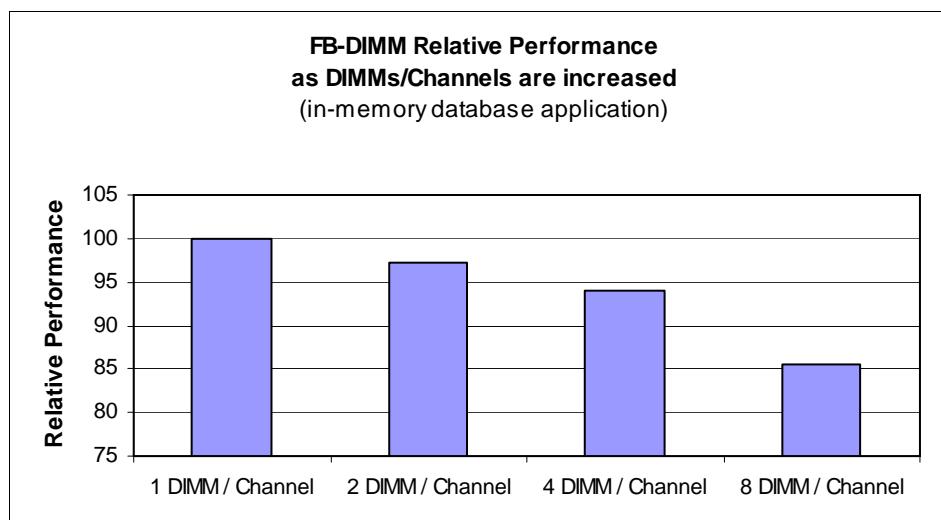


Figure 10-7 FB-DIMM relative performance

#### **10.2.7 MetaSDRAM**

Many DDR2- and DDR3-based servers are now supporting a new type of memory DIMM, known as MetaSDRAM, often shortened to the founding company's name, MetaRAM. This technology's primary benefit is to allow high

capacity DIMM sizes without the exponentially higher prices always commanded by the largest DIMM sizes.

MetaSDRAM accomplishes this by using a custom-designed chipset on each DIMM module which makes multiple, inexpensive SDRAMs appear as a single, large capacity SDRAM. Because this chipset acts as an onboard buffer and appears as a single electrical load on the memory bus, higher frequencies can often be achieved using MetaSDRAM-based DIMMs than would be possible using standard DIMMs, which often have to switch to slower speeds as multiple DIMMs are added per channel.

MetaSDRAM DIMMs offset the additional power gain imposed by the additional SDRAMs by implementing intelligent power management techniques, allowing two to four times the memory capacity to fit within a typical server's power and cooling capability.

Due to the inclusion of the additional MetaSDRAM chipset between the memory bus and the DDR2 or DDR3 SDRAMs, there is some latency added to the memory transactions. The impact of this latency, however, has been measured across a number of commercial application environments to impact overall application performance by just 1% to 2%. Because the technology has the potential to double a server's memory size, this small impact is well under the potential gains achievable from the added memory capacity. However, if MetaSDRAM is being used only to reduce cost in a server's memory subsystem, it is important to realize that performance can be slightly lower than standard DIMMs, depending on configuration.

See 10.12, "Memory rules of thumb" on page 234 for more discussion on gains from increased memory sizes.

### 10.2.8 DIMM nomenclature

The speed of a memory DIMM is most commonly indicated by numeric PC and DDR values on all current DIMM types. The PC value correlates to the theoretical peak transfer rate of the module, whereas the DDR value represents the bus transfer rate in Millions of transfers per second. The tables in this section list the nomenclature and the bus speed, transfer rate, and peak throughput.

With DDR, DDR2, and DDR3, because the SDRAM transfers data on both the falling and the rising edges of the clock signal, transfers speeds are double the memory bus clock speed. The peak throughput per channel can be calculated by multiplying the DDR transfer rate times the transfer size of 64bits (8 Bytes) per transfer.

Table 10-1 on page 197 summarizes the common nomenclatures for DDR memory.

*Table 10-1 DDR memory implementations*

DDR Module name	Bus speed	transfers/sec	Peak throughput
PC1600 (DDR-200)	100 MHz	200 M	1600 MBps
PC2100 (DDR-266)	133 MHz	266 M	2100 MBps
PC2700 (DDR-333)	167 MHz	333 M	2700 MBps
PC3200 (DDR-400)	200 MHz	400 M	3200 MBps

Table 10-2 lists the common DDR2 memory implementations.

*Table 10-2 DDR2 memory implementations*

DDR2 Module name	Bus speed	transfers/sec	Peak throughput
PC2-3200 (DDR2-400)	200 MHz	400 M	3200 MBps
PC2-4300 (DDR2-533)	266 MHz	533 M	4300 MBps
PC2-5300 (DDR2-667)	333 MHz	667 M	5300 MBps
PC2-6400 (DDR2-800)	400 MHz	800 M	6400 MBps

**Note:** Because FB-DIMMs use DDR2 SDRAM, they are typically referenced using the PC2 speed designation plus an FB-DIMM qualifier, for example:

PC2-5300 FB-DIMM

Table 10-3 lists current DDR3 memory implementations.

*Table 10-3 DDR3 memory implementations*

DDR3 Module name	Bus speed	transfers/sec	Peak throughput
PC3-6400 (DDR3-800)	400 MHz	800 M	6400 MBps
PC3-8500 (DDR3-1066)	533 MHz	1066 M	8533 MBps
PC3-10600 (DDR3-1333)	667 MHz	1333 M	10667 MBps
PC3-12800 (DDR3-1600)	800 MHz	1600 M	12800 MBps

You can find more detailed SDRAM specification information at:

<http://developer.intel.com/technology/memory/>

## 10.2.9 DIMMs layout

The DIMM location within the system's DIMM sockets is often mandated by a system's installation guide to ensure that DIMMs are in a supported ordering, can be seen by the memory controller, and are spread across the available memory channels to optimize bandwidth. However, there are also other performance implications for the DIMM layout.

For DDR and DDR2 DIMMs, optimal performance is typically obtained when fully populating the available DIMM slots with equivalent memory size and type of DIMMs. When this is not feasible in a solution, populating in multiples of 4 or 8 DIMMs, all of the same size and type, can often have a similar outcome.

These methods allow the memory controller to maintain the maximum number of open memory pages, and allow the memory controller to optimize throughput and latency by allowing address bit permuting (sometimes called *symmetric mode* or *enhanced memory mapping*). This reorders memory addresses to optimize memory prefetch efficiency and reduces average memory read/write latencies significantly.

Although FB-DIMM and DDR3 DIMMs can also benefit from these optimization mechanisms, they both have caveats that must be balanced to achieve optimized performance. As highlighted in “FB-DIMM performance” on page 195, FB-DIMMs increase in latency as additional DIMMs are added to each memory channel. In this case, using the fewest possible, high capacity DIMMs per channel while maintaining the required memory size will typically yield the best performance.

Similarly, the high speeds of DDR3 memories limit the number of DIMMs, or even total memory ranks, that can be populated per channel at each frequency. While it can vary from one implementation to another and should be verified with the systems configuration guide, it is possible that each DIMM added per channel will drop the memory speed to the next slowest speed.

For either of these DIMM technologies, care must be taken when specifying the number of DIMMs to ensure that the right balance of cost, performance, and memory size is obtained for the solution.

**Tip:** Pay careful attention to the memory layout, specifically to the number of DIMMs per channel, when comparing FB-DIMM-based or DDR3-based systems, because this can have an impact on overall system performance.

The significance of this discussion is that memory performance is highly dependent on not just whether the data is in cache or main memory, but also on how the access patterns appear to the memory controller. The access pattern

will strongly affect how the memory controller reorders or combines memory requests, whether successive requests hit the same page, and so forth. Memory performance is affected by a number of complex factors. Those factors can include the choice of architecture, processor frequency, memory frequency, the number of DIMMs in the system, and whether the system is set up as a NUMA or an SMP system.

### 10.2.10 Memory interleaving

*Interleaving* is a technique that is often used to organize DIMMs on the motherboard of a server to improve memory transfer performance. The technique can be implemented within a single cache line access or across multiple cache lines to improve total memory bandwidth. When two DIMMs are grouped together and accessed concurrently to respond to a single cache line request, the interleave is defined as two-way. When four DIMMs are grouped together and accessed concurrently for a single cache line, the interleave is four-way.

Interleaving improves memory performance because each DIMM in the interleave is given its memory address at the same time. Each DIMM begins the access while the memory controller waits for the first access latency time to expire. Then, after the first access latency has expired, all DIMMs in the interleave are ready to transfer multiple 64-bit objects in parallel, without delay.

Although interleaving was often a tunable parameter in older server architectures, it is typically optimized at system boot time by the BIOS on modern systems, and therefore does not need to be explicitly tuned.

## 10.3 Specifying memory performance

Performance with memory can be simplified into two main areas: bandwidth and latency.

### 10.3.1 Bandwidth

Basically, the more memory bandwidth you have, the better system performance will be, because data can be provided to the processors and I/O adapters faster. Bandwidth can be compared to a highway: the more lanes you have, the more traffic you can handle.

The memory DIMMs are connected to the memory controller through memory channels, and the memory bandwidth for a system is calculated by multiplying

the number of data width of a channel by the number of channels, and then multiplied by the frequency of the memory.

For example, if a processor is able to support up to 400 MHz (DDR-400) registered ECC memory and has two 8-byte channels from the memory controller to access the memory, then the memory bandwidth of the system will be 8 bytes\*2 channels\*400 MHz, or 6.4 GBps.

**Tip:** The theoretical memory bandwidth does not depend on the memory technology (DDR2 or DDR3), but on the memory bus frequency, bus width, and number of channels.

### 10.3.2 Latency

The performance of memory access is usually described by listing the number of memory bus clock cycles that are necessary for each of the 64-bit transfers needed to fill a cache line. Cache lines are multiplexed to increase performance, and the addresses are divided into row addresses and column addresses.

A row address is the upper half of the address (that is, the upper 32 bits of a 64-bit address). A column address is the lower half of the address. The row address must be set first, then the column address must be set. When the memory controller is ready to issue a read or write request, the address lines are set, and the command is issued to the DIMMs.

When two requests have different column addresses but use the same row address, they are said to “occur in the same page.” When multiple requests to the same page occur together, the memory controller can set the column address once, and then change the row address as needed for each reference. The page can be left open until it is no longer needed, or it can be closed after the first request is issued. These policies are referred to as a *page open* policy and a *page closed* policy, respectively.

The act of changing a column address is referred to as Column Address Select (CAS).

There are three common access times:

- ▶ CAS: Column Address Select
- ▶ RAS to CAS: delay between row access and column access
- ▶ RAS: Row Address Strobe

Sometimes these numbers are expressed as *x-y-y* by manufacturers.

These numbers are expressed in clocks, and might be interpreted as wait times or *latency*. Therefore, the lower these numbers are the better, because access times imply data access latency.

CAS Latency (CL) measures the number of memory clocks that elapse between the time a memory controller sets the column address to request a line of data and when the DIMM is able to respond with that data. Even if other latencies are specified by memory manufacturers, CL is the most commonly used when talking about latency. If you look at the sticker on a DIMM (Figure 10-8), it might list the CL value for that particular device.

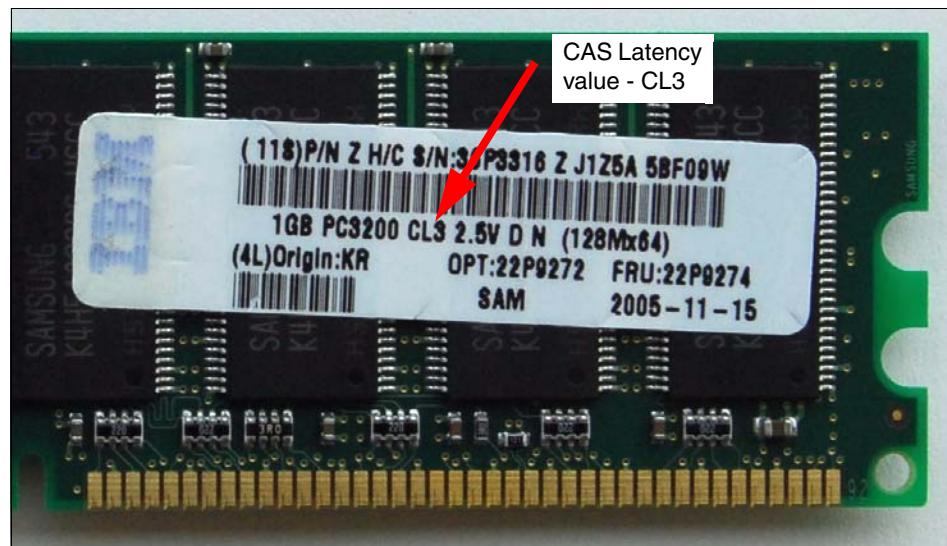


Figure 10-8 CAS Latency value as printed on a PC3200 (400 MHz) DDR DIMM

CL values of 2.5 or 3.0 are typical of 400 MHz technology. With 533 MHz and 667 MHz memory, typical values for the CL are respectively 4 and 5. Numbers with fractions are possible because data can be clocked at a different rate than commands. With DDR memory, data is clocked at double the speed of commands. For example, 400 MHz DDR memory has a data clock of 400 MHz and a native clock (command and address) of 200 MHz. Thus, CL2.5 memory has a CL of 2.5 command clocks, which is equivalent to five data clocks.

It is important to note that system level transaction latency is comprised of many factors, only one of which is the latency of the memory DIMMs themselves.

### 10.3.3 Loaded versus unloaded latency

When talking about latency, manufacturers generally refer to CL, which is a theoretical latency and corresponds to an *unloaded* latency condition. A single access to memory from a single thread, executed by a single processor while no other memory accesses are occurring is referred to as unloaded latency.

As soon as multiple processors or threads concurrently access memory, the latencies increase. This condition is termed *loaded* latency. While the conditions for unloaded latency are easily defined when memory accesses only happen one at a time, loaded latency is much more difficult to specify. Since latency will always increase as load is added, and is also dependent on the characteristics of the load itself (for example, the percentage of the time the data is found within an open page), loaded latency cannot be characterized by any single, or even small group, of data points. For this reason alone, memory latencies are typically compared only in the unloaded state.

Since real systems and applications are almost never unloaded, there is typically very little value in utilizing unloaded latency as a comparison point between systems of different architectures.

### 10.3.4 STREAM benchmark

Many benchmarks exist to test memory performance. Each benchmark acts differently and gives different results because each simulates different workloads. However, one of the most popular and simple benchmarks is STREAM.

STREAM is a synthetic benchmark program that measures sequential access memory bandwidth in MBps and the computational rate for simple vector kernels. The benchmark is designed to work with larger data sets than the available cache on most systems, so the results are indicative of very large vector-style applications. It provides real-world sustained memory bandwidth and not the theoretical peak bandwidth that is provided by vendors.

It is critical to note, however, that very few server applications access large sections of memory sequentially as the STREAM benchmark does. More commonly, real application workloads access memory randomly, and random memory performance can be many times slower than sequential memory bandwidth due to system-level memory access latencies. Such latencies include overheads from the processor cache, bus protocols, memory controller, and the memory latency itself. In addition, tuning the memory subsystem for maximum bandwidth is often done to the detriment of random memory performance, so a system that is better with STREAM could actually perform worse in many real world workloads.

**Note:** Results obtained from the STREAM benchmark, and other benchmarks that operate similarly, can be misleading, because very few application environments perform sustained sequential memory operations as STREAM does.

You can find more information about the STREAM benchmark at:

<http://www.cs.virginia.edu/stream>

Although STREAM provides the maximum sustainable memory bandwidth of a given system, detailed focus on other aspects of server performance should be made as well.

## 10.4 SMP and NUMA architectures

There are two fundamental system architectures used in the x86 server market: SMP and NUMA. Each architecture can have its strengths and limitations, depending on the target workload, so it is important to understand these details when comparing systems.

### 10.4.1 SMP architecture

Prior to the introduction of Intel 5500-series processors, most systems based on Intel processors typically use the Symmetric Multiprocessing (SMP) Architecture. The exception to this is the IBM x3950 M2, which uses a combination of SMP in each node and NUMA architecture between nodes.

SMP systems are fundamentally defined by having one or more Front-Side Buses (FSB) that connect the processors to a single memory controller, also known as a *north bridge*. Although older system architectures commonly deployed a single shared FSB, newer systems often have one FSB per processor socket.

In an SMP architecture, the CPU accesses memory DIMMs through the separate memory controller or north bridge. The north bridge handles traffic between the processor and memory, and controls traffic between the processor and I/O devices, as well as data traffic between I/O devices and memory. Figure 10-9 on page 204 shows the central position of the north bridge and the shared front-side bus. These components play the dominant role in determining memory performance.

Whether the memory controller and processors implement shared, or separate front side buses, there is still a single point of contention in the SMP design. As shown in Figure 10-9, if a single CPU is consuming the capabilities of the FSB, memory controller, or memory buses, there will be limited gain from using a second processor. For applications that have high front-side bus and memory controller utilizations (which is most common in scientific and technical computing environments), this architecture can limit the potential for performance increases. This becomes more of an issue as the number of processor cores are increased, because the requirement for additional memory bandwidth increases for these workloads, as well.

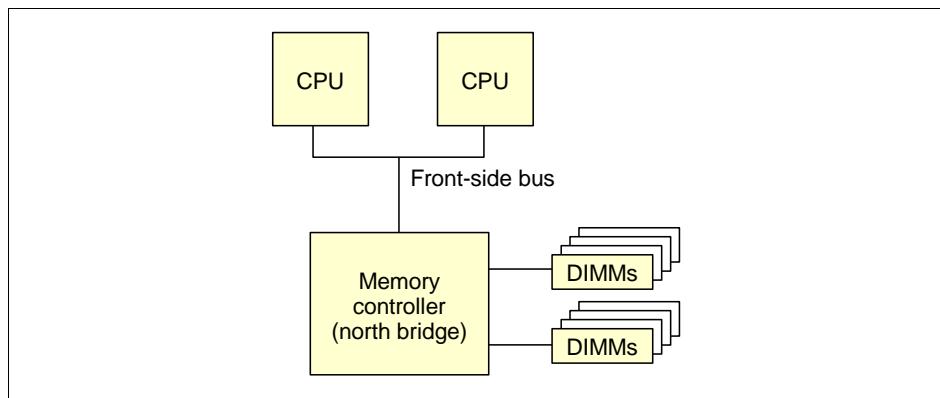


Figure 10-9 An Intel dual-processor memory block

The speed of the north bridge is tied to the speed of the front-side bus, so even as processor clock speeds increase, the latency to memory remains virtually the same. The speed of the front-side bus places an upper bound on the rate at which a processor can send data to or receive data from memory. For this reason, aggregate memory bandwidth of the memory channels attached to the memory controller is always tuned to be equal to or greater than the aggregate bandwidth of the front-side bus interfaces attached to the memory controller.

#### 10.4.2 NUMA architecture

To overcome the limitations of a shared memory controller design, the latest Intel processors, as well as Opteron processors, utilize a Non-Uniform Memory Architecture (NUMA), rather than an SMP design. In NUMA configurations, the memory controller is integrated into the processor, which can be a benefit for two reasons:

- ▶ The memory controller is able to be clocked at higher frequencies, because the communications between the processor cores and memory controller do not have to go out to the system board. In this design, as the processor speed

is increased, the memory controller speed can often also be increased, thus reducing the latency through the memory controller while increasing the sustainable bandwidth through the memory controller.

- ▶ As additional processors are added to a system, additional memory controllers are also added. Because the demand for memory increases as processing capability increases, the additional memory controllers provide linear increases in memory bandwidth to accommodate the load increase of additional processors, thus enabling the potential for very efficient processor scaling and high potential memory bandwidths.

In x86 server architectures, all system memory must be available to all server processors. This requirement means that with a NUMA architecture, we have a new potential performance concern, *remote memory*. From a processor point of view, *local memory* refers to the memory connected to the processor's integrated memory controller. Remote memory, in comparison, is that memory that is connected to another processor's integrated memory controller.

**Note:** The IBM eX4-based servers employ a similar technique when used in multi-node configurations. For example, with a two-node x3950 M2 server, processors and memory are spread over two nodes. Each node contains its own CPUs, memory controller, and memory. When the nodes are merged together to form a single logical system, these resources all communicate as one large system, although they operate on the same principles of local and remote memory.

Figure 10-10 on page 206 shows the architecture of an AMD Opteron processor. As in Figure 10-9 on page 204, there is a processor core and cache. However, in place of a bus interface unit and an external memory controller, there is an integrated memory controller (MCT), an interface to the processor core (SRQ), three HyperTransport (HT) units and a crossbar switch to handle routing data, commands, and addresses between processors and I/O devices.

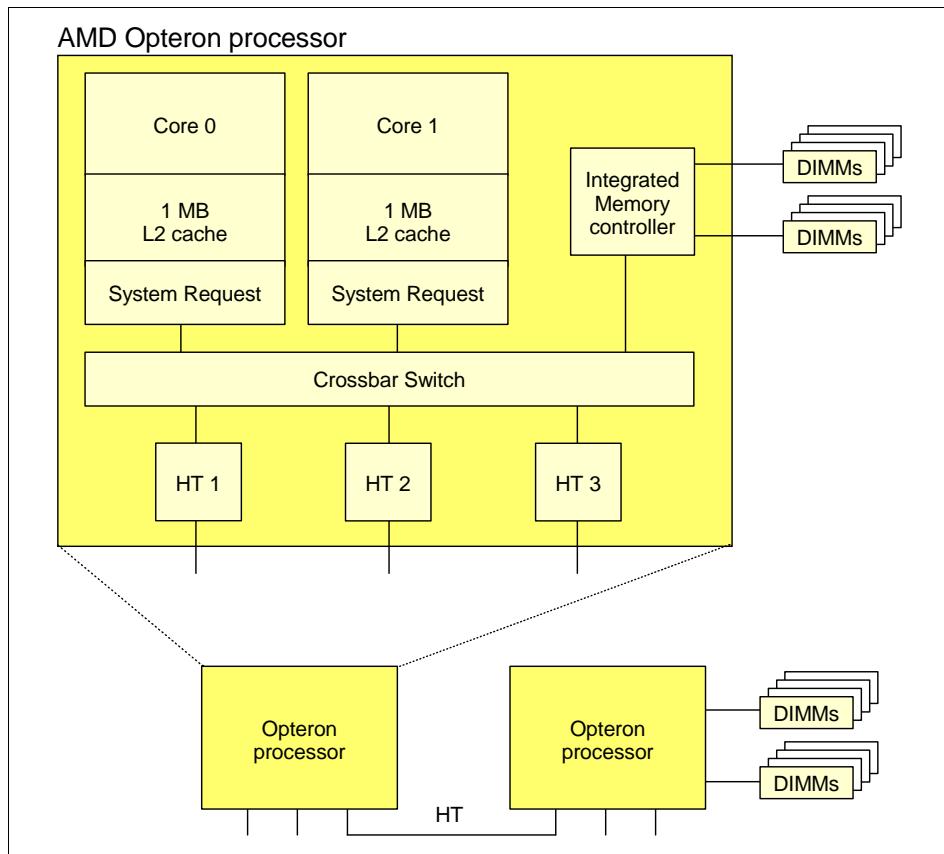


Figure 10-10 An AMD dual-processor memory block

Two of the HyperTransport (HT or cHT) units are typically used for connecting to and communicating with the other processors. The third HT unit is to connect to I/O devices. More detail on AMD and Intel processors can be found in Chapter 6, “Processors and cache subsystem” on page 93.

When a program thread is executing on a processor core, the memory it needs to reference could exist in the memory attached to that processor's integrated memory controller, in the memory attached to a different memory controller, or in both. Because access to remote memory must traverse an additional crossbar switch and HT link twice to retrieve remote data, remote memory accesses can be significantly slower than local memory accesses.

Although modern operating systems work to limit the number of remote memory accesses as much as possible to keep performance as high as possible, not all memory accesses can be guaranteed to get handled from the local memory

controller. Thus, care must be taken when using a NUMA system to ensure maximum performance is achieved.

**Note:** In a NUMA architecture, optimal performance is achieved only when the application and OS work together to minimize remote memory requests.

## 10.5 The 32-bit 4 GB memory limit

A memory address is a unique identifier for a memory location at which a processor or other device can store a piece of data for later retrieval. Each address identifies a single byte of storage. All applications use virtual addresses, not physical. The operating system maps any (virtual) memory requests from applications into physical locations in RAM. When the combined total amount of virtual memory used by all applications exceeds the amount of physical RAM installed in the server, the difference is stored in the page file, which is also managed by the operating system.

32-bit CPUs, such as older Intel Xeon processors, have an architectural limit of only being able to directly address 4 GB of memory. With many enterprise server applications now requiring significantly greater memory capacities, CPU and operating system vendors have developed methods to give applications access to more memory.

The first method was implemented by Microsoft with its Windows NT® 4.0 Enterprise Edition operating system. Prior to Enterprise Edition, the 4 GB memory space in Windows was divided into 2 GB for the operating system kernel and 2 GB for applications. Enterprise Edition offered the option to allocate 3 GB to applications and 1 GB to the operating system.

The 32-bit Linux kernels, by default, split the 4 GB virtual address space of a process in two parts: 3 GB for the user-space virtual addresses and the upper 1 GB for the kernel virtual addresses. The kernel virtual area maps to the first 1 GB of physical RAM and the rest is mapped to the available physical RAM.

The potential issue here is that the kernel maps directly all available kernel virtual space addresses to the available physical memory, which means a maximum of 1 GB of physical memory for the kernel. For more information, see the article *High Memory in the Linux Kernel*, which is available at:

<http://kerneltrap.org/node/2450>

For many modern applications, however, 3 GB of memory is simply not enough. To address more than 4 GB of physical memory on 32-bit operating systems, two

mechanisms are typically used: Physical Address Extension (PAE) and for Windows, Address Windowing Extensions (AWE).

### 10.5.1 Physical Address Extension

32-bit operating systems written for the 32-bit Intel processor use a segmented memory addressing scheme. The maximum directly addressable memory is 4 GB ( $2^{32}$ ). However, an addressing scheme was created to access memory beyond this limit: the Physical Address Extension (PAE).

This addressing scheme is part of the Intel Extended Server Memory Architecture and takes advantage of the fact that the 32-bit memory controller actually has 36 bits that are available for use for memory and L2 addressing. The extra four address bits are normally not used but can be employed along with the PAE scheme to generate addresses above the standard 4 GB address limit.

PAE uses a four-stage address generation sequence and accesses memory using 4 KB pages, as shown in Figure 10-11.

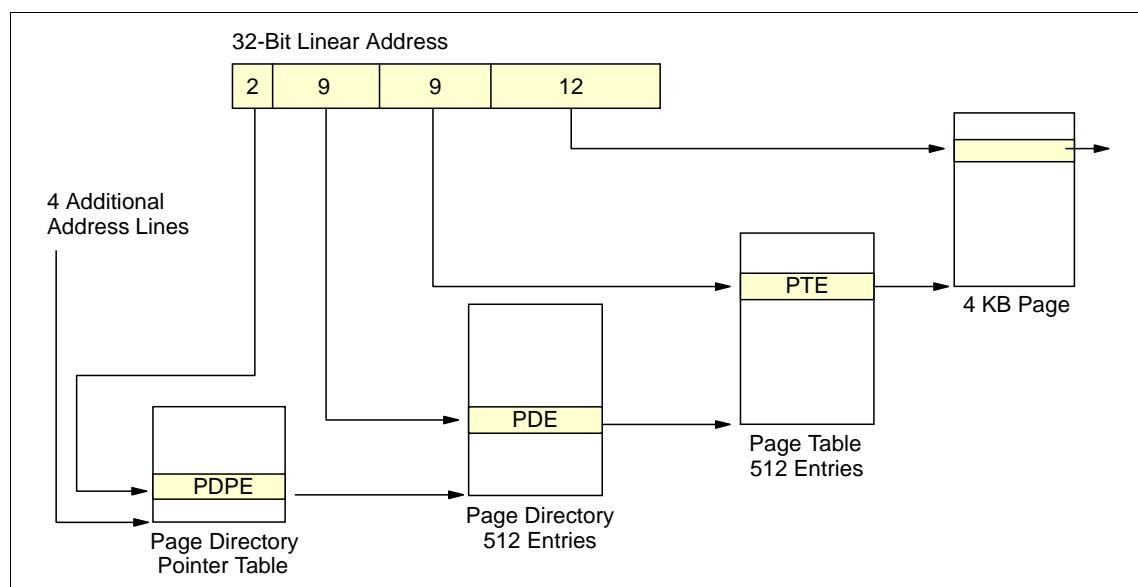


Figure 10-11 PAE-36 address translation

Four reserved bits of control register CR3 pad the existing 32-bit address bus with an additional 4 bits, enabling 36-bit software and hardware addressing to access 64 GB of memory.

PAE maintains the existing Intel 4 KB memory page definition and requires four levels of redirection to generate each physical memory address. However, as memory capacity increases, using a fixed size 4 KB page results in increased memory management overhead, because the number of memory pages grows as the size of maximum addressable memory increases. Using a larger memory page would reduce the total number of pages and the overhead required to point to any one page, because fewer pages would need to be addressed.

## Windows PAE and Address Windowing Extensions

Although recent Service Packs for Enterprise and Datacenter editions of Windows Server 2003 and 2008 now have PAE enabled by default, older versions of these operating systems may not. For these older operating systems, the user must add the /PAE switch to the corresponding entry in the `BOOT.INI` file to access memory above the 4GB boundary.

PAE is supported only on 32-bit versions of the Windows operating systems. 64-bit versions of Windows do not support, nor need, PAE, as sufficient address space is already accommodated.

**Note:** If you are using a processor with the Data Execution Prevention (DEP) feature (Intel processors refer to this as Execute Disable Bit or XD feature and AMD processors call this the no-execute page-protection processor or NX feature) and have it enabled, then Windows Server 2003 32-bit will automatically enable PAE.

To support DEP, Windows will automatically load the PAE kernel no matter how much memory is installed, and you do not have to use the /PAE boot switch in the `boot.ini` file.

The following 32-bit Windows versions support PAE, with the given amount of physical RAM indicated:

- ▶ Windows 2000 Advanced Server (8 GB maximum)
- ▶ Windows 2000 Datacenter Server (32 GB maximum)
- ▶ Windows XP (all versions) (4 GB maximum)
- ▶ Windows Server 2003, Standard Edition (4 GB maximum)
- ▶ Windows Server 2003, Enterprise Edition (32 GB maximum)
- ▶ Windows Server 2003, Enterprise Edition R2 or SP1 (64 GB maximum)
- ▶ Windows Server 2003, Datacenter Edition (64 GB maximum)
- ▶ Windows Server 2003, Datacenter Edition R2 or SP1 (128 GB maximum)
- ▶ Windows Server 2008, Standard and Web Editions (4 GB maximum)
- ▶ Windows Server 2008, Enterprise Edition (64 GB maximum)
- ▶ Windows Server 2008, Datacenter Edition (64 GB maximum)

Although PAE enables the operating system to map to memory regions above the 4 GB limit, other mechanisms must be employed for application code to exceed the 2-3 GB application limit of a 32-bit architecture. Address Windowing Extensions (AWE) accomplish this by using a set of Windows APIs that remap portions of extended memory space into the application's addressable memory space. This remapping process does incur a performance overhead, however, and most applications that originally implemented AWE now recommend usage of 64-bit operating system and application versions.

**Important:** The two BOOT.INI switches /PAE and /3GB interact with each other and in some circumstances should not be used together. See 13.14.1, “Interaction of the /3GB and /PAE switches” on page 393 for details.

## 10.6 64-bit memory addressing

To break through the 4 GB limitations of 32-bit addressing, CPU and operating system vendors extended the x86 specification to 64-bits. Known by many names, this technology is most generally referred to as x86-64 or x64, though Intel refers to it as EM64T, and AMD uses the name AMD64. Fundamentally, this technology enables significantly increased memory addressability for both operating systems and applications.

Table 10-4 illustrates the differences between the 32-bit and 64-bit operating systems.

*Table 10-4 Virtual memory limits*

Description	32-bit	64-bit (x64)
Total virtual address space	4 GB	16 TB
Virtual address space per 32-bit application	2 GB <sup>a</sup>	2 GB <sup>b</sup>
Virtual address space per 64-bit process	Not applicable	8 TB
Virtual address space for the operating system kernel	2 GB <sup>a</sup>	8 TB
Paged pool	470 MB	128 GB
Non-paged pool	256 MB	128 GB
System cache	1 GB	1 TB

a. 3 GB for the application and 1 GB for the kernel if system booted with a /3GB switch.

b. 4 GB if the 32-bit application has the LARGEADDRESSAWARE flag set (LAA).

The width of a memory address dictates how much memory the processor can address. As shown in Table 10-5, a 32-bit processor can address up to  $2^{32}$  bytes or 4 GB. A 64-bit processor can theoretically address up to  $2^{64}$  bytes or 16 Exabytes (or 16777216 Terabytes).

*Table 10-5 Relation between address space and number of address bits*

Bits (Notation)	Address space
8 ( $2^8$ )	256 bytes
16 ( $2^{16}$ )	65 KB
32 ( $2^{32}$ )	4 GB
64 ( $2^{64}$ )	18 Exabytes (EB)

Current implementation limits are related to memory technology and economics. As a result, physical addressing limits for processors are typically implemented using less than the full 64 potential address bits, as shown in Table 10-6.

*Table 10-6 Memory supported by processors*

Processor	Flat addressing
Intel Xeon MP Gallatin (32-bit)	4 GB (32-bit)
Intel Xeon Nocona and Cranford processors (64-bit)	64 GB (36-bit)
Intel 64 Technology (All other x64 processors)	1 TB (40-bit)
AMD Opteron (64-bit)	256 TB (48-bit)

These values are the limits imposed by the processors themselves. They represent the maximum theoretical memory space of a system using these processors.

**Tip:** Both Intel 64 and AMD64 server architectures can utilize either the 32-bit (x86) or 64-bit (x64) versions of their respective operating systems. However, the 64-bit architecture extensions will be ignored if 32-bit operating system versions are employed. For systems using x64 operating systems, it is important to note that the drivers must also be 64-bit capable.

## 10.7 Advanced ECC memory (Chipkill)

All current System x servers implement standard error checking and correcting (ECC) memory. ECC memory detects and corrects any single-bit error. It can also detect double-bit errors, but is unable to correct them. Triple-bit and larger errors might not be detected.

With the increase in the amount of memory that is used in servers, there is a need for better memory failure protection. As the area of DRAM silicon increases and the density of those DRAM components also increases, there is a corresponding increase in multi-bit failures. This situation means that for larger amounts of memory, there is an increasing propensity for failures that occur to affect more than one data bit at a time and, therefore, overwhelm the traditional single-error correct (SEC) ECC memory module.

IBM has developed and uses a technology known as Chipkill Protect ECC DIMM, which allows an entire DRAM chip on a DIMM to fail while the system continues to function. These new DIMMs have been designed so that there is no performance degradation over SEC or standard ECC DIMMs.

Figure 10-12 shows the results of a failure rate simulation for 32 MB of parity memory, 1 GB of standard SEC ECC, and 1 GB of IBM Advanced ECC memory. The simulation was for three years of continuous operation and showed the significant reduction in failures when using advanced ECC (approximately two orders of magnitude).

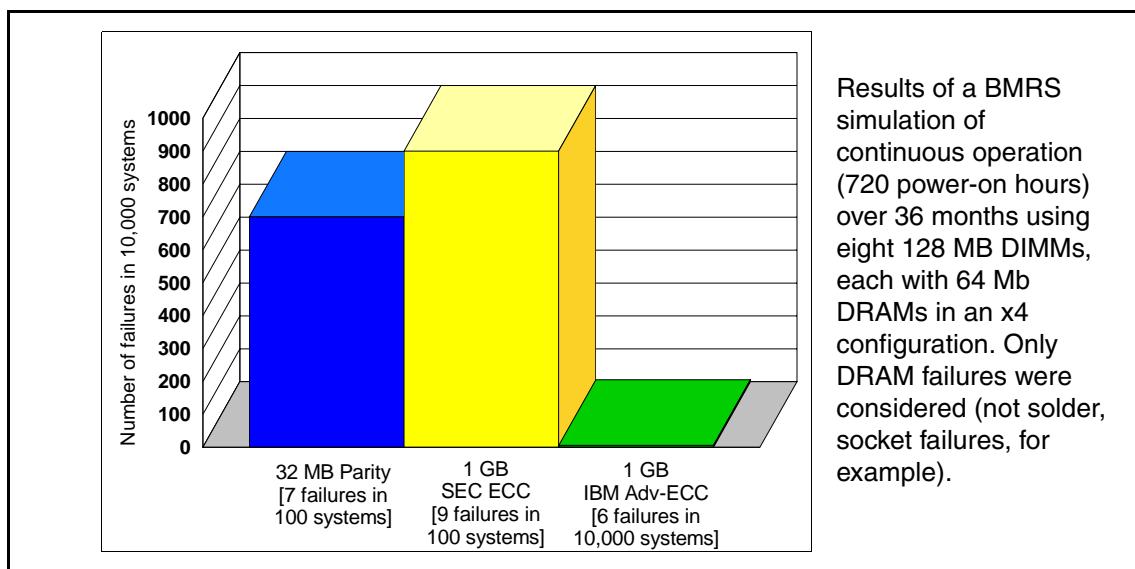


Figure 10-12 Memory failure rate comparison

The capability that the data shows for the memory subsystem is fundamentally the same as RAID technology used for the disk subsystem today. In fact, from a marketing perspective, it could be called RAID-M for Redundant Array of Inexpensive DRAMs for main Memory. This name captures the essence of its function: on-the-fly, automatic data recovery for an entire DRAM failure.

IBM is now offering this advanced ECC technology integrated as an option for several members of the System x family. For more information, read the white paper *IBM Chipkill Memory*, which is available from:

<http://www.ibm.com/systems/support/supportsite.wss/docdisplay?brandind=5000008&lnocid=MCGN-46AMQP>

## 10.8 Memory mirroring

Memory mirroring is similar in function to a disk subsystem using RAID-1 mirroring. The aim of mirroring the memory is to enhance the server's availability. If a memory DIMM fails, then the mirrored DIMM can handle the data. It is a redundant feature that provides failover capabilities.

Mirroring forces memory to be divided into two equal parts. Because the memory pool is divided in two, the total amount of available memory is half the amount of installed memory. Thus, if a server is installed with 64 GB of total RAM and memory mirroring is enabled (in the server's BIOS), the total available memory seen by the operating system will be 32 GB. It is important to verify correct DIMMs placement in the system to ensure availability of this feature.

Although the specific configuration menus are slightly different from one system to the next, Figure 10-13 on page 214 shows that this typically found in the server's BIOS Setup menu if you select **Advanced Settings → Memory Settings**. A window like this should appear, allowing you to change the Memory Configuration option from the default value to Mirrored which is also known in some servers as Full Array Memory Mirroring).

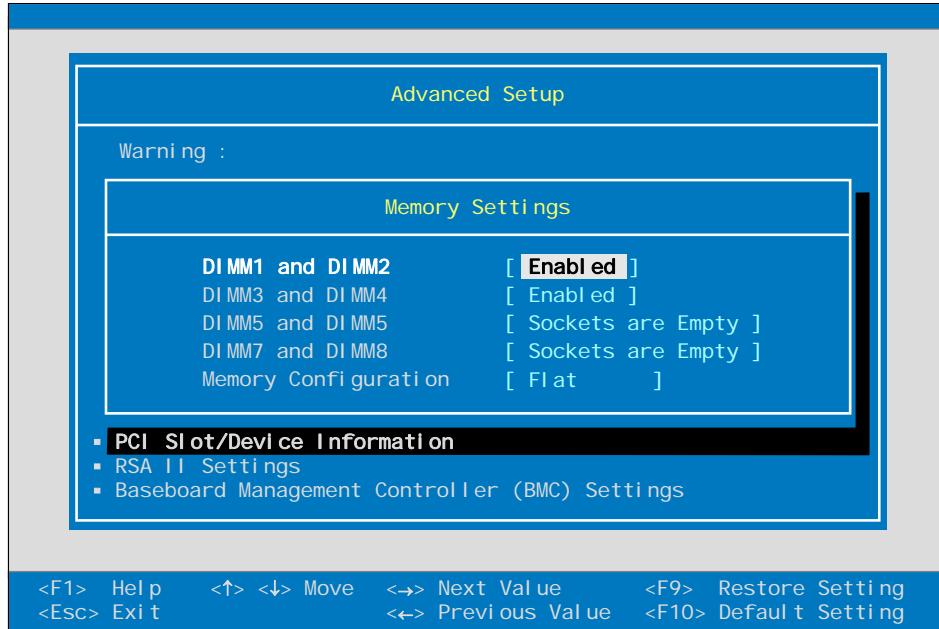


Figure 10-13 Enabling memory mirroring

## Performance of Memory mirroring

Memory mirroring functions by writing each block of data to each of the mirrored portions of memory. Since two physical memory writes occur for each application write request, memory write bandwidth is effectively cut in half. However, since the memory controller can interleave the read requests among both memory mirrors, memory read bandwidth is not reduced. Because real applications balance both memory read and writes, the actual performance impact will be dependent on the application characteristics and sensitivity to memory bandwidth.

Figure 10-14 on page 215 shows the performance impact of enabling memory mirroring in a transactional database environment.

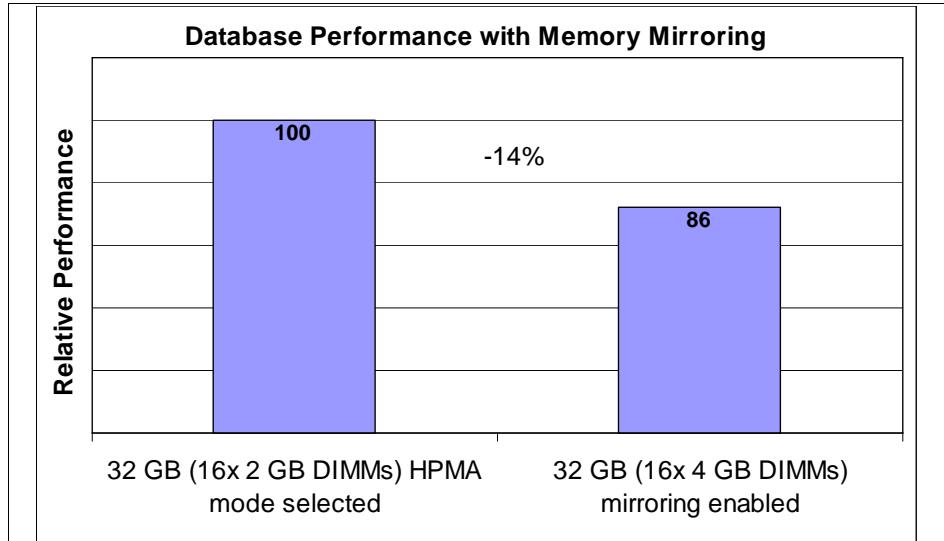


Figure 10-14 Memory Mirroring performance

**Tip:** Unlike RAID-1 disk configurations, memory mirroring will not improve memory bandwidth and could have a negative effect on performance.

## 10.9 Intel Xeon 5500 Series Processors

The Intel Xeon 5500 Processor Series is the family of next-generation quad-core processors targeted at the two-socket server space. It is the common building block across a number of IBM platforms, including the IBM BladeCenter HS22 blade server, the 1U x3550 M2 and 2U x3650 M2 rack servers, and the IBM iDataPlex dx360 M2 server.

With the Xeon 5500 series processors, Intel has diverged from its traditional Symmetric Multiprocessing (SMP) architecture to a Non-Uniform Memory Access (NUMA) architecture. In a two-processor scenario, the Xeon 5500 series processors are connected through a serial coherency link called QuickPath Interconnect (QPI). The QPI is capable of 6.4, 5.6 or 4.8 GT/s (gigatransfers per second), depending on the processor model.

The Xeon 5500 series integrates the memory controller within the processor, resulting in two memory controllers in a two-socket system. Each memory controller has three memory channels and supports DDR-3 memory. Depending

on processor model, the type of memory, and the population of memory, memory may be clocked at 1333MHz, 1066MHz, or 800MHz.

Each memory channel supports up to 3 DIMMs per channel (DPC), for a theoretical maximum of 9 DIMMs per processor or 18 per 2-socket server; refer to Figure 10-15. However, the actual maximum number of DIMMs per system is dependent upon the system design.

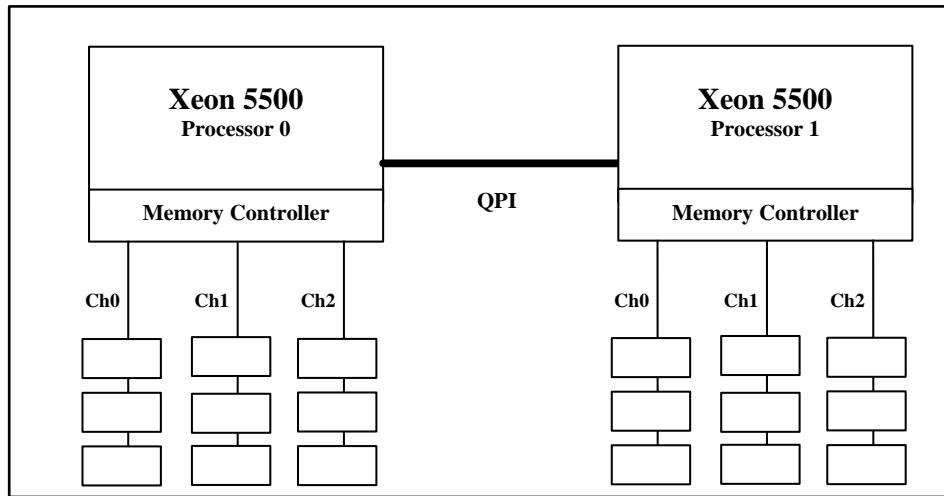


Figure 10-15 Xeon 5500 architecture showing maximum memory capabilities

### 10.9.1 HS22 Blade

HS22 is designed with 12 DIMM slots as shown in Figure 10-16 on page 217 and Figure 10-17 on page 217. The 12-DIMM layout provides 6 DIMMs per socket and 2 DIMMs per channel (DPC).

Figure 10-16 on page 217 illustrates the HS22 DIMM slots architectural layout.

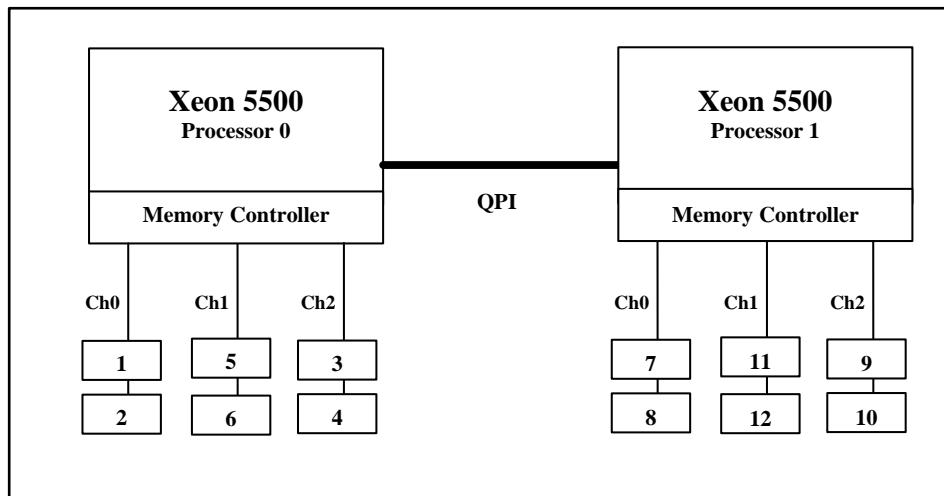


Figure 10-16 HS22 DIMM slots architectural layout

Figure 10-17 illustrates the HS22 DIMM slots physical layout.

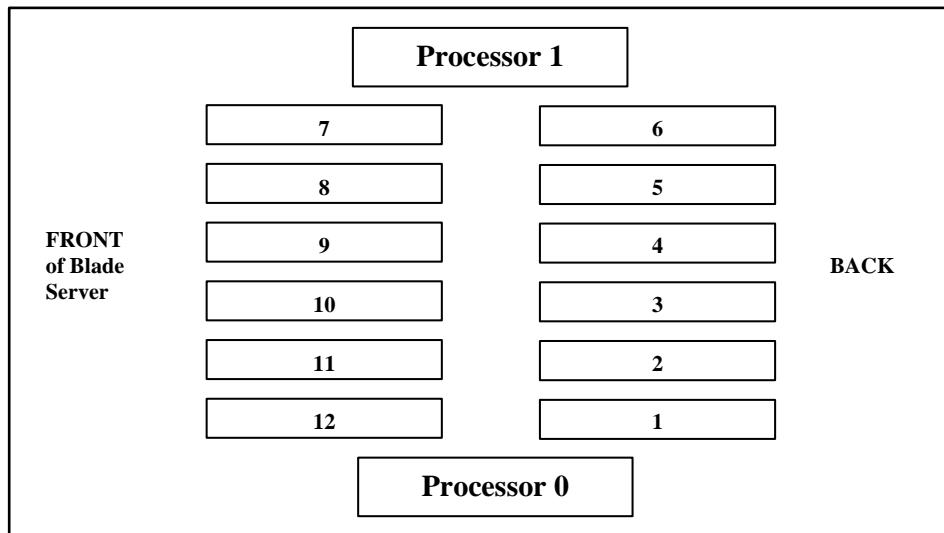


Figure 10-17 HS22 DIMM slots physical layout

### 10.9.2 System x3550 M2, x3650 M2, and iDataPlex dx360 M2

As shown in Figure 10-18 on page 218 and Figure 10-19 on page 219, the other IBM servers containing Xeon 5500 series processors, that is, the System x3550 M2, the x3650 M2, and the iDataPlex dx360 M2, each provide 16 DIMM slots.

Like the HS22, each processor has an equal number of DIMM slots. However, unlike the HS22 all memory channels do not have equal DPC (DIMMs per channel).

Figure 10-18 illustrates the slots architectural layout.

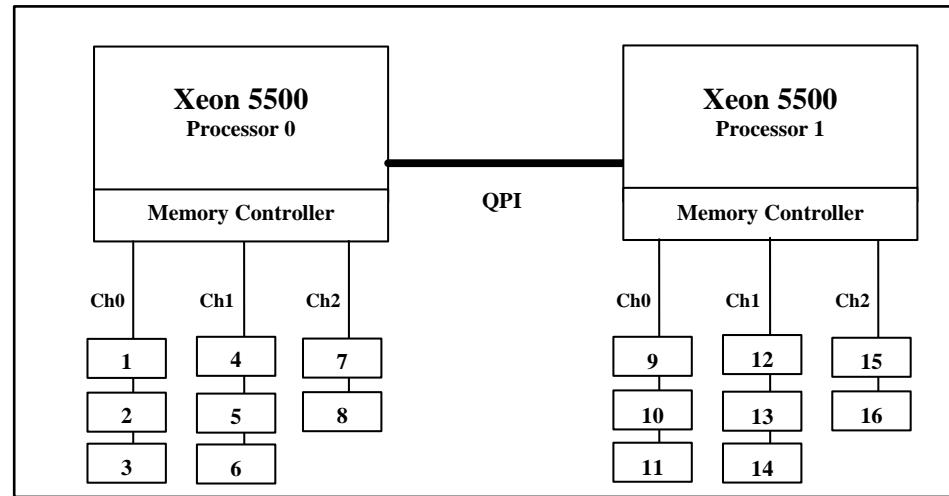


Figure 10-18 x3550 M2/x3650 M2/dx360 M2 DIMM slots architectural layout

Figure 10-19 on page 219 illustrates the slots physical layout.

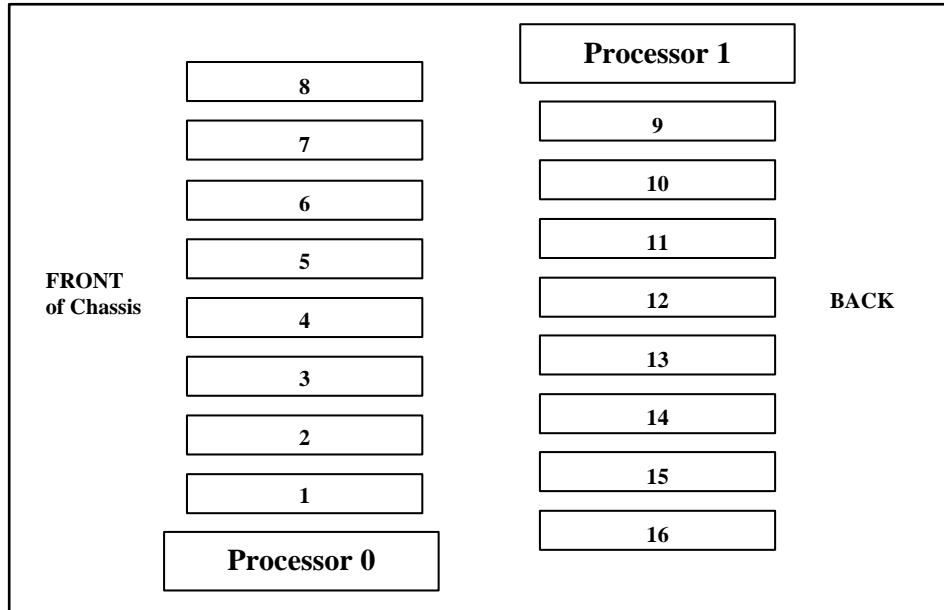


Figure 10-19 x3550 M2/x3650 M2/dx360 M2 DIMM slots physical layout

### 10.9.3 Memory performance

With the varied number of configurations possible in the Xeon 5500 series processor-based systems, a number of variables emerge that influence processor/memory performance. The main variables are memory speed, memory interleaving, memory ranks and memory population across various memory channels and processors. Depending on the processor model and number of DIMMs, the performance of the Xeon 5500 platform will see large memory performance variances. We will look at each of these factors more closely in the next sections.

As mentioned earlier, the memory speed is determined by the combination of three aspects:

- ▶ Processor model

The initial Xeon 5500 series processor-based offerings will be categorized into three bins called Performance, Volume and Value. The bins have the ability to clock memory at different maximum speeds, as listed in Table 10-7 on page 220.

*Table 10-7 Maximum memory speeds*

Xeon 5500 model	Maximum memory speed
X55xx processor models	1333 MHz
E552x or L552x and up	1066 MHz
E550x	800 MHz

The processor model will limit the maximum frequency of the memory.

**Note:** Because of the integrated memory controllers, the former front-side bus (FSB) no longer exists.

► DDR3 DIMM speed

DDR-3 memory will be available in various sizes at speeds of 1333MHz and 1066MHz. 1333MHz represents the maximum capability at which memory can be clocked. However, the memory will not be clocked faster than the capability of the processor model and will be clocked appropriately by the BIOS.

► DIMMs per Channel (DPC)

The number and type of DIMMs and the channels in which they reside will also determine the speed at which memory will be clocked. Table 10-8 describes the behavior of the platform. The table assumes a 1333MHz-capable processor model (X55xx). If a slower processor model is used, then the memory speed will be the lower of the memory speed and the processor model memory speed capability. If the DPC is not uniform across all the channels, then the system will clock to the frequency of the slowest channel.

*Table 10-8 Memory speed clocking (Full-speed configurations highlighted)*

DPC	DIMM speed	Ranks per DIMM	Memory speed
1	1333 MHz	1,2	1333 MHz
2	1333 MHz	1,2	1066 MHz
3	1333 MHz	1,2	800 MHz
1	1333 MHz	4	1066 MHz
2	1333 MHz	4	800 MHz
1	1066 MHz	1,2	1066 MHz

DPC	DIMM speed	Ranks per DIMM	Memory speed
2	1066 MHz	1,2	1066 MHz
3	1066 MHz	1,2	800 MHz
1	1066 MHz	4	1066 MHz
2	1066 MHz	4	800 MHz

### Low-level performance specifics

It is important to understand the impact of the performance of the Xeon 5500 series platform, depending on the memory speed. We will use both low-level memory tools and application benchmarks to quantify the impact of memory speed.

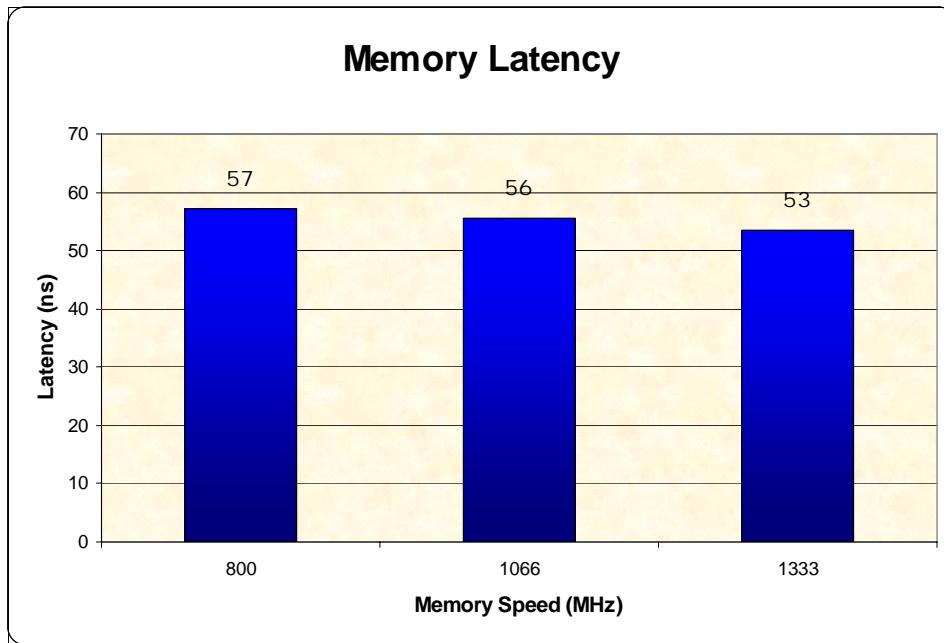
Two of the key low-level metrics that are used to measure memory performance are memory latency and memory throughput. We use a base Xeon 5500 2.93GHz, 1333MHz-capable 2-socket system for this analysis.

The memory configurations for the three memory speeds in the following benchmarks are as follows:

- ▶ 1333 MHz – 6 x 4GB dual-rank 1333 MHz DIMMs
- ▶ 1066 MHz – 12 x 2GB dual-rank DIMMs for 1066 MHz
- ▶ 800 MHz – 12 x 2GB dual-rank DIMMs clocked down to 800 MHz in BIOS

Memory ranks are explained in detail in 10.9.5, “Memory ranks” on page 227.

As shown in Figure 10-20 on page 222, we show the unloaded latency to local memory. The unloaded latency is measured at the application level and is designed to defeat processor prefetch mechanisms. As shown in the figure, the difference between the fastest and slowest speeds is about 10%. This represents the high watermark for latency-sensitive workloads. Another important thing to note is that this is almost a 50% decrease in memory latency when compared to the previous generation Xeon 5400 series processor on 5000P chipset platforms.



*Figure 10-20 Xeon 5500 series memory latency as a function of memory speed*

A better indicator of application performance is memory throughput. We use the triad component of the STREAMS benchmark to compare the performance at different memory speeds. The memory throughput assumes all local memory allocation and all 8 cores utilizing main memory.

As shown in Figure 10-20, the performance gain from running memory at 1066MHz versus 800MHz is 28%, and the performance gain from running at 1333MHz versus 1066MHz is 9%. As a result, the performance penalty of clocking memory down to 800MHz is far greater than clocking it down to 1066MHz.

However, greater memory capacity comes with lower memory speed. Alternatively, it is possible to achieve the same memory capacity at lower cost but at a lower memory speed. So, there is a distinct trade-off of memory capacity, performance, and cost.

Regardless of memory speed, the Xeon 5500 platform represents a significant improvement in memory bandwidth over the previous Xeon 5400 platform. At 1333MHz, the improvement is almost 500% over the previous generation. This huge improvement is mainly due to dual integrated memory controllers and faster DDR-3 1333MHz memory. This improvement translates into improved application performance and scalability.

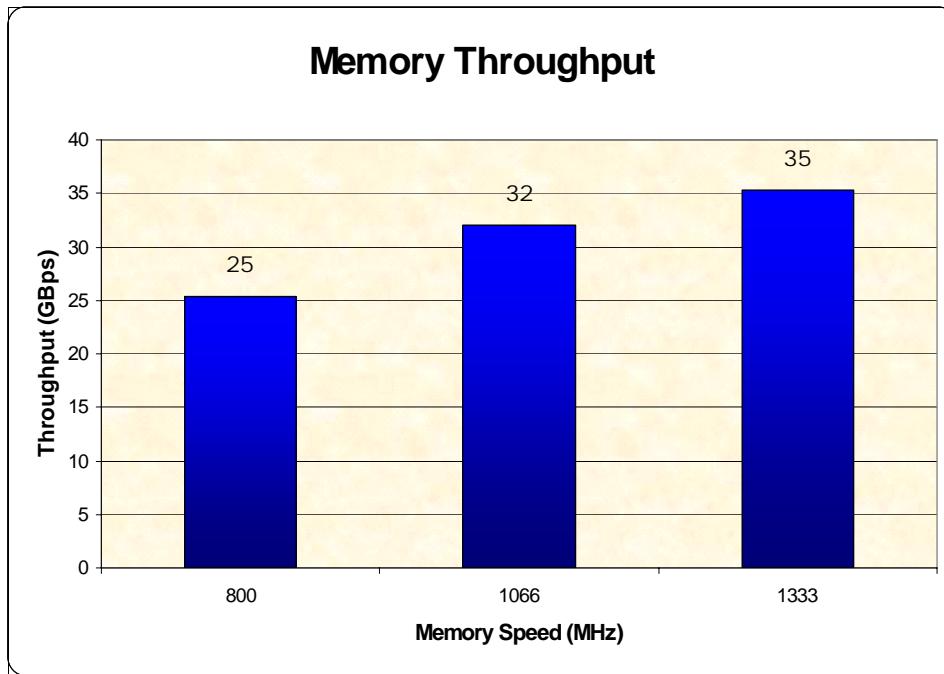


Figure 10-21 Memory throughput using STREAMS Triad

### 3.1.5 Application performance

In this section, we discuss the impact of memory speed on the performance of three commonly used benchmarks: SPECint 2006\_rate, SPECfp 2006\_rate, and SPECjbb 2005.

- ▶ SPECint2006\_rate is typically used as an indicator of performance for commercial applications. It tends to be more sensitive to processor frequency and less to memory bandwidth.

There are very few components in SPECint2006\_rate that are memory bandwidth-intensive and so the performance gain with memory speed improvements is the least for this workload. In fact, most of the difference observed is due to one of the sub-benchmarks that shows a high sensitivity to memory frequency. There is an 8% improvement going from 800MHz to 1333MHz. The improvement in memory bandwidth is almost 40%.

- ▶ SPECfp\_rate is used as an indicator for high-performance computing (HPC) workloads. It tends to be memory bandwidth-intensive and should reveal significant improvements for this workload as memory frequency increases.

As expected, a number of sub-benchmarks demonstrate improvements as high as the difference in memory bandwidth. As shown in Figure 10-22 on page 224, there is a 13% gain going from 800MHz to 1066MHz and another

6% improvement with 1333MHz. SPECfp\_rate captures almost 50% of the memory bandwidth improvement.

- SPECjbb2005 is a workload that does not stress memory but keeps the data bus moderately utilized. This workload provides a middle ground and the performance gains reflect that trend. As shown in Figure 8, there is an 8% gain from 800MHz to 1066MHz and another 2% upside with 1333MHz.

In each case, the benchmark scores are relative to the score at 800MHz, as shown in Figure 10-22.

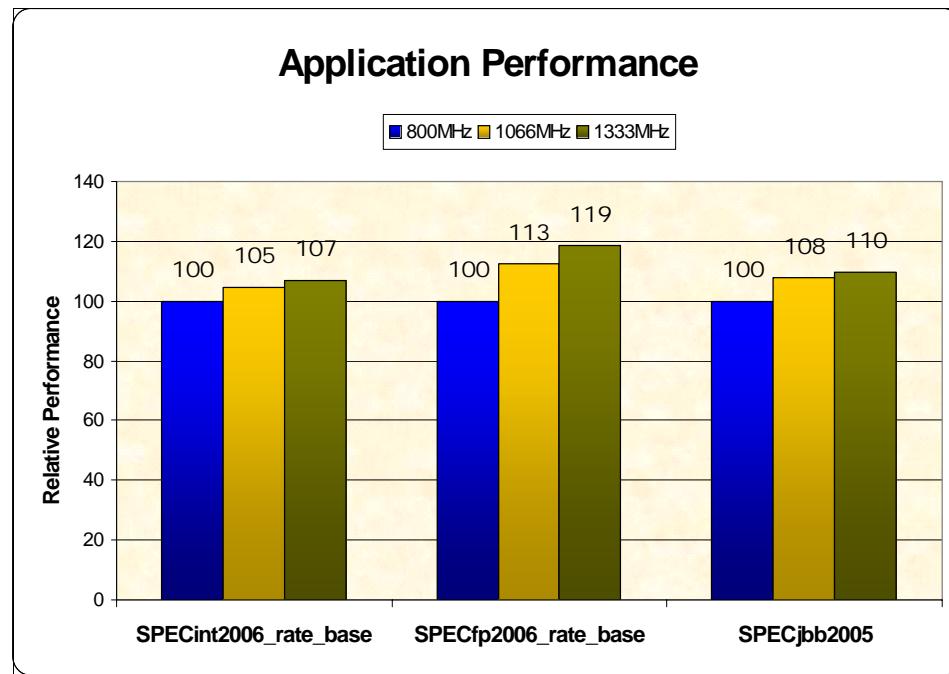


Figure 10-22 Application performance as a function of memory speed

#### 10.9.4 Memory Interleaving

Memory interleaving refers to how physical memory is interleaved across the physical DIMMs. A balanced system provides the best interleaving. A Xeon 5500 series processor-based system is balanced when all memory channels on a socket have the same amount of memory. The simplest way to enforce optimal interleaving is by populating six identical DIMMs at 1333MHz, 12 identical DIMMs at 1066MHz and 18 identical DIMMs (where supported by platform) at 800MHz.

## **HS22 Blade Server**

For HS22, which has a balanced DIMM layout, it is easy to balance the system for all three memory frequencies. The recommended DIMM population is shown in Table 10-9, assuming DIMMs with identical capacities.

*Table 10-9 Memory configurations to produce balanced performance in HS22*

Desired memory speed	DIMMs per channel	DIMM slots to populate
1333 MHz	1	2, 4, 6, 8, 10, and 12
1066 MHz	2	All slots
800 MHz	2	All slots; clock memory speed to 800MHz in BIOS

## **x3650 M2/x3550 M2/dx360 M2 rack systems**

For systems with 16 DIMM slots, care needs to be taken when populating the slots, especially when configuring for large DIMM counts at 800MHz.

When configuring for 800 MHz, with large DIMM counts, it would be a common error to populate all 16 DIMM slots with identical DIMMs. However, such a configuration leads to an unbalanced system where two memory channels have less memory capacity than the other four. (In other words, two channels with, for example, 3 x 4GB DIMMs and one channel with 2 x 4GB DIMMs.) This leads to lessened performance.

Figure 10-23 on page 226 shows the impact of reduced interleaving. The first configuration is a balanced baseline configuration where the memory is down-clocked to 800MHz in BIOS.

The second configuration populates four channels with 50% more memory than two other channels causing an unbalanced configuration.

The third configuration balances the memory on all channels by populating the channels with fewer DIMM slots with a DIMM that is double the capacity of others. (For example, two channels with 3 x 4GB DIMMs and one channel with 1 x 4GB and 1 x 8GB DIMMs.) This ensures that all channels have the same capacity.

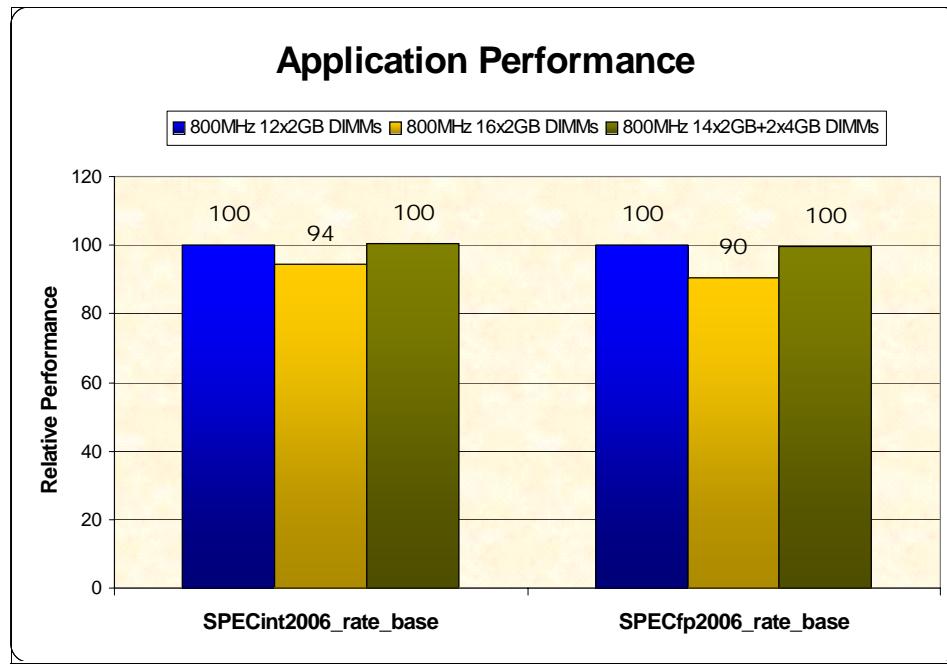


Figure 10-23 Impact of unbalanced memory configuration

As Figure 10-23 shows, the first and third balanced configurations significantly outperform the unbalanced configuration. Depending on the memory footprint of the application and memory access pattern, the impact could be higher or lower than the two applications cited in the figure.

The recommended DIMM population is shown in Table 10-10.

Table 10-10 Memory configs to produce balanced performance in System x servers

Desired memory speed	DIMMs per channel	DIMM slots to populate
1333 MHz	1	3, 6, 8, 11, 14, 16
1066 MHz	2	2, 3, 5, 6, 7, 8, 10, 11, 13, 14, 15, 16
800 MHz	2	2, 3, 5, 6, 7, 8, 10, 11, 13, 14, 15, 16 and clock memory down to 800MHz in BIOS
800MHz	>2	1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15 with DIMMs of size 'x' 8 and 16 with DIMMs of size '2x'

### 10.9.5 Memory ranks

A *memory rank* is simply a segment of memory that is addressed by a specific address bit. DIMMs typically have 1, 2, or 4 memory ranks, as indicated by their size designation.

- ▶ A typical memory DIMM description: 2GB 4R x8 DIMM
- ▶ The 4R designator is the rank count for this particular DIMM (R for rank = 4)
- ▶ The x8 designator is the data width of the rank

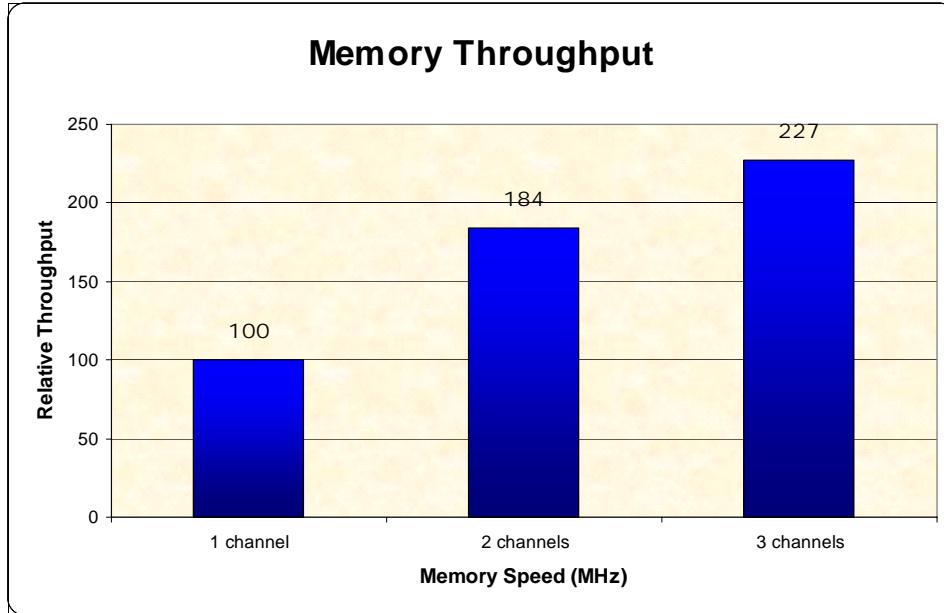
It is important to ensure that DIMMs with the appropriate number of ranks are populated in each channel for optimal performance. Whenever possible, it is recommended to use dual-rank DIMMs in the system. Dual-rank DIMMs offer better interleaving and hence better performance than single-rank DIMMs.

For instance, a system populated with 6 x 2GB dual-rank DIMMs outperforms a system populated with 6 x 2GB single-rank DIMMs by 7% for SPECjbb2005. Dual-rank DIMMs are also better than quad-rank DIMMs because quad-rank DIMMs will cause the memory speed to be down-clocked.

Another important guideline is to populate equivalent ranks per channel. For instance, mixing single-rank and dual-rank DIMMs in a channel should be avoided.

### 10.9.6 Memory population across memory channels

It is important to ensure that all three memory channels in each processor are populated. The relative memory bandwidth is shown in Figure 10-24 on page 228, which illustrates the loss of memory bandwidth as the number of channels populated decreases. This is because the bandwidth of all the memory channels is utilized to support the capability of the processor. So, as the channels are decreased, the burden to support the requisite bandwidth is increased on the remaining channels, causing them to become a bottleneck.



*Figure 10-24 The effect of populating different number of channels*

### 10.9.7 Memory population across processor sockets

Because the Xeon 5500 series uses NUMA architecture, it is important to ensure that both memory controllers in the system are utilized by providing both processors with memory. If only one processor is installed, then only the associated DIMM slots can be used. Adding a second processor not only doubles the amount of memory available for use, but also doubles the number of memory controllers, thus doubling the system memory bandwidth. It is also optimal to populate memory for both processors in an identical fashion to provide a balanced system.

Using Figure 10-25 on page 229 as an example, Processor 0 has DIMMs populated but no DIMMs are populated for Processor 1. In this case, Processor 0 will have access to low latency local memory and high memory bandwidth. However, Processor 1 has access only to remote or “far” memory. So, threads executing on Processor 1 will have a long latency to access memory as compared to threads on Processor 0.

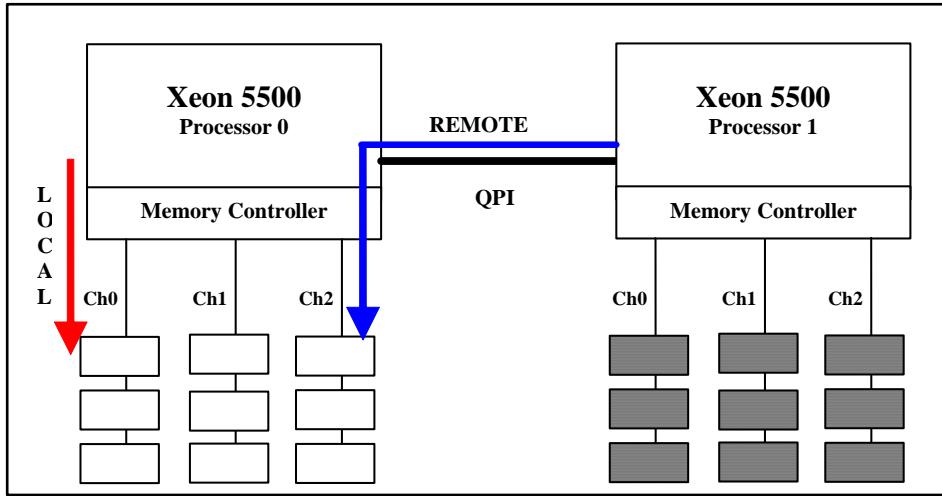


Figure 10-25 Diagram showing local and remote memory access

This is due to the latency penalty incurred to traverse the QPI links to access the data on the remote memory controller. The latency to access remote memory is almost 75% higher than local memory access. The bandwidth to remote memory is also limited by the capability of the QPI links. So, the goal should be to always populate both processors with memory.

### 10.9.8 Best practices

In this section, we recapture the various rules to be followed for optimal memory configuration on the Xeon 5500 based platforms.

#### Maximum performance

Follow these rules for peak performance:

- ▶ Always populate both processors with equal amounts of memory to ensure a balanced NUMA system.
- ▶ Always populate all three memory channels on each processor with equal memory capacity.
- ▶ Ensure an even number of ranks are populated per channel.
- ▶ Use dual-rank DIMMs whenever appropriate.
- ▶ For optimal 1333MHz performance, populate six dual-rank DIMMs (three per processor).
- ▶ For optimal 1066MHz performance, populate 12 dual-rank DIMMs (six per processor).

- ▶ For optimal 800MHz performance with high DIMM counts:
  - On 12 DIMM platforms, populate 12 dual-rank or quad-rank DIMMs (6 per processor).
  - On 16 DIMM platforms:
    - Populate 12 dual-rank or quad-rank DIMMs (6 per processor).
    - Populate 14 dual-rank DIMMs of one size and 2 dual-rank DIMMs of double the size as described in the interleaving section.
- ▶ With the above rules, it is not possible to have a performance-optimized system with 4 GB, 8 GB, 16 GB, or 128 GB. With three memory channels and interleaving rules, customers need to configure systems with 6 GB, 12 GB, 18 GB, 24 GB, 48 GB, 72 GB, 96 GB, and so on for optimized performance.

## Other considerations

Other aspects to consider regarding performance include the following:

- ▶ Plugging order
 

Take care to populate empty DIMM sockets in the specific order for each platform when adding DIMMs to Xeon 5500 series platforms. The DIMM socket farthest away from its associated processor, per memory channel, is always plugged first. Consult the documentation with your specific system for details.
- ▶ Power guidelines
 

This document is focused on maximum performance configuration for Xeon 5500 series processor-based systems. Here are a few power guidelines for consideration:

  - Fewer larger DIMMs (for example 6 x 4GB DIMMs versus 12 x 2GB DIMMs will generally have lower power requirements.
  - x8 DIMMs (x8 data width of rank) will generally draw less power than equivalently sized x4 DIMMs.
  - Consider BIOS configuration settings.
- ▶ Reliability
 

Here are two reliability guidelines for consideration:

  - Using fewer, larger DIMMs (for example 6 x 4 GB DIMMs vs. 12 x 2GB DIMMs is generally more reliable.
  - Xeon 5500 series memory controllers support IBM Chipkill memory protection technology with x4 DIMMs (x4 data width of rank), but not with x8 DIMMs.

- ▶ BIOS configuration settings

There are a number of BIOS configuration settings on servers using the Xeon 5500 series processors that can also affect memory performance or benchmark results. For example, most platforms allow the option of decreasing the memory clock speed below the supported maximum. This may be useful for power savings but, obviously, decreases memory performance.

Meanwhile, options like Hyper-Threading Technology (formerly known as Simultaneous Multi-Threading) and Turbo Boost Technology can also significantly affect benchmark results. Specific memory configuration settings important to performance are listed in Table 10-11.

*Table 10-11 BIOS settings*

BIOS options	Maximum performance setting
Memory Speed	Auto
Memory Channel Mode	Independent
Socket Interleaving	NUMA
Patrol Scrubbing	Off
Demand Scrubbing	On
Thermal Mode	Performance

## 10.10 eX4 architecture servers

The eX4Architecture servers are the fourth-generation Enterprise X-Architecture servers from IBM. There are two servers based on the eX4 Architecture, the System x3850 M2 and the x3950 M2. Many unique memory options and features are available with the eX4 chipset. However, in this section, we focus on the performance aspects only.

Although all servers utilize some amount of main memory and L1, L2, and sometimes L3 caches, the x3850M2 and x3950M2 systems also implement an additional level of memory called the XceL4v Dynamic Cache. This memory space is referred to as an “L4 cache” because it exists between the processor and memory subsystems.

This cache functions as part of the snoop filter of the eX4 chipset technology, and in this generation it is actually carved out of a portion of the system’s main memory. This cache allows minimization of snoop requests to peer-level

processors within a server node, as well as between nodes when x3950 M2 systems are attached together via the scalability ports. The end result enables the x3850 M2 and x3950 M2 to achieve efficient performance scaling for many applications.

The eX4 Architecture servers implement memory using one to four memory cards, each of which holds up to 8 DIMMs. The servers have one or two memory cards installed as standard (model-dependent).

The x3850 M2 and x3950 M2 systems uses ECC DDR2 DIMMs meeting the PC2-5300 standard, and must be populated according to the following rules:

- ▶ A minimum of one memory card containing two DIMMs is required for the server to operate.
- ▶ Each memory card contains two memory channels that each must be populated with DIMMs of identical part number.
- ▶ The installation sequence for DIMMs on the memory cards is 1&5, 2&4, 3&6, and 4&8.
- ▶ For performance-optimized configurations, it is highly recommended that you use two or four memory cards. Best performance is achieved when the DIMM pairs are spread evenly across the memory boards. For example, if 16 DIMMs are to be used with 4 memory cards, you would install 4 DIMMs per card in slots 1, 5, 2, and 6.

**Tip:** The memory controller is best optimized when 4 memory cards are used. For small memory configurations, 2 memory cards can be used at a small loss in performance. Configurations using 1 or 3 memory cards cannot exploit all the memory controller's optimizations, and therefore are *not* recommended in performance-sensitive configurations.

You can find a more detailed description and the exact sequence for installation in the User's Guide for each server.

## Memory configuration in BIOS

Depending on your needs, you can configure the x3850 M2 and x3950 M2 memory in three different ways:

- ▶ Full Array Memory Mirroring (FAMM)
- ▶ Hot Add Memory (HAM)
- ▶ High Performance Memory Array (HPMA)

The Hot Add Memory setting allows most DIMM types to be added to the server without turning off the server. Because this feature requires that only two memory cards be used until the additional memory is needed, it is typically not utilized in performance-sensitive configurations. Many other caveats also exist; refer to your server's installation guide for additional details about this feature.

**Note:** Best memory performance is achieved in the default setting of High Performance Memory Array (HPMA).

Note that Redundant Bit Steering is now enabled within HPMA mode, and therefore no longer needs its own, distinct Memory Configuration BIOS setting.

### **Memory ProteXion: redundant bit steering**

Redundant bit steering (RBS) is the technical term for Memory ProteXion, and is sometimes also known as *memory sparing*. When a single bit in a memory DIMM fails, the function known as redundant bit steering moves the affected bit to an unused bit in the memory array automatically. This removes the need to perform the ECC correction and thereby returns the memory subsystem to peak performance. The number of RBS actions that can be performed depends on the type of DIMMs installed in the server:

- ▶ A pair of single-rank DIMMs can perform one RBS action. Single ranked DIMMs are typically the smaller, 1 GB DIMMs. A pair of single-ranked DIMMs has a single *chip select group* (CSG).
- ▶ A pair of double-ranked DIMMs can perform two RBS actions. Double-ranked DIMMs comprise most of the common memory options available. A pair of double-ranked DIMMs has two chip select groups.

Memory errors are handled as follows:

- ▶ If a single-bit error occurs in a CSG, RBS is used to correct the error.
- ▶ If a second single-bit error occurs in the same CSG, the ECC circuitry is used to correct the error.

So, for example, if an x3850 M2 is configured with 32x dual-rank 2 GB DIMMs, and each dual-ranked DIMM pair corresponds to 2 CSGs, the server has a total of 32 CSGs. This means that the server can survive up to 64 single-bit memory failures: two RBS per pair of dual-rank DIMMs plus one ECC correction per DIMM.

## 10.11 IBM Xcelerated Memory Technology

The x3755 memory subsystem consists of two sets of 4 DIMMs in a daisy chain interconnect, see Figure 10-26.

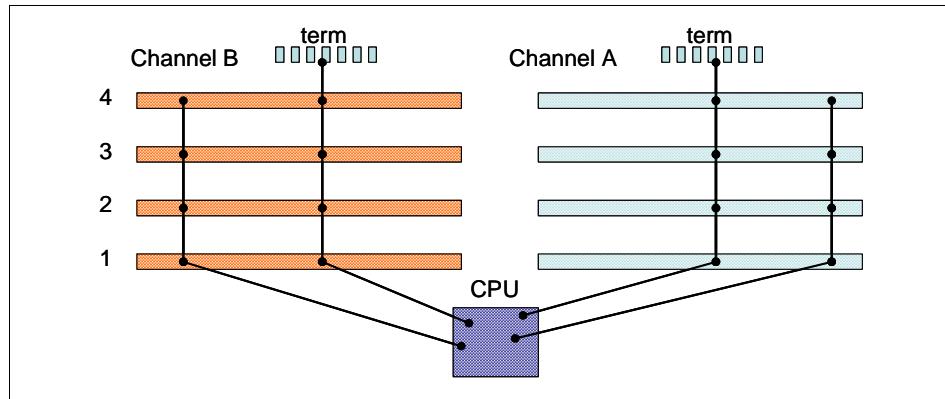


Figure 10-26 x3755 memory subsystem

Normally, if a read or write signal is sent down the memory bus to DIMM socket 4, the DIMM furthest away from the CPU. Then, due to the design of the bus, this signal will be reflected by each DIMM along the bus. This creates additional signals that cause noise and can result in incorrect data reads and writes, which in turn could cause the system to hang.

The AMD design specifications for their second-generation server processors (82xx and 22xx series processors) suggest that if more than 2 DIMMs are added to a memory bus, then the system should lower the memory bus speed from 667 MHz to 533 MHz to minimize the effect of the electrical noise.

IBM developers, however, have found that if a circuit is added to the bus to counteract the noise, we can maintain the timing and electrical integrity of the signal. This in turn allows the x3755 to keep the bus speed at a full 667 MHz for a full population of eight DIMMs on each CPU/memory card. This capability, known as Xcelerated Memory Technology, provides a 25% memory bandwidth advantage over competitor offerings.

## 10.12 Memory rules of thumb

The rules for memory capacity measurement when upgrading servers that are performing well are straightforward. Usually, the quantity of memory for

replacement servers is kept constant or somewhat increased if the number of users and applications does not change. However, this is not always the case.

Most of the memory is used typically for file or data cache for the operating system and applications. The operating system requirement for 256 to 512MB can usually be ignored, assuming this is small fraction of the total server memory.

For most environments, the proper approach for memory sizing is to proportionally scale the amount of memory required for the current number of users based on the expected increase of the number of users. For example, a server with 150 users and 8 GB of memory would need 16 GB to support 300 users. Doubling the number of users requires doubling the amount of server memory. To improve the accuracy of memory requirements, the memory usage of the server that is being replaced must be monitored.

There is no guarantee, however, that an existing server always has optimal memory utilization. For Windows environments, you should monitor memory allocation periodically in the Task Manager to determine the amount of total memory that is installed and the average amount of available memory. Total memory minus available memory equals the amount of memory the server is actually using: the *working set*. Because memory utilization is dynamic, it is best to monitor memory utilization over an extended period of time to arrive at an accurate representation of the memory working set.

A useful rule of thumb to determine the amount of memory that is needed to support twice the number of users is to simply double the peak working set size, and then add 30% as a buffer for growth activity.

Servers should be configured so that the average memory utilization does not exceed 70% of installed memory. Generally, 30% is enough extra memory so that the server will not start paging memory to disk during periods of peak activity. In any event, when you spot excessive memory utilization and the system starts to page, the best fix is to add memory.

The memory rules of thumb are as follows.

- ▶ In general, servers should *never* regularly page memory to disk (unless the application is performing memory mapped file I/O, which is discussed further in 20.3.3, “Memory subsystem” on page 681). Common application environments that use memory mapped files are Lotus Domino® and 32-bit versions of SAP. For details about memory mapped I/O, see:

[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dng\\_enlib/html/msdn\\_manamemo.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dng_enlib/html/msdn_manamemo.asp)

If an application is paging and the maximum amount of memory supported by that application has not been installed, then adding memory is likely to

significantly reduce paging. Unfortunately, some applications will continue to page even after the maximum amount of memory is installed. In this case, the only choice is to optimize the paging device by using a high-speed disk array.

- ▶ Average memory utilization should not exceed 70% of available memory.  
If time is short, simply determine the amount of installed memory on the server being replaced and scale future memory requirements based upon the expected increase in the user community. Memory is relatively inexpensive in comparison to the effort required to accurately predict the exact amount required.

Performance improvements from adding memory can vary greatly because the improvement depends on so many factors, such as the speed of the disk subsystem, the amount of memory the application requires, the speed of your memory subsystem, the speed of the processors, and so forth.

Performance gains from memory can fall into the following categories:

- ▶ Memory added to eliminate paging  
In this case, gains from adding this memory will be the largest, as memory operates many orders of magnitude faster than disk paging can. If significant paging is occurring in the application, increasing memory to eliminate paging can easily multiply performance levels many times over.
- ▶ Memory added to allow caching of the most frequently used data  
Many application workloads have an ideal balanced memory size that allows them to keep the most utilized data objects in memory, while not over-spending on the server's memory subsystem. This is commonly from 2 to 4 GB per processor core, though it could easily be more or less depending on the application. Doubling memory size to get into this ideal range can achieve 25% or higher gains in some environments.
- ▶ Memory added to allow maximum performance  
After the most frequent data objects are cached, the gains from increasing memory further are significantly reduced. Above this level, gains of ~10% for each doubling of memory are common for enterprise environments utilizing very large data sets.



# Disk subsystem

Ultimately, all data must be retrieved from and stored to disk. Disk accesses are usually measured in milliseconds. Memory and PCI bus operations are measured in nanoseconds or microseconds. Disk operations are typically thousands of times slower than PCI transfers or memory accesses. For this reason, the disk subsystem can easily become a major bottleneck for any server configuration.

Disk subsystems are also important because the physical orientation of data stored on disk has a dramatic influence on overall server performance. A detailed understanding of disk subsystem operation is critical for effectively solving many server performance bottlenecks.

This chapter discusses the following topics:

- ▶ 11.1, “Introduction to disk subsystems” on page 238
- ▶ 11.2, “Disk array controller operation” on page 240
- ▶ 11.3, “Direct-attached storage” on page 241
- ▶ 11.4, “Remote storage” on page 250
- ▶ 11.5, “RAID summary” on page 257
- ▶ 11.6, “Factors that affect disk performance” on page 267
- ▶ 11.7, “Disk subsystem rules of thumb” on page 291

## 11.1 Introduction to disk subsystems

A typical *disk subsystem* consists of the physical hard disk and the disk controller. A disk is made up of multiple platters that are coated with a magnetic material to store data. The entire *platter assembly*, which is mounted on a spindle, revolves around the central *axis*. A *head assembly* mounted on an *arm* moves to and from (linear motion) to read the data that is stored on the magnetic coating of the platter.

The linear movement of the head is referred to as the *seek*. The time it takes to move to the exact track where the data is stored is called *seek time*. The rotational movement of the platter to the correct sector to present the data under the head is called *latency*. The ability of the disk to transfer the requested data is called the *data transfer rate*. In measurement terms, low latency figures are more desirable than high latency figures. With throughput, it is the other way around: the higher the throughput, the better.

The most widely used drive technology in servers today is SerialAttached SCSI (SAS) with SATA and Fibre Channel for some applications. As Table 11-1 shows, however, it is not the only standard.

Table 11-1 Storage standards

Storage technology	Direct attach or remote storage	Description
Serial Attached SCSI (SAS)	Direct attach or Remote	SAS is the serial evolution of the SCSI. As the name suggests, it uses serial instead of parallel technology, resulting in faster bus speeds and longer cable length.
SCSI	Direct attach	Probably the most common storage technology in older servers. SCSI has reached the throughput limits of its parallel interconnect technology.
Serial Advanced Technology Attachment (SATA)	Direct attach	SATA is set to replace the old Parallel ATA technology. It is used for optical devices and is found on low-end servers.
Enhanced Integrated Drive Electronics (EIDE)	Direct attach	EIDE uses Parallel ATA technology. Installed in servers to control peripherals such as CD-ROMs and DVDs.
iSCSI (Internet SCSI)	Remote	SCSI encapsulated in TCP/IP packets to enable connections between servers and remote storage over the existing Ethernet network. Can have high latency on 1 Gbps Ethernet networks but significant improvement when run over 10 Gb Ethernet.

Storage technology	Direct attach or remote storage	Description
Fibre Channel (FC)	Remote	Like iSCSI, a method of remote-attaching storage to servers, but it does not use the TCP/IP protocol. Requires a dedicated fibre storage network. Has high throughput and low latency.
Serial Storage Architecture (SSA)	Remote	An alternative to Fibre Channel that provides high capacity storage at remote distances to the server.

Note that these technologies are divided into two groups, as discussed in 11.3, “Direct-attached storage” on page 241 and 11.4, “Remote storage” on page 250.

Both EIDE and SCSI are old technologies using parallel cables to connect the host adapter to the devices. A side effect of increasing the bus speed on parallel cables is also an increase in electromagnetic radiation or “noise” that is emitted from the wires. Because there are physically more wires on parallel cables, there is more noise, and this noise impacts data transmission.

This noise means that cable length and bus speeds have been restricted. Developments such as Low Voltage Differential (LVD) SCSI have been introduced to help overcome some of these restrictions, but even these have reached their limits with parallel technology.

There is now a shift away from parallel to serial technology. With serial technology, far fewer wires are needed, typically one pair for transmit and one for receive. Cable pairs are used to enable differential signals to be sent. Serial technology has resulted in smaller and thinner cables with increased bus speeds and longer cable lengths.

**Note:** When describing throughput of a particular storage system, it is common practice to use megabytes per second (MBps) for parallel devices and megabits per second (Mbps) for serial devices. Not everyone adheres to this convention, however.

To convert Mbps to MBps, divide by 10 (8 bits per byte plus 2 for a typical overhead). Therefore, 1 Gbps is roughly equal to 100 MBps.

## 11.2 Disk array controller operation

The disk controller is tasked with providing the interface between the server and the disks themselves. The following sequence outlines the fundamental operations that occur when a disk-read operation is performed:

1. The server operating system generates a disk I/O read operation by building an I/O control block command in memory. The I/O control block includes the read command, a disk address called a Logical Block Address (LBA), a block count or length, and the main memory address where the read data from disk is to be placed (destination address).
2. The operating system generates an interrupt to tell the disk array controller that it has an I/O operation to perform. This interrupt initiates execution of the disk device driver. The disk device driver (executing on the server's CPU) addresses the disk array controller and sends it the address of the I/O control block and a command instructing the disk array controller to fetch the I/O control block from memory.
3. The disk array controller copies the I/O control block from server memory into its local adapter memory. The onboard microprocessor executes instructions to decode the I/O control block command, to allocate buffer space in adapter memory to temporarily store the read data, and to program the SAS controller chip to initiate access to the SAS or SATA disks including the read data. The SAS controller chip is also given the address of the adapter memory buffer that will be used to temporarily store the read data.
4. At this point, the SAS controller sends the read command, along with the length of data to be read, to the target drives over dedicated paths. The SAS controller disconnects from the dedicated path and waits for the next request from the device driver.
5. The target drive begins processing the read command by initiating the disk head to move to the track including the read data (called a *seek operation*). The average seek time for current high-performance SAS drives is 3 to 5 milliseconds.

This time is derived by measuring the average amount of time it takes to position the head randomly from any track to any other track on the drive. The actual seek time for each operation can be significantly longer or shorter than the average. In practice, the seek time depends upon the distance the disk head must move to reach the track that includes the read data.
6. After the head reaches its destination track, the head begins to read a *servo track* (adjacent to the data track). A servo track is used to direct the disk head

to accurately follow the minute variations of the data signal encoded within the disk surface.

The disk head also begins to read the sector address information to identify the rotational position of the disk surface. This step allows the head to know when the requested data is about to rotate underneath the head. The time that elapses between the point when the head settles and is able to read the data track, and the point when the read data arrives, is called the *rotational latency*. Most disk drives have a specified average rotational latency, which is half the time it takes to traverse one complete revolution. It is half the rotational time because on average, the head will have to wait a half revolution to access any block of data on a track.

The average rotational latency of a 10,000 RPM drive is about 3 milliseconds. The average rotational latency of a 15,000 RPM drive is about 2 milliseconds. The actual latency depends upon the angular distance to the read data when the seek operation completes, and the head can begin reading the requested data track.

7. When the read data becomes available to the read head, it is transferred from the head into a buffer included on the disk drive. Usually, this buffer is large enough to include a complete track of data.
8. The target drive has the ability to re-establish a dedicated path between itself and the SAS controller. The target drive begins to send the read data into buffers on the adapter SAS controller chip. The adapter SAS controller chip then initiates a direct memory access (DMA) operation to move the read data into a cache buffer in array controller memory.
9. Using the destination address that was supplied in the original I/O control block as the target address, the disk array controller performs a PCI data transfer (memory write operation) of the read data into server main memory.
10. When the entire read transfer to server memory has completed, the disk array controller generates an interrupt to communicate completion status to the disk device driver. This interrupt informs the operating system that the read operation has completed.

### 11.3 Direct-attached storage

Table 11-1 on page 238 lists storage technologies and as you can see, SCSI, EIDE and SATA are listed as direct-attached storage (DAS) technologies. SAS is mostly used as DAS, but also has switching technology which allows creation of

a smaller SAN. DAS is connected physically to the server using cables, and is available for the server's exclusive use.

**Tip:** There is no technical reason why Fibre Channel disks could not be attached directly to a server. In practice, Fibre Channel disks are used in enclosures that typically hold many more disks than can be attached directly.

### 11.3.1 SAS

The SCSI parallel bus was the predominant server disk connection technology for a long time. However, SAS has recently replaced SCSI, and System x servers are now offering SAS as the standard storage architecture for both server-internal and in external disk enclosures.

SAS is the follow-on to SCSI and, in fact, retains the SCSI long-established software advantage by using underlying SCSI protocol. SAS also has support for SATA disk drives.

The first generation SAS was 3 Gbps technology with support for dual-port drives and wide ports, enabling full-duplex data transmission plus aggregated bandwidth.

The next generation 6 Gbps SAS was announced in September 2008 by the SCSI Trade Association (STA), a member-run industry association established to support and promote SCSI technology.

6 Gbps SAS has many enhancements beyond 3 Gbps SAS. It has more bandwidth per connection, greater scalability, and enhanced features. 3 Gbps SAS usage models will be preserved in 6 Gbps SAS along with the retention of 1.5 Gbps and 3 Gbps SAS/SATA compatibility. There are many other targeted improvements beyond first generation 3 Gbps SAS, assuring enterprise storage users that SAS technology will continue to meet their needs.

Almost all server disk controllers implement the SCSI communication between the SAS disk controller and disk drives. SCSI is an intelligent interface that allows simultaneous processing of multiple I/O requests. This is the single most important advantage of using SAS controllers on servers. Servers must process multiple independent requests for I/O. The ability of SCSI to concurrently process many different I/O operations makes it the optimal choice for servers.

SAS array controllers consist of the following primary components, as shown in Figure 11-1 on page 243:

- ▶ PCI bus interface/controller
- ▶ SAS controllers and SAS channels

- ▶ Microprocessor
- ▶ Memory (microprocessor code and cache buffers)
- ▶ Internal bus (connects PCI interface, microprocessor, and SAS controllers)

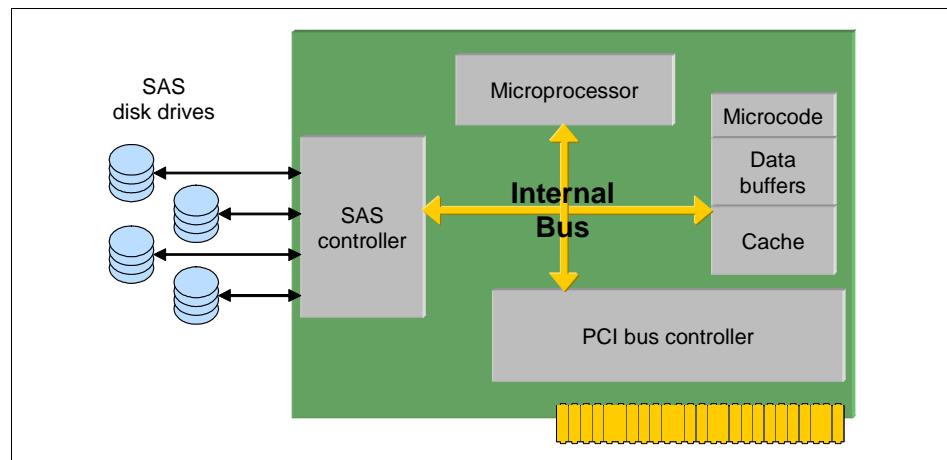


Figure 11-1 Architecture of a disk array controller

## SAS protocols and layers

SAS uses three protocols to define how transfers are handled between different devices:

- ▶ Serial SCSI Protocol (SSP) supports SAS hard drives and tape devices. SSP is full duplex, so frames can be sent in both directions simultaneously.
- ▶ SATA Tunneled Protocol (STP) supports SATA Hard drives. STP is half duplex, so frames can only be sent in one direction at a time.
- ▶ Management Protocol (SMP) supports SAS expanders. SMP is a simple protocol that allows initiators to view and configure details about expanders and devices.

The SAS protocol has the following layers:

- ▶ Application layer

The application layer receives commands from the device driver. It then sends the requests (command) to Transport Layer using the appropriate protocol (SSP, STP, or SMP).

- ▶ Transport layer

The transport layer is the interface between the application layer and the Port layer. It defines the frame formats. SAS formats are based on Fibre Channel.

An example of a common SSP frame format:

- Frame header: 24 bytes

- Information Unit: 0 to 1024 bytes
- Fill bytes: 0 to 2 bytes
- CRC: 4 bytes
- ▶ Port layer
 

The port layer creates command queues and requests available *phys*. A phy is the SAS terminology for a port. Ports are abstractions that include phys.
- ▶ Link layer
 

The link layer manages connections between a SAS initiator phy and a SAS target phy. It also arbitrates fairness and deadlocks, and closes connections.
- ▶ Phy layer
 

Out of band (OOB) signaling handles speed negotiation and 8b10b encoding. An OOB signal is a pattern of idle times and burst times. 8b10b coding converts 8 bit bytes into 10-bit data characters for transmission on a wire.
- ▶ Physical layer
 

These are the physical SAS and SATA cables and connectors.

## SAS and SATA speed negotiation

SAS and SATA speed negotiation occurs in the Phy layer. For SAS, to negotiate the connection speed, both devices start at the slowest rate and then increase the rate to find the fastest rate window.

The following steps describe the process:

1. Both devices send ALIGN(0)s.
2. If ALIGNED(0)s are received, send ALIGNED(1)s.
3. If ALIGNED(1)s are received, the current rate windows is successful; otherwise, the current rate windows is unsuccessful.
4. If the current rate window is successful, use a faster rate window and repeat steps 1 through 3 until the fastest supported rate is found.
5. Set the connection speed to the fastest supported rate.

For SATA, to negotiate the connection speed, both devices start at the fastest rate and then decrease the rate if necessary to find the fastest rate window. The following steps describe the process:

1. The SATA target device sends ALIGN primitives at the fastest supported rate.
2. The SATA target waits for the host to reply with ALIGNs.
3. If no reply is received, the SATA target sends ALIGN primitives at the next slower supported rate.

4. Steps 1 through 3 are repeated until the host replies with ALIGN(0)s, and the fastest supported rate is found.
5. Set the connection speed to the fastest supported rate.

### SAS inter-enclosure multi-lane cables

SAS inter-enclosure multi-lane cables are used to connect SAS and SAS RAID controllers to external EXP3000 enclosures. Each cable provides four SAS connections. At 3 Gbps per SAS lane, each cable can support up to 12 Gbps of throughput in each direction.

IBM SAS and SAS RAID controllers use two different types of SAS inter-enclosure multi-lane cables:

- ▶ Mini-SAS Molex or SFF-8088 (Figure 11-2) connects the SAS HBA controller and ServeRAID MR10M to an EXP3000 SAS drive enclosure. This cable also connects EXP 3000 enclosures to each other for cascading.

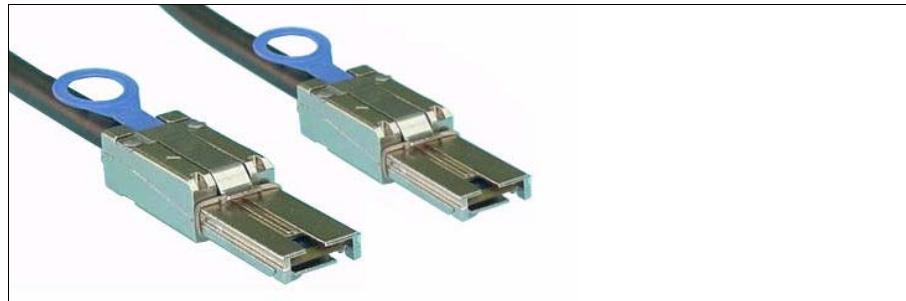


Figure 11-2 Mini-SAS Molex cable

- ▶ Infiniband-Mini-SAS Molex cable or SFF-8470 (Figure 11-3) connects MegaRAID 8480E to EXP3000 SAS drive enclosures.



Figure 11-3 Infiniband-Mini-SAS Molex cable

### 11.3.2 Serial ATA

Serial ATA (SATA) is the Serial Advanced Technology Attachment interface specification that offers increased data rate performance over its parallel equivalent, EIDE (now also known as Parallel ATA or PATA)

Developed by a group of leading technology vendors, SATA was designed to overcome the performance barriers of Parallel ATA technologies while maintaining their benefits and cost-efficiency. The SATA Working Group introduced the first SATA specification, SATA 1.0, in 2001, with plans for future versions.

**Tip:** The terms Parallel ATA and EIDE are interchangeable and relate to the same technology.

For more information about the SATA Working Group, visit:

<http://www.sata-io.org>

SATA was introduced in 2001 and initially offered a maximum data rate of 150 MBps. The SATA 1.0 specifications allowed for thinner, more flexible cables and lower pin counts, thus enabling easier, more flexible cable routing management and the use of smaller connectors than was possible with the existing EIDE/Parallel ATA technology.

In 2002, a second SATA specification was launched called SATA II, which provided enhancements to the previous specification, with a focus on the area of networked storage. New extensions included solutions for backplane interconnect for hot-swap drive racks, complete enclosure management, and performance enhancements.

**Note:** SATA II was the name of the organization that was formed to create SATA specifications. The group is now called the SATA International Organization (SATA-IO).

SATA II does not refer to 3 Gbps data transfer rate for SATA. The increased data transfer rate is one of several features included in SATA specifications subsequent to the SATA 1.0 specification.

A new specification for Serial ATA storage interface, SATA 3.0 will double the maximum transfer speed from 3 Gbps to 6 Gbps. SATA 6 Gbps technology will enable large amounts of data to be moved at even faster rates, which is a key benefit as users amass ever-increasing amounts of high-resolution photos, videos, music, and other multimedia files.

The characteristics of SATA make it an ideal solution for new applications such as low-cost secondary storage in networked storage environments.

SATA features include:

- ▶ Lower voltage

SATA operates at 250 millivolts, and Parallel ATA is based on 5 volt signaling. This low voltage results in lower power consumption, which means lower cooling needs, thus making SATA attractive for multi-drive RAID arrays.

- ▶ Data transfer rates

Parallel ATA is limited to data transfer rates of 133 MBps. The initial SATA implementation has a data transfer rate of 150 MBps. Although this transfer rate seems lower, the SATA road map calls for 300 MBps (3 Gbps), and then 600 MBps (6 Gbps) data transfer capability.

- ▶ Point-to-point connectivity

The master and slave shared connectivity approach is replaced with a point-to-point connection scheme supporting only one device per cable. This connectivity allows each drive to communicate directly with the system at any time. Because there is no sharing on the bus, performance scales linearly. Thus, adding a disk on a SATA system gives you the additional maximum throughput of the added disk, provided that the limitations of the storage controller are not exceeded.

- ▶ Serial transmission

Serial transmission is used in many recent technologies including Gigabit Ethernet, USB 2.0, IEEE 1394, and Fibre Channel. In fact, serial is used for most of the fastest data transfer technology and enables SATA to rival SCSI and Fibre Channel in speed.

- ▶ Cyclic redundancy checking (CRC)

This provides improved data protection and integrity over PATA and confers to SATA another feature already found in SCSI.

- ▶ Improved performance with hot-swappable drives

SATA features greater performance and hot-swappable drives. This feature enables you to swap out a drive without taking the system offline or rebooting. This characteristic makes SATA a viable option for enterprise solutions where system down time is usually not an option.

- ▶ Improved cabling and connector

A simplified cabling scheme offers a narrow serial cable with compact connectors for improved connectivity and ventilation, facilitating improved product design, and hardware assembly.

Practically, the connector size is reduced from 40 pins with Parallel ATA to 7 pins with SATA. Parallel ATA uses 16 separate wires to send 16-bits of data and thus must use a bulky flat cable, which is the cause of electromagnetic interference that compromises data integrity.

- ▶ Backward compatibility with older ATA storage devices

SATA is designed to be backward-compatible with previous Parallel ATA devices. To system software, SATA is not different from PATA.

## **SATA and ServeRAID**

SATA protocol support is included in SAS protocol. SAS controllers support SATA drives, and all IBM SAS ServeRAID cards support SATA devices. Due to differences in protocol, size and rotational speed, RAID arrays cannot mix SAS and SATA drives.

There can also be limitations in mixing SAS and SATA drives within some disk enclosures. This is due to different vibrations of the drives, and SAS drives' ability to operate at higher vibration level. SAS disks have stronger and more reliable mechanics, whereas SATA is more sensitive to these types of environmental conditions.

For more information about the ServeRAID family, see *ServeRAID Adapter Quick Reference*, TIPS0054, which is available online at:

<http://www.redbooks.ibm.com/abstracts/tips0054.html>

## **An alternative for enterprise storage**

With the availability of SATA and its established road map for future specification revisions and enhancements, customers have a viable alternative for many enterprise storage applications. As a result, SATA technology is now increasingly found in storage arrays and entry-level servers.

It is common to see mixing of SATA and SAS supported within many storage devices. IBM Systems Storage DS3000 series controllers, EXP3000 disk enclosures, and the BladeCenter S chassis support both SAS and SATA drives.

For future compatibility details please refer to IBM System Storage™ DS3000 Interoperability Matrix, available from:

<http://ibm.com/systems/storage/product/interop.html>

For more information about SATA, refer to *Introducing IBM TotalStorage FASST EXP100 with SATA Disks*, REDP-3794, which is available from:

<http://www.redbooks.ibm.com/abstracts/redp3794.html>

### 11.3.3 NL SAS

When comparing desktop class (SATA) and enterprise (SAS) HDDs, there are clear differences in design and usability. SATA has attractive characteristics, specifically large capacity and low price per GB. SAS has performance and reliability advantages for server workloads. Enterprise class SAS drives have faster motors that are designed to run longer; smaller discs platter design with full media certification; and better head stack design. These features make Enterprise class SAS drives more costly than SATA. There are also many advantages in SAS protocol when compared to SATA.

To provide the best of both Enterprise class SAS and SATA drives, Near-Line (NL) SAS HDDs were created. Basically, NL SAS drives have SATA HDD mechanical designs but with a SAS controller added to the drive. This provides many mission-critical features to existing SATA drives:

- ▶ Full Duplex (Bidirectional) I/O
- ▶ Enterprise Command Queuing
- ▶ Full SCSI command set
- ▶ Variable Sector Size
- ▶ 100% Phy-Compatible
- ▶ SAS level of ECC

NL SAS is designed for use in enterprise, mission-critical 24/7 environments. It still lacks the performance of a SAS mechanical design, but gives much lower price per GB. When mixed with SAS in remote storage systems, there is better choice for applications to choose the right characteristics for storage, either top IOPS or reasonable performance with more space for data.

### 11.3.4 Solid State Drive

A new category of hard disk drives has emerged based on flash memory. Instead of using rotating media like on conventional HDDs, Solid State Drive (SSD) uses flash memory with a special controller that allows the system to treat this flash memory like a regular disk drive.

There are several reasons to use SSD technology:

- ▶ Lower power usage
- ▶ Faster data access
- ▶ Higher reliability

Lower power usage is discussed in Chapter 5, “Energy efficiency” on page 49. The SSD drive can consume as little as 0.5W in idle mode and less than 2W under load, in comparison to more than 10W on conventional HDDs. This, combined with much higher performance, provides a new, high efficiency solution

for certain applications where high IOPS performance is required. The downside of such a disk subsystem is that SSD drives typically have a smaller capacity per disk, but some high performance applications only demand large numbers of HDDs to cover the performance requirements, and only utilize a small portion of the physical disk space anyway.

Because there is no delay due to rotational latency, read access is much faster on SSD than on mechanical drives. Although older SSD drives have historically had very high write latencies, newer generations of SSD have included technologies to increase write speeds such that the newer SSD drives will outperform SAS drives in random read/write workloads.

Using OLTP (67% Read, 33% Write, 100% Random) performance tests, SSD has as much as 4 times higher IOPS throughput than SAS HDD on 4k transfer. However, the percentage performance advantage decreases as transfer size increases and the impact of rotational latency is less of a problem, because the SSD advantage declines to 30% for 64 k transfers.

Reliability of the drives is also better as there are no moving parts in the device itself. Smart logic on the disk controller, called wear-leveling algorithms, are used to distribute write segments across all memory addresses on the device to avoid specific memory cells exceeding their usage limits, which could occur if the writes would consistently access the first memory blocks of the device.

**Note:** SSDs should primarily be used for workloads that are characterized by high random IOps such as a database or mail server, or low disk write latencies such as some Citrix environments. SSDs may not provide any significant gains over HDD drives for sequential access environments.

## 11.4 Remote storage

*Remote storage* refers to storage that is physically separate from the server and connected by fiber optics, a LAN infrastructure, or through SAS. Remote storage is often shared between multiple servers. SAS storage carries attributes of both direct attached and remote storage, for instance sharing storage between multiple servers and extending connection distance. In this section, we cover Fibre Channel and iSCSI, as well as the new IBM XIV® Storage System.

One point to remember is that Fibre Channel and iSCSI are used to transfer the data between the server and remote storage device. The remote storage device might actually be using Fibre Channel, SAS, or SATA disks.

### 11.4.1 Differences between SAN and NAS

Before describing the different remote disk technologies, it is worth discussing Storage Attached Network (SAN) and Network Attached Storage (NAS) and how they differ. Both of our remote storage topics, Fibre Channel and iSCSI, are forms of SANs. Although iSCSI works over the existing Ethernet network, it is not an NAS system.

**Tip:** To learn more about SAN and NAS implementations, refer to *IBM System Storage Solutions Handbook*, SG24-5250, which you can find online at:

<http://www.redbooks.ibm.com/abstracts/sg245250.html>

#### SAN

A SAN is a specialized, dedicated high-speed storage network. Servers, switches, hubs, and storage devices can attach to the SAN; it is sometimes called “the network behind the servers.” Like a LAN, a SAN allows any-to-any connection across the network, using interconnect elements such as routers, gateways, hubs, and switches. Fibre Channel is the de facto SAN networking architecture, although you can use other network standards.

Fibre Channel is a multi-layered network, based on a series of ANSI standards. These define characteristics and functions for moving data across the network. As with other networks, information is sent in structured packets or frames, and data is serialized before transmission. However, unlike other networks, the Fibre Channel architecture includes a significant amount of hardware processing. The maximum data rate currently supported is 8 Gbps or 800 MBps full duplex.

However, a SAN implementation does not come without a price. Because of the complexities involved, a SAN can be an expensive investment. Storage management becomes a consideration. A high level of skill is needed to maintain and manage a SAN. It is therefore worth investing a significant amount of time in planning the implementation of a SAN.

#### Designing a SAN

Designing and implementing a SAN requires a knowledge of the fundamental storage principles that are needed to create a storage subsystem that can handle the I/O requirements of an enterprise production environment. For a review of storage fundamentals see 11.6, “Factors that affect disk performance” on page 267.

You should also consider how multiple sets of users, applications, and servers accessing the storage pool will affect performance. Conceptually split the SAN into three different zones:

- ▶ Backend zone

The backend zone is defined as the hardware from the hard disk drives to the backend of the remote storage controllers. The backend zone must include the optimal drive technology, number of drives, and sufficient bandwidth capacity up to the remote storage to satisfy the I/O requests from all servers and applications that have access to a particular backend zone.

The potential for creating a bottleneck in a SAN is very high in the backend zone. One reason for this is because backend technology such as hard disk drives and disk drive enclosures are typically the last components to complete a technology jump such as 4 Gbit to 8 Gbit Fibre Channel.

Furthermore, it is very expensive and time-consuming to upgrade backend technology compared to frontend and middle zone technology. Therefore, the front and middle zones of a SAN might include 8 Gbit FC technology, while the backend remains at 4 Gbit FC technology.

Because SANs must accommodate the full spectrum of different workload characteristics, the potential for a bottleneck in streaming environments such as backups, restores, and table scans is high. At a minimum, users might not realize the full potential of SAN performance capabilities if one zone is populated with inferior technology.

**Important:** The hardware in the backend zone is critical to ensuring optimal performance of a SAN, because the backend zone is where data begins its journey up to the servers. If the drive technology is insufficient or if the bandwidth technology is inferior to the bandwidth technology in the middle and frontend zones, then the performance of the entire SAN could potentially be gated.

- ▶ Middle zone

The middle zone includes the hardware from the remote storage controllers up to the backend of any switches, hubs, or gateway hardware.

Sufficient bandwidth capacity must exist in the middle zone to allow for sustainable throughput coming from the backend zone en route to the frontend zone.

- ▶ Frontend zone

The frontend zone includes the hardware from the front of any switches, hubs, or gateways up to and including the host bus adapter (HBA), host channel adapter (HCA), or adapter used to feed the servers replies from I/O requests.

Again, sufficient bandwidth capacity must exist in the frontend zone to allow for sustainable throughput coming from the middle zone. For example, if four 8 Gbit connections exist in the middle zone, and there are only two 8 Gbit host connections in the frontend zone, then a bottleneck could easily develop in a streaming-intensive workload environment.

In addition to considering fundamental storage principles and the hardware in the three different zones, it is just as important to consider the load placed on the SAN. The aggregate load on the SAN must be balanced across hard drives, remote storage controllers, links, and switches up through each zone of the SAN. Unbalanced loads will cause portions of the SAN to be underutilized, and other portions of the SAN to be overutilized.

## NAS

Storage devices that optimize the concept of file sharing across the network have come to be known as NAS. NAS solutions use the mature TCP/IP network technology of the Ethernet LAN. Data is sent to and from NAS devices over the LAN using TCP/IP protocol. By making storage devices LAN addressable, the storage is freed from its direct attachment to a specific server, and any-to-any connectivity is facilitated using the LAN fabric.

In principle, any user running any operating system can access files on the remote storage device. This is done by means of a common network access protocol, for example, NFS for UNIX servers, and CIFS for Windows servers. In addition, a task, such as backup to tape, can be performed across the LAN, using software like Tivoli Storage Manager, enabling sharing of expensive hardware resources, such as automated tape libraries, between multiple servers.

A storage device cannot just attach to a LAN. It needs intelligence to manage the transfer and the organization of data on the device. The intelligence is provided by a dedicated server to which the common storage is attached. It is important to understand this concept. NAS comprises a server, an operating system, plus storage which is shared across the network by many other servers and clients. So an NAS is a *device*, rather than a network infrastructure, and shared storage is attached to the NAS server.

A key difference between an NAS disk device and Direct Attached Storage or other network storage solutions, such as SAN or iSCSI, is that all I/O operations use file-level I/O protocols. File I/O is a high-level type of request which, in essence, specifies only the file to be accessed but does not directly address the storage device. Directly addressing the storage device is done later by other operating system functions in the remote NAS appliance.

A file I/O specifies the file and also indicates an offset into the file. For instance, the I/O might specify “Go to byte ‘1000’ in the file (as though the file were a set of contiguous bytes), and read the next 256 bytes beginning at that position.”

Unlike block I/O, a file I/O request has no awareness of disk volume or disk sectors. Inside the NAS appliance, the operating system keeps tracks of where files are located on disk. The operating system issues a block I/O request to the disks to fulfill the file I/O read and write requests it receives.

In summary, the network access methods, NFS and CIFS, can only handle file I/O requests to the remote file system, which is located in the operating system of the NAS device. I/O requests are packaged by the initiator into TCP/IP protocols to move across the IP network. The remote NAS file system converts the request to block I/O and reads or writes the data to the NAS disk storage.

To return data to the requesting client application, the NAS appliance software repackages the data in TCP/IP protocols to move it back across the network. A database application that is accessing a remote file located on an NAS device, by default, is configured to run with file system I/O. It cannot use a *raw I/O* to achieve improved performance.

Because NAS devices attach to mature, standard LAN infrastructures and have standard LAN addresses, they are, typically, extremely easy to install, operate, and administer. This “plug and play” operation results in low risk, ease of use, and fewer operator errors, so it contributes to a lower cost of ownership.

### 11.4.2 Fibre Channel

Fibre Channel introduces different techniques to attach storage to servers. As a result, it has unique performance issues that affect the overall performance of a server. This section provides a brief introduction to the motivation behind Fibre Channel, explains how Fibre Channel affects server performance, and identifies important issues for configuring Fibre Channel for optimal performance.

Fibre Channel was designed to be a transport for both network traffic and an I/O channel for attaching storage. In fact, the Fibre Channel specification provides for many protocols such as 802.2, Internet Protocol (IP), and SCSI. Our discussion in this book is limited to its use for disk storage attachment.

Fibre Channel provides low latency and high throughput capabilities. As a result, Fibre Channel is the dominant I/O technology used to connect servers and high-speed storage. Fibre Channel addresses many of the shortcomings of direct-access storage with improvement in the following areas:

- ▶ Bandwidth
- ▶ Reliability

- ▶ Scalability
- ▶ Resource utilization/sharing

Fibre Channel provides the capability to use either a serial copper or fiber optic link to connect the server with storage devices. Fiber optic technology allows for storage to be located a maximum distance of up to 10 kilometers away from the attaching server.

A significant advantage of Fibre Channel is its ability to connect redundant paths between storage and one or more servers. Redundant Fibre Channel paths improve server availability because cable or connector failures do not cause server down time because storage can be accessed by a redundant path. In addition, both Fibre Channel and SAS throughput can scale by utilizing multiple channels or buses between the servers and storage.

In addition to a simpler cable scheme, Fibre Channel offers improved scalability due to several very flexible connection topologies. Basic point-to-point connections can be made between a server and storage devices providing a low-cost simple stand-alone connection.

Fibre Channel can also be used in both loop and switch topologies. These topologies increase server-to-storage connection flexibility. The Fibre Channel loop allows up to 127 devices to be configured to share the same Fibre Channel connection. A device can be a server, storage subsystem, drive enclosure, or disk. Fibre Channel switch topologies provide the most flexible configuration scheme by, theoretically, providing the connection of up to 16 million devices.

### 11.4.3 iSCSI

iSCSI is an industry standard that allows SCSI block I/O protocols (commands, sequences and attributes) to be sent over a network using the TCP/IP protocol. This is analogous to the way SCSI commands are already mapped to Fibre Channel, parallel SCSI, and SSA media (do not confuse this with the SCSI cabling transport mechanism; here we are addressing protocols).

iSCSI is a network transport protocol for SCSI that operates on top of TCP. It encapsulates SCSI protocols into a TCP/IP frame, so that storage controllers can be attached to IP networks.

Unlike Fibre Channel, iSCSI uses the existing Gigabit Ethernet LAN as a medium to transfer data from the iSCSI appliance, known as the *target*, to the file or application server. At the server end, either a software iSCSI driver or a dedicated iSCSI adapter can be used to encapsulate the iSCSI blocks. This is known as the *initiator*. If a software initiator is used, then the iSCSI traffic will be

transmitted through the existing network adapters. If a hardware initiator is installed, then it will need its own Ethernet network connection.

## Performance

So, what sort of performance can you expect from an iSCSI SAN? That depends on the disk subsystem speed and also on the number of links used to attach the storage server.

For a relative comparison we used existing IBM Systems Storage Controllers from the DS3000 family. Figure 11-4 shows the relative interface throughput comparisons for DS3200 SAS (2 ports), DS3300 iSCSI (4 ports of 1 Gbps Ethernet), and DS3400 (2 ports of 4 Gbps FC).

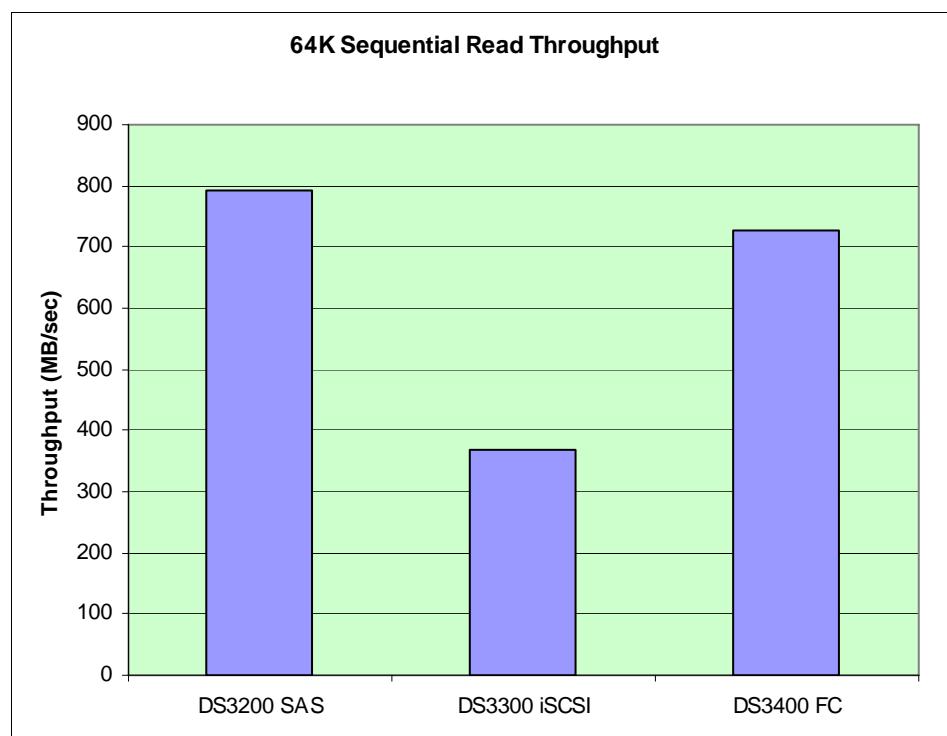


Figure 11-4 64 K Sequential Read Throughput comparison

Note that the DS3200 is limited in this case by the SAS controller on the storage controller, and not by the SAS link bandwidth. Further, actual sustainable performance for any given workload depends heavily on the characteristics of the workload, the latency of the interface, and on the technology and number of links used. For FC storage, the newer 8 Gbps technology could theoretically double the throughput shown here.

With 10 Gbps Ethernet, iSCSI will benefit significantly in throughput as well as latency, because this is a major step forward in Ethernet performance. This leads to new technologies like Converged Enhanced Ethernet (CEE) which is discussed in Chapter 12, “Network subsystem” on page 293.

## Security

Many IT managers would have serious reservations about running mission-critical corporate data on an IP network that is also handling other traffic. iSCSI introduces the possibility of an IP network SAN, which could be shared. To alleviate these worries, iSCSI can be encrypted using IPSEC.

For information about iSCSI from a networking point of view, see 12.4, “Internet SCSI (iSCSI)” on page 339.

### 11.4.4 IBM XIV Storage System

The recently offered IBM XIV Storage System is based on a revolutionary high-end disk storage architecture. It can attach to both Fibre Channel and iSCSI-capable hosts.

This storage system incorporates a variety of features designed to uniformly distribute data across key internal resources. This unique data distribution method fundamentally differentiates the XIV Storage System from conventional storage subsystems, thereby effecting numerous availability, performance, and management benefits across both physical and logical elements of the system.

The concept of parallelism pervades all aspects of the XIV Storage System architecture by means of a balanced, redundant data distribution scheme in conjunction with a pool of distributed (or grid) computing resources.

XIV architecture is discussed in the IBM Redbooks publication, *IBM XIV Storage System: Concepts, Architecture, and Usage*, SG24-7659. It covers architecture, implementation, and performance considerations of the new architecture. This book is available from:

<http://www.redbooks.ibm.com/abstracts/sg247659.html>

## 11.5 RAID summary

Most of us have heard of Redundant Array of Independent Disks (RAID) technology. Unfortunately, there is still significant confusion about the performance implications of each RAID strategy. This section presents a brief overview of RAID and the performance issues as they relate to commercial server environments.

RAID is a collection of techniques that treat multiple, inexpensive disk drives as a unit, with the object of improving performance and reliability. Table 11-2 lists the RAID levels offered by RAID controllers in IBM System x servers.

*Table 11-2 RAID summary*

RAID level	Fault tolerant?	Description
RAID-0	No	All data evenly distributed (striping) to all drives. See 11.5.1, “RAID-0” on page 258.
RAID-1	Yes	A mirrored copy of one drive to another drive (two disks). See 11.5.2, “RAID-1” on page 259.
RAID-1E	Yes	All data is mirrored (more than two disks). See 11.5.3, “RAID-1E” on page 260.
RAID-5	Yes	Distributed checksum. Both data and parity are striped across all drives. See 11.5.4, “RAID-5” on page 261.
RAID-5E RAID-5EE	Yes	Distributed checksum and hot spare. Data, parity and hot spare are striped across all drives. See 11.5.5, “RAID-5EE and RAID-5E” on page 262.
RAID-6	Yes	Distributed checksum. Both data and parity are striped across all drives - twice, to provide two-drive failure fault tolerance. See 11.5.6, “RAID-6” on page 265.
RAID-10	Yes	Striping (RAID-0) across multiple RAID-1 arrays. See 11.5.7, “RAID-10, RAID-50 and other composite levels” on page 266.
RAID-50	Yes	Striping (RAID-0) across multiple RAID-5 arrays. See 11.5.7, “RAID-10, RAID-50 and other composite levels” on page 266
RAID-60	Yes	Striping (RAID-0) across multiple RAID-6 arrays

In the following sections we discuss these levels in more detail.

### **11.5.1 RAID-0**

RAID-0 is a technique that stripes data evenly across all disk drives in the array. Strictly, it is not a RAID level, because no redundancy is provided. On average, accesses are random, thus keeping each drive equally busy. SCSI has the ability to process multiple, simultaneous I/O requests, and I/O performance is improved because all drives can contribute to system I/O throughput. Because RAID-0 has no fault tolerance, when a single drive fails, the entire array becomes unavailable.

RAID-0, as illustrated in Figure 11-5, offers the fastest performance of any RAID strategy for random commercial workloads. RAID-0 also has the lowest cost of implementation because redundant drives are not supported.

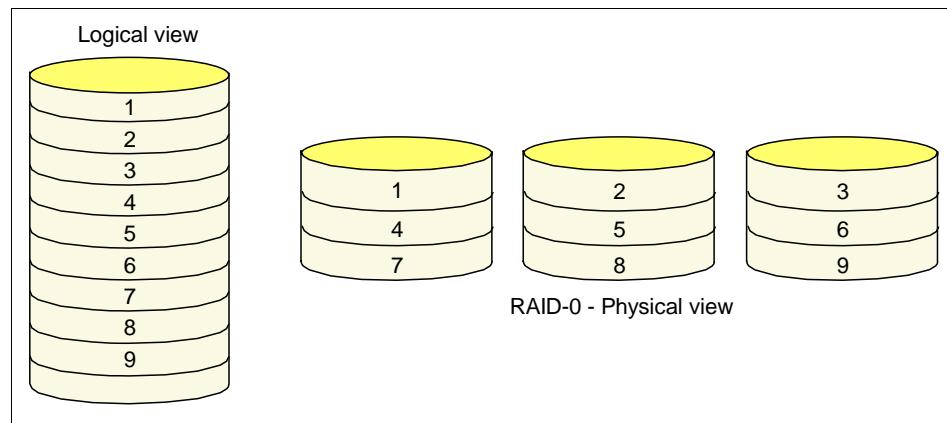


Figure 11-5 RAID-0: All data evenly distributed across all drives but no fault tolerance

### 11.5.2 RAID-1

RAID-1 provides fault tolerance by mirroring one drive to another drive. The mirror drive ensures access to data if a drive should fail. RAID-1 also has good I/O throughput performance compared to single-drive configurations because read operations can be performed on any data record on any drive contained within the array.

Most array controllers (including the ServeRAID family) do not attempt to optimize read latency by issuing the same read request to both drives in the mirrored pair. The drive in the pair that is least busy is issued the read command, leaving the other drive to perform another read operation. This technique ensures maximum read throughput.

Write performance is somewhat reduced because both drives in the mirrored pair must complete the write operation. For example, two physical write operations must occur for each write command generated by the operating system.

RAID-1, as illustrated in Figure 11-6 on page 260, offers significantly better I/O throughput performance than RAID-5. However, RAID-1 is somewhat slower than RAID-0.

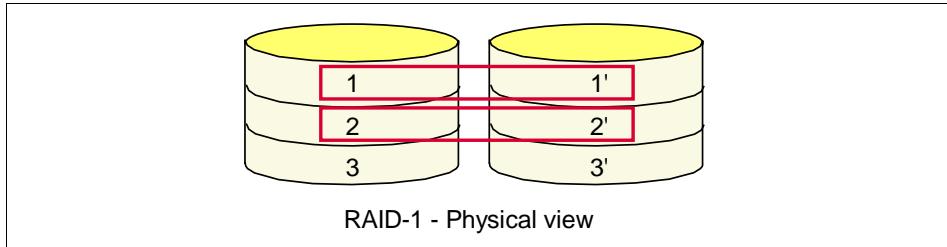


Figure 11-6 RAID-1: Fault-tolerant; a mirrored copy of one drive to another drive

### 11.5.3 RAID-1E

RAID-1 Enhanced, or more simply, RAID-1E, is only implemented by the IBM ServeRAID adapter and allows a RAID-1 array to consist of three or more disk drives. “Regular” RAID-1 consists of exactly two drives. RAID-1E is illustrated in Figure 11-7.

The data stripe is spread across all disks in the array to maximize the number of spindles that are involved in an I/O request to achieve maximum performance. RAID-1E is also called *mirrored stripe*, because a complete stripe of data is mirrored to another stripe within the set of disks. Like RAID-1, only half of the total disk space is usable; the other half is used by the mirror.

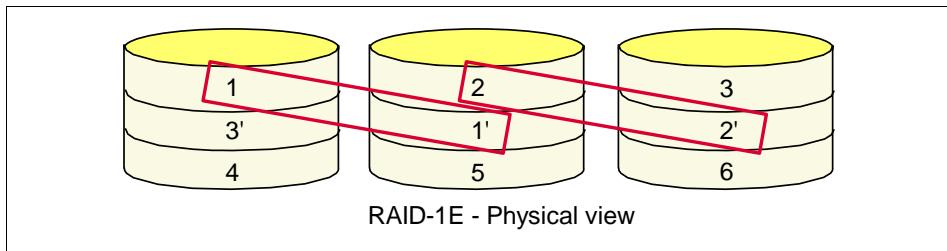


Figure 11-7 RAID-1E: Mirrored copies of each drive

Because you can have more than two drives (up to 16), RAID-1E will outperform RAID-1. The only situation where RAID-1 performs better than RAID-1E is in the reading of sequential data. The reason for this is because when a RAID-1E reads sequential data off a drive, the data is striped across multiple drives. RAID-1E interleaves data on different drives, so seek operations occur more frequently during sequential I/O. In RAID-1, data is not interleaved, so fewer seek operations occur for sequential I/O.

RAID-10 can also be used to increase the number of drives in a RAID-1 array. This technique consists of creating a RAID-0 array and mirroring it with another collection of similar-capacity drives. Thus, you can configure two sets of five

drives each in a RAID-0 configuration, and mirror the two sets of drives. This configuration would deliver the same performance for most commercial applications as a 10-drive RAID-1E configuration, but RAID-1E lacks one added benefit. Each of the RAID-0 arrays in the RAID-10 configuration can be contained in two different drive enclosures. Thus, if one drive enclosure fails because of a bad cable or power supply, the other mirror set can provide data access. With RAID-10, an entire set of drives (five, in this case) can fail and the server can still access the data.

#### 11.5.4 RAID-5

RAID-5, as illustrated in Figure 11-8, offers an optimal balance between price and performance for most commercial server workloads. RAID-5 provides single-drive fault tolerance by implementing a technique called *single equation single unknown*. This technique implies that if any single term in an equation is unknown, the equation can be solved to exactly one solution.

The RAID-5 controller calculates a *checksum* (parity stripe in Figure 11-8) using a logic function known as an exclusive-or (XOR) operation. The checksum is the XOR of all data elements in a row. The XOR result can be performed quickly by the RAID controller hardware, and is used to solve for the unknown data element.

In Figure 11-8, addition is used instead of XOR to illustrate the technique: stripe 1 + stripe 2 + stripe 3 = parity stripe 1-3. If drive one should fail, stripe 1 becomes unknown and the equation becomes X + stripe 2 + stripe 3 = parity stripe 1-3. The controller solves for X and returns stripe 1 as the result.

A significant benefit of RAID-5 is the low cost of implementation, especially for configurations requiring a large number of disk drives. To achieve fault tolerance, only one additional disk is required. The checksum information is evenly distributed over all drives, and checksum update operations are evenly balanced within the array.

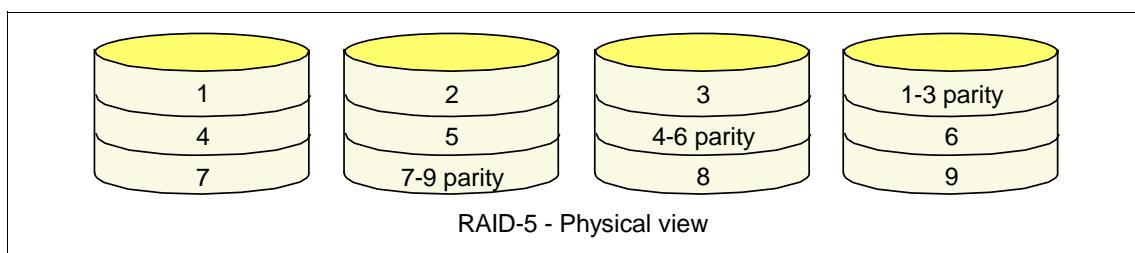


Figure 11-8 RAID-5: both data and parity are striped across all drives

However, RAID-5 yields lower I/O throughput than RAID-0 and RAID-1. This is due to the additional checksum calculation and write operations required. In

general, I/O throughput with RAID-5 is 30% to 50% lower than with RAID-1 (the actual result depends upon the percentage of write operations). A workload with a greater percentage of write requests generally has a lower RAID-5 throughput. RAID-5 provides I/O throughput performance similar to RAID-0 when the workload does not require write operations (read only).

### 11.5.5 RAID-5EE and RAID-5E

Although not available on the most current-generation products, RAID-5E and 5EE still ship with some older IBM products, so this section is left in for reference purposes.

IBM research invented RAID-5E, which is illustrated in Figure 11-9. RAID-5E distributes hot spare drive space over the n+1 drives that comprise a typical RAID-5 array plus standard hot spare drive. RAID-5EE was first implemented in ServeRAID firmware V3.5 and was introduced to overcome the long rebuild times that are associated with RAID-5E in the event of a hard drive failure. Some older ServeRAID adapters only support RAID-5E. For more information, see *ServeRAID Adapter Quick Reference*, TIPS0054, which is available online at:

<http://www.redbooks.ibm.com/abstracts/tips0054.html>

Adding a hot spare drive to a server protects data by reducing the time spent in the critical state. However, this technique does not make maximum use of the hot spare drive because it sits idle until a failure occurs. Often many years can elapse before the hot spare drive is ever used. IBM invented a method to use the hot spare drive to increase performance of the RAID-5 array during typical processing and preserve the hot spare recovery technique, and this method of incorporating the hot spare into the RAID array is called RAID-5E.

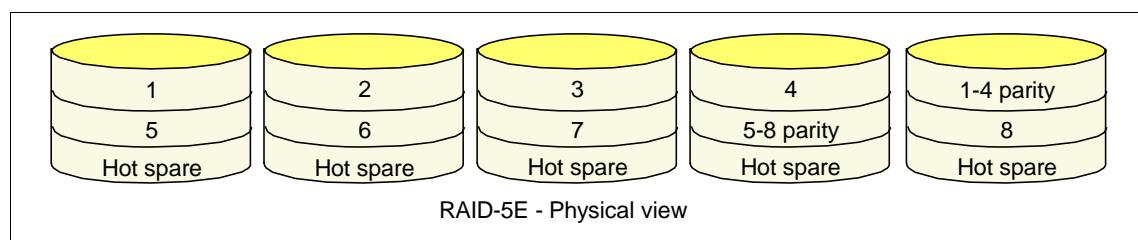


Figure 11-9 RAID-5E: The hot spare is integrated into all disks, not a separate disk

RAID-5EE is slightly different from RAID-5E in that the hot spare segments are distributed through the drives with the parity segments, instead of existing at the end of each disk's space as shown in the figure.

RAID-5E is designed to increase the normal operating performance of a RAID-5 array in two ways:

- ▶ The hot-spare drive includes data that can be accessed during normal operation. The RAID array now has an extra drive to contribute to the throughput of read and write operations. Standard 10 000 RPM drives can perform more than 100 I/O operations per second, so RAID-5 array throughput is increased with this extra I/O capability.
- ▶ The data in RAID-5E is distributed over  $n+1$  drives instead of  $n$  as is done for RAID-5. As a result, the data occupies fewer tracks on each drive. This has the effect of physically utilizing less space on each drive, keeping the head movement more localized and reducing seek times.

Together, these improvements yield a typical system-level performance gain of about 10% to 20%.

A disadvantage of RAID-5EE is that the hot spare drive cannot be shared across multiple physical arrays, as can be done with standard RAID-5 plus a hot spare. This RAID-5 technique is more cost-efficient for multiple arrays because it allows a single hot spare drive to provide coverage for multiple physical arrays.

This reduces the cost of using a hot spare drive, but the downside is the inability to handle separate drive failures within different arrays. IBM ServeRAID adapters offer increased flexibility by offering the choice to use either standard RAID-5 with a hot spare or the newer integrated hot spare provided with RAID-5EE.

Although RAID-5EE provides a performance improvement for most operating environments, there is a special case where its performance can be slower than RAID-5. Consider a three-drive RAID-5 with hot spare configuration, as shown in Figure 11-10. This configuration employs a total of four drives, but the hot spare drive is idle. Thus, for a performance comparison, it can be ignored. A four-drive RAID-5EE configuration would have data and checksum on four separate drives.

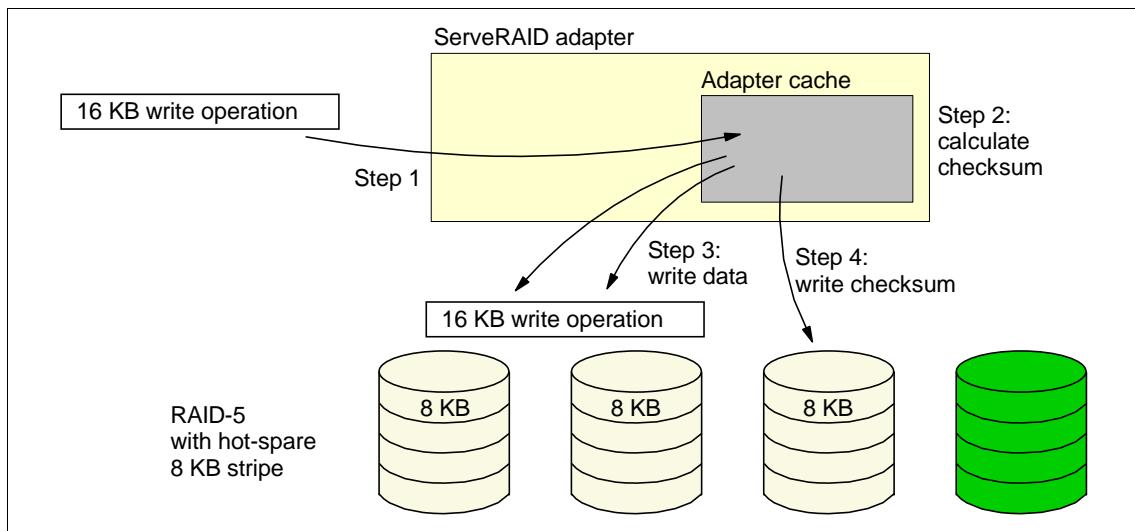


Figure 11-10 Writing a 16 KB block to a RAID-5 array with an 8 KB stripe size

Referring to Figure 11-10, whenever a write operation is issued to the controller that is *two times* the stripe size (for example, a 16 KB I/O request to an array with an 8 KB stripe size), a three-drive RAID-5 configuration would not require any reads because the write operation would include all the data needed for each of the two drives. The checksum would be generated by the array controller (step 2) and immediately written to the corresponding drive (step 4) without the need to read any existing data or checksum. This entire series of events would require two writes for data to each of the drives storing the data stripe (step 3) and one write to the drive storing the checksum (step 4), for a total of three write operations.

Contrast these events to the operation of a comparable RAID-5EE array which includes four drives, as shown in Figure 11-11 on page 265. In this case, in order to calculate the checksum, a read must be performed of the data stripe on the extra drive (step 2). This extra read was not performed with the three-drive RAID-5 configuration, and it slows the RAID-5EE array for write operations that are twice the stripe size.

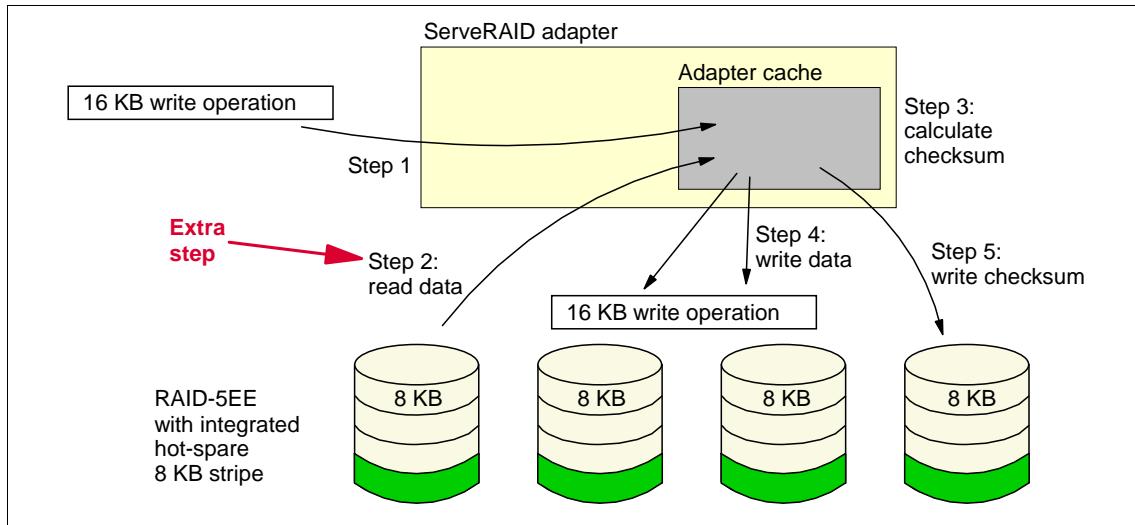


Figure 11-11 Writing a 16 KB block to a RAID-5EE array with a 8 KB stripe size

You can avoid this issue with RAID-5E by selecting the proper stripe size. By monitoring the average I/O size in bytes or by knowing the I/O size that is generated by the application, you can select a large enough stripe size so that this performance degradation rarely occurs.

### 11.5.6 RAID-6

RAID-6 provides higher fault tolerance that allows for two drives to fail, or a single drive failure and subsequent bad block failure. The fault tolerance of the second drive failure is achieved by implementing a second distributed parity method across all of the drives in a RAID-6 array. RAID-6 requires a minimum of four drives.

The two-drive fault tolerance provided by RAID-6 is computed using Galois field algebra. Refer to documentation about group and ring theory for an in-depth examination of Galois field algebra.

The rebuild process for a single drive failure is not as complex as the rebuild process for two-drive failures. Remember that performance degrades during rebuild time due to the RAID controller devoting cycles to restoring data as well as simultaneously processing incoming I/O requests. Therefore, users must decide if the extra fault tolerance is worth degraded performance for a longer rebuild time following the failure of two drives.

Alternatively, with the increasing popularity of less expensive, less robust hard disk drive technology such as SATA, a RAID-6 configuration might be worth the longer rebuild times. In addition, the ever-increasing capacities of hard disk drives could potentially increase the chances of another disk failure or bad block failure during longer rebuild times due to the larger capacity. As always, performance must be weighed against potential downtime due to drive failures.

### 11.5.7 RAID-10, RAID-50 and other composite levels

The ServerRAID adapter family supports composite RAID levels. This means that it supports RAID arrays that are joined together to form larger RAID arrays.

Figure 11-12 illustrates a ServeRAID RAID-10 array.

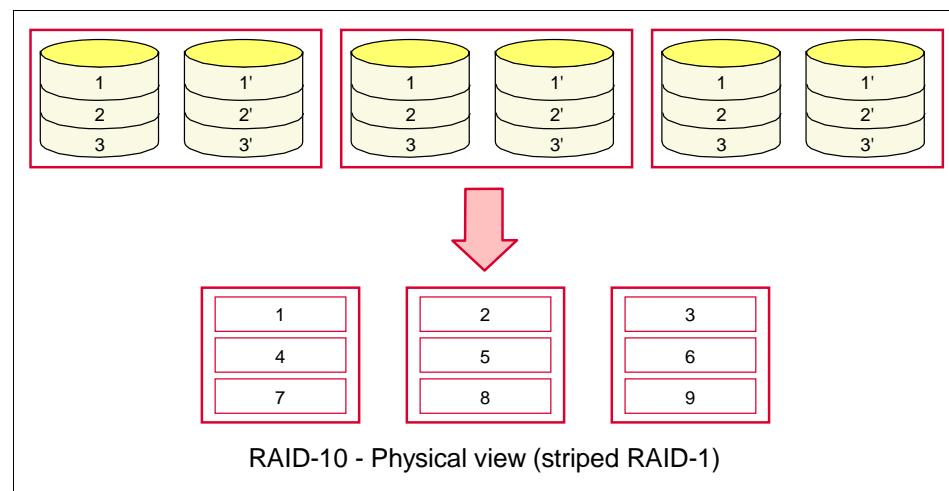


Figure 11-12 RAID-10: a striped set of RAID-1 arrays

Likewise, Figure 11-13 shows a striped set of RAID-5 arrays.

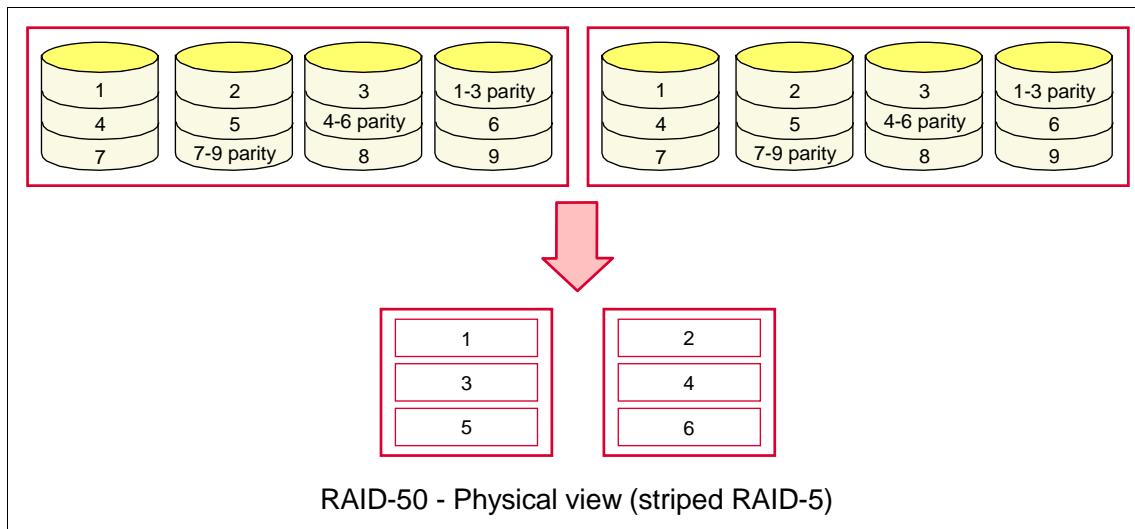


Figure 11-13 RAID-50: a striped set of RAID-5 arrays

Many of the ServeRAID adapters support the combinations that are listed in Table 11-3.

Table 11-3 Composite RAID spans supported by ServeRAID adapters

RAID level	The sub-logical array is	The spanned array is
RAID-10	RAID-1	RAID-0
RAID-50	RAID-5	RAID-0
RAID-60	RAID-6	RAID-0

## 11.6 Factors that affect disk performance

Many factors can affect disk performance. The most important considerations for configuring storage are:

- ▶ 11.6.1, “RAID strategy” on page 268
- ▶ 11.6.2, “Number of drives” on page 269
- ▶ 11.6.3, “Active data set size” on page 271
- ▶ 11.6.4, “Drive performance” on page 273
- ▶ 11.6.5, “Logical drive configuration” on page 274
- ▶ 11.6.6, “Stripe size” on page 275
- ▶ 11.6.7, “Disk cache write-back versus write-through” on page 281

- ▶ 11.6.8, “RAID adapter cache size” on page 282
- ▶ 11.6.9, “Rebuild time” on page 284
- ▶ 11.6.10, “Device drivers and firmware” on page 284
- ▶ 11.6.11, “Fibre Channel performance considerations” on page 285

The topics that we list here are relevant to all disk subsystems, whether these disk subsystems are attached directly or remote. The last section also looks at how to configure Fibre Channel to achieve the best overall disk performance.

### 11.6.1 RAID strategy

The RAID strategy should be carefully selected because it significantly affects disk subsystem performance. Figure 11-14 on page 269 illustrates the relative performance between RAID-0, RAID-10, RAID-5, and RAID-6 using ServeRAID MR10M. The chart shows the RAID-0 configuration has very good I/Os per second performance, but this level does not provide fault tolerance.

RAID-0 has no fault tolerance and is, therefore, best used for read-only environments when downtime for possible backup recovery is acceptable or specific applications where high I/Os per second are required and fault tolerance is built into the application.

You need to select RAID-10, RAID-5, or RAID-6 for applications that require fault tolerance. RAID-10 is usually selected when the number of drives is low (less than six) and the price for purchasing additional drives is acceptable. RAID-10 offers faster performance than RAID-5 or RAID-6. You must understand these performance considerations before you select a fault-tolerant RAID strategy.

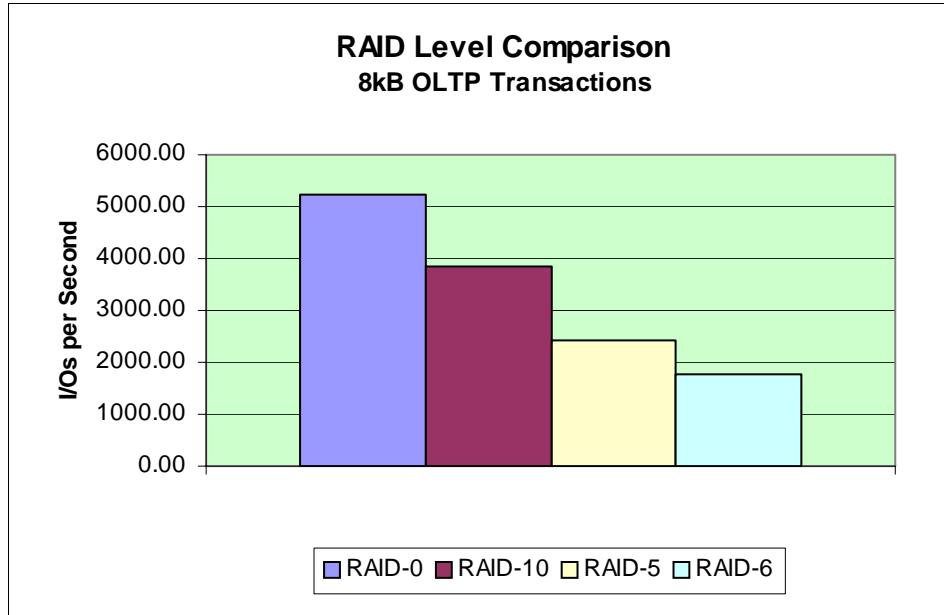


Figure 11-14 Comparing RAID levels (ServeRAID MR10M)

In many cases, RAID-5 is the best choice in small arrays because it provides the best price and performance combination for configurations requiring capacity five or more disk drives. RAID-5 performance approaches RAID-0 performance for workloads where the frequency of write operations, and thus parity calculations, is low. Servers executing applications that require fast read access to data and high availability in the event of a drive failure should employ RAID-5.

RAID-6 is used on complex arrays using 12-24 drives. RAID-6 significantly enhances reliability by providing protection against simultaneous two-drive failure by storing two sets of distributed parity. It is also recommended for arrays using SATA drives with lower duty cycle that might be more likely to fail in 24/7 or business-critical applications.

## 11.6.2 Number of drives

The number of disk drives affects performance significantly, because each drive contributes to total system throughput. Capacity requirements are often the only consideration used to determine the number of disk drives configured in a server. Throughput requirements are usually not well understood or are completely ignored. Capacity is used because it is estimated easily and is often the only information available.

The result is a server configured with sufficient disk space, but insufficient disk performance to keep users working efficiently. High-capacity drives have the lowest price per byte of available storage and are usually selected to reduce total system price. This often results in disappointing performance, particularly if the total number of drives is insufficient.

It is difficult to specify server application throughput requirements accurately when attempting to determine the disk subsystem configuration. Disk subsystem throughput measurements are complex. To express a user requirement in terms of “bytes per second” is meaningless because the disk subsystem’s byte throughput changes as the database grows and becomes fragmented, and as new applications are added.

The best way to understand disk I/O and user throughput requirements is to monitor an existing server. You can use tools such as the Windows Performance console to examine the logical drive queue depth and disk transfer rate (described in 17.1, “Reliability and Performance Monitor console” on page 534). Logical drives that have an average queue depth much greater than the number of drives in the array are very busy, which indicates that performance can be improved by adding drives to the array.

**Tip:** In general, adding drives is one of the most effective changes that you can make to improve server performance.

Measurements show that server throughput for most server application workloads increases as the number of drives configured in the server is increased. As the number of drives is increased, performance usually improves for all RAID strategies. Server throughput continues to increase each time drives are added to the server, as shown in Figure 11-15 on page 271.

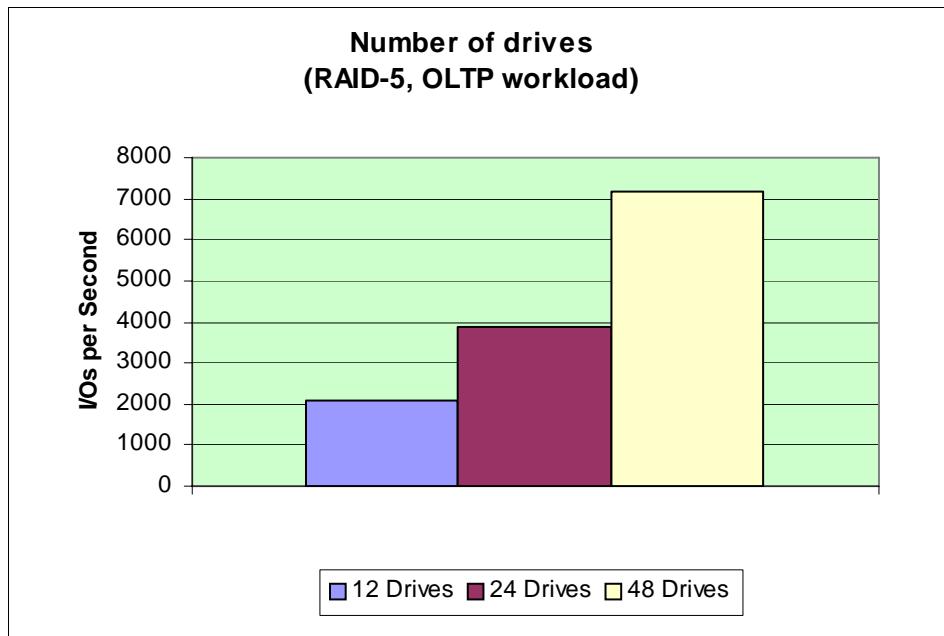


Figure 11-15 Improving performance by adding drives to arrays

This trend continues until another server component becomes the bottleneck. In general, most servers are configured with an insufficient number of disk drives. Therefore, performance increases as drives are added. Similar gains can be expected for all I/O-intensive server applications such as office-application file serving, Lotus Notes, Oracle, DB2, and Microsoft SQL Server.

**Rule of thumb:** For most server workloads, when the number of drives in the active logical array is doubled, server throughput improves by about 50% or until other bottlenecks occur.

If you are using one of the IBM ServeRAID family of RAID adapters, you can use the *logical drive migration* feature to add drives to existing arrays without disrupting users or losing data.

### 11.6.3 Active data set size

The active data set is the set of data that an application uses and manipulates on a regular basis. In benchmark measurements, the active data set is often referred to as *stroke*, as in 10% stroke, meaning that the data is stored on only 10% of the disk surface.

As discussed earlier, many drive configurations are based on capacity requirements, which means that the data is stored over a large percentage of the total capacity of the drives. The downside of filling the disks with data is that in most production environments, it translates into reduced performance due to longer seek times.

Figure 11-16 illustrates the performance degradation of a disk drive with respect to the size of the active data set.

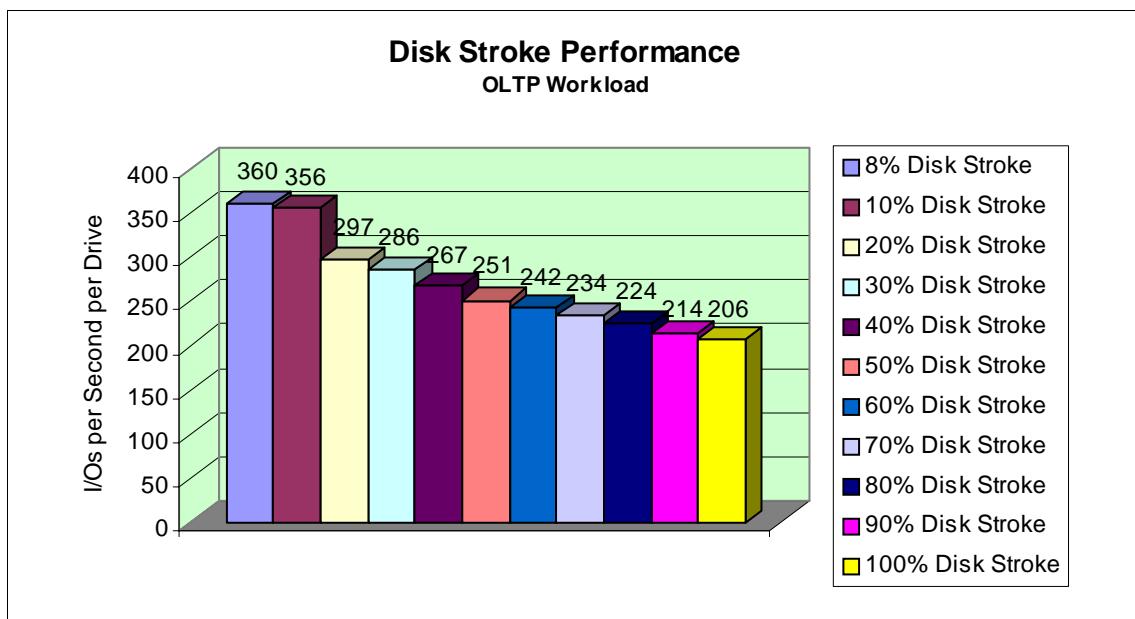


Figure 11-16 Hard drive performance with respect to active data set size

If the active data set spans 50% of the drive capacity, then a 15 K RPM drive is capable of achieving approximately 251 I/Os per second for a simulated database workload that consists of 67% reads and 33% writes that are randomly accessed. Adding enough drives so that the active data set spans only 20% of the drive capacity would increase the drive performance by 18%, to 297 I/Os per second. As shown in Figure 11-16, spreading the active data set across more drives minimizes the seek time, and therefore improves performance.

Disk fragmentation also degrades performance. Over time, files become fragmented on the hard drive, which means that the data in those files is not arranged contiguously on the hard drive. Consequently, a request for a fragmented file will result in multiple seeks in order to satisfy the request. You need to use a disk defragmentation tool on a regular basis to maintain

contiguous geometry of data within a file, which can ensure optimized performance.

#### 11.6.4 Drive performance

Drive performance contributes to overall server throughput because faster drives perform disk I/O in less time. There are four major components to the time it takes a disk drive to execute and complete a user request:

- ▶ Command overhead

This is the time it takes for the drive's electronics to process the I/O request. The time depends on whether it is a read or write request and whether the command can be satisfied from the drive's buffer. This value is of the order of 0.1 ms for a buffer hit to 0.5 ms for a buffer miss.

- ▶ Seek time

This is the time it takes to move the drive head from its current cylinder location to the target cylinder. As the radius of the drives has been decreasing, and drive components have become smaller and lighter, so too has the seek time been decreasing. Average seek time is usually 3-5 ms for most current drives used in servers today.

- ▶ Rotational latency

When the head is at the target cylinder, the time it takes for the target sector to rotate under the head is called the rotational latency. Average latency is half the time it takes the drive to complete one rotation, so it is inversely proportional to the RPM value of the drive:

- 15,000 RPM drives have a 2.0 ms latency
- 10,000 RPM drives have a 3.0 ms latency
- 7200 RPM drives have a 4.2 ms latency
- 5400 RPM drives have a 5.6 ms latency

- ▶ Data transfer time

This value depends on the *media data rate*, which is how fast data can be transferred from the magnetic recording media, and the *interface data rate*, which is how fast data can be transferred between the disk drive and disk controller (for example, the SAS transfer rate). The sum of these two values is typically 1 ms or less.

As you can see, the significant values that affect performance are the seek time and the rotational latency. For random I/O (which is normal for a multiuser server), this is true. Reducing the seek time will continue to improve performance as the physical drive attributes become fewer.

For sequential I/O (such as with servers with small numbers of users requesting large amounts of data) or for I/O requests of large block sizes (for example, 64 KB), the data transfer time does become important when compared to seek and latency.

Likewise, when caching and read-ahead is employed on the drives themselves, the time taken to perform the seek and rotation is eliminated, so the data transfer time becomes very significant.

In addition to seek time and rotational latency, current IBM disk drives improve performance by employing advanced I/O command optimization. These drives achieve high performance in part because of a rotational positioning optimization (RPO) scheme. RPO utilizes an onboard microprocessor to sort incoming I/O commands to reduce disk head movement and increase throughput.

For example, assuming the disk head is at track number 1 and sector number 1, IBM drives would optimize the following three I/O requests from:

1. Read track 1 sector 2
2. Write track 1 sector 50
3. Read track 3 sector 10

to:

1. Read track 1 sector 2
2. Read track 3 sector 10
3. Write track 1 sector 50

The optimization algorithm of the IBM drives reorders the I/O requests whenever a seek to another track can be accomplished before the disk rotates to the sector of the next I/O request. This technique effectively increases the drive's throughput by processing an I/O command while waiting for the rotational latency of the next I/O request to expire.

The easiest way to improve disk performance is to increase the number of accesses that can be made simultaneously by using many drives in a RAID array and spreading the data requests across all drives. See 11.6.2, “Number of drives” on page 269, for more information about this topic.

### 11.6.5 Logical drive configuration

Using multiple logical drives on a single physical array is convenient for managing the location of different file types. However, depending on the configuration, it can significantly reduce server performance.

When you use multiple logical drives, you are physically spreading the data across different sections of the array disks. If I/O is performed to each of the

logical drives, the disk heads have to seek further across the disk surface than when the data is stored on one logical drive. Using multiple logical drives greatly increases seek time and can slow performance by as much as 25%.

An example of this is creating two logical drives in the one RAID array and putting a database on one logical drive and the transaction log on the other. Because heavy I/O is being performed on both, the performance will be poor. If the two logical drives are configured with the operating system on one and data on the other, then there should be little I/O to the operating system code after the server has booted, so this type of configuration would work.

It is best to put the page file on the same drive as the data when using one large physical array. This is counterintuitive, in that many people think the page file should be put on the operating system drive because the operating system will not see much I/O during runtime. However, this causes long seek operations as the head swings over the two partitions. Putting the data and page file on the data array keeps the I/O localized and reduces seek time.

Of course, this is not the most optimal case, especially for applications with heavy paging. Ideally, the page drive will be a separate device that can be formatted to the correct stripe size to match paging. In general, most applications will not page when given sufficient RAM, so usually this is not a problem.

The fastest configuration is a single logical drive for each physical RAID array. Instead of using logical drives to manage files, create directories and store each type of file in a different directory. This will significantly improve disk performance by reducing seek times because the data will be as physically close together as possible.

If you really want or need to partition your data and you have a sufficient number of disks, you need to configure multiple RAID arrays instead of configuring multiple logical drives in one RAID array. This improves disk performance; seek time is reduced because the data is physically closer together on each drive.

### 11.6.6 Stripe size

*Striping* is the process of storing data across all the disk drives that are grouped in an array. With RAID technology, data is striped across an array of hard disk drives.

The granularity at which data from one file is stored on one drive of the array before subsequent data is stored on the next drive of the array is called the *stripe unit* (also referred to as *interleave depth*). For the ServeRAID adapter family, the stripe unit can be set to a stripe unit size of up to 64 KB for older adapters, and

up to 1024 KB for current adapters. With Fibre Channel, a stripe unit is called a *segment*, and segment sizes can also be up to 1 MB.

The collection of these stripe units, from the first drive of the array to the last drive of the array, is called a *stripe*.

Figure 11-17 shows the stripe and stripe unit.

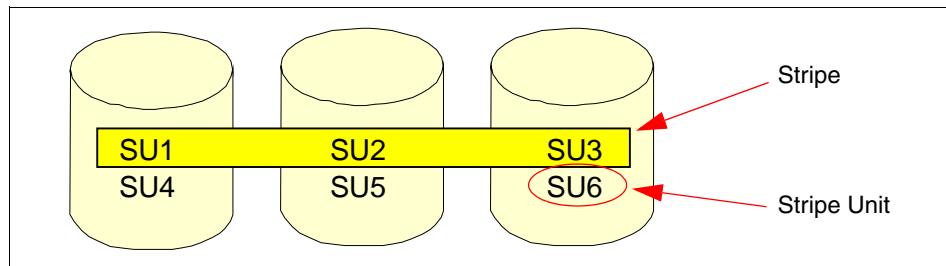


Figure 11-17 RAID stripes and stripe units

**Note:** The term *stripe size* should really be *stripe unit size* because it refers to the length of the stripe unit (the piece of space on each drive in the array).

Using stripes of data balances the I/O requests within the *logical drive*. On average, each disk will perform an equal number of I/O operations, thereby contributing to overall server throughput. Stripe size has no effect on the total capacity of the logical disk drive.

### Selecting the correct stripe size

The selection of stripe size affects performance. In general, the stripe size should be at least as large as the median disk I/O request size generated by server applications.

- ▶ Selecting too small a stripe size can reduce performance. In this environment, the server application requests data that is larger than the stripe size, which results in two or more drives being accessed for each I/O request. Ideally, only a single disk I/O occurs for each I/O request.
- ▶ Selecting too large a stripe size can reduce performance because a larger than necessary disk operation might constantly slow each request. This is a problem, particularly with RAID-5 where the complete stripe must be read from disk to calculate a checksum. If you use too large a stripe, extra data must be read each time the checksum is updated.

**Note:** It is better to choose too large a stripe unit size rather than too small. The performance degradation from too small a stripe unit size is more likely to be greater than the degradation caused by too large a stripe unit size.

Selecting the correct stripe size is a matter of understanding the predominate request size performed by a particular application. Few applications use a single request size for each and every I/O request. Therefore, it is not possible to always have the ideal stripe size. However, there is always a best-compromise stripe size that will result in optimal I/O performance.

There are two ways to determine the best stripe size:

- ▶ Use a rule of thumb, as listed in Table 11-4.
- ▶ Monitor the I/O characteristics of an existing server.

The first and simplest way to choose a stripe size is to use Table 11-4 as a guide. This table is based on tests performed by the System x Performance Lab.

*Table 11-4 Stripe size setting for various applications*

Applications	Stripe size
Groupware on Windows (Lotus Domino, Exchange, and so on)	32 KB to 64 KB
Database server on Windows (Oracle, SQL Server, DB2, and so on)	32 KB to 64 KB
File server (Windows)	16 KB
File server (NetWare)	16 KB
File server (Linux)	64 KB
Web server	8 KB
Video file server (streaming media)	64 KB to 1 MB
Other	16 KB

**Important:** Table 11-4 is only relevant if no information is available with respect to the production environment. The more information that you can determine about the production application and its average data request size, the more accurate the stripe size setting can become.

In general, the stripe unit size only needs to be at least as large as the I/O size. Having a smaller stripe size implies multiple physical I/O operations for each logical I/O, which will cause a drop in performance. Using a larger stripe size implies a read-ahead function which might or might not improve performance.

Table 11-4 on page 277 offers rule-of-thumb settings. There is no way to offer the precise stripe size that will always give the best performance for every environment without doing extensive analysis on the specific workload.

Further, due to advances in RAID performance, most modern RAID adapters have minimum stripe sizes of around 64 kB. As you can see from Table 11-4 on page 277, this stripe would handle most random workloads today. Nevertheless, there can be significant gains in sequential access workloads using modern ServeRAID controllers like the MR10m when using a 128 kB stripe. For this reason, it is recommended that 128 kB stripes be used where mixed workloads may exist.

The second way to determine the correct stripe size involves observing the application while it is running using the Windows Performance console. The key is to determine the average data transfer size being requested by the application and select a stripe size that best matches. Unfortunately, this method requires the system to be running, so it either requires another system running the same application, or reconfiguring the existing disk array after the measurement has been made (and therefore backup, reformat, and restore operations).

The Windows Performance console can help you determine the proper stripe size. Select:

- ▶ The object: PhysicalDisk
- ▶ The counter: Avg. Disk Bytes/Transfer
- ▶ The instance: the drive that is receiving the majority of the disk I/O

Then, monitor this value. As an example, the trend value for this counter is shown as the thick line in Figure 11-18 on page 279. The running average is shown as indicated. The figure represents an actual server application.

You can see that the application request size (represented by Avg. Disk Bytes/Transfer) varies from a peak of 64 KB to about 20 KB for the two run periods.

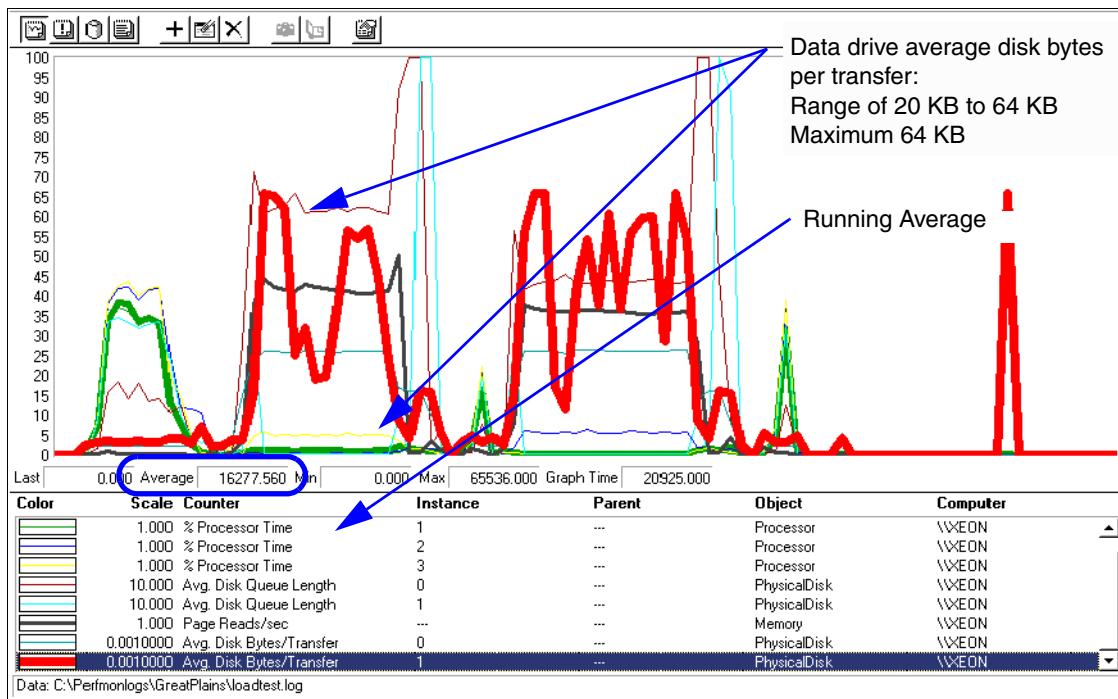


Figure 11-18 Average I/O size

**Note:** This technique is not infallible. It is possible for the Bytes per Transfer counter to have a very high degree of variance. When this occurs, using an average value to select the stripe size is less precise than using a distribution.

However, most generally available monitoring software is limited to providing average values for bytes per transfer. Fortunately, using a simple average is sufficient for most applications.

As previously mentioned, in general the stripe size should be at least as large as the median disk I/O request size generated by the server application.

This particular server was configured with an 8 KB stripe size on an older RAID controller, which produced very poor performance. Increasing the stripe size to 16 KB would improve performance, and increasing the stripe size to 32 KB would increase performance even more. The simplest technique would be to place the time window around the run period and select a stripe size that is at least as large as the average size shown in the running average counter.

### **Activating disk performance counters:**

Windows Server 2003 and Windows Server 2008 have both the logical and physical disk counters enabled by default.

In Windows 2000, physical disk counters are enabled by default. The logical disk performance counters are disabled by default and might be required for some monitoring applications. If you require the logical counters, you can enable them by typing the command **DISKPERF -yv** then restarting the computer.

Keeping this setting on all the time draws about 2% to 3% CPU, but if your CPU is not a bottleneck, this is irrelevant and can be ignored. Enter **DISKPERF /?** for more help regarding the command.

In Windows Server 2003 and 2008, it is even more important to use the Windows System Monitor to determine the average data transfer size. Both operating systems file data can be cached in 256 KB chunks in the system address space. This allows file data to be read from and written to the disk subsystem in 256 KB data blocks. Therefore, using System Monitor to set the stripe size to match the data transfer size is important.

Server applications that serve video and audio are the most common workloads that can have average transfer sizes larger than 64 KB. However, a growing number of real-world applications that store video and audio in SQL databases also have average transfer sizes larger than 64 KB.

In cases where the Windows Performance console reports average data transfer sizes that are greater than 64 KB, the stripe unit size should be increased appropriately.

### **Page file drive**

Windows Server performs page transfers at up to 64 KB per operation, so the paging drive stripe size can be as large as 64 KB. However, in practice, it is usually closer to 32 KB because the application might not make demands for large blocks of memory which limits the size of the paging I/O.

Monitor average bytes per transfer as described in “Selecting the correct stripe size” on page 276. Setting the stripe size to this average size can result in a significant increase in performance by reducing the amount of physical disk I/O that occurs because of paging.

For example, if the stripe size is 8 KB and the page manager is doing 32 KB I/O transfers, then four physical disk reads or writes must occur for each page per

second that you see in the Performance console. If the system is paging 10 pages per second, then the disk is actually doing 40 disk transfers per second.

### 11.6.7 Disk cache write-back versus write-through

Most people think that write-back mode is always faster because it allows data to be written to the disk controller cache without waiting for disk I/O to complete.

This is usually the case when the server is lightly loaded. However, as the server becomes busy, the cache fills completely, thereby causing data writes to wait for space in the cache before being written to the disk. When this happens, data write operations slow to the speed at which the disk drives empty the cache. If the server remains busy, the cache is flooded by write requests, which results in a bottleneck. This bottleneck occurs regardless of the size of the adapter's cache.

In write-through mode, write operations do not wait in cache memory that must be managed by the processor on the RAID adapter. When the server is lightly loaded (shown in the left side of Figure 11-19), write operations take longer because they cannot be quickly stored in the cache. Instead, they must wait for the actual disk operation to complete. Thus, when the server is lightly loaded, throughput in write-through mode is generally lower than in write-back mode.

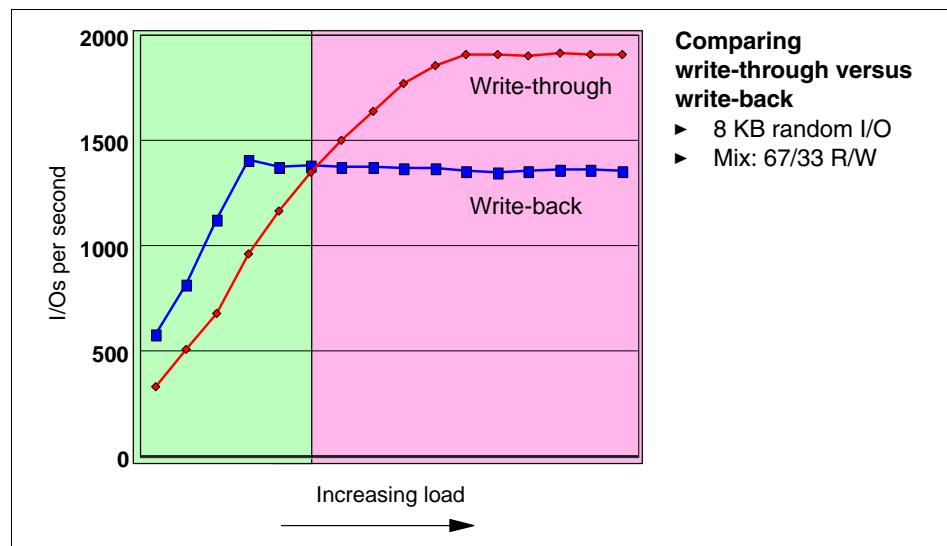


Figure 11-19 Comparing write-through and write-back modes under increasing load

However, when the server becomes very busy (shown in the right side of Figure 11-19), I/O operations do not have to wait for available cache memory. I/O

operations go straight to disk, and throughput is usually greater for write-through than in write-back mode.

Write-through is also faster when battery-backup cache is installed; this is due partly to the fact that the cache is mirrored. Data in the primary cache has to be copied to the memory on the battery-backup cache card. This copy operation eliminates a single point of failure, thereby increasing the reliability of the controller in write-back mode. However, this takes time and slows writes, especially when the workload floods the adapter with write operations.

The difficult part is to determine where this crossover point is. There is no set rule of thumb because each server and corresponding workload is different. However, as a starting point, use Performance Monitor to determine whether the counter Average Disk sec/write (write response time) is greater than 40 ms. If this is the case and more drives cannot be added to reduce the response time, we recommend that you use the write-through cache policy.

**Rules of thumb:** Based on Figure 11-19, the following rules of thumb are appropriate.

- ▶ If the disk subsystem is very busy (write response time greater than 40 ms), use write-through mode.
- ▶ If the disks are configured correctly, and the server is not heavily loaded, use write-back mode.

### 11.6.8 RAID adapter cache size

Tests show that RAID adapters with very large caches (that is, 512 MB) do not outperform adapters with more typical cache sizes of 256 MB for most real-world application workloads. After the cache size has exceeded the minimum required for the job, the extra cache usually offers little additional performance benefit.

The cache influence depends on specific workloads. With most RAID processors on current controllers fast enough to handle calculations in real time, cache memory speed and bus width become important.

Larger caches have the potential to increase performance by providing data very quickly that would otherwise be accessed from slower disk drives. However, in real-world applications, total data space is almost always so much larger than disk cache size that, for most operations, there is very little statistical chance of finding the requested data in the cache.

For example, a 256 GB database would not be considered very large by today's standards. A typical database of this size might be placed on an array consisting

of four or more 73 GB drives. For random accesses to such a database, the probability of finding a record in the cache would be the ratio of 256 MB:256 GB, or approximately 1 in 1000 operations. If you double the cache size, this value is decreased by half, which is still a very discouraging hit-rate. It would take a very large cache to increase the cache hit-rate to the point where caching becomes advantageous for random accesses.

In RAID-5 mode, significant performance gains from write-back mode are derived from the ability of the disk controller to merge multiple write commands into a single disk write operation. In RAID-5 mode, the controller must update the checksum information for each data update. Write-back mode allows the disk controller to keep the checksum data in adapter cache and perform multiple updates before completing the update to the checksum information contained on the disk. In addition, this does not require a large amount of RAM.

In most cases, disk array caches can usually provide high hit rates only when I/O requests are sequential. In this case, the controller can pre-fetch data into the cache so that on the next sequential I/O request, a cache hit occurs. Pre-fetching for sequential I/O requires only enough buffer space or cache memory to stay a few steps ahead of the sequential I/O requests. This can be done with a small circular buffer, so very large caches have little benefit here.

Having a large cache often means more memory to manage when the workload is heavy; during light loads, very little cache memory is required.

Most people do not invest the time to think about how cache works. Without much thought, it is easy to reach the conclusion that “bigger is always better.” The drawback is that larger caches take longer to search and manage. This can slow I/O performance, especially for random operations, because there is a very low probability of finding data in the cache.

Benchmarks often do not reflect a customer production environment. In general, most “retail” benchmark results run with very low amounts of data stored on the disk drives. In these environments, a very large cache will have a high hit-rate that is artificially inflated compared to the hit rate from a production workload. Unfortunately, this can cause incorrect assumptions about drive caches to be made, because few customers have the resources to build test environments that mimic their production workloads.

In identical hardware configurations, it takes more CPU overhead to manage 512 MB of cache than it does to manage 256 MB. In a production environment, an overly large cache can actually slow performance because the adapter continuously searches the cache for data that is never found, before it starts the required disk I/O. This is the reason why many array controllers turn off the cache when the hit rates fall below an acceptable threshold.

## 11.6.9 Rebuild time

Rebuild time is an important part of the overall performance of a RAID subsystem. The longer the disk subsystem spends recovering from a drive failure, the longer the subsystem is vulnerable to losing data if another drive failure occurs. In addition, the performance of a RAID controller that is busy doing a rebuild will also be lower.

Rebuild time varies depending on the RAID controller, the capacity of the drive, the number of drives in an array, the I/O workload during rebuild, and the RAID level. The smaller the drive, the shorter the rebuild time.

## 11.6.10 Device drivers and firmware

Device drivers play a major role in performance of the subsystem to which the driver is associated. A *device driver* is software written to recognize a specific device. Most of the device drivers are vendor-specific; these drivers are supplied by the hardware vendor (such as IBM, in the case of ServeRAID). For IBM ServeRAID and Fibre Channel products, go to the IBM support site to download the latest version:

<http://ibm.com/support>

The same applies to firmware. The firmware is stored on the disk controller itself (for example, the ServeRAID adapter) and is often the source of significant performance improvements. Firmware updates are also available from the site mentioned, but you should review the Read Me file associated with the upgrade to determine if performance improvements are likely.

Wherever it is practical, we recommend that you always maintain your servers with the latest version of driver and firmware. It might not be practical if your system requires a high level of uptime and there is currently no performance or access problem. Specifically for ServeRAID, we recommend that you ensure that the driver, firmware, and ServeRAID Manager code are always at the latest level.

Also note that sometimes the latest driver is not the best or correct driver to use. This point is especially important to understand with specific hardware configurations that are certified by an application vendor. An example of this circumstance is the Microsoft Cluster Server. You must check the certified configuration to determine what driver level is supported.

### 11.6.11 Fibre Channel performance considerations

Now we examine what happens when a read I/O operation is requested to a Fibre Channel subsystem, and the data requested is not located in the RAID controller disk cache:

1. A read command is generated by the server, and the read command includes the logical block address of the data being requested.
2. The SCSI command is encapsulated by Fibre Channel frames and transmitted by the Fibre Channel host adapter to the RAID controller over the Fibre Channel link.
3. The RAID controller parses the read command and uses the logical block address to issue the disk read command to the correct drive.
4. The disk drive performs the read operation and returns the data to the RAID controller.
5. The Fibre Channel electronics within the RAID controller encapsulate the data Fibre Channel frames. The data is transferred to the server over the Fibre Channel link.
6. When in the Fibre Channel adapter, the data is transferred over the PCI bus into memory of the server.

A large amount of the detail was left out in this description, but this level of observation is sufficient to understand the most important performance implication of Fibre Channel.

The Fibre Channel link, like most network connections, sustains a data transfer rate that is largely determined by the payload of the frame. Stated another way, the throughput of Fibre Channel is a function of the disk I/O size being transferred. This is because Fibre Channel frames have a maximum data payload of 2112 bytes. Data transfers for larger data sizes require multiple Fibre Channel frames.

Figure 11-20 on page 286 illustrates the effects of disk request size on Fibre Channel throughput. At small disk request sizes such as 2 KB, the maximum Fibre Channel throughput is about 225 MBps or about 30% the maximum achievable bandwidth that four 2 Gbps links provide. These values provide critical information, because many people think the maximum bandwidth of a Fibre Channel link or the maximum aggregate bandwidth of multiple links is obtained for all operations.

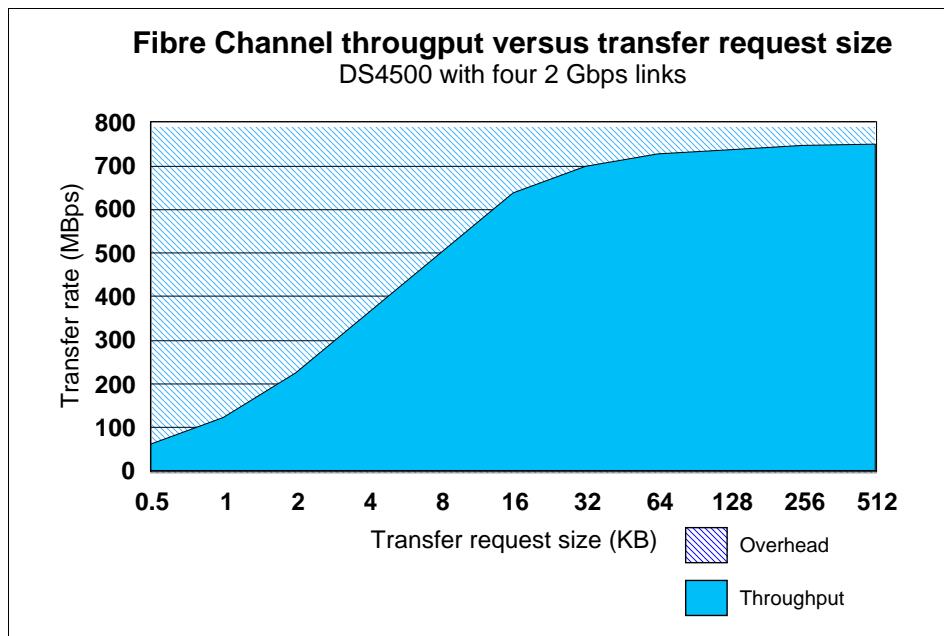


Figure 11-20 Fibre Channel throughput versus transfer request size

Only when the transfer request size is as large as 64 KB does Fibre Channel begin to reach its maximum sustainable transfer rate over four 2 Gbps links. In this case, the maximum transfer rate is approximately 750 MBps.

Why is the maximum transfer rate of four 2 Gbps links not equal to 800 MBps ( $4 \times 2$  Gbit links equals 800 MBps, taking into account 2-bit serial overhead for every byte)? The maximum theoretical throughput for a 1 Gbps Fibre Channel link is 92 MBps, which is 92% of the maximum theoretical throughput of a 1 Gbit link (100 MBps).

It is interesting to note that the maximum throughput of four 2 Gbps links (750 MBps) is approximately 94% of the maximum theoretical throughput of four 2 Gbit links (800 MBps). This overhead has remained nearly constant over time, including current 8-Gbit technologies. Therefore, a single 8-Gbit FC link will have similar performance capabilities as four 2-Gbit FC links.

The difference between the measured result and the theoretical maximum throughput can be explained by overhead of command and control bits that accompany each Fibre Channel frame. This is discussed in the following sections.

## Fibre Channel protocol layers

We can get a better appreciation for the overhead described in the previous section by taking a brief look at the Fibre Channel layers and the Fibre Channel frame composition.

The Fibre Channel specification defines five independent protocol layers, as illustrated in Figure 11-21. These layers are structured so that each layer has a specific function to enable reliable communications for all of the protocols supported by Fibre Channel standard.

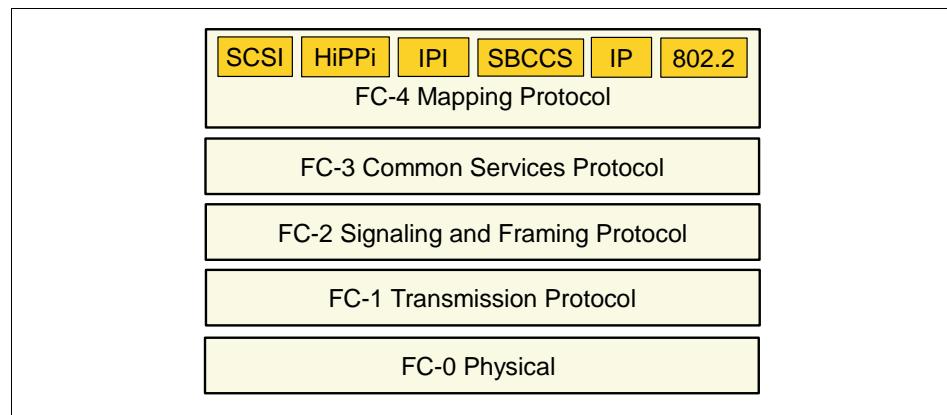


Figure 11-21 Fibre Channel functional levels

The five independent layers are described here:

- ▶ FC-0 is the physical layer. It is comprised of the actual wire or optical fibre over which data travels.
- ▶ FC-1 is the transmission protocol. The Transmission layer is responsible for encoding of the bits on the physical medium, for data transmission error detection, and for signal clock generation.
- ▶ FC-2 is important from a performance perspective because this is the layer that is responsible for building the data frames that flow over the Fibre Channel link. FC-2 is also responsible for segmenting large transfer requests into multiple Fibre Channel frames.
- ▶ FC-3 defines the common services layer. This layer is responsible for defining the common services that are accessible across all Fibre Channel ports. One of these services is the Name Server. The Name Server provides a directory of all the Fibre Channel nodes accessible on the connection. For example, a Fibre Channel switch would be a name server and maintain a directory of all the ports attached to that switch. Other Fibre Channel nodes could query the switch to determine what node addresses are accessible through that switch.

- ▶ FC-4 defines the protocol standards that can be used to transport data over Fibre Channel. Some of these protocols include:
  - SCSI (Small Computer Systems Interface)
  - HiPPI (High Performance Parallel Interface)
  - IPI (Intelligent Peripheral Interface)
  - SBCCS (Single Byte Command Code Set) to support ESCON®
  - IP (Internet Protocol)
  - 802.2

Our discussion is limited to SCSI because the IBM DS4000® RAID controller products are based upon the SCSI protocol. Fibre Channel allows the SCSI protocol commands to be encapsulated and transmitted over Fibre Channel to SCSI devices connected to the RAID controller unit. This is significant because this technique allows Fibre Channel to be quickly developed and function with existing SCSI devices and software.

### The importance of the I/O size

Considering the shape of the throughput chart in Figure 11-20 on page 286, we can deduce that the throughput of Fibre Channel is clearly sensitive to the disk access size. Small disk access sizes have low throughput; larger blocks have greater overall throughput. The reason for this can be seen by looking at the read command example discussed in 11.6.11, “Fibre Channel performance considerations” on page 285.

In the case of a 2 KB read operation, the sequence is:

1. A SCSI read command is issued by the device driver to the Fibre Channel host adapter at level FC-4.
2. On the host side, the SCSI read command must flow down from FC-4 to FC-0 before it is transferred over the Fibre Channel link to the external RAID controller.
3. The RAID controller also has a Fibre Channel interface that receives the read command at FC-0 and sends it up through FC-1, FC-2, FC-3, to the SCSI layer at FC-4.
4. The SCSI layer then sends the read command to the Fibre Channel RAID controller.
5. The SCSI read command is issued to the correct disk drive.
6. When the read operation completes, data is transferred from the drive to SCSI layer FC-4 of the Fibre Channel interface within the RAID controller.
7. Now the read data must make the return trip down layers FC-4, FC-3, FC-2, FC-1 on the RAID controller side and onto the Fibre Channel link.

8. When the data arrives on the Fibre Channel link, it is transmitted to the host adapter in the server.
9. Again it must travel up the layers to FC-4 on the server side before the SCSI device driver responds with data to the requesting process.

Contrast the 2 KB read command with a 64 KB read command, and the answer becomes clear.

Like the 2 KB read command, the 64 KB read command travels down FC-4, FC-3, FC-2, and to FC-1 on the server side. It also travels up the same layers on the RAID controller side.

However, here is where things are different. After the 64 KB read command completes, the data is sent to FC-4 of the Fibre Channel interface on the RAID controller side. The 64 KB data travels down from FC-4, FC-3 and to FC-2. At layer FC-2, the 64 KB data is formatted into a 2112-byte payload to be sent over the link. However, 64 KB do not fit into a 2112-byte payload. Therefore, layer FC-2 performs segmentation and breaks up the 64 KB disk data into 32 separate Fibre Channel frames to be sent to the IBM DS4000 controller.

Of the 32 frames, 31 frames never had to traverse layers FC-4 and FC3 on the RAID controller side. Furthermore, 31 of these frames never required a separate read command to be generated at all. They were transmitted with one read command.

Thus, reading data in large blocks introduces significant efficiencies because much of the protocol overhead is reduced. Any transfer exceeding the 2112 byte payload is shipped as “low-cost” frames back to the host. This explains why throughput at smaller frame sizes (see Figure 11-20 on page 286) is so low and throughput for larger frames improves as disk I/O size increases. The overhead of the FC-4 and FC-3 layers and the additional SCSI read or write commands slow throughput.

## **Configuring Fibre Channel for performance**

The important thing to understand is that degradation of throughput with smaller I/O sizes occurs. You must use that information to better configure your Fibre Channel configuration.

One way to improve performance is to profile an existing server to get an idea of the average disk transfer size by using the Performance console and by examining the following physical disk counters:

- ▶ Average disk bytes/transfer

This counter can be graphed versus time to tell you the predominant transfer size for the particular application. This value can be compared to

Figure 11-20 on page 286 to determine the maximum level of throughput a single Fibre Channel link can sustain for a particular application.

- ▶ Disk bytes/second

This counter tells you what the current disk subsystem is able to sustain for this particular application. This value can also be compared to the maximum throughput obtained from Figure 11-20 on page 286 to determine whether multiple links should be used to reach the target level of throughput demanded for the target number of users.

As well as adding a PCIe host adapter, you can improve performance by balancing workload over multiple storage controllers. Throughput nearly doubles for all transfer sizes when a second controller is used in the storage system, as shown in Figure 11-22.

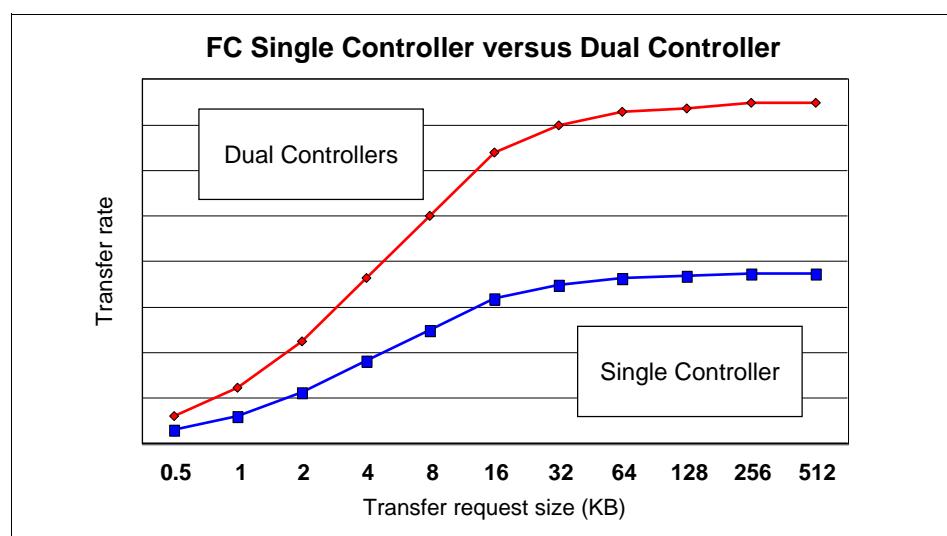


Figure 11-22 Comparing single versus dual controller throughputs

**Rules of thumb:**

- ▶ Double the number of users requires double the amount of disk I/O.
- ▶ Use Figure 11-20 on page 286 to determine the maximum sustainable throughput. If your expected throughput exceeds this value, add another RAID controller.
- ▶ Using a second RAID controller module doubles the throughput, provided no other bottleneck is created in the process.

The remainder of the challenges of optimizing Fibre Channel are similar to those present when configuring a standard RAID controller. Disk layout and organization, such as RAID strategy, stripe size and the number of disks, all affect performance of the IBM Fibre Channel RAID controller in much the same way that it does for ServeRAID. You can use the same techniques that you used to determine these settings for ServeRAID to optimize the IBM Fibre Channel RAID controller solution.

Using a large number of drives in an array is the best way to increase throughput for applications that have high I/O demands. These applications include database transaction processing, decision support, e-commerce, video serving, and groupware such as Lotus Notes and Microsoft Exchange.

## 11.7 Disk subsystem rules of thumb

A performance relationship can be developed for the disk subsystem. This relationship is based upon the RAID strategy, number of drives, and the disk drive model. Table 11-5 lists disk subsystem rules of thumb.

*Table 11-5 Disk subsystem rules of thumb*

Performance of this configuration	Is equivalent to...
RAID-0	33% to 50% more throughput than RAID-1 (same number of drives)
RAID-1E	33% to 50% more throughput than RAID-5 (same number of drives)
RAID-5E	10% to 20% more throughput than RAID-5
RAID-5	30% to 50% more throughput than RAID-6
Doubling number of drives	50% increase in drive throughput (until disk controller becomes a bottleneck)
One 10,000 RPM drive	10% to 50% improvement over 7200 RPM drives (50% when considering RPM only, 10% when comparing with 7200 RPM drives with rotational positioning optimization)
One 15,000 RPM drive	10% to 50% improvement over 10,000 RPM drives





# Network subsystem

Because all server applications provide services to users who are connected through a network, the network subsystem and the network itself play a crucial role in server performance from the user's point of view.

This chapter discusses the following topics:

- ▶ 12.1, “LAN operations” on page 294
- ▶ 12.2, “Factors affecting network controller performance” on page 300
- ▶ 12.3, “Advanced network features” on page 316
- ▶ 12.4, “Internet SCSI (iSCSI)” on page 339
- ▶ 12.5, “New trends in networking” on page 345

**Note:** Throughout this book, B represents bytes and b represents bits:

- ▶ MB, MBps, KB, KBps refer to bytes (megabytes per second, for example)
- ▶ Mb, Gb, Mbps, Kbps refer to bits (megabits per second, for example)

## 12.1 LAN operations

The Ethernet protocol is the dominant networking technology in most of today's local area networks (LANs). Competing technologies such as token-ring, FDDI and asynchronous transfer mode (ATM) are no longer widely used for LANs.

ATM is still a prevalent choice for running high-bandwidth networks that span large areas. With current transfer rates of more than 10 Gbps, many Internet service providers and telecommunications companies have implemented ATM as their backbone technology.

With its low cost, ease of use and backward-compatibility, Ethernet is the number one choice for most LANs today. Features such as Quality of Service (QoS) that used to be in the domain of newer technologies such as ATM are incorporated into the Ethernet standard as the technology evolves. And today's speed on Ethernet networks also offers new ways of using this technology.

The network adapter is the pathway into the server. All requests to the server and all responses from the server must pass through the network adapter. Its performance is key for many server applications.

Most LAN adapters are comprised of components that perform functions related to the following:

- ▶ Network interface/control
- ▶ Protocol control
- ▶ Communication processor
- ▶ PCI bus interface
- ▶ Buffers/storage

The relationship between the network client workstation and the server is done with a *request/response protocol*. Any time a network client wants to access data on the server, it must issue a request. The server then locates the data, creates a data packet, and issues the transmit command to the server LAN adapter to acquire that data from memory and transmit it to the network client workstation (response).

The “network” itself, and its components (switches, routers, and so on) is intelligent enough to handle the transport of the information from and to the requester and receiver. This request/response relationship is a fundamental characteristic that also affects server performance.

**Note:** In this chapter, we refer to data chunks transmitted over the network by different names, depending on their respective position in the OSI reference model. These are:

- ▶ Ethernet frames (Layer 2)
- ▶ IP datagrams (Layer 3)
- ▶ TCP segments (Layer 4)

Although the term *packet* is often used for IP datagrams, we use this term for any data type that resides on Layer 3 or above.

The LAN adapter performs two fundamental operations for each packet:

- ▶ The LAN adapter communication processor must execute firmware code necessary to prepare each packet to be moved to or from system memory.

This is called *adapter command overhead*. Every LAN adapter has a limit on the number of packets per second that it can process. Because it takes a certain amount of time to execute this code, the adapter packet throughput rate is reached when the onboard communication processor reaches 100% utilization.

In addition to adapter communication processing, each packet that is received or sent by the server requires device driver, TCP/IP, and application processing. Each of these components requires server CPU utilization. Often, one or more of these components can cause packet rate bottlenecks that can result in high CPU utilization and high interrupt frequency on the server.

These bottlenecks can occur when a server is sending or receiving a high percentage of small packets (that is, packets of less than 512 bytes). In this case, LAN utilization might be quite low because the packets are small in size and the amount of data traversing the LAN is small.

**Note:** Never assume that you do not have a LAN bottleneck by simply looking at LAN sustained throughput in bytes per sec. LAN adapter bottlenecks often occur at low LAN utilization but at high sustained packet rates. Observing packets per second often yields clues to these types of bottlenecks.

- ▶ The LAN adapter must also act as a PCI bus master and copy all packets to or from memory. The speed at which this can be sustained determines the adapter DMA throughput capability.

When a LAN adapter is moving large amounts of data per packet (approximately 1000 bytes or more), the amount of time spent processing firmware commands, device driver, and TCP/IP commands overhead is small

compared to the time necessary to copy the large packet to or from server memory. Consequently, it is the DMA performance of the adapter, not the onboard communication processor, that limits packet throughput of the adapter transferring large packets.

This is also true for the CPU time of the server processors. With large-size packets, the server CPU must spend the majority of time copying data from device driver buffers to TCP/IP buffers and from TCP/IP to the file system buffers. Because of all the server copies, sustained throughput is often determined by the speed of the front-side bus and memory, or simply, the speed at which the server CPUs can move data from one buffer to the next; in other words, how many bytes per second the adapter can move to or from memory.

### 12.1.1 LAN and TCP/IP performance

With the Internet, TCP/IP has replaced most other protocols as the network protocol of choice for networks of all sizes. Windows and Linux servers use TCP/IP as their default network protocol. TCP/IP enables server software vendors to standardize on one common protocol instead of having to support three or four protocols in their products.

Although most applications still support other protocols in addition to TCP/IP, support for these protocols is gradually phasing out and features are often only supported with TCP/IP. For example, both Lotus Domino clustering and Microsoft clustering require TCP/IP to function properly.

TCP/IP processing accounts for a large amount of overhead in network adapter operations, and it is often the limiting factor when it comes to network throughput. To understand this process, let us take a look at the processing that takes place during TCP/IP operations.

Figure 12-1 shows the primary components that make up the server. More specifically, it shows the PCI bridges in a dual-peer PCI bridge architecture with a single LAN adapter installed in the right PCI bus segment.

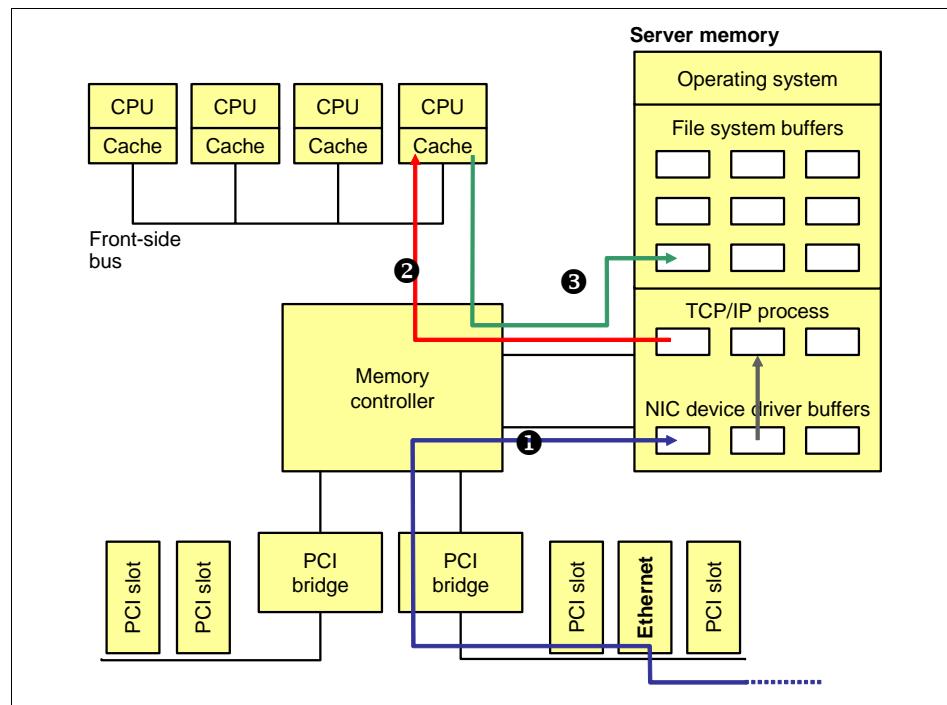


Figure 12-1 Internal data path for TCP/IP

The flow of traffic is as follows:

1. An application that executes in the network client workstation makes a request to the server for data. The networking layer in the client workstation builds a network frame with the address of the target server. The LAN adapter in the client workstation sends the frame as a serial data stream over the network using the address of the target server.
2. The frame arrives at the server LAN adapter as a serial bit stream. It is validated for correctness by the protocol control logic and assembled into a frame in adapter storage.
3. An interrupt or handshake is generated by the LAN adapter to gain service from the server CPU. The server CPU executes the LAN adapter device driver, which responds to the interrupt by building a receive-frame command in a buffer in server memory. The receive-frame command includes a server memory address (destination address for the incoming frame) that tells the LAN adapter where to store the received frame.

4. The LAN adapter device driver instructs the LAN adapter to gain access to the PCI bus and retrieve the receive-frame command for processing. The LAN adapter gains access to the PCI bus and copies the receive-frame command from server memory to the LAN adapter's onboard buffer. The LAN adapter's communication processor parses the receive-frame command and begins to perform the receive-frame operation.
5. The LAN adapter gains access to the PCI bus (bus master) and uses the destination (server memory) address as the location to store the received frame. The bus master then moves the received frame's contents into the server's receive-buffer memory for processing using direct memory access (DMA).

This part of the process is shown by line number one (❶) in Figure 12-1 on page 297, which points to the NIC device driver buffer. This step is where the speed of the PCI bus really matters. A higher speed PCI bus enables faster transfers between the LAN adapter and server memory. However, after the data is transferred into the buffer, the PCI bus is no longer used.

6. After the data arrives into the buffer, the adapter generates an interrupt to inform the device driver that it has received packets to process.
7. The IP protocol and TCP process the packet to complete the transport protocol process.
8. After the TCP/IP protocol processing is finished, the NDIS driver sends an interrupt to the application layer that it has data for the server application. The NDIS driver copies the data from the TCP/IP buffer into the file system or application buffers.

In Figure 12-1 on page 297, this part of the process is shown by the line pointing into the CPU (❷) and the line pointing into the file system buffers (❸). Each of these copies is also executed by a server CPU.

**Note:** The Network Driver Interface Specification (NDIS) driver is used by Microsoft operating systems. In Linux, there are drivers that perform a similar task, but they use other standards for drivers like the Uniform Driver Interface (UDI). The way they work at a high level is very similar, but this description is mainly based on the Microsoft Windows behavior.

As shown in Figure 12-1 on page 297, TCP/IP requires up to three transfers for each packet. A server CPU executes all but one of these transfers or copies. Remember that this data must travel over the front-side bus up to two times, and the PCI bus is used for one transfer. A server that is moving 75 MBps over the LAN is doing three times that amount of traffic over the memory bus. This transfer rate is over 225 MBps and does not include the instructions that the CPUs must fetch and the overhead of each packet.

Some network adapters that have advanced network features support will greatly improve performance here. For more information, see 12.3, “Advanced network features” on page 316.

Frames are transmitted to network clients by the server LAN adapter in a similar manner. The process is as follows:

1. When the server operating system has data to transmit to a network client, it builds a transmit command that is used to instruct the LAN adapter to perform a transmit operation.

Included inside the transmit command is the address of the transmit frame that has been built in server memory by the network transport layer and initiated by the operating system or server application. The transmit frame includes data that was requested by the network client workstation and placed in server memory by the server operating system or application.
2. After receiving the transmit command, the LAN adapter gains access to the PCI bus and issues the address of the transmit frame to the PCI bus to access server memory and to copy the transmit frame contents into its onboard buffer area.
3. The communication processor on the LAN adapter requests access to the network.
4. When network access is granted, the data frame is transmitted to the client workstation.

This explanation is an oversimplification of a very complex process. It is important, however, to gain a high-level understanding of the flow of data from the LAN adapter through the server and the contribution of TCP/IP overhead to server performance. Much of the TCP/IP processing has been omitted, because that processing is complex and beyond the scope of this book.

For more detail about Windows Server 2003 TCP/IP operation, see the Microsoft white paper *Microsoft Windows Server 2003 TCP/IP Implementation Details*, which is available from:

<http://www.microsoft.com/downloads/details.aspx?FamilyID=06c60bfe-4d37-4f50-8587-8b68d32fa6ee&displaylang=en>

The most important point to remember is that data is passed from the LAN adapter to a device driver buffer by LAN adapter bus master transfers. These transfers consume little, if any, server CPU cycles because they are performed entirely by the LAN bus master adapter. After the LAN adapter has moved data into device driver buffers, the transport protocol stack processes the data. After this data is copied from the server transport buffers into the file system memory, the CPU processes the data again.

These CPU copies of the data can consume a significant amount of server CPU utilization and front-side bus bandwidth, and can create bottlenecks within the server that often limit LAN throughput scalability.

## 12.2 Factors affecting network controller performance

There are a number of aspects of a server's configuration that will affect the potential data throughput of a Gigabit Ethernet controller. The factors discussed here are:

- ▶ 12.2.1, "Transfer size" on page 300
- ▶ 12.2.2, "Number of Ethernet ports" on page 304
- ▶ 12.2.3, "Processor speed" on page 310
- ▶ 12.2.4, "Number of processors or processor cores" on page 311
- ▶ 12.2.5, "Jumbo frame" on page 312
- ▶ 12.2.6, "10 Gigabit Ethernet adapters" on page 313
- ▶ 12.2.7, "LAN subsystem performance summary" on page 314

### 12.2.1 Transfer size

It is often stated that a Gigabit Ethernet controller can transfer 1000 Mbps (bits) or 100 MBps (bytes). Depending on the type of traffic that needs to be transmitted or received, this transfer rate can actually be more or significantly less, even under ideal conditions.

The amount of data that can be transferred over an Ethernet connection depends greatly on the average size of the data packets that need to be transmitted. Because the size of an Ethernet packet is fixed and the CPU and network adapter will have to process each packet regardless of the amount of payload data it carries, small data sizes can overload the CPU subsystem before the maximum theoretical throughput can be reached.

By comparison, a full duplex Gigabit Ethernet connection can transmit and receive data at the same time, allowing for transmissions of more than 100 MBps.

Figure 12-2 shows the data throughput and corresponding CPU utilization for a dual processor server with a single Gigabit Ethernet controller. Throughput and CPU utilization are measured for increasingly larger application I/O transfer sizes.

Applications with small transfer sizes include instant messaging and simple file/print. Examples of applications with large transfer sizes include a database and virtualization software.

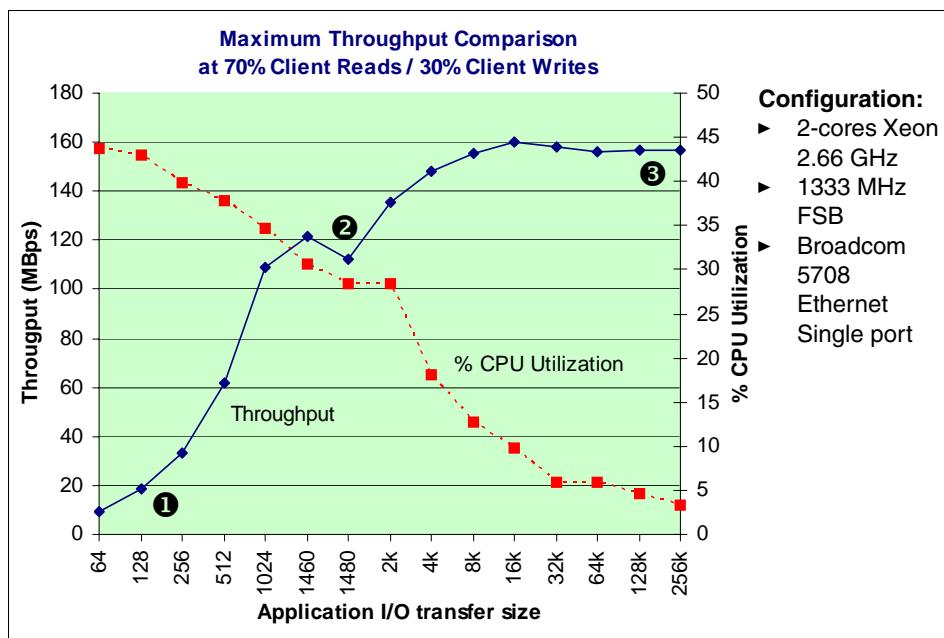


Figure 12-2 Network throughput dependency on transfer size

The left Y axis in Figure 12-2 represents throughput in MBps and is shown as the solid line. The throughput plot is the aggregate throughput that is sustained by the server. Throughput continues to increase to about 160 MBps. This increase is because 160 MBps is about the maximum throughput that we can expect for a single Gigabit Ethernet adapter.

The right Y axis is the overall CPU utilization. CPU utilization when packet sizes are small is high and can be the source of a bottleneck. CPU utilization for large packet transfers is much lower.

To understand LAN throughput behavior, we examine this data more closely, starting with small packet sizes.

**Note:** The mix of traffic that passes through the adapter has a significant effect on maximum throughput. At a 50-50 ratio of client read-write, throughput would approach 225 MBps (which is the maximum possible throughput of the adapter), and a 100-0 read-write mix yields a maximum of about 110 MBps. However, a more realistic load of 70-30 client read-write (as shown in Figure 12-2), throughput peaks at approximately 160 MBps.

## Small packet sizes

In the case of a very small packet, 128 bytes (❶ in Figure 12-2 on page 301), we see that the server sustains about 18 MBps of total throughput. Any application that sends data in small packets cannot expect to scale because the red dotted line (the right Y axis) shows that two CPUs are at high utilization because it is processing application, TCP/IP, and device driver packet overhead as fast as the small packets can be received and sent. Throughput at 128 byte packets is dominated by adapter and server CPU processing. These components limit throughput to about 18 MBps.

One MBps is equal to  $1024 \times 1024$  bytes or 1,048,576 bytes per second. At 18 MBps, the server is supporting a throughput of  $18 \times 1,048,576 = 18,874,368$  bps. If each packet is 128 bytes, then the server is handling 147,456 packets per second. The worst case scenario is that each packet requires a server interrupt to receive the command and one server interrupt to receive or transmit the data, so this server might be executing over  $147,456 \times 2 = 294,912$  interrupts per second, which is a massive overhead.

Most vendors optimize the LAN device driver to process multiple frames per interrupt to reduce this overhead. However, in many cases, it is not unreasonable to see servers that execute many thousands of interrupts per second. Assuming that all is operating correctly, when you see this level of interrupt processing, you can be fairly sure that the server is processing many small network packets.

Usually, the only solution is to obtain a newer, optimized NIC driver capable of servicing multiple packets per interrupt or upgrade to a significantly faster CPU. Care should be exercised when upgrading to a faster CPU because system implementation of memory controller and PCI bus bridge will often limit how fast the CPU can communicate with the PCI LAN adapter. Often this latency can be a significant component of LAN device driver processing time. A faster CPU has little effect on how quickly the CPU can address the PCI LAN adapter on the other side of the memory controller. This bottleneck will usually limit LAN throughput gains when upgrading to faster CPUs.

## Larger packet sizes

As we work our way to the right of the chart to study performance for larger packet sizes (❸ in Figure 12-2 on page 301), notice that throughput for the single LAN adapter increases to a maximum of about 160 MBps. In full-duplex mode, this is about the most we should expect from a single 100 MBps Ethernet NIC. In general, when it comes to network performance, the actual amount of data that can be sent across the network depends greatly on the type of application.

For example, consider a chat server application that typically sends and receives small messages. Sending a chat message such as “Hey Lou, what are you doing for lunch?” is never going to be as large as 8 KB in size. As a result, a chat server application that uses TCP/IP and Ethernet would not scale throughput beyond a single 100 MBps Ethernet NIC because the single NIC would consume all the CPU power most modern servers can provide executing the application, TCP, IP, and device driver overhead. This is not necessarily undesirable. After all, 8 MBps represents a large number of chat messages. However, do not make the mistake of expecting the same performance improvement or scaling for all application environments.

## Inefficient frame payloads

Note the dip in throughput at the 1480 byte packet size (❹ in Figure 12-2 on page 301). This drop in throughput is related to the Ethernet protocol. Standard Ethernet adapters have maximum frame size of 1518 bytes. This is 1500 bytes of maximum transmission unit (MTU) + 14 bytes of Ethernet header + 4 byte Ethernet CRC. About 40 of these bytes must be used for IP and TCP addressing, header, and checksum information. This leaves 1460 bytes for data to be carried by the packet.

**Note:** The original Ethernet standard defined the maximum frame size as 1518 bytes. This was later extended to 1522 bytes to allow for VLANs. Both variants can carry a maximum payload of 1500 bytes.

A 1480 byte request overflows the 1460 bytes of data payload of a single packet by 20 bytes. This forces the transport to use two different packets for each 1480 bytes of data being requested; 1460 bytes fill one full packet and the remaining 20 bytes are transmitted in a second packet to complete each 1480 byte request. This requires two trips down to the LAN adapter. The overhead of the second 20-byte frame is what causes the drop in throughput because the server CPU must now work twice as hard to send 1480 bytes, as compared to sending 1460 bytes that fit into a single Ethernet packet.

These numbers translate to a production environment. If you are building a Web server, for example, keeping the size of your images to an integral multiple of 1460 bytes in size can maximize server throughput because each binary object

could fit in a single or multiple full Ethernet frames. This increases throughput of the server because the Ethernet connection is running at maximum efficiency.

### **Efficiency versus utilization**

It is very common to become confused between efficiency and utilization. We often think that if a CPU or other hardware component is heavily used, the it is performing very efficiently, but this may not be the case.

Efficiency refers to the speed or the time it takes for a task to be done. Utilization refers to the use or non-use of one resource. In a network operation, having the CPU busy does not necessarily mean that the CPU is efficient at its work, because probably it is spending computing cycles performing many tasks related to the network request that may not be the network request itself.

#### **12.2.2 Number of Ethernet ports**

Adding Ethernet controllers or using more ports in a single controller is an effective way of increasing network throughput. This increase can prove to be beneficial as long as the bottleneck is the network adapter and not another subsystem.

The performance benefit of adding ports depends very much on the packet size being used. For large packets, throughput scales very well. For small packets, however, the benefit is less. Figure 12-3 on page 305 shows that going from one port to two ports doubles the throughput.

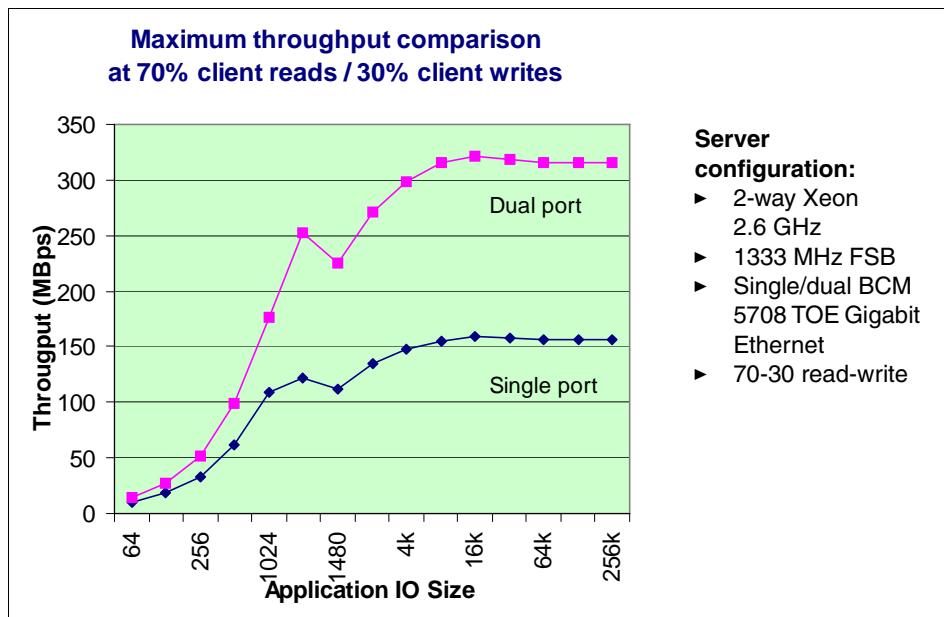


Figure 12-3 Throughput comparison of one Ethernet port versus two ports

Assuming the CPU is not a bottleneck (front-side bus is rarely a bottleneck in modern servers), adding 4 ports increases the throughput in MBps, as shown in Figure 12-4.

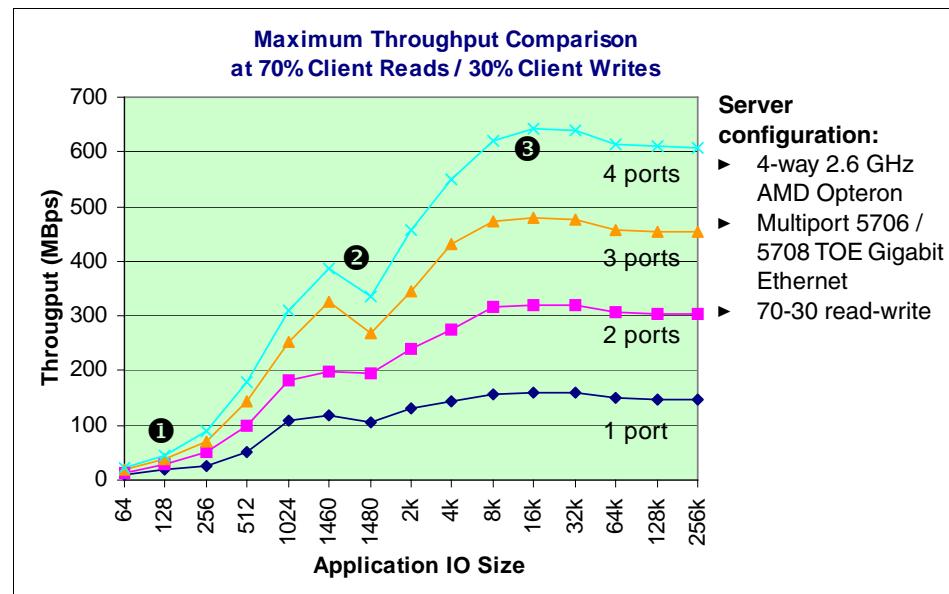


Figure 12-4 Throughput effect of adding more Ethernet ports

As you increase the number of Ethernet connections, CPU utilization also increases proportionally, as shown in Figure 12-5. As explained in “Small packet sizes” on page 302, CPU utilization is highest with small packet sizes. However, in this example CPU utilization is only a bottleneck with four Ethernet ports and small packet sizes.

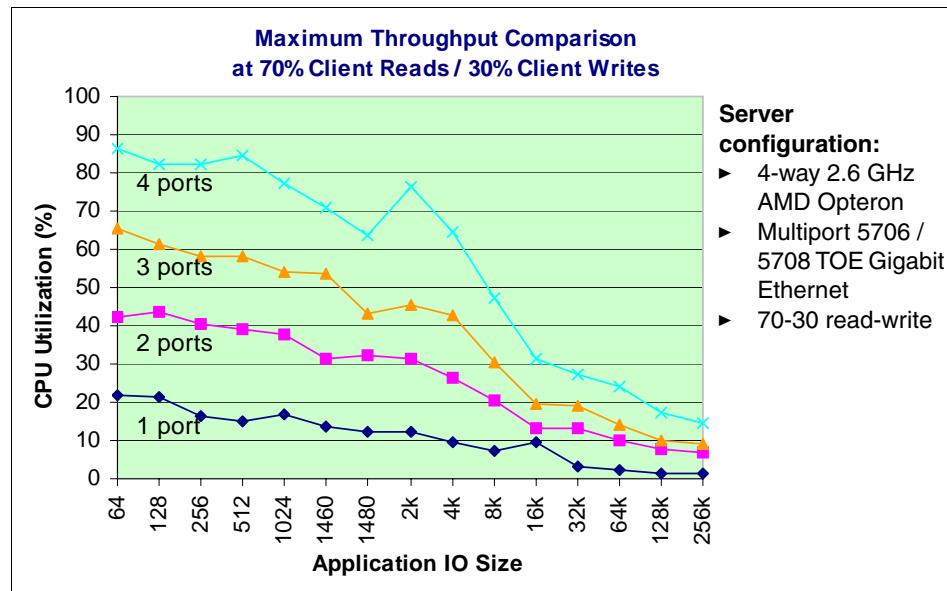


Figure 12-5 CPU utilization effect of adding more Ethernet ports

### Small transfer sizes

Examine the chart at the 128 byte transfer size (❶ in Figure 12-4 on page 306). Regardless of the number of network adapters installed, throughput is always under 45 MBps. For small transfer sizes, the CPU is often the bottleneck. However, notice that in this instance, the CPU is not the bottleneck (Figure 12-5 shows the CPU is not a bottleneck for 1, 2, or 3 ports).

In this configuration, the Ethernet adapters have TCP/IP Offload Engine (TOE) capability and are operating in that mode. In this case, the limitation is simply the TOE processor on the network card. This is positive, as far as CPU is concerned. If this configuration did not use TOE, the CPU utilization would most likely be the bottleneck.

## Larger transfer sizes

With applications that use large transfer sizes, however, network throughput scales almost linear as additional network cards are installed in the server. Throughput for an application that uses 16 KB packets would scale throughput very nicely up to four Ethernet adapters (see ❸ in Figure 12-4 on page 306).

The dip in throughput as described in “Inefficient frame payloads” on page 303 is more apparent in ❷ in Figure 12-4 on page 306, especially with four network cards.

## Linear scaling

As mentioned earlier, the increase in throughput in this configuration is linear across all transfer sizes when no other subsystems are the bottleneck. The same data shown in Figure 12-4 on page 306 plotted differently shows the linear scaling from 1 port through to 4 ports, as illustrated in Figure 12-6 on page 309.

At the smallest transfer sizes, four ports gives a 2.5x improvement in throughput. However, at the largest transfer sizes, it is a 4x improvement. The scaling is typically less at small transfer sizes either because the TOE processor on the network adapter is the bottleneck (as is the case here) or the server’s CPUs are the bottleneck (as is often the case with non-TOE network controllers and less CPU capacity).

Another factor when considering network scaling is the new capabilities of operating systems such as Windows Server 2003 with Receive-side scaling (RSS) as described in 12.3.5, “Receive-side scaling” on page 334.

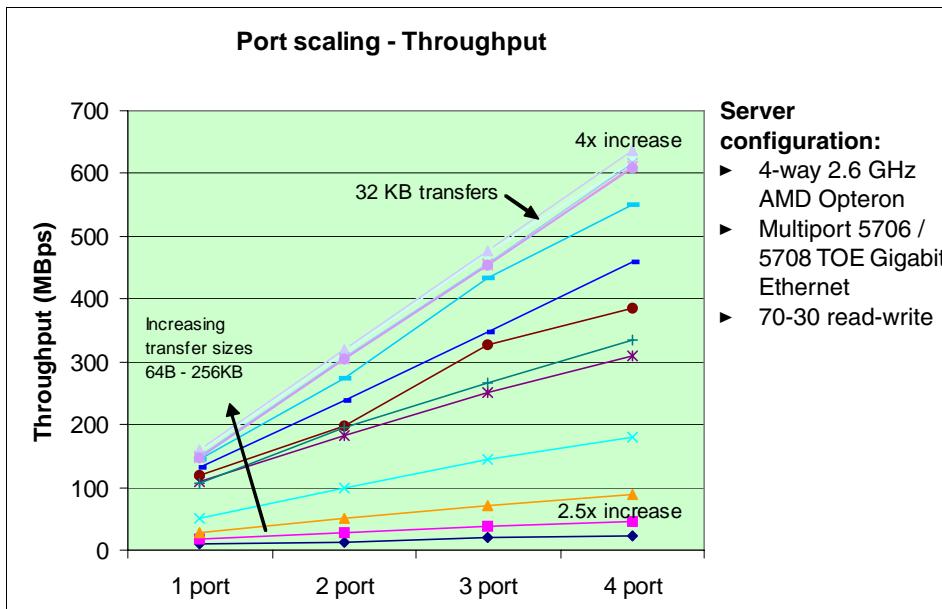


Figure 12-6 Throughput scales between 2.5x and 4x, going from 1 port to 4 ports

If you compare this four-way x3755 AMD Opteron system with an older 2-way 3.2 GHz Xeon system (compare the red throughput line in Figure 12-7), you can see that scaling is linear up to three Gigabit ports but CPU utilization peaks and limits any further improvement. This shows that the CPU subsystem plays an important part in the performance subsystem.

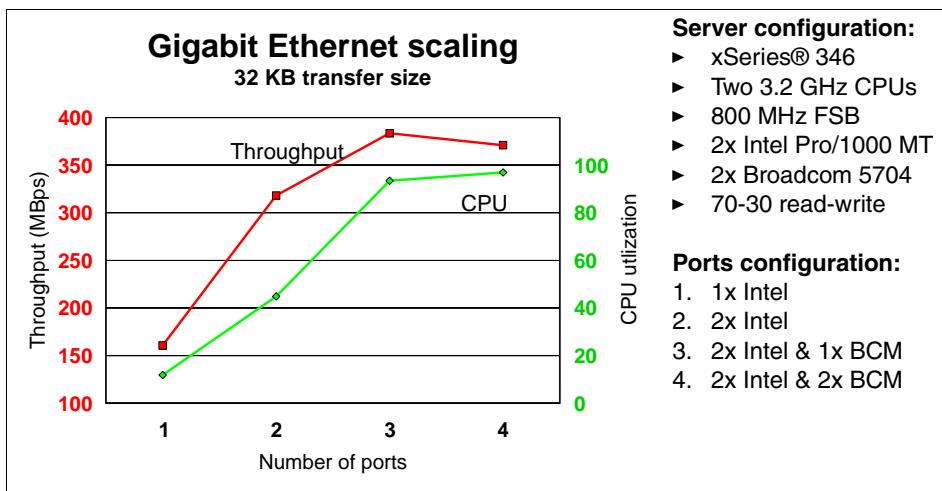


Figure 12-7 Benefit of adding multiple adapters (70-30 read-write mix)

### 12.2.3 Processor speed

Now that we have taken a look at existing LAN subsystem performance, we can discuss more in detail the contribution of the server to sustained LAN throughput.

As explained in 12.1.1, “LAN and TCP/IP performance” on page 296, sustained throughput for Ethernet running TCP/IP is largely dependent on how fast the processors can do buffer copies (*bcopies*) of data between device driver, TCP/IP, and file system buffers. The bcopy speed is dependent largely upon the speed of the front-side bus and the speed of main memory. The faster the bus and memory, the faster the processors can move the data.

When comparing processor impact on performance, and removing the front-side bus and adapter as a bottleneck, as a rule of thumb, the expected improvement in performance is half the difference in the speed increase in the processor.

The larger the block size, the less the CPU is a bottleneck. So, the benefit of the processor speed is diminished. In Figure 12-8 on page 311 you see the effect of a higher CPU speed on network throughput. The chart shows a comparison of a 2.66 GHz processor versus a 3.0 GHz processor (a 12% increase) that results in about a 5% increase in throughput at low transfer sizes.

A performance benefit is observed at smaller block sizes because the CPU is the bottleneck and the higher clock speed means it is able to process more headers per second. At larger block sizes, performance is almost identical because the CPU is copying data more and is not the bottleneck.

**Tip:** The 2:1 rule of thumb (a 2% increase in CPU speed results in a 1% increase in network throughput) only applies when the front-side bus speed is constant, and it only applies for smaller transfer sizes where CPU is the bottleneck.

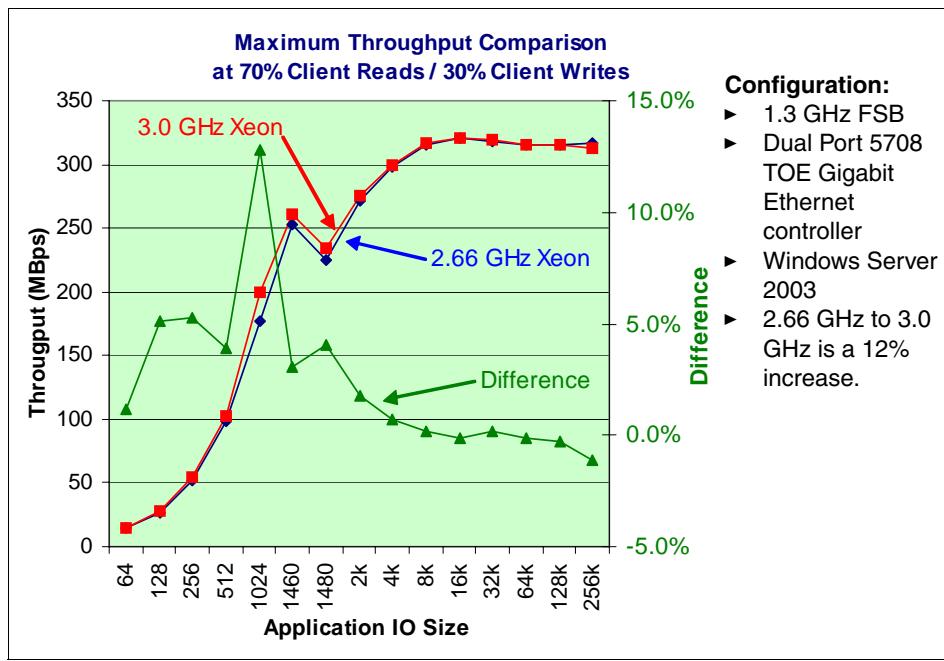


Figure 12-8 Processor speed scaling (70-30 read-write mix)

#### 12.2.4 Number of processors or processor cores

Adding a second processor will not improve network throughput in most cases, unless the number of network ports is also added. This is because NICs are not able to use multiple processors.

**Note:** Microsoft offers the Scalable Network Pack with a receive-side scaling (RSS) feature. If this software is installed, a NIC is no longer associated to a single CPU. If the CPU is the bottleneck on a multi-processor server, RSS improves the performance dramatically. For more information, see 12.3.5, “Receive-side scaling” on page 334.

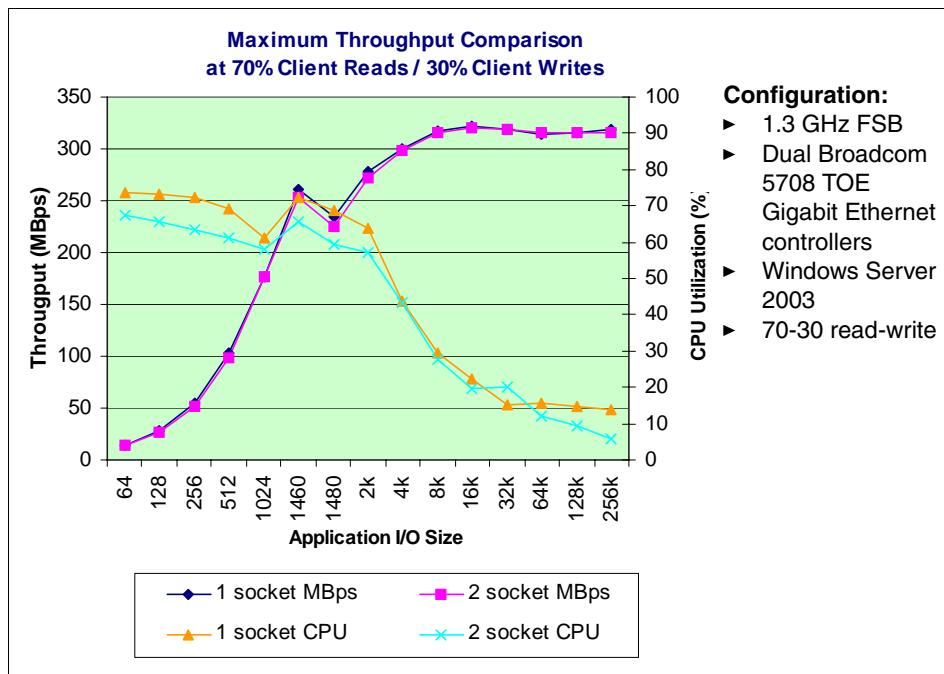


Figure 12-9 Throughput on a one-CPU and two-CPU system (70-30 read-write mix)

### 12.2.5 Jumbo frame

The term *jumbo frame* refers to an Ethernet frame in excess of the standard 1500-byte size. Bigger packets mean bigger payloads and consequently fewer packet headers per second to process. Using 9000-byte frames can increase network packet throughput, while simultaneously decreasing CPU utilization.

Jumbo frame technology is not strictly an alternative to TOE and I/OAT, described in 12.3, “Advanced network features” on page 316, because TOE and I/OAT do not offload processing onto the network controller. Both TOE and I/OAT provide throughput equivalent to what jumbo frames offers, and CPU offloading in excess of what jumbo frames offers. However, for those servers that *lack* TOE or I/OAT controllers, jumbo frames can offer the following benefits:

- ▶ The transmit and receive throughput can improve by up to 5% due to better packing of frames.
- ▶ Jumbo frames can reduce CPU utilization, as compared with standard frames, by transmitting or receiving large chunks of data without requiring segmentation and reassembly up and down the TCP/IP stack.

For customers who have end-to-end support for jumbo frames in their infrastructure, it is available in specific IBM System x servers, including those that use Broadcom controllers.

Note that the requirement for jumbo frames is an end-to-end support for the technology throughout the entire infrastructure. If you want to use jumbo frames, every network controller, switch, storage device, and other network element that is connected through Ethernet needs to have support for jumbo frames.

Jumbo frames improve throughput by about 5% at larger blocks due to better packing of frames, and they decrease CPU utilization by transferring large blocks. The decrease in CPU utilization is the main reason why jumbo frames are deployed.

### 12.2.6 10 Gigabit Ethernet adapters

10 Gigabit Ethernet adapters are the latest iteration of network adapters that deliver increased network throughput. They follow the IEEE 802.3ae standards and vary slightly from previous Ethernet adapters because they only operate in full duplex mode, thus making collision-detection protocols unnecessary. Adapters are available with optical fiber or RJ45 copper connectivity.

10 Gigabit Ethernet is capable of operating at in excess of 2.5 GBps during burst mode. This, of course, presumes that the other subsystems are capable of supporting that throughput; at this rate the old PCI bus, front-side bus, and memory bus, as well server CPUs, would be saturated. If you consider deploying 10 Gbps adaptors in old machines, you should consider the impact on the overall performance and eventually upgrade the servers to newer ones that implement faster buses and CPUs.

Table 12-1 on page 314 shows the impact of 10 Gigabit Ethernet on various subsystems. This assumes that the sustained throughput will be 80% of burst mode and full-duplex. When reviewing this table, it is important to remember that the first data transfer to memory is DMA and, therefore, does not have to travel across the front-side bus. There are four remaining data moves across the front-side bus to move the data through to the application.

There are also two rows in the table for data moving across the front-side bus. This shows the difference between using a TOE-enabled adapter and a non-TOE adapter. The number of data transfers to memory remains the same, but the data transfers across the front-side bus will be reduced. All 10 Gigabit Ethernet adapters should be TOE-compliant.

*Table 12-1 10 Gigabit Ethernet throughputs*

Subsystem	Burst Mode	Sustained
PCI adapter	2.24 GBps	1.79 GBps
Front-side bus (without TOE)	8.96 GBps	7.16 GBps
Memory bus	11.2 GBps	8.96 GBps
Front-side bus (with TOE)	4.48 GBps	3.58 GBps
Front-side bus speed minimums	1200 MHz	1200 MHz

### **12.2.7 LAN subsystem performance summary**

The following key points are raised in this section:

- ▶ A server with a bottleneck will run only as fast as the bottlenecked component will allow, no matter how fast the other parts run.

For example, a 64-bit 66 MHz PCI bus can burst 533 MBps. A single-port Gigabit Ethernet adapter at a sustained throughput of 140 MBps to 160 MBps is far from saturating that PCI bus. Even a dual-port Gigabit Ethernet adapter that can sustain 300 MBps to 320 MBps will not saturate the PCI bus.

- ▶ Do not test server throughput by performing a single user file copy.

LAN adapters function using a request-response protocol. This means the client makes a request for data from the server, and the server responds by sending data. In most cases, applications do not flood the server with requests. They typically wait for a response before sending the next request.

Therefore, a single client will almost never load the network or server to its maximum throughput. It takes many clients to show maximum network and server throughput. Therefore, do not run a copy command from your workstation over an Ethernet or any network to the server and expect to see wire speed.

- ▶ Although the PCI, memory, and front-side bus are capable of supporting the sustained throughput of the Gigabit Ethernet adapter, other bottlenecks might prevent this maximum from being reached.
- ▶ Applications that transfer data using small packet sizes will result in low throughput and high CPU overhead.

Applications that request small blocks of data require the LAN adapter processor to spend a larger percentage of time executing overhead code while the server processor is executing a high percentage of interrupts. Almost every LAN adapter is unable to sustain wire speed at small packet sizes (less than 512 bytes).

- ▶ Most NIC device drivers do not scale well with SMP, although this is changing

Only one thread can communicate with an adapter hardware interface at any one time, so having more than two CPUs does not usually produce significant improvements in throughput. Usually, the only solution for increasing performance for small frame environments is to obtain a newer, optimized NIC driver that is capable of servicing multiple packets per interrupt, or to upgrade to a significantly faster CPU.

Use care when you upgrade to a faster CPU, because system implementation of the memory controller and PCI bus bridge will often limit how fast the CPU can communicate with the PCI LAN adapter. Often this latency can be a significant component of LAN device driver processing time. A faster CPU has little effect on how quickly the CPU can address the PCI LAN adapter on the other side of the PCI bus. This bottleneck often limits LAN throughput gains for small packet environments when upgrading to faster CPUs. Solutions exist to help with SMP scalability. See 12.3.5, “Receive-side scaling” on page 334 for information.

However, the operating system improvements made in Microsoft Windows and Linux provide solutions to improve this scalability, and RSS on Windows and certain Extended Message Signaled Interrupts (MSI-X) implementations on Linux scale really well.

- ▶ Transfer size makes a significant difference to throughput.

LAN adapters are efficient when applications generate requests for large packets. Ethernet has a payload of 1448 bytes. Planning your objects to be no larger than 1448 bytes or an even multiple of 1448 bytes is best for Ethernet.

- ▶ Windows Server 2003/2008 uses packet segmentation to offload CPU utilization to the Ethernet adapter.

This feature offloads the segmentation of large packet requests onto the Ethernet adapter. Basically, this means that rather than have the server CPU segment large transfer requests into suitable size packets, the Ethernet adapter can accept one large transfer request and break the data up into multiple packets. This usually occurs for requests larger than 2 KB in size and explains why CPU utilization begins to decrease after the 2 KB size requests.

- ▶ Windows Server 2003/2008 and checksum offload

Most Gigabit Ethernet adapters support a Windows Server 2003 and 2008 function called *checksum offload*. When packet size exceeds a predetermined threshold, the adapter assumes offload of the checksum function from the server CPU. The checksum is a calculated value that is used to check data integrity.

The reason why the checksum is offloaded to the adapter for larger packets and not for small packets has to do with performance. At 128-byte packet sizes, a 100 MBps Ethernet adapter might be doing as many as 134,217,728

send and receive packets per second. The processing of checksums for such a high packet rate is a significant load on the LAN adapter processor, so it is better to leave that to the server CPU. As the packet size gets larger, fewer packets per second are being generated (because it takes a longer time to send and receive all that data) and it is prudent to offload the checksum operation to the adapter.

LAN adapters are efficient when network applications requesting data generate requests for large frames. Applications that request small blocks of data require the LAN adapter communication processor to spend a larger percentage of time executing overhead code for every byte of data transmitted. This is why most LAN adapters cannot sustain full wire speed for all frame sizes. In this case, the solutions are new applications (difficult and perhaps impossible) or additional subnetworks using multiple LAN adapters. Faster LAN adapter technology could be used, but the gains would be minimal. Faster LAN technology offers higher data rate, but when the frames are small, a greater percentage of time is spent in adapter overhead and not in data transmission.

## 12.3 Advanced network features

In many cases, it is not the NIC itself but other server components that could be the bottleneck. As a result, new advanced network technologies have been developed that free the CPU and bus subsystems from the heavy workload, thereby increasing performance. This section discusses some of those advanced network features.

**Note:** Technologies such as TOE and I/OAT provide the most benefit at large block sizes. In fact, at small block sizes, there is very little benefit. In addition, the benefit is generally a drop in CPU utilization, while the throughput is unaffected.

### 12.3.1 TCP offload engine

Processing TCP/IP traffic can consume significant network, memory, CPU and front-side bus resources. As described in 12.1.1, “LAN and TCP/IP performance” on page 296, a TCP/IP request requires multiple trips into and out of the CPU.

When processing TCP/IP requests, the CPU is involved in the following activities:

- ▶ Packet processing
- ▶ Data movement
- ▶ Context switching
- ▶ Interrupt processing

TCP offload engine (TOE) is a hardware-based solution that removes the burden of IP processing from the CPU on a server and moves it down to the NIC. Data is written directly to the NIC, and it handles the IP processing that is necessary to transmit and to receive in the network. TOE frees up CPU for small blocks, but there is still considerable overhead from interrupt processing and context switching on the CPU. For large blocks, the data movement is far more efficient because a single copy is completely eliminated.

Figure 12-10 on page 317 compares the dataflow of a traditional Ethernet transfer versus that using a TOE controller.

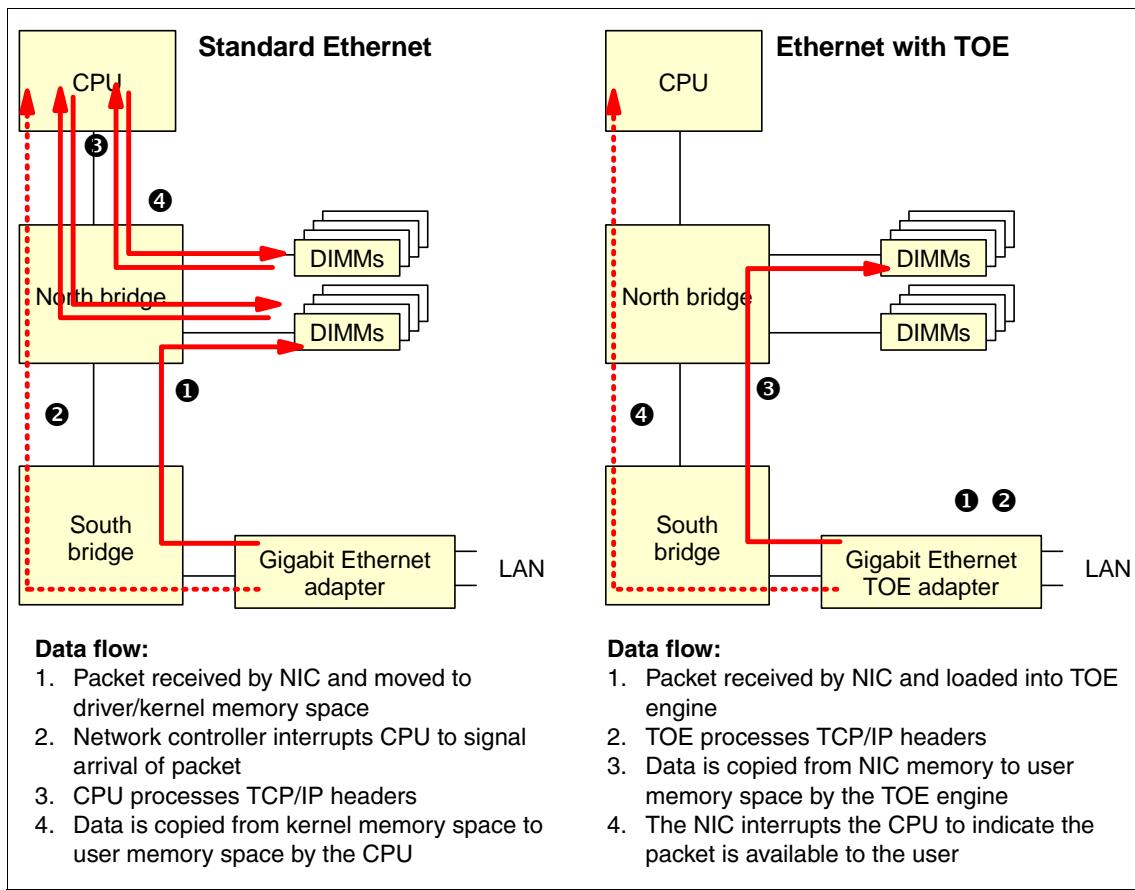


Figure 12-10 Comparing standard Ethernet data flow with that of a TOE-enabled system

TOE has two potential benefits: reduced CPU utilization, and improved network throughput. As you can see in the right side of Figure 12-10, there is very little CPU involvement in the TOE dataflow. The reduction in CPU utilization comes about because it no longer needs to perform the read/modify/write memory

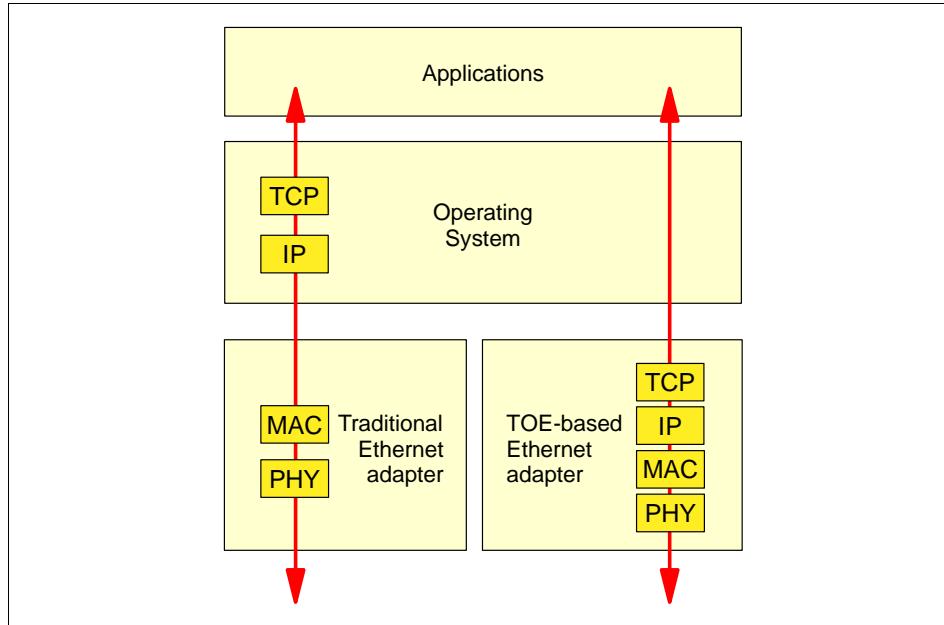
sequences that were shown in the standard Ethernet networking model (left side of Figure 12-10). This can be important in a server environment if there are other workloads that are restricted by lack of CPU processing power. TOE can improve network throughput by reducing the interaction required from the CPU. The more efficient data movement structure can allow for better flow speeds.

If a server has a Gigabit Ethernet adapter operating at a maximum throughput of 220 MBps, it needs the CPU to process approximately 150 000 memory copies per second (assuming 1460-byte packets -  $220M/1460$ ) to make the data available to the application without delay, not including the CPU cycles required to process error checking, TCP or checksum validation. This processing places a high load on network resources.

Using an adapter that supports TOE can dramatically reduce the impact on the CPU by changing the packet transfer model. TOE will reduce the number of copies down to a single DMA operation to the user space, provided the application posts buffers. Thus, TOE will result in 0 copies per second being needed.

Using these calculations, the number of I/O operations that the CPU would have to process for the Gigabit Ethernet adapter would be reduced from 613,000 I/Os per second to approximately 306,000 I/Os per second, effectively cutting in half the impact on the CPU to process the data.

Figure 12-11 shows the operations that are managed traditionally by the operating system as now managed by the TOE adapter.



*Figure 12-11 A TCP Offload Engine-enabled network adapter offloading operating system functions*

This technology decreases the workload that the CPU and front-side bus need to do, thereby enabling the server to dedicate potentially strained resources to other tasks. 10 Gigabit adapters are capable of flooding the CPU, PCI bus, memory, and front-side bus. Using a TOE adapter in this instance will help to reduce the impact on these subsystems.

Figure 12-12 compares the network throughput on a system with TOE enabled versus the same system with TOE disabled. The chart shows there are gains at low-medium transfer sizes (as much as 30% at 1024 bytes), but at large transfer sizes, TOE does not provide much gain in throughput.

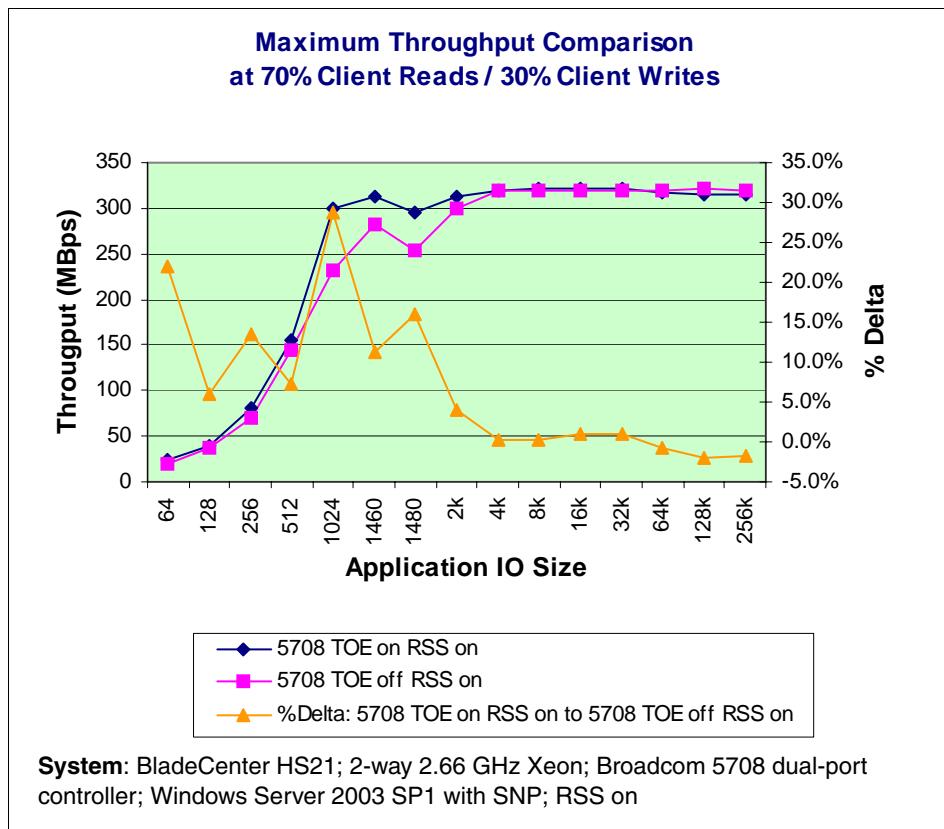


Figure 12-12 Throughput comparison - TOE enabled versus disabled

However, comparing CPU utilization in Figure 12-13, you can see that TOE lowers the demand on CPU capacity. The chart shows that there is a drop in CPU utilization at high transfer sizes.

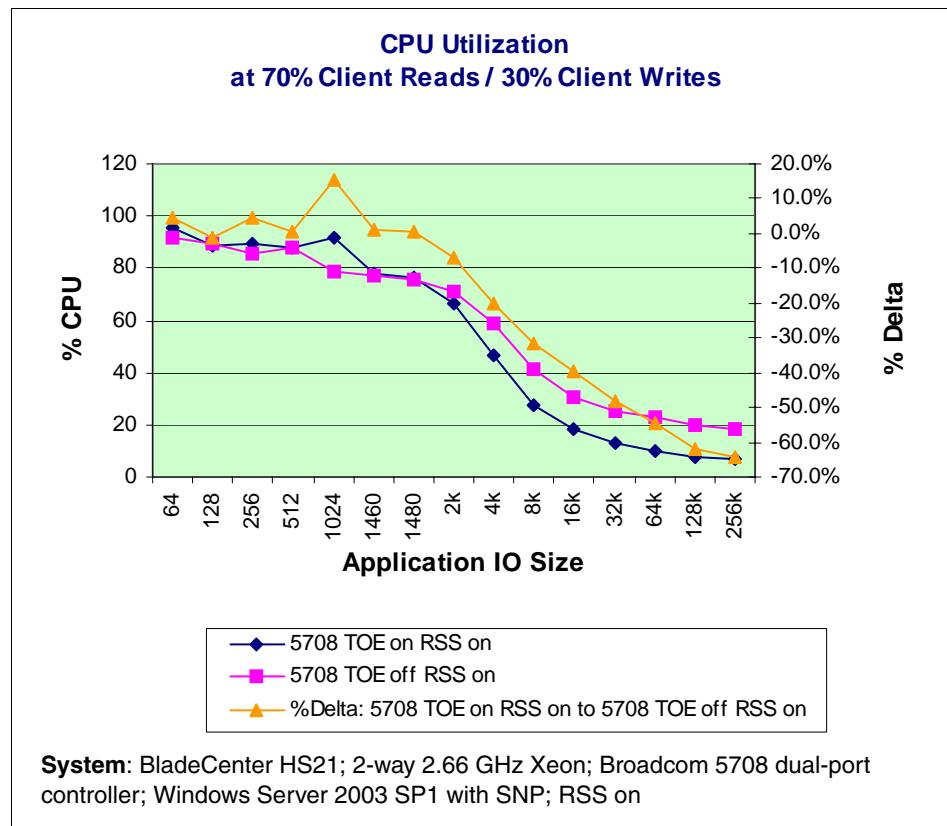


Figure 12-13 CPU utilization comparison - TOE enabled versus disabled

A better way of showing the effect on CPU utilization is to plot the *CPU efficiency* which is Throughput/CPU utilization, where the higher the number the better. This is shown in Figure 12-14. CPU efficiency is equal at low transfer sizes (comparing TOE enabled versus TOE disabled), while CPU efficiency is markedly higher at large transfer sizes when TOE is enabled.

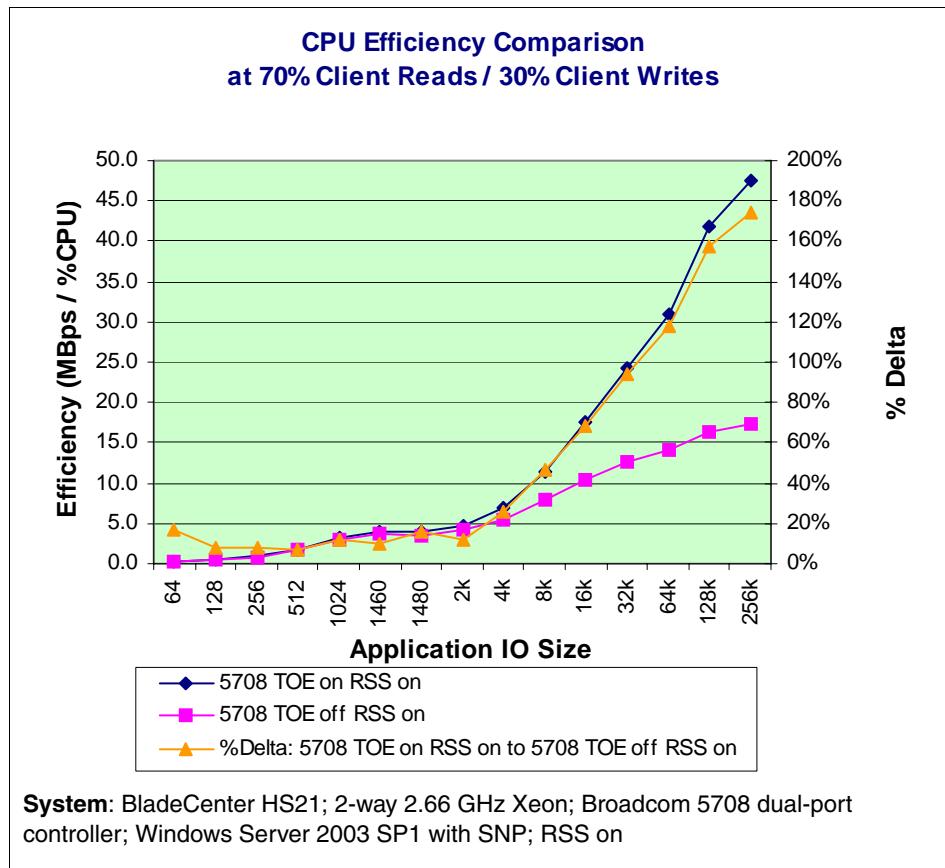


Figure 12-14 CPU efficiency comparison (70/30% client read/write)

The packet processing capability of a TOE engine is definitely less than a pair of Xeon processors. At small blocks, the number of packets per second is the highest, so TOE will generate a lower throughput level than the host CPUs. So, the host CPU essentially consumes proportionately higher CPU for the higher bandwidth that it delivers. Also, with small blocks, there is no data copying done, only protocol offload. As the block size increases, the TOE adapter will be able to offload data movement and protocol processing, so it can generate lower CPU utilization.

The preceding charts are for a client read/write ratio of 70/30, which is common for most production workloads. TOE is most effective, however, when client writes are at 100%, as shown in Figure 12-15.

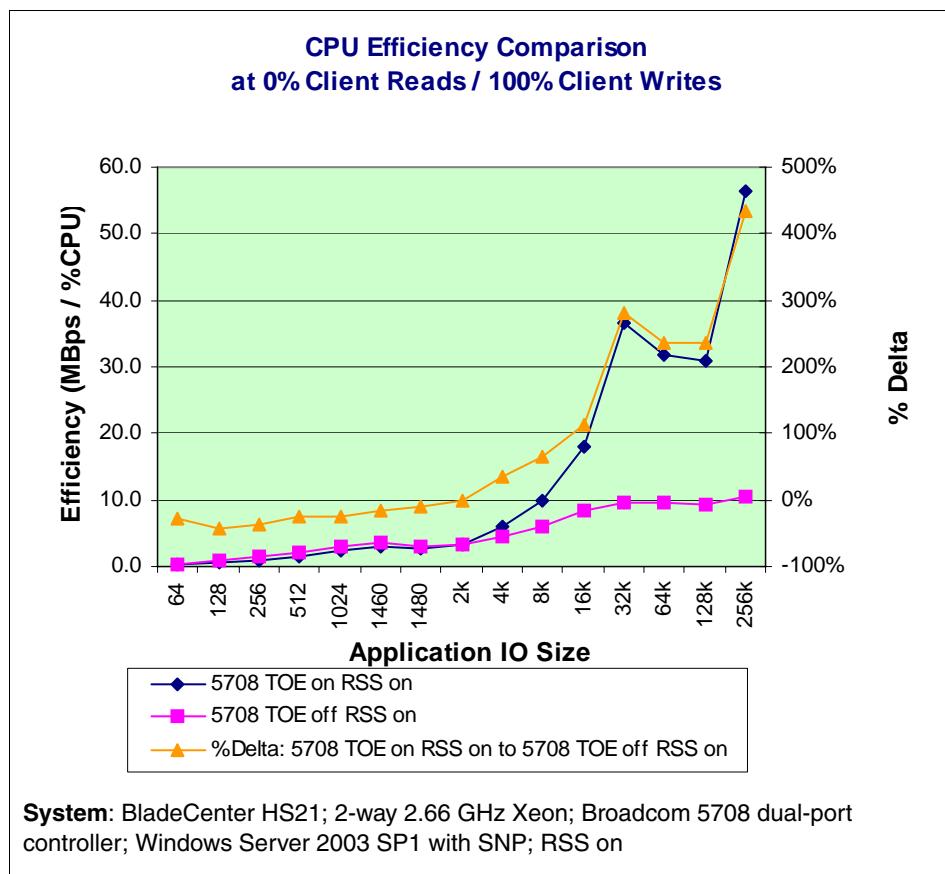


Figure 12-15 CPU efficiency comparison (100% client writes)

Because the data movement steps are modified (compare the two data flows in Figure 12-10 on page 317), and occur in a different sequence, TOE requires support by driver and operating system. Currently, Windows Server 2003 and Windows Server 2008 offer support. Some adapter vendors provide TOE drivers for Linux, but the Red Hat and SUSE Linux distributions do not support TOE natively. See 12.3.6, “Operating system considerations” on page 339 for details.

In terms of bandwidth without TOE, there is a single DMA write, followed by copy from the operating system kernel to user space. This is equivalent to just two memory reads followed by a write on the front-side bus, which is almost 3x. The memory bus sees two reads and two writes, which is almost 4x the speed. The

bandwidth in this case depends on whether buffers are posted or not. In the case that they are not, the result is as if TOE was not used. If buffers are actually posted, then there is a single DMA write. This will still pull the data into the processor caches to be read, so it will result in one read on the FSB and one read plus one write on the memory bus.

TOE is supported by many Ethernet controllers integrated into System x servers, including those with the Broadcom 5708 Ethernet chipset. TOE is also a feature of a number of Gigabit Ethernet adapters including the NetXtreme II 1000 Express Ethernet adapter.

### 12.3.2 I/O Accelerator Technology

I/O Acceleration Technology (I/OAT or IOAT) is an Intel technology that was developed as an alternative to TOE. Similar to TOE, it reduces performance bottlenecks by offloading work from the processor in several ways.

**Tip:** I/OAT is also known as NetDMA.

I/OAT is implemented as a chipset in the server. It implements enhanced data moving capabilities, a new API to control data flow, and a new driver. I/OAT is supported by both Windows and Linux.

I/OAT provides the following:

- ▶ An optimized protocol stack to significantly reduce protocol processing cycles
- ▶ Header splitting, which allows processing of packet headers and payloads on parallel paths
- ▶ Interrupt modulation, to prevent excessive interrupts
- ▶ Use of direct memory access (DMA) to reduce the latency (number of CPU cycles consumed) while waiting for a memory access to finish

Even though TCP/IP has been upgraded and enhanced, the core of the TCP/IP protocol stack has remained unchanged since 1977. To drive improvements in TCP/IP performance to keep up with advances in CPU, memory, and PCI technology, Intel has implemented several enhancements to streamline the network stack:

- ▶ Separate data and control paths

These paths are enabled by header-splitting in the network adapter's media access controller (MAC).

- ▶ Cache-aware data structures

These increase the percentage of cache hits and reduce the number of memory accesses required to process a packet.
- ▶ Improved exception testing

This reduces the length of the path the packet must travel through the protocol stack.

Header splitting reduces latency in packet processing by eliminating the time wasted by the CPU looking at both the header and payload as a single entity. The CPU needs to look only at the header to start the delivery process. Other system components can handle the packet payload more efficiently.

Header splitting also helps in improving the cacheability of headers. Because the headers in a connection are positioned in consecutive addresses, the headers are moved into the cache for processing through processor prefetches.

Interrupt modulation allows the CPU to spend more time on other tasks, rather than having to acknowledge each packet.

Direct memory access (DMA) bypasses the latency that is caused by data movement between memory buffers. Using DMA, as soon as the CPU sends one request off, it can move on to start processing another task. It no longer needs to be involved directly in the movement of data through the processors.

The I/OAT work flow is shown in Figure 12-16.

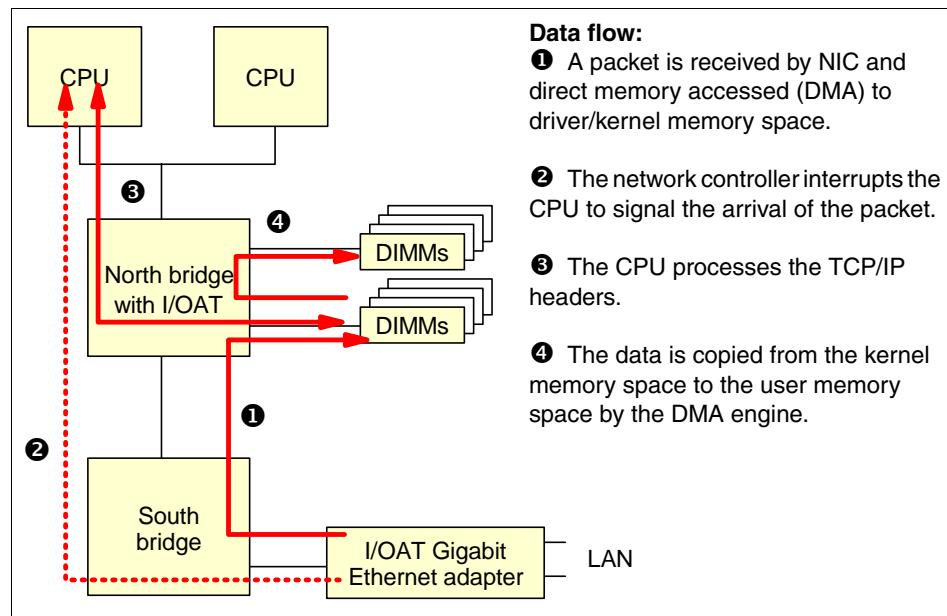


Figure 12-16 Data flow for data received

I/OAT is supported in System x servers with the use of the Intel PRO/1000 PT Dual Port Server Adapter. I/OAT also requires an operating system level update, as described in 12.3.4, “TCP Chimney Offload” on page 332.

IOAT is most beneficial when most I/O traffic is written from the client to the server, because the DMA engine is useful only for this workload and RSS helps only this workload. With I/OAT, there is no optimization for client reads.

The following three charts compare a system with an I/OAT enabled adapter (the Intel PRO/1000 PT Dual Port Server Adapter) with standard Gigabit adapter (the Broadcom 5708 without TOE enabled)

Figure 12-19 on page 329 shows that I/OAT results in lower CPU utilization for large block sizes, as much as half for large block sizes.

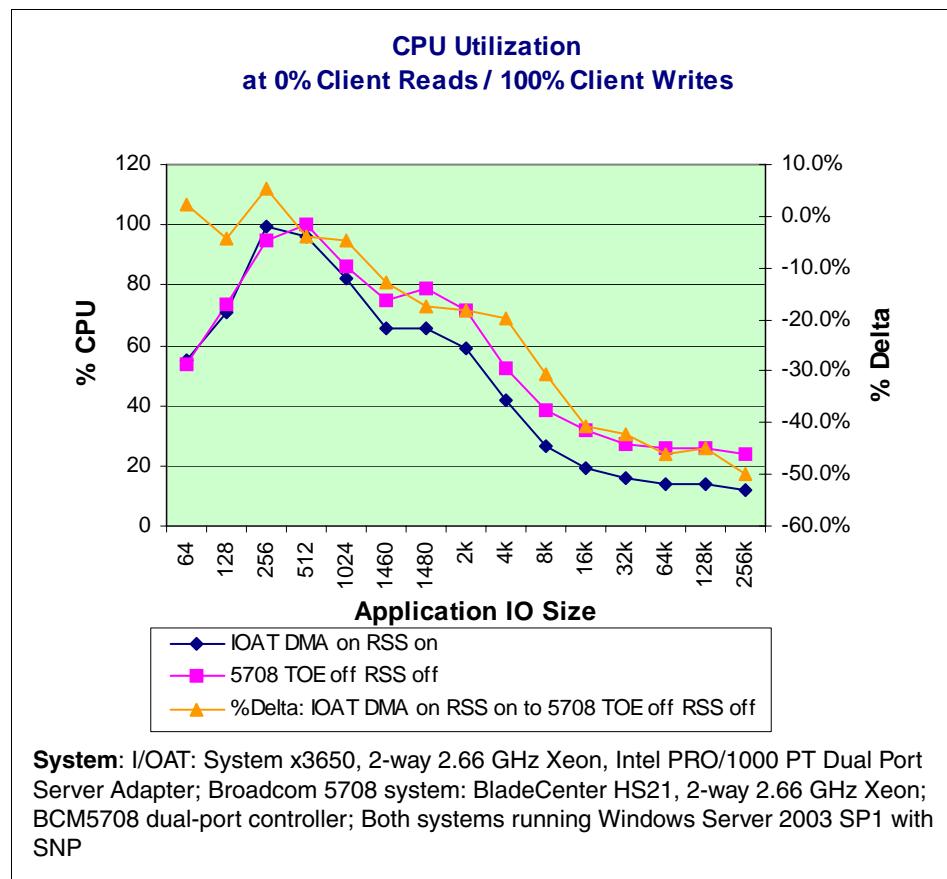


Figure 12-17 CPU utilization comparison - I/OAT enabled versus disabled

I/OAT results in higher CPU efficiency compared to a non-I/OAT system, as shown in Figure 12-18.

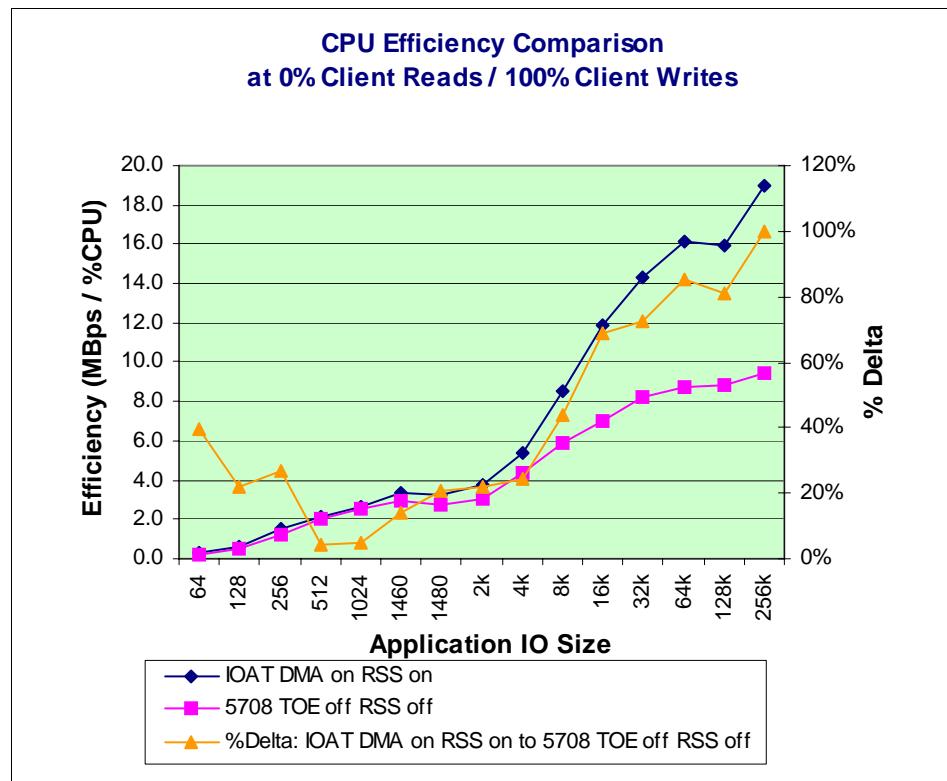


Figure 12-18 CPU efficiency comparison - I/OAT enabled versus disabled

As discussed, throughput does not benefit from technologies such as I/OAT, as shown in Figure 12-19.

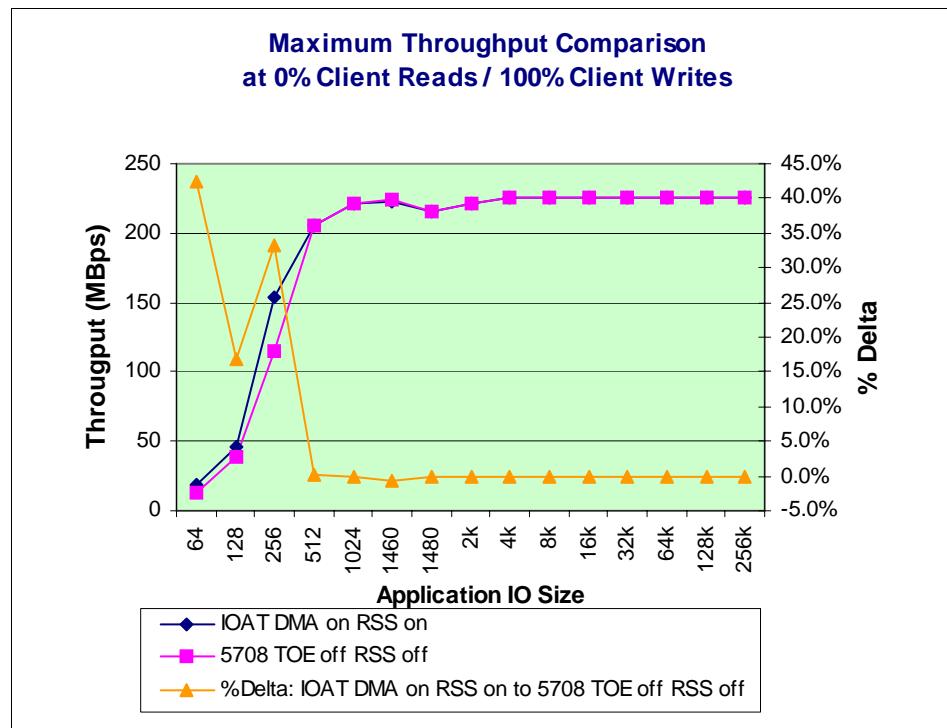


Figure 12-19 Throughput comparison - IOAT enabled versus disabled

### 12.3.3 Comparing TOE and I/OAT

TOE and I/OAT are competing technologies that aim to accomplish the same goal: improving system performance by offloading communication workload from the system processor to other components. However, the way that they achieve this goal is different.

Each technology has advantages and disadvantages when compared with the other. Either technology might be acceptable for you, depending on your needs and infrastructure; however, TOE and I/OAT do differ in several respects:

- ▶ TOE accelerates both transmit and receive traffic. I/OAT accelerates only receive traffic.
- ▶ TOE offloads data movement and protocol processing. I/OAT offloads only data movement.

- ▶ I/OAT is supported by Linux. TOE currently is only supported by Red Hat Enterprise Linux (RHEL) 5.
- ▶ I/OAT is a stateless offload; that is, it does not require state to be stored in the offload engine that scales the number of connections that are offloaded. In contrast, TOE is not a stateless offload.

Both TOE and IOAT require the Linux kernel to be modified. In the case of IOAT, the Linux stack is modified so that during server receives, it instructs the DMA engine in the memory controller to copy the data from the kernel buffers to the user buffers. This is a relatively minor change to the Linux kernel.

However, for TOE, the Linux stack has to be rearchitected because essentially TOE is a TCP/IP stack running in parallel to the Linux TCP/IP stack. The stack has to be rearchitected to support NICs that use the Linux TCP/IP stack, TOE adapters that use portions of the kernel stack and offload portions of it, and full offload adapters that complete bypass the standard Linux TCP/IP stack. This would be a major change to the Linux stack and it has not yet been implemented.

Many customers have already standardized on Ethernet products from either Intel or Broadcom. Ultimately, this standardization might have more bearing on whether you choose TOE or I/OAT than the relative merits of one technology over the other.

Both offload technologies offer the biggest benefit with the server is doing large-block receives (that is, 100% client write at large transfer sizes). TOE offloads data movement and protocol processing. I/OAT offloads data movement only.

Figure 12-20 provides a comparison of I/OAT and TOE showing that CPU utilization is lowered more with TOE than with I/OAT at large block sizes.

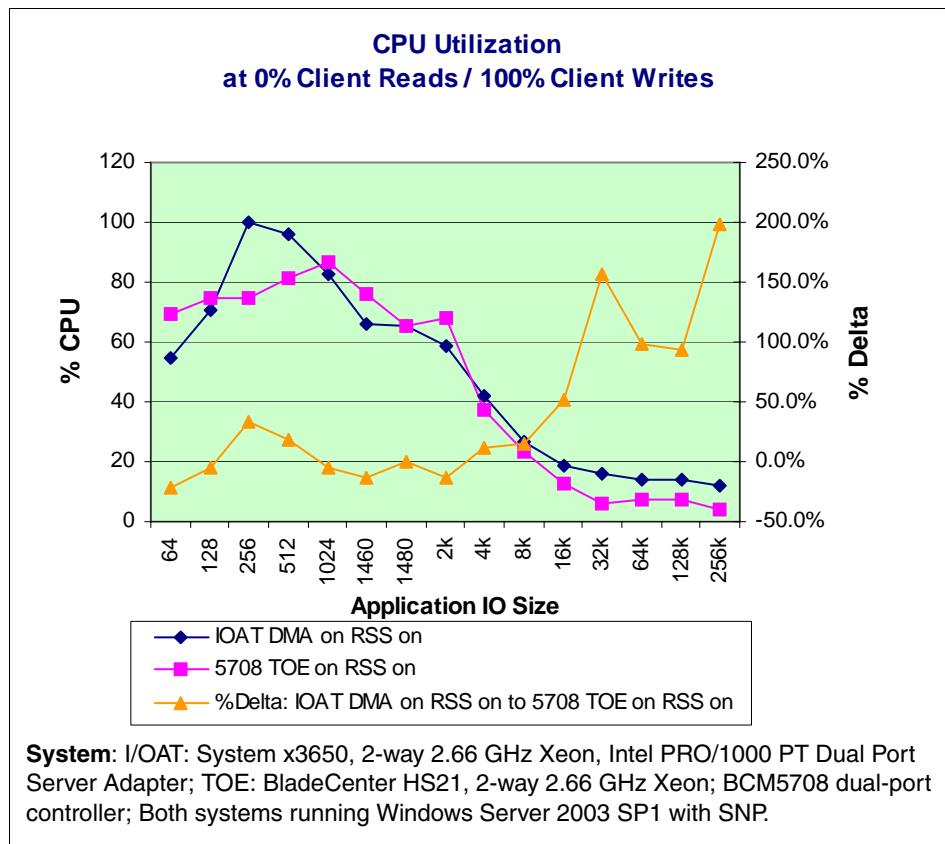


Figure 12-20 Throughput comparison - IOAT versus TOE

Table 12-2 shows the current status of the operating systems supporting I/OAT and TOE.

Table 12-2 Operating system support

Operating system	Intel I/OAT	TOE
Microsoft Windows Server 2008	Yes	Yes
Windows Server 2003 with the Scalable Network Pack	Yes	Yes
SUSE Linux Enterprise Server 10	Yes	Yes <sup>a</sup>
Red Hat Enterprise Linux 5	Yes	Yes <sup>a</sup>

- a. Support for TOE in Linux is directly from the adapter card vendors. There is no native support for TOE in either the Red Hat or SUSE Linux distributions.

#### 12.3.4 TCP Chimney Offload

TCP Chimney Offload is a Microsoft technology that optimizes server performance when processing network traffic. It is implemented either natively in the operating system or with the addition of a Microsoft component, Scalable Networking Pack (SNP). This feature, combined with TOE network adapters, removes existing operating system bottlenecks such as CPU processing overhead related to use network packet processing and the ability to use multiple processors for incoming network traffic. Applications that are currently bound with network processing overhead will generally scale better when used with TCP Chimney.

TCP Chimney Offload is supported natively in Windows Server 2008, although disabled by default, as described in this Windows KB entry:

<http://support.microsoft.com/kb/951037>

The Scalable Networking Pack is available for the following operating systems:

- ▶ Windows Server 2003 SP1 x32 and Windows Server 2003 R2 x32
- ▶ Windows Server 2003 SP1 x64 and Windows Server 2003 R2 x64
- ▶ Windows XP x64

You can download the Scalable Networking Pack from:

<http://support.microsoft.com/kb/912222>

TCP Chimney Offload includes network acceleration technology as well as support for hardware offloading based on the TCP/IP protocol supporting IPv4 and IPv6, as shown in Figure 12-21 on page 333.

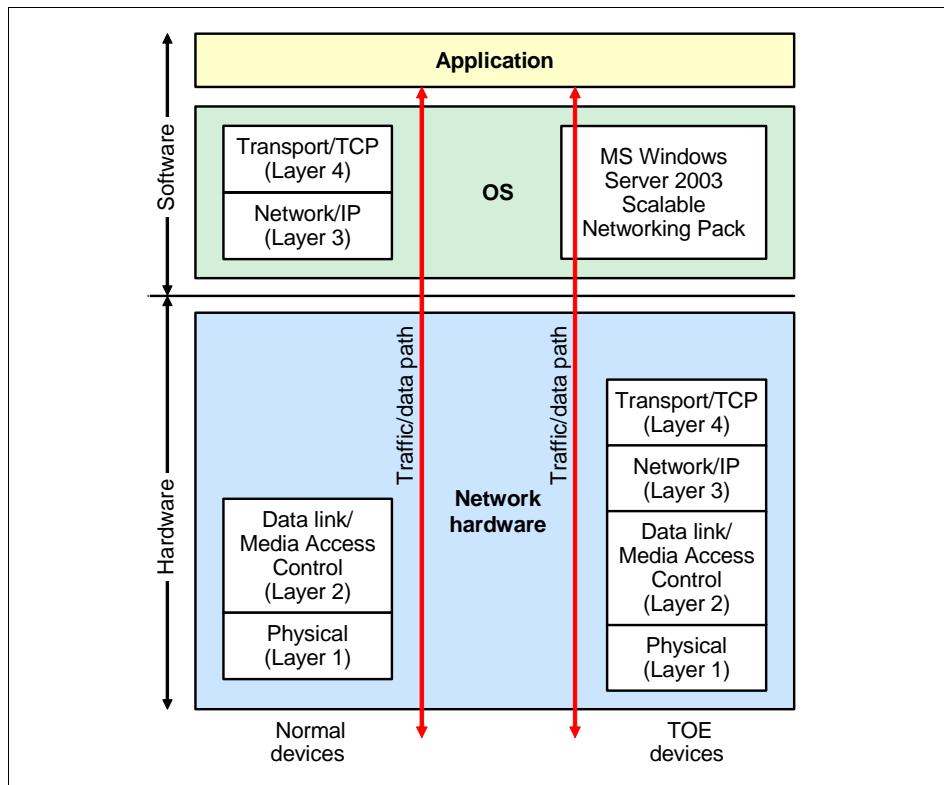


Figure 12-21 Microsoft Scalable Network Pack implementation

Figure 12-21 shows that the Scalable Network Pack is embedded in the first two software layers of the OSI reference model. TCP Chimney Offload creates a software switch shown in Figure 12-22 on page 334, between the top of the protocol stack and the software drivers. Incoming data is transferred directly to the top of the protocol stack, without moving through the intermediate protocol layers. That is also the reason why this technology is called *chimney*—it ascends like smoke in a chimney.

The key concept of the chimney is that data transfer only occurs through the top or the bottom. At the top of the chimney is the implemented switch, which is managed by the operating system. The data that is coming in at the physical layer is getting transferred directly through the chimney to the switch. There, Windows decides either to offload the data back through the chimney to the TOE engine or to process the data itself. If it offloads the data to the TOE engine, it increases the host performance as described in 12.3.1, “TCP offload engine” on page 316.

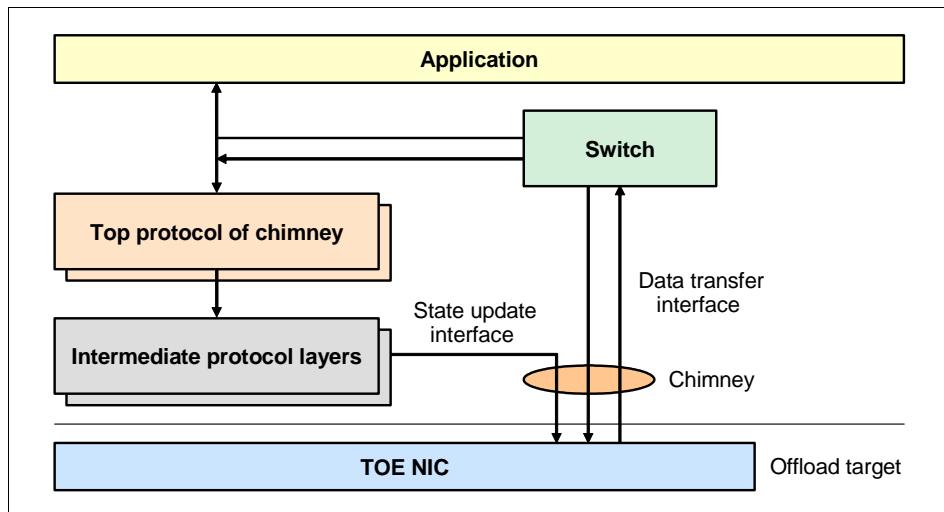


Figure 12-22 Chimney Offload block diagram

### 12.3.5 Receive-side scaling

As discussed in 12.2.3, “Processor speed” on page 310, adding a second CPU to a server does not increase networking performance even if the CPU is the bottleneck, because a network adapter in a multi-core server running Windows is associated with a single core. This limitation means that the associated CPU must handle all the traffic, regardless of whether there are other CPUs available. If there is so much incoming traffic that the TOE and the associated CPU are not able to handle all traffic fast enough, the network adapter discards the traffic, resulting in retransmissions and decreased performance.

Receive-side scaling (RSS) attempts to solve this problem. RSS is a new Network Driver Interface Specification (NDIS) 6.0 technology. It is primarily a software enhancement that takes advantage of multi-core platforms by distributing network processing across the cores. RSS enables packet receive-processing to scale according to the number of available computer processors.

RSS is only available for Windows and is implemented in the Scalable Networking Pack for Windows Server 2003 and natively in Windows Server 2008 (although disabled by default).

RSS offers the following benefits:

- ▶ Parallel execution
  - Receive packets from a single network adapter can be processed concurrently on multiple CPUs, while preserving in-order delivery.
- ▶ Dynamic load balancing
  - As the system load on the host varies, RSS rebalances the network processing load between the processors.
- ▶ Cache locality
  - Because packets from a single connection are always mapped to a specific processor, the state for a particular connection never has to move from one processor's cache to another, thus minimizing cache thrashing and promoting improved performance.
- ▶ Send-side scaling
  - TCP is often limited in how much data it can send to the remote peer. When an application tries to send a buffer larger than the size of the advertised receive window, TCP sends part of the data and then waits for an acknowledgment before sending the balance of the data. When the TCP acknowledgement arrives, additional data is sent in the context of the deferred procedure call in which the acknowledgment is indicated. Thus, scaled receive processing can also result in scaled transmit processing.
- ▶ Secure hash
  - The default generated RSS signature is cryptographically secure, making it much more difficult for malicious remote hosts to force the system into an unbalanced state.

To optimize the performance of this parallel processing of received packets it is critical to preserve in-order delivery. If packets are distributed among the cores of a server, and packets of one connection are processed on different CPUs, it is not possible to enforce that older packets get processed first, and performance would decrease because of that.

RSS assures an in-order packet delivery by ensuring that only one CPU processes packets for a single TCP/IP connection. This means a single TCP/IP connection will always be handled by the same CPU, but different TCP/IP connections can be handled in parallel on other CPUs.

RSS requires that the network adapter check each packet header to create an hash result that is used as an index into the indirection table, and then added to a base CPU number to determine which processor the packet should be processed on. Because the host protocol stack can change the contents of the

table at any time, the TCP/IP stack can dynamically balance the processing load on each CPU.

RSS is most beneficial when CPU is the bottleneck and there are addition CPUs that are currently not being used for I/O processing, as shown in Figure 12-23.

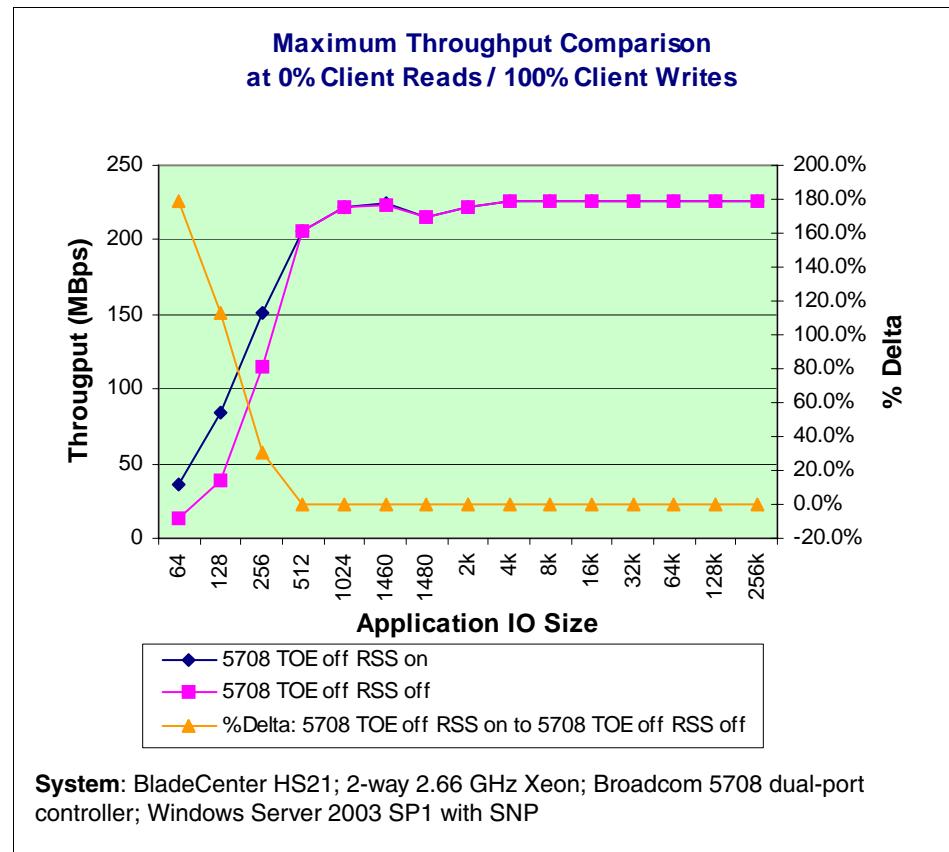


Figure 12-23 RSS benefits most at small block sizes (where CPU is the bottleneck)

As shown in Figure 12-24, with RSS on, more CPU power can now be used to process I/O requests.

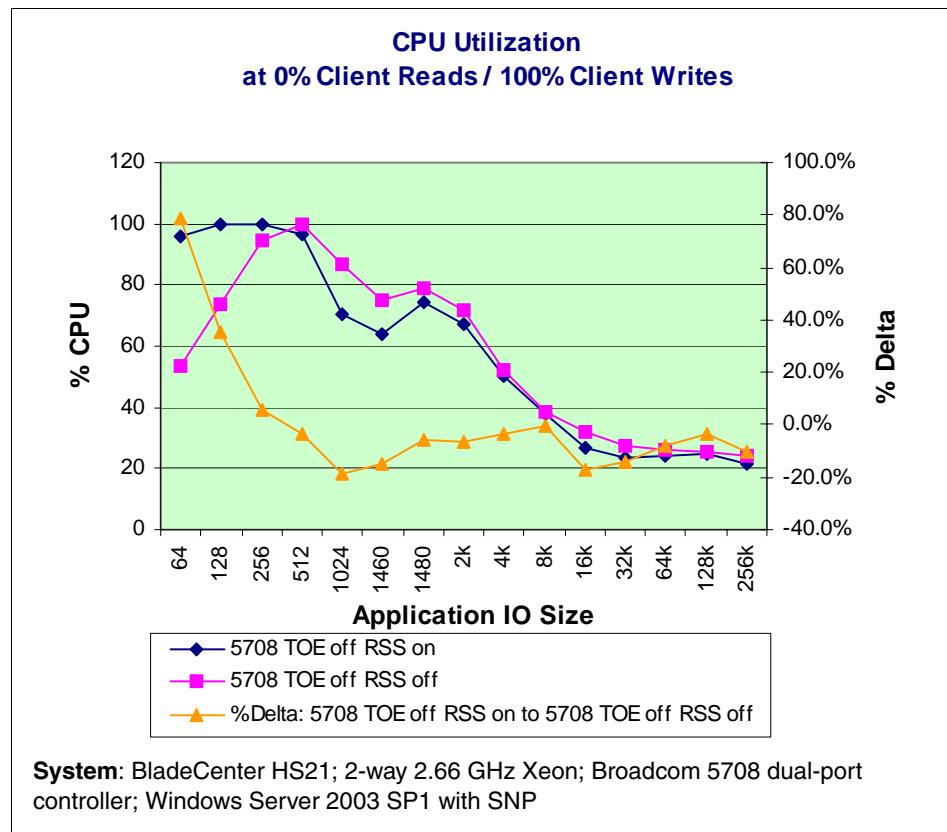


Figure 12-24 RSS benefits most at small block sizes (where CPU is the bottleneck)

The result is a more efficient use of the CPU at small transfers because of RSS, as shown in Figure 12-25.

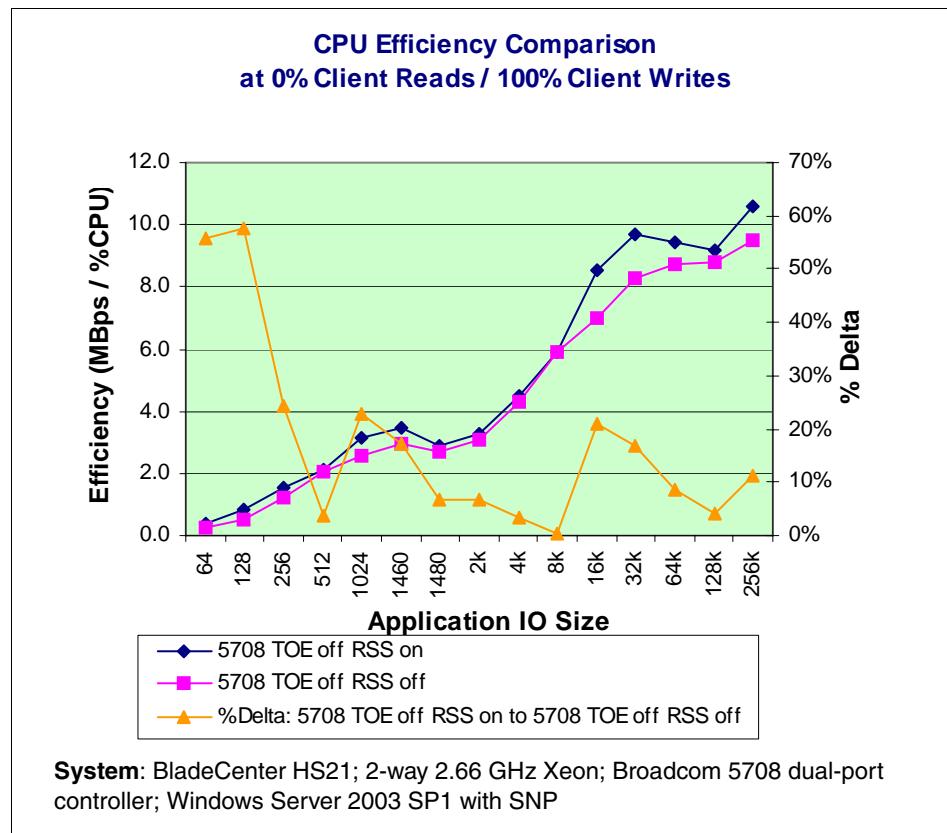


Figure 12-25 RSS benefits most at small block sizes (where CPU is the bottleneck)

For more information about RSS, read the Microsoft White Paper *Scalable Networking with RSS*, which is available from:

[http://www.microsoft.com/whdc/device/network/NDIS\\_RSS.mspx](http://www.microsoft.com/whdc/device/network/NDIS_RSS.mspx)

**Note:** Chimney Offload and RSS are two independent features. The use of RSS does not require the use of Chimney Offload.

### 12.3.6 Operating system considerations

To benefit from these new networking technologies, the operating system view to the network interface must be modified.

- ▶ Microsoft implements support for these technologies either natively (Windows Server 2008) or using the Scalable Networking Pack upgrade and is described in 12.3.4, “TCP Chimney Offload” on page 332.
- ▶ The Linux community currently has no concrete plans to support TOE in the kernel. However, some adapter card vendors do provide TOE support in their drivers.

Table 12-3 lists the support for the advanced networking features that are described in this section.

*Table 12-3 Expected operating system support for advanced networking features*

Operating System	TOE	I/OAT (NetDMA)	Jumbo frames	RSS
Red Hat Enterprise Linux 5	Yes <sup>a</sup>	Yes	OS independent	Yes
SUSE Linux Enterprise Server 10	Yes <sup>a</sup>	Yes	OS independent	Yes
Windows 2000 Server	No	No	OS independent	No
Windows Server 2003 Scalable Networking Pack	Yes	Yes	OS independent	Yes
Windows Server 2008	Yes	Yes	OS independent	Yes

a. Support for TOE in Linux is directly from the adapter card vendors. There is, however, no native support for TOE in either the Red Hat or SUSE Linux distributions.

## 12.4 Internet SCSI (iSCSI)

iSCSI is a transport protocol that carries SCSI commands from an initiator to a target. It is a data storage networking protocol that transports SCSI block I/O protocol requests (commands, sequences, and attributes) over TCP/IP. SCSI data and commands are encapsulated in TCP/IP packets, which means that iSCSI enables Ethernet-based Storage Area Networks (SANs) as opposed to Fibre Channel-based SANs.

iSCSI is well suited to run over almost any physical network. By eliminating the need for a second network technology just for storage, it can lower the cost of implementing network storage. It also offers the capability to extend beyond the

confines of a LAN, to include metropolitan area networks (MANs) and wide area networks (WANs).

The iSCSI technology is a native IP interconnect that wraps SCSI data and commands in TCP/IP packets. The receiving device takes the command out of the IP packet and passes it to the SCSI controller, which forwards the request to the storage device. When the data is retrieved, it is again wrapped in an IP packet and returned to the requesting device.

Do not confuse these IP-based SANs with network attached storage (NAS). NAS permits users to use a LAN file system, where for example the access authorization gets managed by the NAS-box itself; it cannot be formatted or have a file system loaded on it. iSCSI, on the other hand, delivers a block-based file system.

The operating system running on the connected server treats the iSCSI device like an attached hard disk; it is able to format the disk, has control of the access rights, and so on. You are also able to boot from the attached iSCSI device. For example, an IBM TotalStorage DS300 is an iSCSI target server. For more information about iSCSI performance, see 11.4.3, “iSCSI” on page 255.

With iSCSI technology, you can create a SAN from existing, familiar and inexpensive Ethernet components, and quickly develop SAN skills without significant retraining. It affords administrators the ability to centrally control storage devices, to pool storage resources, to integrate NAS appliances into the SAN, and to apply familiar IP security methods to the SAN.

An iSCSI SAN not only offers the same kinds of remote backup, clustering, mirroring, disaster recovery and business continuance capabilities as an FC SAN, but it also improves on FC by making the offsite distance essentially unlimited. Finally, a SAN can improve the performance of not only the storage and retrieval of data, but of the user IP network.

### 12.4.1 iSCSI Initiators

The device at the server end of an iSCSI connection is called an *iSCSI initiator*, and it can be either hardware-based or software-based. The iSCSI initiator is responsible for initiating the SCSI request over IP to a target server. Every host that requires access to the iSCSI target must have at least one initiator installed.

As shown in Figure 12-26, the firmware running on the target device manages the SCSI over IP requests. The initiator intercepts disk access requests and sends commands to the target. Later, when the target responds with disk information, the initiator receives the responses and passes them back to the requestor. iSCSI connections use layer 5 (session layer) of the OSI seven-layer reference model.

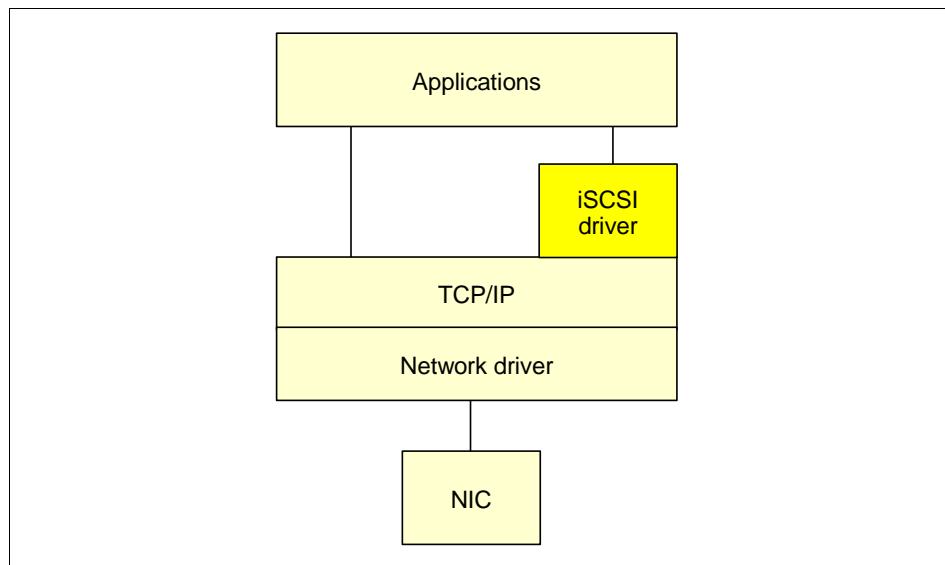


Figure 12-26 iSCSI initiators in the OSI layer 7 reference model

Figure 12-27 shows the encapsulation of iSCSI in TCP/IP packets. The upper packet shows the composition of a normal TCP/IP packet, as used for normal data transfers over TCP/IP Ethernet connections. The lower packet shows how the iSCSI initiator and the SCSI data are encapsulated in the TCP/IP packet.

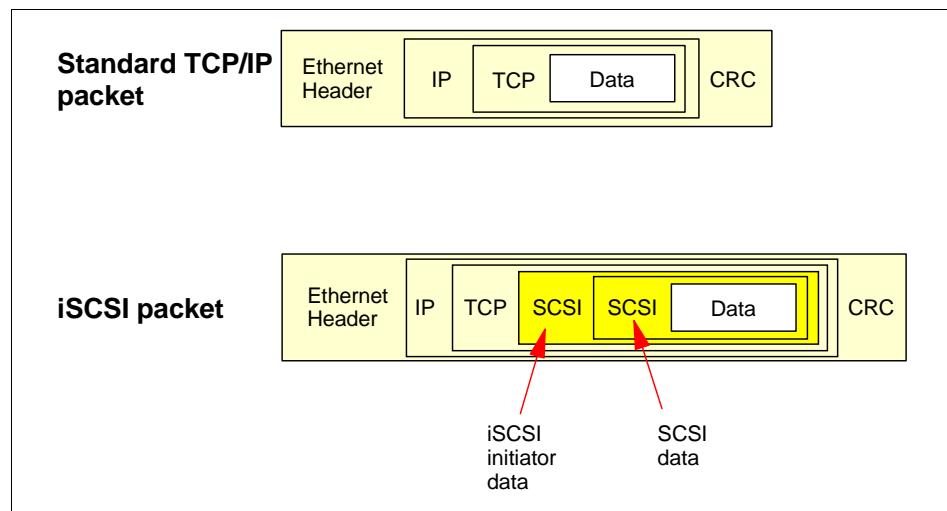


Figure 12-27 iSCSI data packet encapsulation

## Software initiators

Microsoft and Linux operating systems have iSCSI software initiators. An iSCSI software initiator is an iSCSI driver that works with the TCP/IP stack, network drivers, and NICs to provide iSCSI connectivity to other iSCSi devices through the IP network. Because an iSCSI software initiator depends on the operating system's IP stack, if the IP stack fails, access to the remote disk is lost. As a result, software initiators are not ideal for booting the server.

An iSCSI software initiator is an inexpensive alternative to an iSCSI hardware initiator. It is able to provide iSCSI support through software instead of through a separate iSCSi adapter. However, the transfer of the iSCSI IP packets impacts server performance, as shown in 12.1.1, “LAN and TCP/IP performance” on page 296. It adds additional workload to the server processor, TCP/IP stack, and network driver to handle the additional SCSI protocol and data traffic.

Because the host CPU is responsible for processing TCP/IP requests, iSCSI can suffer performance degradation, especially in high traffic settings. This performance limitation is especially significant when compared with Fibre Channel, which does not have TCP/IP overhead. However, iSCSI software initiators are a potential fit for casual demands for storage access.

One way to address the performance problem is to increase the speed of your host processor. Another method is to use an Ethernet controller with TCP/IP Offload Engine (TOE).

### TOE in combination with iSCSI

We describe TOE in 12.3.1, “TCP offload engine” on page 316. Using TOE in combination with iSCSI can improve your server performance. TOE offloads, as shown in the IP ① and TCP ② processing from the CPU and improves performance. However, the iSCSI software initiator ③ is still required: iSCSI connections use layer 5, the TOE handles processing up to layer 4, and then transmits the packets to the application, as shown in Figure 12-28. TOE handles the IP processing, but the iSCSI initiator is still handled by the host system.

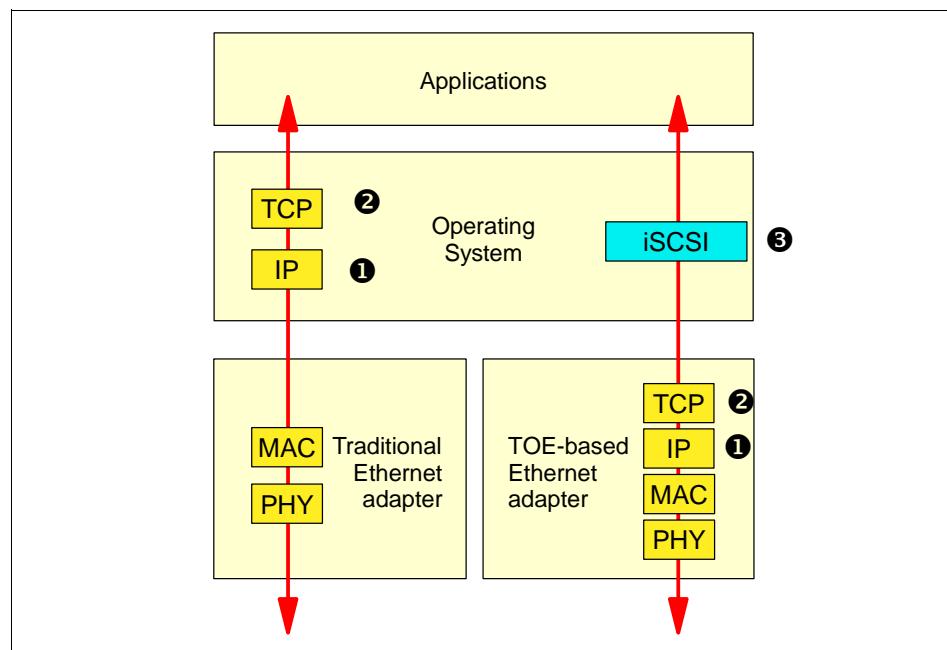


Figure 12-28 TOE in combination with iSCSI

**Note:** If you are using iSCSI in combination with TOE, it might not allow you to run your standard TCP/IP connections off the NIC unless the vendor has provided some type of filter driver to intercept the standard TCP/IP requests.

### Hardware initiator

The iSCSI initiator or driver can be implemented in a hardware adapter card, rather than in the host software. You can implement the adapter card using an

iSCSI host bus adapter (HBA). The iSCSI processing is offloaded to the hardware adapter card instead of processing the iSCSI protocol in the host software. iSCSI TCP/IP processing is also offloaded to the TOE engine on the HBA. With both TCP and iSCSI processing on the adapter card, high-speed transport of block data with minimal CPU overhead is possible.

The hardware initiator is the more expensive of the iSCSI initiator options because it requires the purchase of an iSCSI HBA. However, it is the most capable and the best performer. All of the SCSI block-processing and TOE functions are integrated into the HBA. This frees the host CPU from having to perform any of the iSCSI processing.

#### **12.4.2 iSCSI network infrastructure**

In theory, an iSCSI SAN does not need a separate network infrastructure. It is able to run over the existing network switches, similar to the normal network traffic. However, in the case of high throughput demand, iSCSI produces a high network load and other protocols and applications will suffer because of that load. Thus, for the iSCSI network traffic, it is advisable to use a separate switch or a separate VLAN and separate NICs to raise the transfer rate. This way, no other protocols or applications will suffer a decrease in performance, and you will have higher security for your delivered iSCSI data.

If you are not able to use a separate LAN or if your iSCSI packets have to be securely delivered over the Internet, you might need to use IPsec (IP Security). IPsec is a set of cryptographic protocols for securing packet flows and key exchange. Although Microsoft supports IPsec, at the time of writing, the large hardware vendors do not currently support IPsec in combination with iSCSI. In the near future, it is expected that there will be iSCSI HBAs available that will be able to offload the IPsec encryption and decryption from the host CPU and increase server performance.

Be aware that because accessing iSCSI-attached storage requires traversing a network, there is additional latency when compared to a SCSI or SAS solution. In addition, Ultra320 SCSI (320 MBps peak) is faster than a Gigabit Ethernet connection (a peak of 112 MBps).

Another use of iSCSI is remote boot capability. Because SCSI traditionally is used for operating system boot as well as data storage, iSCSI can be used for boot support. At a high level, this is similar to existing network boot methods such as Preboot Execution Environment (PXE) and BootP. IBM support for iSCSI remote boot is limited currently to blade servers and optional iSCSI adapters.

## 12.5 New trends in networking

In this section we review the protocols and technologies that will impact networking space in the short term, even if the actual challenges to be faced in terms of performance are still unknown. We discuss the 10 Gbps technology and two of the main industry directions over Ethernet: Converged Enhanced Ethernet and Fibre Channel over Ethernet initiatives.

### 12.5.1 10 Gbps Ethernet

Ethernet is probably the most extended and widely used network protocol. Since its adoption, it has been widely deployed and has evolved to meet the new trends and needs of network operations, especially in speed and the transport media supported. The latest Ethernet standard that is being deployed now is 10 Gbps, but there is work in progress to make it work at 100 Gbps and faster in the future.

Adapters and switches currently available support speeds up to 10 Gbps full duplex. These products are now less expensive than they once were because no collision detection hardware or special protocols are needed anymore. When the original draft of the 10 Gbps standard was written in 2002, only fiber-optic cables were supported; now copper cables and existing infrastructure are supported. This makes for an easy upgrade path from Gigabit Ethernet.

IBM has partnered with the key network interface manufacturers to offer 10 Gbps Ethernet products to customers, and these offerings will expand in the immediate future. Figure 12-29 shows early IBM benchmark results of 10 Gbps interface cards on Linux.

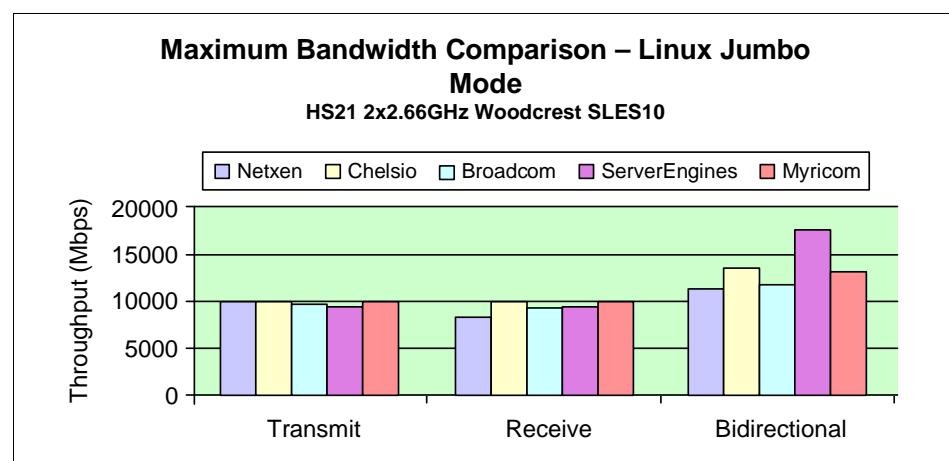


Figure 12-29 Maximum bandwidth comparison

In the next section we discuss one of the main consequences of this evolution, because it enables new and creative uses of this extremely high bandwidth.

## 12.5.2 Converged Enhanced Ethernet

With the recent advances in the speed of Ethernet, industry groups have been examining the ways of converging different I/O applications and protocols over Ethernet, thereby reducing costs and making the network infrastructure easier to manage.

With this idea, IEEE has proposed the creation of the Converged Enhanced Ethernet (CEE, and also called Data Center Ethernet) standard to unify LAN, SAN, and HPC needs over the Ethernet protocol, and extend the existing standards related to Ethernet with new proposals like FC over Ethernet (FCoE).

With existing technology, data centers have different networks for each interconnect technology used. So, for storage needs, FC networks will be used; Ethernet for packet networks; or Myrinet for HPC low latency needs. With the actual speed and bandwidth of the Ethernet networks, going up to 10 Gbps today or even 40 Gbps or 100 Gbps in a near future, there may be sufficient bandwidth to be able to handle all the different I/O needs over the same Ethernet channel.

Configuring these networks today would require a dedicated adapter for each protocol to use. For example, one server connected to one storage fabric over FC, to two different networks and in a HPC cluster, would need at least one dedicated adapter per network, as explained here:

- ▶ At least one Ethernet adapter if the chipset provides two ports, or two Ethernet adapters, if each has only one port
- ▶ At least one Fibre Channel host-bus adapter (HBA)
- ▶ At least one low latency adapter (Myrinet or other)

In the CEE world, only one adapter would be able to handle the Ethernet, SAN and HPC needs using a single very high speed and low latency port, reducing the cost in number of adapter, cabling, external switches, and so on. This basically means:

- ▶ One single Ethernet interface visible from the network side
- ▶ As many interfaces as needed from the server side

The CEE adaptors are called Converged Network Adapters (CNAs). They will typically appear to be a Fibre Channel HBA with an Ethernet NIC on the server side, but will look like a single Ethernet NIC to the network. This will simplify the network infrastructure and possibly create new performance and optimization challenges.

## **FC over Ethernet**

Fibre Channel over Ethernet (FCoE) is a proposed extension of the IEEE 802.3 network standard that enables 10 Gbps networks to use the FC protocol. This is made possible due to three main extensions in the Ethernet specification:

- ▶ Encapsulation of a native FC frame into an Ethernet frame
- ▶ Enablement of lossless Ethernet fabric
- ▶ Replacement of the FC's WWNs with MAC addresses

Basically, this means that FCoE will map FC over the existing Ethernet protocol, while keeping it fully functional. The main difference of this method against iSCSI is that FCoE will be implemented at the IP level as another protocol, when iSCSI is relying on the TCP layer in the Ethernet frames. Because of this, FCoE is not a routeable protocol at the IP level and will not work across IP routed networks.

The use of FCoE can be particularly useful in data centers with fully loaded racks with complex hardware configurations, where it is crucial to optimize the performance/cost ratio by integrating all the connectivity options over one single kind of network, thus simplifying infrastructure management and optimizing the use of the network elements such HBAs and switches.



# Operating systems

The operating system is an integral part of the server. All applications and data have to pass through the operating system from the server hardware to the applications and back again. Just as with the hardware and applications, you need to tune the operating system for performance.

In this part we discuss the following operating systems:

- ▶ Chapter 13, “Microsoft Windows Server 2003” on page 351
- ▶ Chapter 14, “Microsoft Windows Server 2008” on page 425
- ▶ Chapter 15, “Linux” on page 453
- ▶ Chapter 16, “VMware ESX 3.5” on page 501





# Microsoft Windows Server 2003

For the last several years, Windows Server 2003<sup>1</sup> has been Microsoft's mainstream server operating system. It offers considerable improvements in stability, performance, security, scalability, and manageability over previous versions of the Windows server operating system, including Windows 2000 Server and Windows NT Server.

Since the last iteration of this chapter, Microsoft has announced three important enhancements to the core Windows 2003 server operating system. These are:

- ▶ Windows Server 2003, Service Pack 1 for 32-bit (x86) Editions
- ▶ Windows Server 2003, x64 (64-bit) Editions
- ▶ Windows Server 2003, Release 2 (R2) for 32-bit (x86) & 64-bit (x64) Editions

This chapter builds upon the performance tuning techniques detailed in its previous release to also emphasize the performance benefits that can be realized from these important product releases.

---

<sup>1</sup> Product screen captures and content reprinted with permission from Microsoft Corporation.

## 13.1 Introduction to Microsoft Windows Server 2003

Microsoft Windows Server 2003 is designed to be a largely “self-tuning” operating system. A standard “vanilla” installation of the operating system will yield sound performance results in most circumstances. In some instances, however, specific settings and parameters can be tuned to optimize server performance even further. This chapter describes in detail many tuning techniques, any of which can become another weapon in your arsenal of methods to extract the best performance possible from your Windows server.

**Tip:** As with all configuration changes, you need to implement the following suggestions one at a time to see what performance improvements are offered. If system performance decreases after making a change, then you need to reverse the change.

Many of the changes listed in this chapter might only offer marginal performance improvements in and of themselves. However, the real benefits of server tuning are realized when multiple tuning improvements are made and combined with one another. For a given server function, not all tuning techniques listed in this chapter will be appropriate. The challenge for the server engineer or architect is to determine which of these techniques, when combined, will yield the greatest performance enhancements. Many factors will play into this, including the server function, the underlying hardware and how this has been configured, and the network and storage infrastructure that the server is connected to.

It is also well worth noting that some of the performance tuning techniques outlined in this chapter might no longer be relevant in the x64 (64-bit) versions of Windows Server 2003. Several of these techniques described throughout are used to tweak and work around the limitations of the x86 (32-bit) architecture. As a result, in the x64 versions, they are no longer relevant. Where known, this has been outlined.

### The Windows Server 2003 family - 32-bit (x86) Editions

Windows Server 2003 comes in four different 32-bit (x86) versions. These are:

- ▶ Windows Server 2003, Web Edition
- ▶ Windows Server 2003, Standard Edition
- ▶ Windows Server 2003, Enterprise Edition
- ▶ Windows Server 2003, Datacenter Edition

Each of these has support for different hardware configurations and features, and largely determines how scalable the server is. Table 13-1 on page 353 compares the capabilities of the various versions available in the 32-bit (x86) versions of Windows Server 2003.

Note that the maximum amount of memory and the number of CPUs supported in the Enterprise and Datacenter editions of the 32-bit editions of Windows Server 2003 has increased with the release of Service Pack 1 (SP1) and Release 2 (R2).

*Table 13-1 Windows Server 2003 Family - 32-bit (x86) Editions*

Requirement	Web Edition	Standard Edition	Enterprise Edition	Datacenter Edition
Maximum supported RAM	2 GB	4 GB	32/64* GB	64/128* GB
Number of supported processors	1 to 2	1 to 4	1 to 8	8 to 32/64** 8-way capable***
Server clustering	No	No	Up to 8 node	Up to 8 node
Support for /3GB switch	No	No	Yes	Yes
Support for /PAE switch	No	No	Yes	Yes
* Maximum physical memory (RAM) supported has increased from 32 GB to 64 GB for Enterprise Edition with R2 and from 64 GB to 128 GB for Datacenter Edition with R2.				
** Maximum CPU support has increased from 32 to 64 CPUs for Datacenter Edition with R2.				
*** Windows Server 2003 Datacenter Edition requires a server that is 8-way-capable but only requires a minimum of four processors in the actual system.				

## The Windows Server 2003 family - 64-bit (x64) Editions

Microsoft has not released the Web Edition of Windows Server 2003 in the 64-bit (x64) family of server operating systems. The editions that are available are:

- ▶ Windows Server 2003, Standard x64 Edition
- ▶ Windows Server 2003, Enterprise x64 Edition
- ▶ Windows Server 2003, Enterprise Itanium Edition
- ▶ Windows Server 2003, Datacenter x64 Edition
- ▶ Windows Server 2003, Datacenter Itanium Edition

Because it is a considerably later release, much of the code base for the 64-bit (x64) editions of Windows Server 2003 is based on the same code that makes up the Service Pack 1 editions of Windows Server 2003. As a result, Service Pack 1 is not an option for the 64-bit (x64) editions. Release 2 (R2) is an optional extra for the 64-bit (x64) editions of Windows Server 2003, although it is expected that most customers would install R2 by default to access the many extra features available within this latest product offering.

Due to the fundamental architectural differences of 64-bit computing, vastly higher memory thresholds are available in the 64-bit (x64) editions of Windows Server 2003, as evidenced in Table 13-2.

Table 13-2 Windows Server 2003 Family - 64-bit (x64) Editions

Requirement	Standard x64 Edition	Enterprise x64 Edition	Datacenter x64 Edition	Enterprise Itanium Edition	Datacenter Itanium Edition
Maximum supported RAM	32 GB*	1 TB	1 TB	1 TB	1 TB
Number of supported processors	1 to 4	1 to 8	8 to 64	1 to 8	8 to 64 8-way-capable*
Server clustering	No	Up to 8 node	Up to 8 node	Up to 8 node	Up to 8 node

\* Windows Server 2003 Datacenter Edition requires a server that is 8-way-capable but only requires a minimum of four processors in the actual system.

A more thorough comparison of all the feature differences between the various versions of the Windows Server 2003 operating system for both 32-bit and 64-bit editions can be found at:

<http://www.microsoft.com/windowsserver2003/evaluation/features/comparefeatures.mspx>

## 13.2 Windows Server 2003 - 64-bit (x64) Editions

After years of working around the limitations of the 32-bit processor architecture, Microsoft released the much-awaited 64-bit editions of Windows Server 2003 in April 2003. Although the Itanium version has been available for some time, it is the release of the editions of Windows Server 2003 to support x64 processors that will see 64-bit computing finally transition into the Windows server mainstream.

In a relatively short period of time, it is expected that the 64-bit editions of Windows Server 2003 will displace the 32-bit editions. This is largely assisted by the high level of compatibility that Microsoft has built into the 64-bit (x64) operating system, offering true backward compatibility for 32-bit applications with little to no degradation in performance.

### **13.2.1 32-bit limitations**

The amount of virtual memory that can be addressed by the 32-bit versions of Windows Server 2003 is 4 GB, through a virtual address space. On a standard implementation, this 4 GB is divided into 2 GB for kernel mode processes and 2 GB for application (user) mode processes.

In Windows Server 2003 32-bit editions, it is possible to increase the amount of memory available from 2 GB to 3 GB for 32-bit applications that have been designed to use more than 2 GB, through the use of the /3GB and /PAE switches, as explained in 13.13, “The /3GB BOOT.INI parameter (32-bit x86)” on page 390 and 13.14, “Using PAE and AWE to access memory above 4 GB (32-bit x86)” on page 391.

This increase of available user memory from 2 GB to 3 GB presents the following problems, however:

- ▶ It imposes limitations on the amount of memory available to kernel mode processes to 1 GB.
- ▶ It does not work around the architectural limit of the total 4 GB virtual address space.
- ▶ It increases the difficulty of developing applications because they need to make use of the Addressable Windows Extensions (AWE) application programming interface (API) to take advantage of Physical Address Extensions (PAE).
- ▶ It has not removed the physical memory constraint of 64 GB.

With the upgrade to Windows Server 2003 64-bit (x64) editions, these limitations no longer exist and there are opportunities for significant improvements in server performance.

### **13.2.2 64-bit benefits**

The single largest performance increase of the 64-bit architecture is the amount of memory that can now be addressed. With Windows Server 2003 x64 Editions, the addressable virtual memory space increases from the 32-bit limit of just 4 GB to 16 TB. Entire databases, data sets, directory stores, indexes, Web caches and applications can now be loaded completely into memory, delivering often staggering processing performance improvements and vastly increased scalability.

It is worth noting that the current Windows Server 2003 x64 editions actually only use 40 bits for addressing memory, offering an address space of  $2^{40}$ , or 16 TB. 16 Exabytes is that actual theoretical maximum of a full 64-bit address space.

This virtual address space is divided evenly between user mode and kernel mode, as with 32-bit Windows. This provides native 64-bit applications with 8 TB of virtual address space. Even 32-bit applications that have been written to take advantage of memory greater than the 2 GB limitation of the 32-bit architecture can benefit from this immediately because they can now address 4 GB of virtual address space as this space no longer needs to be shared with kernel mode processes.

Table 13-3 highlights some of the key difference between 32-bit and 64-bit memory limits. Each of the notable improvements in these memory limits for the 64-bit (x64) platform offers real scalability and performance enhancements.

*Table 13-3 Memory limitations of 32-bit (x86) and 64-bit (x64) Windows Server 2003*

Memory Limit	32-bit (x86)	64-bit (x64)
Total virtual address space	4 GB	16 TB
Virtual address space per 32-bit process	2 GB	4 GB
Virtual address space per 64-bit process	Not applicable	8 TB
Paged Pool	491 MB	128 GB
Non-paged Pool	256 MB	128 GB
System Page Table Entry (PTE)	660 MB to 990 MB	128 GB

The Windows on Windows 64 emulator (WOW64) allows 32-bit applications to run on Windows Server 2003 x64 Editions exactly as they might run on a 32-bit edition. This has been written so optimally that any overhead imposed in emulation activities is very marginal and in many cases imperceptible. Even with the emulation between 32-bit and 64-bit, in several cases 32-bit applications will run faster on Windows Server 64-bit (x64) Editions due to other improvements in the operating system, offering another notable benefit to this new operating system version.

Another notable advantage of the 64-bit (x64) editions is the greatly increased amount of physical RAM that can be supported, offering huge scalability benefits. With the Standard Edition of Windows Server 2003 x64, the maximum supported memory is 32 GB. With the Enterprise and Datacenter Editions, however, it is a considerably larger 1 TB of RAM. When compared to the previous memory maximums of 4 GB, 64 GB, and 128 GB of the Standard, Enterprise and Datacenter editions of Windows Server 2003 R2, the increase in supportable memory, and thus performance, is significant.

Overall performance improvements in disk and network I/O throughput and efficiency should be evident in the 64-bit (x64) editions of Windows Server 2003. Greater physical memory support and addressable memory space means that caches can be considerably larger, improving I/O performance. An increased number of larger (wider) processor registers will also deliver notable performance gains to applications because data does not need to be written out to memory as frequently and function calls can process more arguments.

Windows Server 2003 64-bit (x64) Edition also delivers improved reliability over previous versions. Based on the exactly the same source code as Windows Server 2003 Service Pack 1 32-bit editions (x86), the newer edition will offer the same reliability that this platform has offered to date. In addition, Windows Server 2003 64-bit (x64) editions include Patch-Guard, a technology that protects poorly-written third-party code from patching the Windows kernel, which in turn could destabilize or crash a server.

Security improvements are also available with the 64-bit (x64) edition of Windows Server 2003. Building on the benefits of Data Execution Prevention (DEP) released first in Windows Server 2003 Service Pack 1 (32-bit x86), the 64-bit (x64) editions all include this feature as a standard. DEP protects Windows against buffer overflows, effectively stopping malicious code from being able to execute from memory locations it should not.

All these features of Windows Server 2003 64-bit (x64) editions serve to make this new version of the operating system the most high-performing, scalable, stable, and secure version released to date.

### 13.2.3 The transition to 64-bit computing

With the release of Intel 64 Technology (previously known as EMT64T) and AMD AMD64, server hardware vendors have made the transition from 32-bit (x86) to 64-bit (x64) processing a very straightforward process. These processors support both 32-bit and 64-bit operating systems and applications, making the migration path an easy one.

With the 64-bit (x64) versions of Windows Server 2003 able to run 32-bit applications directly, often at a much higher level of performance, the move to the optimal native 64-bit computing should present few hurdles.

### 13.2.4 Acknowledgements

Much of this material for this section on Windows Server 2003 64-bit (x64) editions has been collated from two key articles available from the Microsoft Web site. More detail and case studies of the benefits of Windows Server 2003 64-bit (x64) computing can be found by referring to these two papers:

- ▶ *Benefits of Microsoft Windows x64 Editions*  
<http://www.microsoft.com/windowsserver2003/techinfo/overview/x64benefits.mspx>
- ▶ *Windows Server 2003 x64 Editions Deployment Scenarios*  
<http://www.microsoft.com/windowsserver2003/64bit/x64/deploy.mspx>

## 13.3 Windows Server 2003, Release 2 (R2)

Windows Server 2003, R2 is an update release for the Windows Server 2003 operating system. This release brings an impressive array of additional features to the native operating system.

This release is different from a Service Pack in that it brings new features and functionality to the operating system while as Service Pack is a rollup of fixes, updates and patches at a given point in time. That said, the installation of R2 is dependent on Windows Server 2003, Service Pack 1 already being installed.

R2 offers enhancements to Windows Server 2003 in the following main areas:

- ▶ Simplified branch office server management
- ▶ Improved identity and access management
- ▶ Reduced storage management costs
- ▶ Rich Web platform
- ▶ Cost effective server virtualization
- ▶ Seamless UNIX / Windows Interoperability

For more detail on the features delivered by R2, visit the following links:

<http://www.microsoft.com/windowsserver2003/R2/whatsnewinr2.mspx>  
<http://download.microsoft.com/download/7/f/3/7f396370-86ba-4cb5-b19e-e7e518cf53ba/WS03R2RevGuide.doc>

The components of R2 that offer notable performance benefits are those included to improve branch office server manageability, such as the following:

- ▶ Robust file replication

The replication engine for the Distributed File System (DFS) has been completely rewritten in Windows Server 2003 R2. DFS is multimaster file

replication service, significantly more scalable and efficient in synchronizing file servers than File Replication Services (FRS), its predecessor. DFS schedules and throttles replication processes, supports multiple replication topologies, and utilizes Remote Differential Compression (RDC) to increase WAN efficiency. If WAN connections fail, data can be stored and forwarded when WAN connections become available. Through the efficiency gains of these new features in R2 DFS, the performance of core user-facing processes improves.

- ▶ Advanced compression technologies

Remote Differential Compression (RDC) is a WAN-friendly compression technology that replicates only the changes needed to ensure global file consistency. Any WAN performance improvements often serve to improve the user experience.

## 13.4 Processor scheduling

Windows uses *preemptive multitasking* to prioritize process threads that the CPU has to attend to. Preemptive multitasking is a methodology whereby the execution of a process is halted and another is started, at the discretion of the operating system. This prevents a single thread from monopolizing the CPU.

Switching the CPU from executing one process to the next is known as *context-switching*. The Windows operating system includes a setting that determines how long individual threads are allowed to run on the CPU before a context-switch occurs and the next thread is serviced. This amount of time is referred to as a *quantum*.

This setting lets you choose how processor quanta are shared between foreground and background processes. Typically for a server, it is not desirable to allow the foreground program to have more CPU time allocated to it than background processes. That is, all applications and their processes running on the server should be given equal contention for the CPU.

We recommend selecting **Background services** so that all programs receive equal amounts of processor time.

To change this:

1. Open the **System** Control Panel.
2. Select the Advanced tab.
3. Within the Performance frame, click **Settings**.
4. Select the Advanced tab. The window shown in Figure 13-1 on page 360 opens.

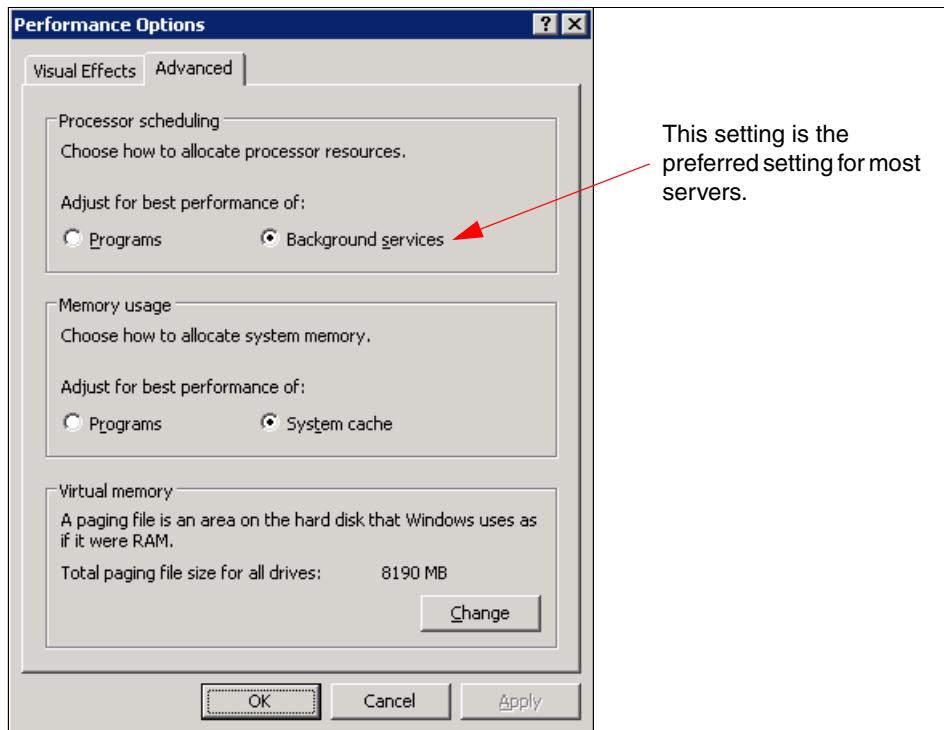


Figure 13-1 Configuring processor scheduling

Modifying the value using the control panel applet as described modifies the following registry value to affect the duration of each quanta:

`HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\PriorityControl\Win32PrioritySeparation`

The Win32PrioritySeparation Registry values in Windows Server 2003 are:

- ▶ 0x00000026 (38) for best performance of Programs
- ▶ 0x00000018 (24) for best performance of Background services

These values remain the same between the 32-bit (x86) and 64-bit (x64) editions of the Windows Server 2003 operating system.

We strongly recommend you use only the control panel applet shown in Figure 13-1 for these settings to always get valid, appropriate, operating system revision-specific, and optimal values in the registry.

## 13.5 Virtual memory

Memory *paging* occurs when memory resources required by the processes running on the server exceed the physical amount of memory installed. Windows, like most other operating systems, employs *virtual memory* techniques that allow applications to address greater amounts of memory than what is physically available. This is achieved by setting aside a portion of disk for paging. This area, known as the *paging file*, is used by the operating system to page portions of physical memory contents to disk, thereby freeing up physical memory to be used by applications that require it at a given time. The combination of the paging file and the physical memory installed in the server is known as *virtual memory*. Virtual memory is managed in Windows by the Virtual Memory Manager (VMM).

Physical memory can be accessed at exponentially faster rates than disk. Every time a server has to move data between physical memory and disk will introduce a significant system delay. Although some degree of paging is normal on servers, excessive, consistent memory paging activity is referred to as *thrashing* and can have a very debilitating effect on overall system performance. Thus, it is always desirable to minimize paging activity. Ideally servers should be designed with sufficient physical memory to keep paging to an absolute minimum.

The paging file, or pagefile, in Windows, is named PAGEFILE.SYS.

Virtual memory settings are configured through the System control panel.

To configure the page file size:

1. Open the System Control Panel.
2. Select the Advanced tab.
3. Within the Performance frame, click **Settings**.
4. Select the Advanced tab.
5. Click **Change**. The window shown in Figure 13-2 on page 362 opens.

Windows Server 2003 has several options for configuring the page file that previous versions of Windows did not. Windows Server 2003 has introduced new settings for virtual memory configuration, including letting the system manage the size of the page file, or to have no page file at all. If you let Windows manage the size, it will create a pagefile of a size equal to physical memory + 1 MB. This is the minimum amount of space required to create a memory dump in the event the server encounters a STOP event (blue screen).

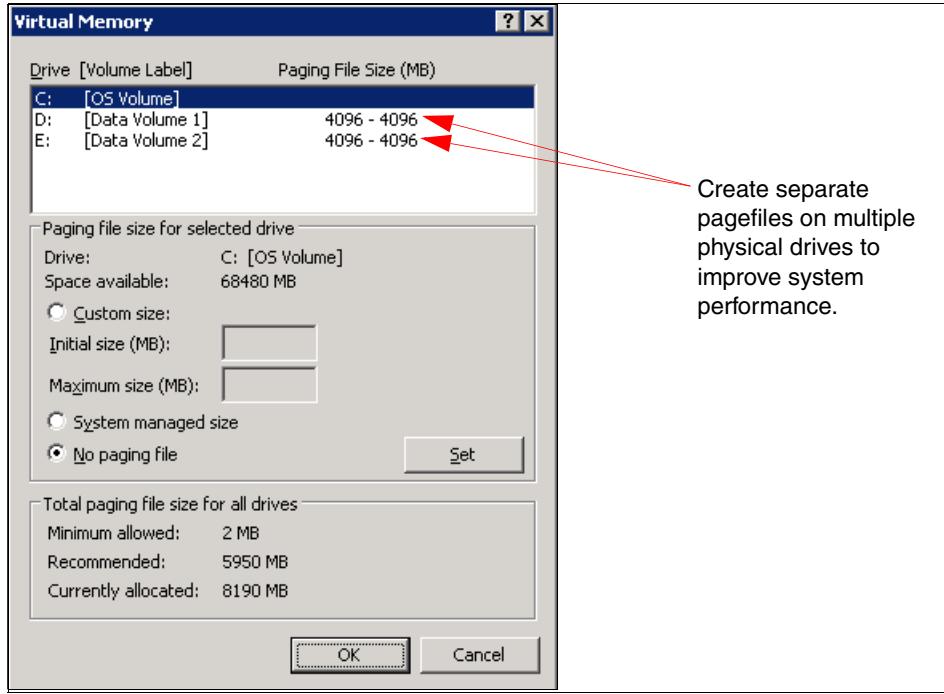


Figure 13-2 Virtual memory settings

A pagefile can be created for each individual volume on a server, up to a maximum of sixteen page files and a maximum 4 GB limit per pagefile. This allows for a maximum total pagefile size of 64 GB. The total of all pagefiles on all volumes is managed and used by the operating system as one large pagefile.

If a pagefile is split between smaller pagefiles on separate volumes as described, when it needs to write to the pagefile, the virtual memory manager optimizes the workload by selecting the least busy disk based on internal algorithms. This ensures the best possible performance for a multiple-volume pagefile.

While not best practice, it is possible to create multiple page files on the same operating system volume. This is achieved by placing the page files in different folders on the same volume. This change is carried out through editing the system registry rather than through the standard GUI interface. The process to achieve this is outlined in Microsoft KB article 237740:

<http://support.microsoft.com/?kbid=237740>

We do not recommend this approach because no performance gain will be achieved by splitting the page file into segments on the same volume regardless of the underlying physical disk or array configuration.

### 13.5.1 Configuring the pagefile for maximum performance gain

Optimal pagefile performance will be achieved by isolating pagefiles to dedicated physical drives running on RAID-0 (striping) or RAID-1 (mirroring) arrays, or on single disks without RAID at all. Redundancy is not normally required for pagefiles, although performance might be improved through the use of some RAID configurations. By using a dedicated disk or drive array, this means PAGEFILE.SYS is the only file on the entire volume and risks no fragmentation caused by other files or directories residing on the same volume. As with most disk arrays, the more physical disks in the array, the better the performance. When distributed between multiple volumes on multiple physical drives, the pagefile size should be kept uniform between drives and ideally, on drives of the same capacity and speed.

We strongly recommend against the use of RAID-5 arrays to host pagefiles because pagefile activity is write-intensive and thus not suited to the characteristics of RAID-5.

Where pagefile optimization is critical, do not place the pagefile on the same physical drive as the operating system, which happens to be the system default. If this must occur, ensure that the pagefile exists on the same volume (typically C:) as the operating system. Putting it on another volume on the same physical drive will only increase disk seek time and reduce system performance because the disk heads will be continually moving between the volumes, alternately accessing the page file, operating system files, and other applications and data.

Remember too that the first partition on a physical disk is closest to the outside edge of the physical disk, the one typically hosting the operating system, where disk speed is highest and performance is best.

Note if you do remove the paging file from the boot partition, Windows cannot create a crash dump file (MEMORY.DMP) in which to write debugging information in the event that a kernel mode STOP error message (“blue screen of death”) occurs. If you do require a crash dump file, then you will have no option but to leave a page file of at least the size of physical memory + 1 MB on the boot partition.

We recommend setting the size of the pagefile manually. This normally produces better results than allowing the server to size it automatically or having no page file at all. Best-practice tuning is to set the initial (minimum) and maximum size settings for the pagefile to the same value. This ensures that no processing resources are lost to the dynamic resizing of the pagefile, which can be intensive. This is especially important given that this resizing activity is typically occurring when the memory resources on the system are already becoming constrained. Setting the same minimum and maximum page file size values also ensures that

the paging area on a disk is one single, contiguous area, improving disk seek time.

Windows Server 2003 automatically recommends a total paging file size equal to 1.5 times the amount of installed RAM. On servers with adequate disk space, the pagefile on all disks combined should be configured up to twice (that is, two times) the physical memory for optimal performance. The only drawback of having such a large pagefile is the amount of disk space consumed on the volumes used to accommodate the page file(s). Servers of lesser workloads or those tight on disk space should still try to use a pagefile total of at least equal to the amount of physical memory.

### 13.5.2 Creating the pagefile to optimize performance

Creating the whole pagefile in one step reduces the possibility of having a partitioned pagefile, and therefore improves system performance.

The best way to create a contiguous static pagefile in one step is to follow this procedure for each pagefile configured:

1. Remove the current page files from your server by clearing the Initial and Maximum size values in the Virtual Memory settings window or by clicking **No Paging File**, then clicking **Set** (Figure 13-2 on page 362).
2. Reboot the machine and click **OK**. Ignore the warning message about the page file.
3. Defragment the disk you want to create the page file on. This step should give you enough continuous space to avoid partitioning of your new page file.
4. Create a new page file with the desired values as described in 13.5.2, “Creating the pagefile to optimize performance” on page 364.
5. Reboot the server.

An even better approach is to reformat the volume entirely and create the pagefile immediately before placing any data on the disk. This ensures the file is created as one large contiguous file as close to the very outside edge of the disk as possible, ensuring no fragmentation and best disk access performance. The effort and time involved in moving data to another volume temporarily to achieve this outcome often means, however, that this procedure is not always achievable on a production server.

### 13.5.3 Measuring pagefile usage

A useful metric for measuring pagefile usage is Paging file: %Usage Max in the Windows System Monitor. If this reveals consistent use of the page file, then

consider increasing the amount of physical memory in the server by this amount. For example, if a pagefile is 2048 MB (2 GB) and your server is consistently showing 10% usage, it would be prudent to add an additional 256 MB RAM or so.

Although today it is often considered an easy and relatively inexpensive way to upgrade a server and improve performance, adding additional memory should only be done after investigating all other tuning options and after it has been determined that memory is indeed the system bottleneck. There is simply no benefit to be gained by having more physical memory in a server than it can put to use. Additional, unused memory might be better put to use in another server.

## 13.6 File system cache

The file system cache is an area of physical memory set aside to dynamically store recently accessed data that has been read or written to the I/O subsystem, including data transfers between hard drives, networks interfaces, and networks. The Windows Virtual Memory Manager (VMM) copies data to and from the system cache as though it were an array in memory.

The file system cache improves performance by reducing the number of accesses to physical devices attached to the I/O subsystem of the server. By moving commonly used files into system cache, disk and network read and write operations are reduced and system performance is improved. You can optimize Windows server performance by tuning the file system cache.

Performance of the file system cache is greatly improved in the 64-bit (x64) editions of Windows Server 2003. The default 2 GB kernel maximum virtual memory address space in the 32-bit (x86) editions of Windows is a major bottleneck because this same space is shared by the system page table entries (PTE), page pool memory, non-paged pool memory and file system cache.

Using the /3GB switch on 32-bit (x86) systems can improve application performance (as described in 13.9, “Optimizing the protocol binding and provider order” on page 376 and 13.10, “Optimizing network card settings” on page 378) but this forces the Windows kernel to operate in only 1 GB of virtual address space, potentially making the situation worse. These same constraints no longer apply in the 64-bit (x64) editions of Windows, thereby greatly enhancing system performance.

**Tip:** The method for managing the file system cache has changed in Windows Server 2003. There are now two applets, not one as in previous versions of Windows.

In previous versions of the Windows server operating system, one Control Panel applet was used for managing the file system cache. Now, in Windows Server 2003, two configuration options exist that determine how much system memory is available to be allocated to the working set of the file system cache versus how much memory is able to be allocated to the working set of applications, and the priority with which they are managed against one another.

The selection made in these dialogs will depend on the intended server function.

The configuration options are:

- ▶ File and Printer Sharing for Microsoft Networks (Network Control Panel applet).
- ▶ System (System Control Panel applet), which is also referred to in Microsoft documentation as “Performance Options” because it has consolidated several performance applets into one location.

The File and Printer Sharing dialog can be accessed as follows:

1. Click **Start** → **Control Panel** → **Network Connections**.
2. While still in the Start menu context, right-click Network Connections and choose **Open**.
3. Select any of Local Area Connections including any teamed network interface. This setting affects all LAN interfaces, so which LAN connection you choose in the preceding steps is not important.
4. Right-click the selected connection object and choose **Properties**.
5. Select **File and Printer Sharing for Microsoft Networks**.
6. Click **Properties**. The window that is shown in Figure 13-3 on page 367 opens.

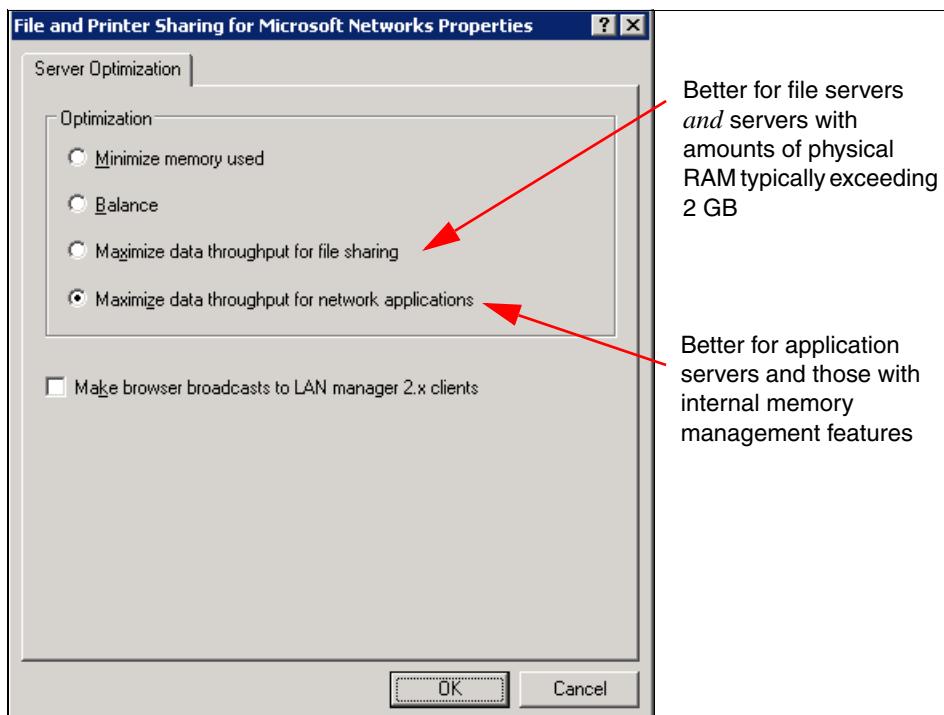


Figure 13-3 File and Print Sharing for Microsoft Networks applet: server optimization

The four options here modify two registry entries:

- ▶ HKLM\System\CurrentControlSet\Services\LanmanServer\Parameters\Size
- ▶ HKLM\System\CurrentControlSet\Control\Session Manager\Memory Management\LargeSystemCache

The value of the registry entries will be set depending on the option selected in the control panel as listed in Table 13-4.

Table 13-4 Registry effect of Server Optimization option selected

Server optimization option selected	LanmanServer Size	LargeSystemCache
Minimize memory used	1	0
Balance	2	0
Maximize data throughput for file sharing	3	1

Server optimization option selected	LanmanServer Size	LargeSystemCache
Maximize data throughput for network applications	3	0

These values are the same for both the 32-bit (x86) and 64-bit (x64) editions of Windows Server 2003.

The file system cache has a working set of memory, like any other process. The option chosen in this dialog effectively determines how large the working set is allowed to grow to and with what priority the file system cache is treated by the operating system relative to other applications and processes running on the server.

Typically, only one of the bottom two options in the control panel is employed for an enterprise server implementation and thus, they are the only two detailed here:

- ▶ **Maximize throughput for file sharing**

This option is the default setting. It instructs the operating system to give the working set of the file system cache a higher priority for memory allocation than the working sets of applications. It will yield the best performance in a file server environment that is not running other applications.

If other applications are running on the server, it will require sufficient physical memory to obtain the maximum performance gain because more memory is set aside for the file system cache than for applications. As a result, the option “maximize throughput for network applications” is typically used. This “file sharing” option might, however, be the best option to use on servers with large quantities of physical memory as described next.

- ▶ **Maximize throughput for network applications**

This choice is the recommended setting for machines running applications that are memory-intensive. With this option chosen, the working set of applications will have a priority over the working set of the file system cache. This setting is normally the best setting to use for all servers *except* those with a) dedicated file servers or with applications exhibiting file server-like characteristics; or b) those with significant amounts of memory (see below).

The second control panel used for managing the file system cache in Windows Server 2003 is within the System applet:

1. Click **Start → Control Panel → System**.
2. Select the **Advanced** tab.
3. Within the **Performance** frame, click **Settings**.

4. Select the **Advanced** tab. The window shown in Figure 13-4 opens.

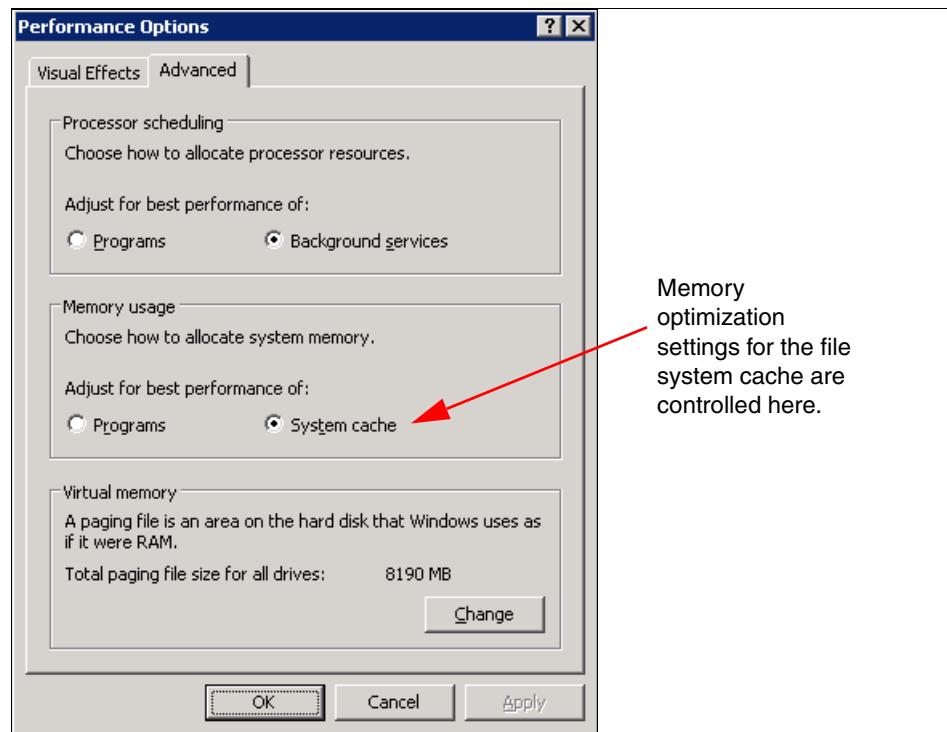


Figure 13-4 System applet: memory usage

The System applet reflects and modifies the value included within the same LanmanServer “LargeSystemCache” registry key as described for File and Printer Sharing in the Network applet in Figure 13-3 on page 367. Making a change to the LargeSystemCache through this applet, however, does so without impacting the MemoryManagement “Size” value that File and Printer Sharing does.

Given that most users only use the “Maximize throughput for network applications” or the “Maximize throughput for file sharing” options for enterprise servers, the Size value remains same, a value of 3. This setting means that using the System applet to adjust the LargeSystemCache value is redundant because it is just as easily set by using File and Print Sharing. For this reason, we recommend that you use the first control panel as previously described and leave this second control panel untouched.

It would seem that the only advantage to using both Control Panel applets in conjunction would allow you to have the applets actually indicate Maximize throughput for network applications and simultaneously indicate memory

usage favors System cache. This same effect to the registry is achieved by selecting **Maximize throughput for file-sharing** (as per Table 13-4 on page 367)—visually, it simply does not say “Maximize throughput for network applications.” If you want this change purely for aesthetic reasons, then make sure you set the first Network applet *before* the second System applet because the first selection overrides the second selections, but the reverse does *not* occur.

### 13.6.1 Servers with large amounts of free physical memory

How much memory is able to be allocated to the file system cache depends on how much physical memory exists in the server and the file system cache option selected in the preceding dialogs.

With Windows Server 2003, when **Maximize data throughput for file sharing** is selected (LargeSystemCache set to 1), the maximum the file system cache can grow to is 960 MB. When **Maximize throughput for network applications** is selected (LargeSystemCache set to 0), then the maximum that the file system cache can grow to is 512 MB (refer to Microsoft KB article 837331 at the following URL for more information about this topic).

Depending on the selection you make here, it is possible that adding more physical memory up to this point will allow the file system cache to grow even larger, up to these stated maximums.

On a server with physical memory from, for example, 2 GB and upwards, it might be preferable to leave the “Maximize data throughput for file sharing” option selected, providing that the total amount of memory used by the operating system and server applications does *not* exceed the amount of physical RAM minus 960 MB. In fact, any application server that can be determined to have 960 MB or more of RAM unused will likely be given a performance boost by enabling the large system cache.

By enabling this, all the disk and network I/O performance benefits of using a large file system cache are realized and the applications running on the server continue to run without being memory constrained.

Some applications have their own memory management optimizers built into them, including Microsoft SQL Server and Microsoft Exchange. In such instances, the setting above is best set to **Maximize throughput for network applications**, and then let the application manage memory and its own internal system cache as it deems appropriate.

See Microsoft Knowledge Base article 837331 for more information:

<http://support.microsoft.com/?kbid=837331>

**Note:** Keep in mind that the maximum size of the file system cache increases from 960 MB in the 32-bit (x86) edition of Windows Server 2003 to 1 TB in the 64-bit (x64) editions. This has potential to yield enormous performance improvements on systems where the file system cache is actively used.

## 13.7 Disabling or removing unnecessary services

When Windows is first installed, many services are enabled that might not be necessary for a particular server. Although in Windows Server 2003 many more services are disabled by default than in previous editions of the server operating system, there still remains on many systems an opportunity for improving performance further by examining running services.

Inexperienced users might also, when installing or updating the operating system, inadvertently add additional services that are not actually required for a given system. Each service requires system resources and for this reason, it is best to disable unnecessary services to improve performance.

However, use care when disabling services. Unless you are completely certain of the purpose of a given service, research it further before choosing to disable it. Disabling some services that the operating system requires to be running can render a system inoperable and possibly unable to boot.

To view the services running in Windows, complete the following steps:

1. Right-click **My Computer** and select **Manage**.
2. Expand the Services and Applications icon.
3. Select the Services icon.
4. Click the Standard tab at the bottom of the right pane. A window similar to Figure 13-5 on page 372 opens. All the services installed on the system are displayed. The status, running or not, is shown in the third column.
5. Click twice on **Status** at the top of the third column. This sorts together all running (Started) services from those that are not running.

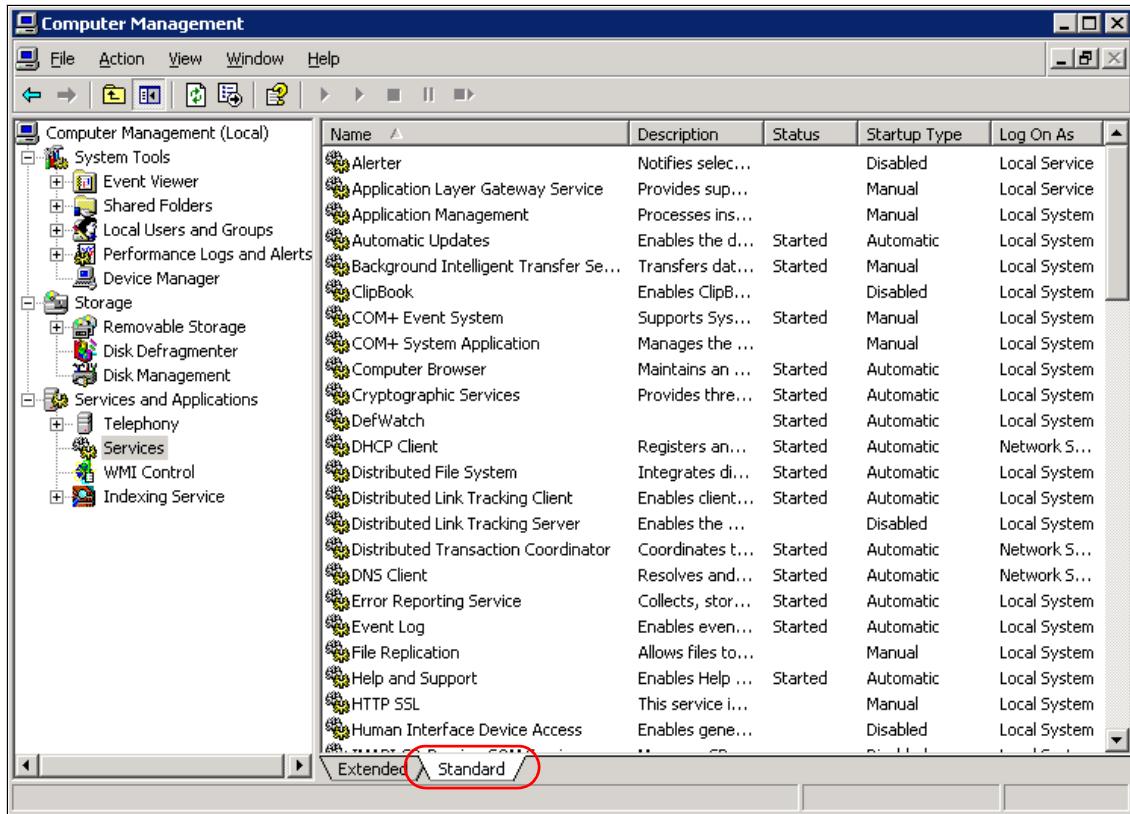


Figure 13-5 Windows Services

From this dialog, all services that are not required to be running on the server should be stopped and disabled. This will prevent the service from automatically starting at system boot time. To stop and disable a service, do the following:

1. Right-click the service and click **Properties**.
2. Click **Stop** and set the Startup type to **Disabled**.
3. Click **OK** to return to the Services window.

If a particular service has been installed as part an application or Windows component and is not actually required on a given server, a better approach is to remove or uninstall this application or component altogether. This is typically performed through the **Add or Remove Programs** applet in Control Panel.

Some services might not be required at system boot time but might be required to start by other applications or services at a later time. Such services should be set to have a startup type of **Manual**. Unless a service is explicitly set to have a

startup type of **Disabled**, it can start at any time and perhaps unnecessarily use system resources.

Windows Server 2003 comes installed with many services that Windows 2000 Server and Windows NT Server did not. Designed as a significantly more secure operating system than its predecessors, many of the services have their startup type set to Disabled or Manual by default. Nonetheless, there remain several services enabled on a standard installation that can likely be disabled on many servers. For example, the Print Spooler service is enabled by default but is not usually required if the server is not functioning as a print server or does not have local printing requirements.

Table 13-5 lists services on a standard Windows Server 2003 installation that should be reviewed for their requirement on your systems. This is not a definitive list of all services, it simply shows those that should be considered for their applicability on an enterprise server. This list includes only services that are not already disabled by default on Windows Server 2003 and might therefore be candidates for disabling.

Be aware that these services might still be required for your environment, depending on the particular function of the server and the applications it is running. For example, the File Replication service (FRS) is normally required on an Active Directory domain controller, but its inclusion with other server types should be questioned. Each server is different and implementing the following recommendations should be tested before changing.

*Table 13-5 Windows service startup recommendations*

Service	Default startup type	Recommended setting
Application Management	Manual	Disabled
Alerter	Automatic	Disabled
Clipbook	Disabled	Disabled
Computer Browser	Automatic	Disabled
Distributed file system	Automatic	Disabled
Distributed link tracking client	Automatic	Disabled
Distributed transaction coordinator	Automatic	Manual
Error Reporting Service	Automatic	Disabled
Fax Service	Manual	Disabled
File Replication	Manual	Disabled

Service	Default startup type	Recommended setting
Help and Support	Automatic	Disabled
HTTP SSL	Manual	Disabled
License Logging	Manual	Disabled
Logical Disk Manager	Automatic	Manual
Messenger	Automatic	Disabled
Portable Media Serial Number Service	Manual	Disabled
Shell Hardware Detection	Automatic	Disabled
Windows Audio	Automatic	Disabled
Wireless Configuration	Automatic	Disabled

## 13.8 Removing unnecessary protocols and services

Windows servers often have more network services and protocols installed than are actually required for the purpose or application for which they have been implemented. Each additional network client, service, or protocol places additional overhead on system resources. In addition, each protocol generates network traffic. By removing unnecessary network clients, services and protocols, system resources are made available for other processes, excess network traffic is avoided, and the number of network bindings that must be negotiated is reduced to a minimum.

TCP/IP is largely viewed as the de facto enterprise network protocol in modern networks. Unless integration with other systems is required, it is likely sufficient now to just have TCP/IP loaded as the only network protocol on your server.

To view the currently installed network clients, protocols and services:

1. Click **Start** → **Control Panel** → **Network Connections**.
2. While still in the Start menu context, right-click **Network Connections** and choose **Open**.
3. Click **Properties**.
4. Right-click **Local Area Connection** (or the entry for your network connection).
5. Click **Properties**. The window shown in Figure 13-6 on page 375 opens.

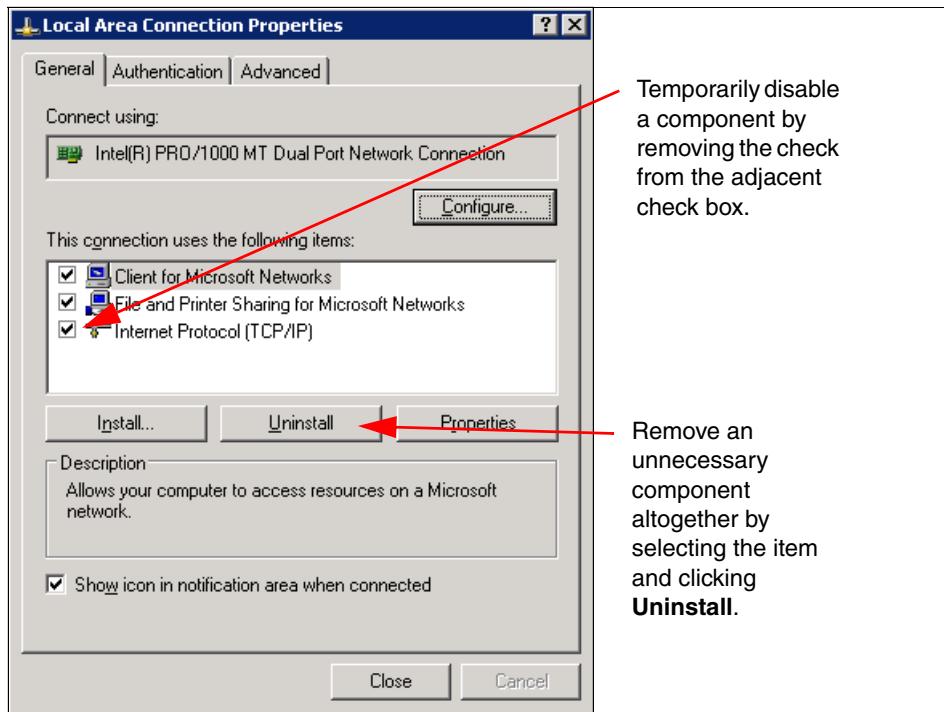


Figure 13-6 Network clients, services and protocols

To remove an unnecessary item, select it and click **Uninstall**. To **disable** the item temporarily without completely uninstalling it, simply remove the check from the check box beside it. This approach (disable rather than uninstall) might be a more appropriate method in determining which services, protocols and clients are actually required on a system. When it has been determined that disabling an item has no adverse affect on the server, it can then be uninstalled.

In many instances, the three components listed in Figure 13-6 are often sufficient for a file and print server on a standard TCP/IP-based network. That is:

- ▶ Client for Microsoft Networks
- ▶ File and Printer Sharing for Microsoft Networks
- ▶ Internet Protocol (TCP/IP)

## 13.9 Optimizing the protocol binding and provider order

Optimizing the protocol binding order and the provider order can also make a difference to performance.

### Protocol binding order

On a system supporting more than one network protocol, the order in which they are bound to the network clients and services running on the server is important. All network communications for a given service or client start with the protocol listed at the top of the binding list. If after a given period, no response is received, then communications are routed to the next protocol in the list until all protocols are exhausted.

As a result, it is crucial to ensure that the most frequently used protocol for a given client or service is moved to the top of the binding list to offer the best network I/O performance possible.

To view the order of network bindings, do the following:

1. Click **Start** → **Control Panel** → **Network Connections**.
2. While still in the Start menu context, right-click Network Connections and choose **Open**.
3. Click **Properties**.
4. From the menu bar, click **Advanced** → **Advanced Settings**. The window shown in Figure 13-7 on page 377 opens.

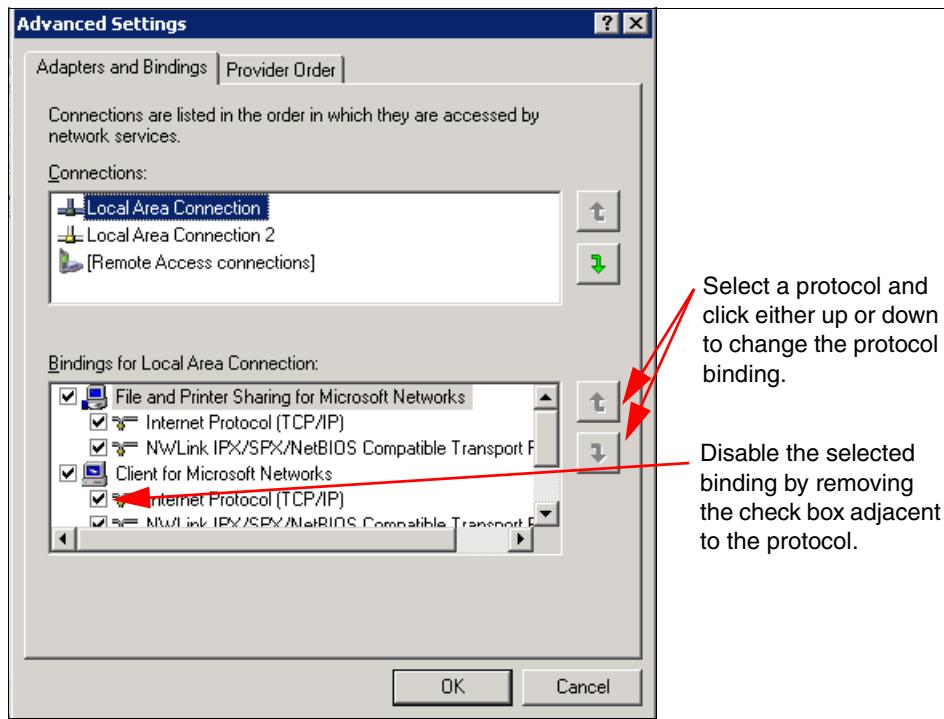


Figure 13-7 Windows protocol binding order

By selecting a protocol and clicking the up and down buttons, you can change the binding priority of your protocols.

If an installed protocol is not required by a particular service or client, it should be disabled. Do so by removing the check in the check box next to the protocol in question. This will improve system performance and possibly improve security.

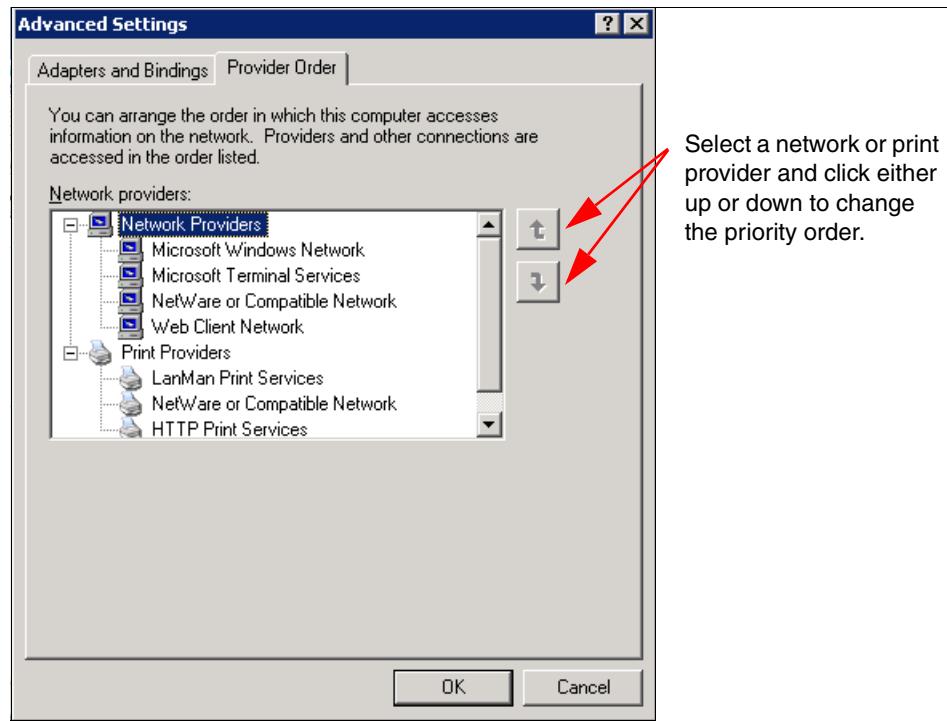
### Network and print provider order

Servers will often have multiple network and print providers installed. Similar to network bindings, the order in which these are configured will determine how quickly they respond to client requests for services running on the server. It will also affect how quickly the server itself connects to hosts when functioning as a client. The most commonly used network providers should be moved to the top of the list with the remaining ones ordered down by decreasing priority.

To access the network provider order configuration:

1. Click **Start** → **Control Panel** → **Network Connections**.

2. While still in the Start menu context, right-click **Network Connections** and choose **Open**.
3. Click **Properties**.
4. From the menu bar, click **Advanced → Advanced Settings**.
5. Select the **Network Provider** tab. The window shown in Figure 13-8 opens.



*Figure 13-8 Windows network provider order*

By selecting a network or print provider and clicking the up and down buttons, you can change the order in which the computer responds to client requests.

## 13.10 Optimizing network card settings

Many network interface cards in servers today have settings that can be configured through the Windows interface. Setting these optimally for your network environment and server configuration can significantly affect the performance of network throughput. Of all the performance tuning features outlined in this chapter, it is the ones in this section that have been noted to have the biggest improvement on system performance and throughput.

To access this range of settings, following these steps:

1. Click **Start** → **Settings** → **Network Connections**.
2. Click **Properties**.
3. Right-click **Local Area Connection** (or the name of your network connection).
4. Click **Properties**. The window shown in Figure 13-9 opens.

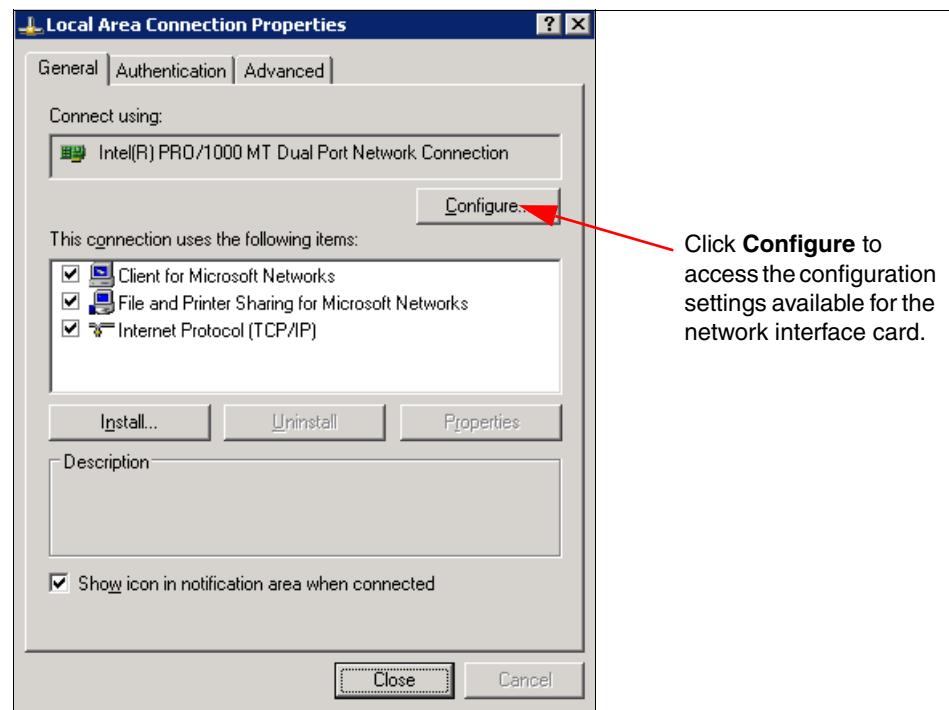


Figure 13-9 Accessing the network interface card configuration

5. Click **Configure**.

6. Click the **Advanced** tab. A dialog box similar to that in Figure 13-10 opens, depending on the network adapter your system is using.

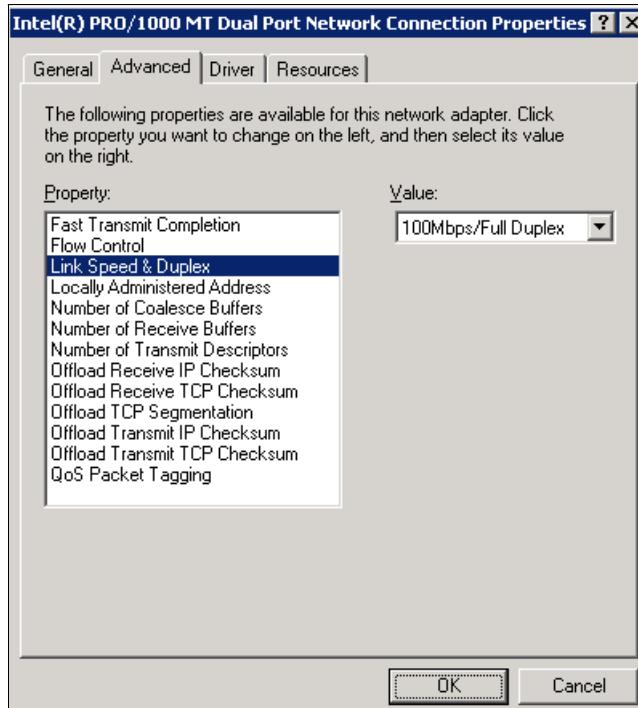


Figure 13-10 Network interface card advanced settings configuration

The exact configuration settings available differ from one network interface card to another. However, a handful of settings are common between most Intel-based cards in the IBM System x range of servers.

**Note:** You apply these settings for each physical network interface, including the individual cards within a set *teamed* of interfaces that are configured for aggregation, load balancing, or fault tolerance. With some teaming software, you might need to apply these settings to the team also. Note also that some network interface cards are largely self-tuning and do not offer the option to configure parameters manually.

The following settings are the ones that can have the most dramatic impact to performance:

► **Link Speed and Duplex**

Experience suggests that the best practice for setting the speed and duplex values for each network interface in the server is to configure them in one of two ways:

- Set to auto-negotiation if, and only if, the switch port is also set to auto negotiation also. The server and switch should then negotiate the fastest possible link speed and duplex settings.
- Set to the same link speed and same duplex settings as those of the switch. These settings will, of course, normally yield the best performance if set to the highest settings that the switch will support.

We do *not* recommend the use of auto-negotiation on the server network interface combined with manually setting the parameter on the switch, or vice versa. Using such a combination of settings at differing ends of the network connection to the server has often found to be the cause of poor performance and instability in many production environments and should definitely be avoided.

To reiterate, use either auto-negotiation at both interfaces, or hard-code the settings at both interfaces, but not a mix of both of these.

For more information, see the following Cisco Web site:

[http://www.cisco.com/warp/public/473/46.html#auto\\_neg\\_valid](http://www.cisco.com/warp/public/473/46.html#auto_neg_valid)

► **Receive Buffers**

This setting specifies the number of memory buffers used by the network interface driver when copying data to the protocol memory. It is normally set by default to a relatively low setting. We recommend setting this value as high as possible for maximum performance gains. On servers low on physical memory, this can have a negative impact because these buffers are taken from available physical memory on the system. On most modern systems, however, the maximum setting can be implemented without any notable impact to memory resources.

The amount of memory used by modifying this setting can easily be determined by watching the appropriate metrics in the Task Manager or System Monitor before and after making the changes. Monitor this impact before making the change permanent.

► **Coalesce Buffers**

Map registers are system resources used in physical-to-virtual address conversion with bus mastering cards like the ones in some IBM System x servers. Coalesce buffers are those available to the network driver if the driver

runs out of map registers. We recommend setting this value as high as possible for maximum performance gains. Note that the same impact to memory as with receive buffers is possible when increasing the number of coalesce buffers.

- ▶ **Transmit Descriptors/Transmit Control Blocks**

This setting specifies how many transmit control buffers the driver allocates for use by the network interface. This directly reflects the number of outstanding packets the driver can have in its “send” queue. We recommend setting this value as high as possible for maximum performance gains. Note that the same impact to memory as with receive buffers is possible when increasing the number of transmit descriptors and transmit control blocks.

- ▶ **Offload features**

In almost all instances there will be benefit derived from enabling network interface offload features. In some cases the network interface might *not* be able to handle the offload capabilities at high throughput. However, as a general rule, enabling offload will benefit overall system performance. Some network interfaces have separate options or parameters to enable or disable offloading for send and receive traffic.

Other advanced settings are often available with network interface cards in addition to those described here. Consult the documentation for the network interface to understand the meaning and impact of changing each setting.

Where possible, use these settings to move network processing requirements away from the server itself and to the network interface. That is, “offload” the network requirements from the server CPU where possible and try to have the network interface do as much of the processing as it can. This will ensure optimal performance.

## 13.11 Process scheduling, priority levels, and affinity

The scheduler is a component of the Windows operating system kernel. It is the scheduler that coordinates the servicing of the processes and their threads that are waiting and ready to use the system CPUs. The kernel schedules ready threads based upon their individual *dynamic priority*. The dynamic priority is a number between 0 and 31 that determines the importance of threads relative to one another. The higher the priority value, the higher the priority level. For example, a thread with a priority of 15 is serviced more quickly than a thread with a priority of 10.

Even if it requires preempting a thread of lower priority, threads with the highest priority always run on the processor. This activity ensures that Windows pays

attention to critical system threads required to keep the operating system running. A thread will run on the processor for either the duration of its CPU quantum (or time slice, described in 13.2, “Windows Server 2003 - 64-bit (x64) Editions” on page 354) or until it is preempted by a thread of higher priority.

Task Manager allows you to easily see the priority of all threads running on a system. To do so, open Task Manager, and click **View** → **Select Columns**, then select **Base Priority**, as shown in Figure 13-11.

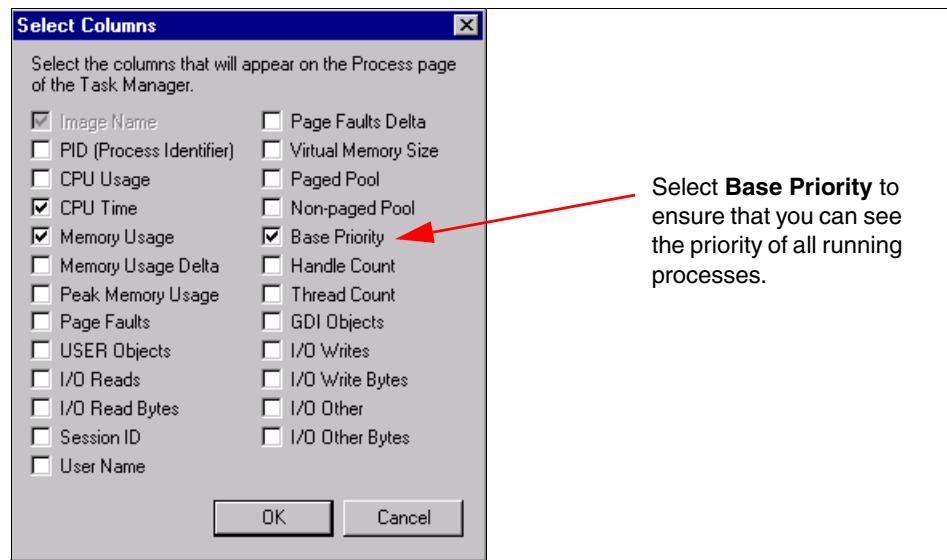
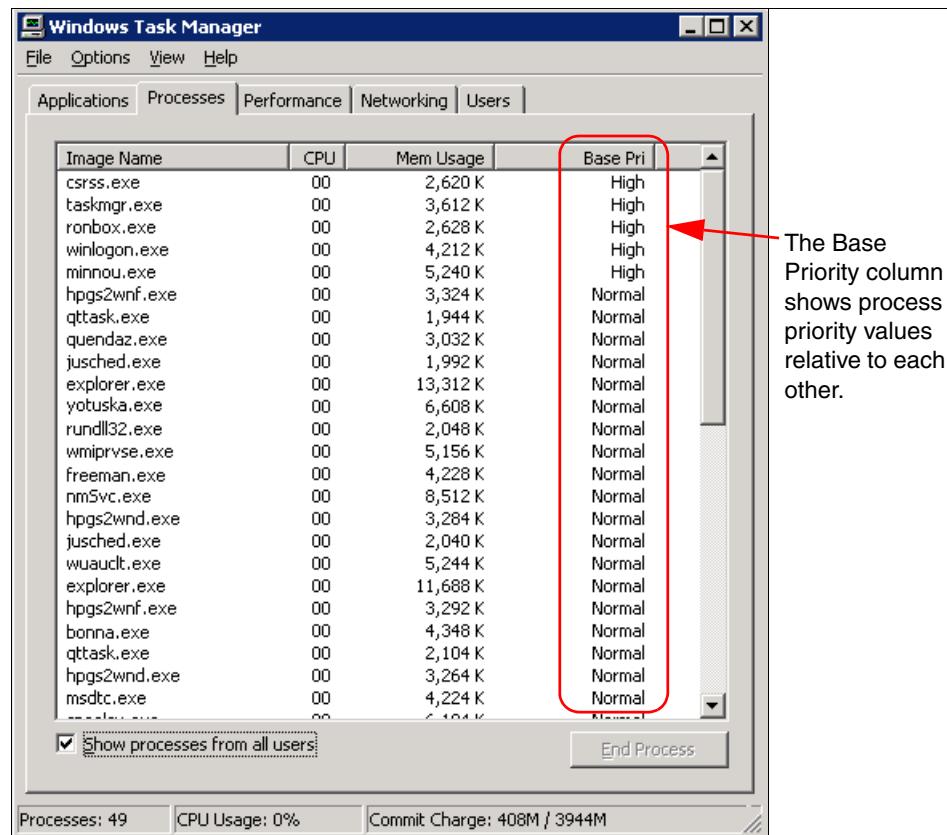


Figure 13-11 Selecting Base Priority in Task Manager

This displays a column in Task Manager, as shown in Figure 13-12, that allows you to see the relative priority of processes running on the system.



The image shows the Windows Task Manager window with the "Processes" tab selected. A red box highlights the "Base Pri" column, which contains values like "High" and "Normal". An arrow points from the text "The Base Priority column shows process priority values relative to each other." to the red box. The table lists various system processes with their base priority levels.

Image Name	CPU	Mem Usage	Base Pri
csrss.exe	00	2,620 K	High
taskmgr.exe	00	3,612 K	High
ronbox.exe	00	2,628 K	High
winlogon.exe	00	4,212 K	High
minnou.exe	00	5,240 K	High
hpgs2wnf.exe	00	3,324 K	Normal
qtask.exe	00	1,944 K	Normal
qundaz.exe	00	3,032 K	Normal
jusched.exe	00	1,992 K	Normal
explorer.exe	00	13,312 K	Normal
yotuska.exe	00	6,608 K	Normal
rundll32.exe	00	2,048 K	Normal
wmiprvse.exe	00	5,156 K	Normal
freeman.exe	00	4,228 K	Normal
nm5vc.exe	00	8,512 K	Normal
hpgs2wnd.exe	00	3,284 K	Normal
jusched.exe	00	2,040 K	Normal
wuauctl.exe	00	5,244 K	Normal
explorer.exe	00	11,688 K	Normal
hpgs2wnf.exe	00	3,292 K	Normal
bonna.exe	00	4,348 K	Normal
qtask.exe	00	2,104 K	Normal
hpgs2wnd.exe	00	3,264 K	Normal
msdtc.exe	00	4,224 K	Normal
	00	6,104 K	Normal

Show processes from all users

Processes: 49 CPU Usage: 0% Commit Charge: 408M / 3944M

Figure 13-12 Windows Task Manager displaying the Base Priority column

Most applications loaded by users run at a *normal* priority, which has a base priority value of 8. Task Manager also allows the administrator the ability to change the priority of a process, either higher or lower.

To change the priority of a process, right-click the process in question and click **Set Priority** from the drop-down menu as shown in Figure 13-13 on page 385. Then click the new priority that you want to assign to the process.

**Note:** This procedure changes the priority of actual processes running on the system, but the change only lasts as long as the life of the selected process.

If you want to launch a process with a non-normal priority, you can do so using the START command from a command-prompt. Type START /? for more information about how to do this.

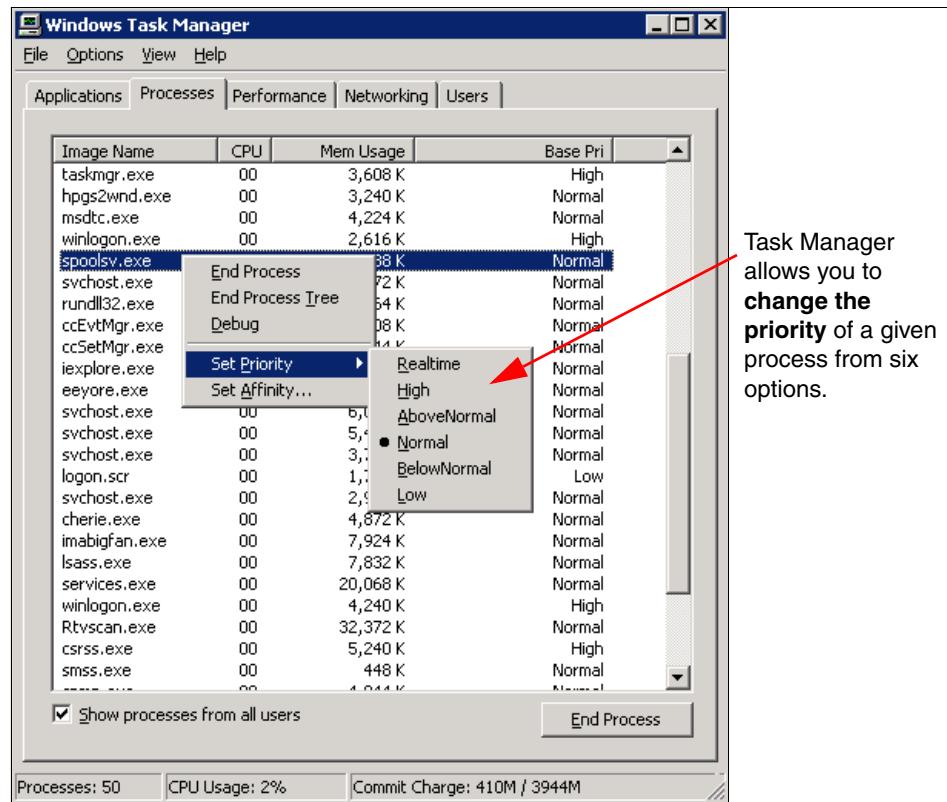


Figure 13-13 Changing the priority of a process using Task Manager

Threads, as a sub-component of processes, inherit the base priority of their parent process. The four priority classes are:

- ▶ Idle
- ▶ Normal
- ▶ High
- ▶ Realtime

Each process's priority class sets a range of priority values (between 1 and 31), and the threads of that process have a priority within that range. If the priority class is Realtime (priorities 16 to 31), the thread's priority can never change while it is running. A single thread running at priority 31 will prevent all other threads from running.

Conversely, threads running in all other priority classes are variable, meaning the thread's priority can change while the thread is running. For threads in the Normal or High priority classes (priorities 1 through 15), the thread's priority can be raised or lowered by up to a value of 2 but cannot fall below its original, program-defined base priority.

When should you modify the priority of a process? In most instances, do so as rarely as possible. Windows normally does a very good job of scheduling processor time to threads. Changing process priority is not an appropriate long-term solution to a bottleneck on a system. If you are suffering performance problems related to processes not receiving sufficient processing time, eventually additional or faster processors will be required to improve the situation.

Normally the only conditions under which the priority of a process should be modified are when the system is CPU-bound. Processor utilization, queue length, and context switching can all be measured using System Monitor to help identify processor bottlenecks.

In a system with plenty of spare CPU capacity, testing has shown that changing the base priority of a process offers marginal, if any, performance improvements. This is because the processor is comfortable with the load it is under and able to schedule time appropriately to threads running on the system. Conversely, on a system suffering heavy CPU-load, CPU time being allocated to nominated processes will likely benefit from changing the base priority. On extremely busy systems, threads with the lowest priority will be serviced infrequently, if at all.

Modifying process priorities can be an effective troubleshooting technique for solving short-term performance problems; however, it is rarely an adequate long-term solution.

**Important:** Be aware that changing priorities might destabilize the system. Increasing the priority of a process might prevent other processes, including system services, from running. In particular, be careful not to schedule many processes with the High priority and avoid using the Realtime priority altogether. Setting a processor-bound process to Realtime could cause the computer to stop responding altogether.

Decreasing the priority of a process might prevent it from running at all, and not merely force it to run less frequently. In addition, lowering priority does not

necessarily reduce the amount of processor time a thread receives; this happens only if it is no longer the highest-priority thread.

### 13.11.1 Process affinity

On symmetric multi-processing (SMP) systems, the Windows scheduler distributes the load of ready threads over all available processors based on thread priority. Even though Windows will often try to associate known threads with a specific CPU (called *soft affinity*), threads invariably end up distributed among multiple processors.

*Hard affinity* can be applied to permanently bind a process to a given CPU or set of CPUs, forcing the designated process to always return to the same processor. The performance advantage in doing this is best seen in systems with large Level 2 caches as the cache hit ratio will improve dramatically.

Assigning hard processor affinity to assign processes to CPUs is not typically used as a method for improving system performance. The only circumstances under which it will occasionally be employed are those servers where multiple instances of the same application are running on the same system, such as SQL Server or Oracle. As with all tuning techniques, the performance of the system should be measured to determine whether using process affinity has actually offered any tangible benefit. Determining the correct mix of processes assigned to the CPUs in the system can be time-consuming.

Some applications, like SQL Server, provide internal options to assign themselves to specific CPUs. The other method for setting affinity is through Task Manager, as explained here:

1. Right-click the process in question.
2. Click **Set Affinity** as shown in Figure 13-14 on page 388.
3. Select the CPUs to which you want to restrict the process and click **OK**.

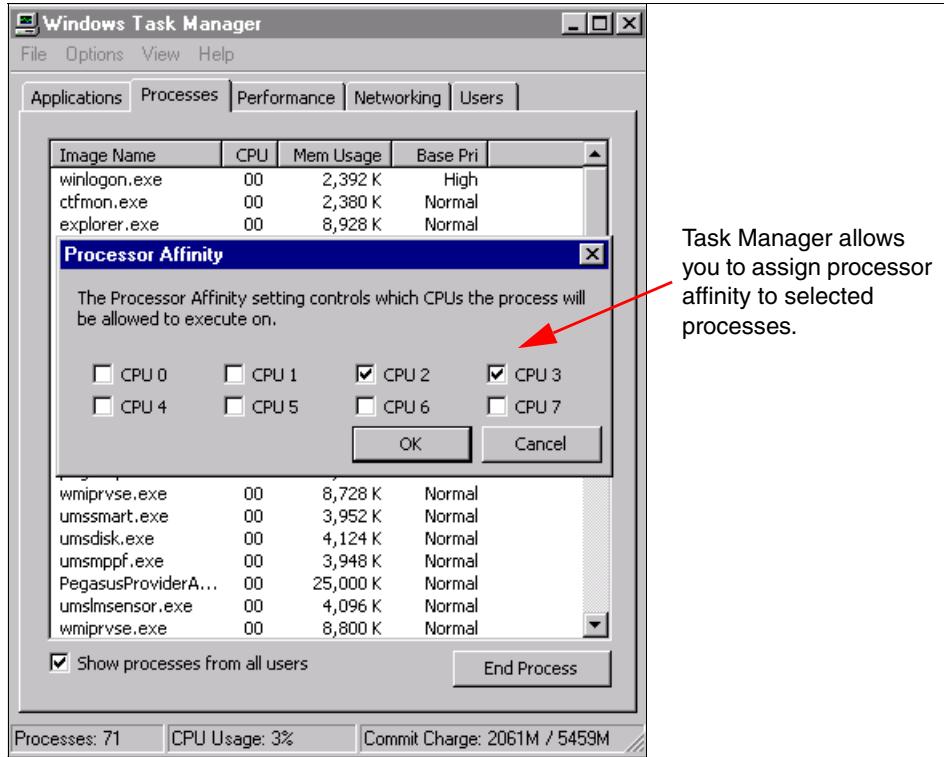


Figure 13-14 Assigning Processor Affinity to a selected process

Note that like changing the process's priority, changing process affinity in this manner will only last for the duration of the process. If the process ends or the system is rebooted, the affinity will need to be reallocated as required. Note also that not all processes permit affinity changes.

Also be aware that the **Set Affinity** option in the Task Manager application will only display in the context menu on a system with multiple logical or physical processors, including processors with multiple cores.

## 13.12 Assigning interrupt affinity

Microsoft offers a utility called Intfiltr that allows the binding of device interrupts to specific system processors. This partitioning technique can be employed to improve system performance, scaling, and the partitioning of large servers.

Specifically for server performance tuning purposes, Intfiltr allows you to assign the interrupts generated by each network adapter to a specific CPU. Of course, it

is only useful on SMP systems with more than one network adapter installed. Binding the individual network adapters in a server to a given CPU can provide large performance efficiencies.

Intfiltr uses plug-and-play features of Windows that permit affinity for device interrupts to particular processors. Intfiltr binds a filter driver to devices with interrupts, and is then used to set the affinity mask for the devices that have the filter driver associated with them. This permits Windows to have specific device interrupts associated with nominated processors.

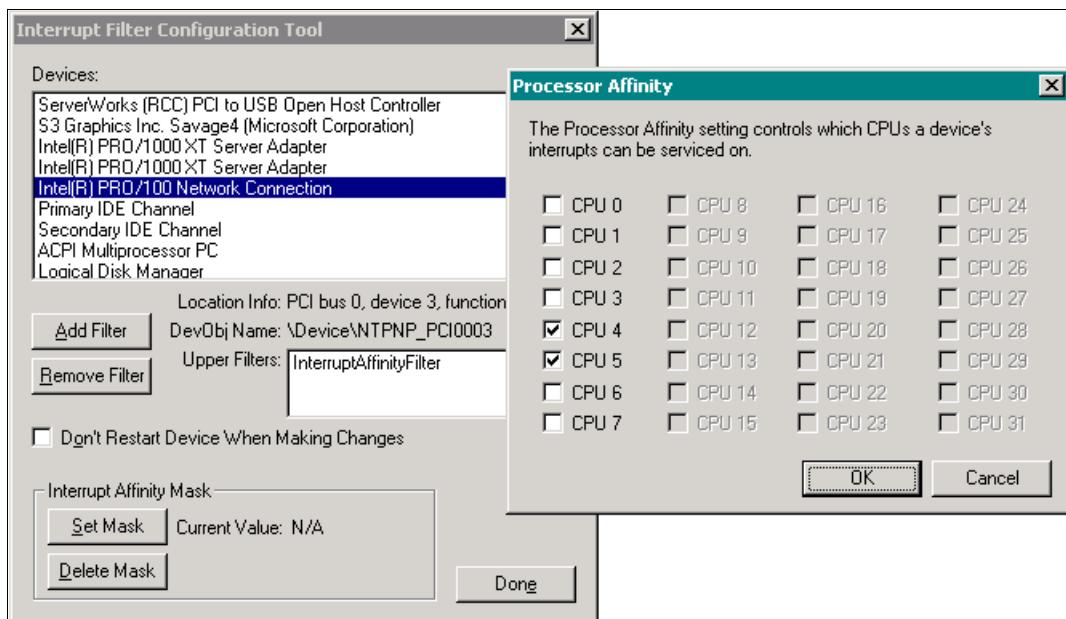


Figure 13-15 Assigning processor affinity using the INTFILTR tool

Interrupt filtering can affect the overall performance of your computer, in both a positive and negative manner. Under normal circumstances, there is no easy way to determine which processor is best left to handle specific interrupts.

Experimentation and analysis will be required to determine whether interrupt affinity has yielded performance gains. To this end, by default, without tools like Intfiltr, Windows directs interrupts to any available processor.

Note that some consideration needs to be made when configuring Intfiltr on a server with CPUs that supports Hyper-Threading to ensure that the interrupts are assigned to the correct physical processors desired, and not to the logical processors. Assigning interrupt affinity to two logical processors that actually refer to the same physical processor will obviously offer no benefit and can even detract from system performance.

Interrupt affinity for network cards can offer definite performance advantages on large, busy servers with many CPUs. Our recommendation is to try Intfiltr in a test environment to associate specific interrupts for network cards with selected processors.

This test environment should simulate your production environment as closely as possible, including hardware, operating system, and application configuration. This will allow you to determine if using interrupt affinity is going to offer a performance advantage for your network interfaces.

**Note:** You can use Intfiltr to create an affinity between CPUs and devices other than network cards, such as disk controllers. Experimentation again is the best way to determine potential performance gains. To determine the interrupts of network cards or other devices, use Windows Device Manager or, alternately, run System Information (WINMSD.EXE).

The Intfiltr utility and documentation is available free of charge from Microsoft:

<ftp://ftp.microsoft.com/bussys/winnt/winnt-public/tools/affinity/intfiltr.zip>

For more information, see:

<http://support.microsoft.com/?kbid=252867>

### 13.13 The /3GB BOOT.INI parameter (32-bit x86)

By default, the 32-bit (x86) editions of Windows can address a total of 4 GB of virtual address space. This is a constraint of the 32-bit (x86) architecture.

Normally, 2 GB of this is reserved for the operating system kernel requirements (privileged-mode) and the other 2 GB is reserved for application (user-mode) requirements. Under normal circumstances, this creates a 2 GB per-process address limitation.

Windows provides a /3GB parameter to be added to the BOOT.INI file that reallocates 3 GB of memory to be available for user-mode applications, and reduces the amount of memory for the system kernel to 1 GB. Some applications, such as Microsoft Exchange and Microsoft SQL Server, written to do so, can derive performance benefits from having large amounts of addressable memory available to individual user-mode processes. In such instances, having as much free space for user-mode processes is desirable.

Given the radically increased memory capability of 64-bit (x64) operating systems, neither the /3GB switch nor the /PAE switch (described in 13.14, “Using

PAE and AWE to access memory above 4 GB (32-bit x86)" on page 391) are for use on the 64-bit (x64) editions of the Windows Server 2003 operating system.

To edit the BOOT.INI file to make this change, complete the following steps:

1. Open the System Control Panel.
2. Select the Advanced tab.
3. Within the Startup and Recovery frame, click **Settings**.
4. Click **Edit**. Notepad opens, and you can edit the current BOOT.INI file.
5. Edit the current ARC path to include the /3GB switch, as shown in Figure 13-16.
6. To have the change take effect, restart the server.

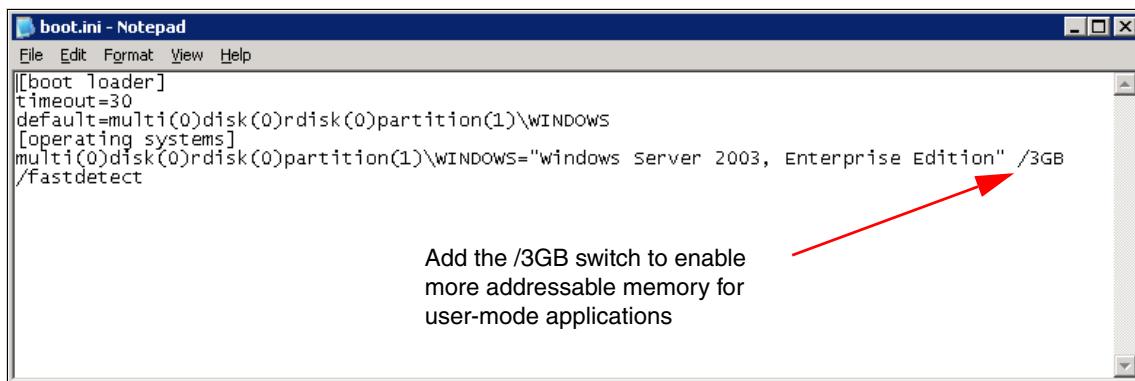


Figure 13-16 Editing the BOOT.INI to include the /3GB switch

You normally use this switch only when a specific application recommends its use. Typically, you use it where applications have been compiled to use more than 2 GB per process, such as some components of Exchange.

For more information, see:

<http://support.microsoft.com/kb/291988>  
<http://support.microsoft.com/kb/851372>  
<http://support.microsoft.com/kb/823440>

## 13.14 Using PAE and AWE to access memory above 4 GB (32-bit x86)

As described in 13.13, "The /3GB BOOT.INI parameter (32-bit x86)" on page 390, the native 32-bit architecture of the x86 processor allows a maximum

addressable memory space of 4 GB. The Intel Physical Address Extension (PAE) is a 36-bit memory addressing mode that allows 32-bit (x86) systems to address memory above 4 GB.

PAE requires appropriate hardware and operating system support to be implemented. Intel introduced PAE 36-bit physical addressing with the Intel Pentium® Pro processor. Windows has supported PAE since Windows NT Server 4.0 Enterprise Edition, and it is supported with the Advanced and Datacenter Editions of Windows 2000 Server and the Enterprise and Datacenter Editions of Windows Server 2003.

Windows uses 4 KB pages with PAE to map up to 64 GB of physical memory into a 32-bit (4 GB) virtual address space. The kernel effectively creates a “map” in the privileged mode addressable memory space to manage the physical memory above 4 GB.

The 32-bit (x86) editions of Windows Server 2003 allow for PAE through use of a /PAE switch in the BOOT.INI file. This effectively allows the operating system to use physical memory above 4 GB. Because the 64-bit (x64) editions of Windows are not bound by this same memory architecture constraint, the PAE switch is not used in these versions of the Windows Server 2003 operating system.

Even with PAE enabled, the underlying architecture of the system is still based on 32-bit linear addresses. This effectively retains the usual 2 GB of application space per user-mode process and the 2 GB of kernel mode space because only 4 GB of addresses are available. However, multiple processes can immediately benefit from the increased amount of addressable memory because they are less likely to encounter physical memory restrictions and begin paging.

Address Windowing Extensions (AWE) is a set of Windows APIs that take advantage of the PAE functionality of the underlying operating system and allow applications to directly address physical memory above 4 GB.

Some applications like SQL Server 2000 Enterprise Edition have been written with these APIs, and they can harness the significant performance advantages of being able to address more than 2 GB of memory per process. More recent applications, like SQL Server 2005 x64 Editions, have been written to take advantage of the 64-bit (x64) memory architecture and can deliver enormous performance gains over their 32-bit (x86) counterparts. They do not need to use AWE to address memory above 4 GB.

To edit the BOOT.INI file to enable PAE, complete the following steps:

1. Open the System Control Panel.
2. Select the Advanced tab.
3. Within the Startup and Recovery frame, click **Settings**.

4. Click **Edit**. Notepad opens, and you can edit the current BOOT.INI file.
5. Edit the current ARC path to include the /PAE switch as shown in Figure 13-17.
6. To have the change take effect, restart the server.

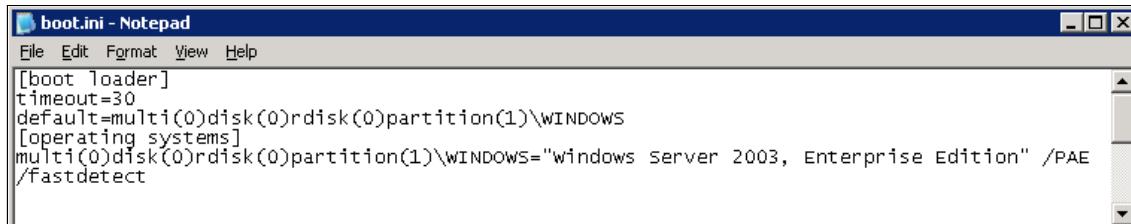


Figure 13-17 Editing the BOOT.INI to include the /PAE switch

For more information, see:

<http://support.microsoft.com/kb/283037>  
<http://support.microsoft.com/kb/268363>  
<http://support.microsoft.com/kb/823440>

### 13.14.1 Interaction of the /3GB and /PAE switches

There is often confusion between when to use the /3GB switch and when to use the /PAE switch in the BOOT.INI file. In some cases it is desirable to use both.

Recall the following information previously covered:

- ▶ The /3GB switch reallocates the maximum 4 GB addressable memory from the normal 2 GB for user-mode applications and 2 GB for the kernel to allow 3 GB of physical memory to be used for applications, leaving 1 GB for the system kernel.
- ▶ PAE permits the operating system to see and make use of physical memory above and beyond 4 GB. This is achieved through the use of the 1 or 2 GB of kernel addressable memory (depending on the use of the /3GB switch) to “map” and manage the physical memory above 4 GB.
- ▶ Applications written using AWE make use of PAE to allow individual applications (processes) to use more than the 2 GB limitation per process.

On a server with between 4 GB and 16 GB of RAM hosting applications that have been compiled or written with AWE to use more than 2 GB of RAM per process or hosting many applications (processes), each contending for limited physical memory, it would be desirable to use both the /3GB and /PAE switches. This will deliver the best performance possible for such a system.

Servers with more than 16 GB of physical memory should *not* use both the /3GB switch and the /PAE switch. The /PAE switch is obviously required to make use of all physical memory above 4 GB. Remember, however, that PAE uses the kernel addressable memory to manage the physical memory above 4 GB.

When physical memory exceeds 16 GB, the 1 GB of memory allocated to the kernel when the /3GB switch is used is insufficient to manage all the additional physical memory above 4 GB. Thus, only the /PAE switch should be used in such a case to avoid the system running out of kernel memory.

For more information, see:

<http://msdn2.microsoft.com/en-us/library/ms175581.aspx>

## 13.15 TCP/IP registry optimizations

Windows has several registry parameters that can be modified to optimize the performance of the TCP/IP protocol for your environment. In most networks, Windows TCP/IP is tuned by the operating system to achieve maximum performance; however, there might be settings that can be made to improve performance on your network.

When changing these parameters, understand what the parameters are doing and what will be the ramification of changing them. If you find that any of these changes cause throughput of your server to decrease, then you should reverse the changes made.

**Tip:** Several of the registry values described in this chapter do not exist in the registry by default and must be added manually. For these values, there is a system default which is overridden when you create the registry value.

This section and the next section list a series of modifications that can, however, be made to the system registry that under certain circumstances might offer system performance improvements. Almost all of these modifications are ones for which there is no corresponding control panel or other GUI utility built natively into Windows to allow these values to be tuned without using the registry directly.

Modifying the registry is not for the faint-hearted and should only be done by experienced technical people. Incorrectly modifying the registry can cause serious system instability and in some circumstances, stop a server from booting correctly. Before implementing any of the suggestions made here, ensure you completely understand their impact and any downstream effects they might have. Further reading from the Microsoft Web site or elsewhere is recommended to

gain a deeper understanding of the parameter changes listed in this section. Where useful further references exist, these have been provided.

Note that several of these changes might not take effect until the server is rebooted subsequent to making the change. Note too that when several of these changes are made and hard-coded into the registry, they restrict Windows from self-tuning this value to what might, under different circumstances, be a more optimal setting. Thus, any modifications made should only be to affect the server performance for the role it is currently hosting.

**Note:** Tuning Windows TCP/IP settings can have a significant impact on memory resources. Monitoring memory usage is very important if you choose to implement any of the settings suggested this section.

### 13.15.1 TCP window size

The TCP receive window specifies the maximum number of bytes that a sender can transmit without receiving an acknowledgment from the receiver. The larger the window size, the fewer acknowledgements are sent back, and the more optimal the network communications are between the sender and receiver. Having a smaller window size reduces the possibility that the sender will time-out while waiting for an acknowledgement, but will increase network traffic and reduce throughput.

TCP dynamically adjusts to a whole multiple of the maximum segment size (MSS) between the sender and receiver. The MSS is negotiated when the connection is initially set up. By adjusting the receive window to a whole multiple of the MSS, the number of full-sized TCP segments used during data transmission is increased, improving throughput.

By default, TCP will try to automatically negotiate the optimal window size depending on the maximum segment size. It initially starts at 16 KB and can range to a maximum of 64 KB. The TCP window size can also be statically specified in the registry, permitting potentially larger or more efficient values than can be negotiated dynamically.

The maximum TCP window size is normally 65535 bytes (64 KB). The maximum segment size for Ethernet networks is 1460 bytes. The maximum multiple of increments of 1460 that can be reached before exceeding this 64 KB threshold is 62420 bytes. This value of 62420 can thus be set in the registry for optimal performance on high-bandwidth networks. The value does not ordinarily exist in the registry and must be added.

The TcpWindowSize registry value can be set at a global or per-interface level. Interface-level settings will override the global value. To achieve the maximum window size, we recommended that you set this only at the global level.

The registry value recommendation is as follows:

Key:	HKLM\SYSTEM \CurrentControlSet \Services\Tcpip\Parameters
Value:	TCPWindowSize
Data type:	REG_DWORD
Range:	0x0 to 0xFFFF
Default:	0x4470 (17520 bytes, the Ethernet MSS (1470)) multiple closest to 16 K
Recommendation:	0xFAF0 (62420)
Value exists by default:	No, needs to be added.

For more information, see:

<http://support.microsoft.com/?kbid=263088>

<http://support.microsoft.com/?kbid=224829>

### 13.15.2 Large TCP window scaling and RTT estimation (time stamps)

The following TCP features are described in RFC 1323.

For more efficient use of high bandwidth networks, an even larger TCP window size can be used than described in 13.15.1, “TCP window size” on page 395. This feature is new to Windows 2000 and Windows Server 2003 and is referred to as TCP window scaling. It is used to increase the maximum TCP window size from the previous limit of 65535 bytes (64 KB) up to 1073741824 bytes (1 GB).

With large window (scaling window) support enabled, Windows can dynamically recalculate and scale the window size. This enables more data to be transmitted between acknowledgements, thereby increasing throughput and performance.

The amount of time used for round-trip communications between a sender and receiver is referred to by TCP as the round-trip time (RTT). A time stamp option available to TCP improves the accuracy of RTT values by calculating it more frequently. This option is particularly helpful in estimating RTT over longer round-trip WAN links, and will more accurately adjust TCP retransmission time-outs. This time stamp option provides two time stamp fields in the TCP header, one to record the initial transmission time, and the other to record the time on the receiver.

The time stamp option is particularly valuable when window scaling support is enabled to ensure the integrity of the much larger packets being transmitted

without acknowledgement. Enabling time stamps might actually have a slight impact to throughput because it adds 12 bytes to the header of each packet. The value of data integrity versus maximum throughput will thus need to be evaluated. In some instances, such as video streaming, where large TCP window size might be advantageous, data integrity might be secondary to maximum throughput. In such an instance, window scaling support can be enabled without time stamps.

Support for scaling window size and time stamps is negotiated at connection setup between the sender and receiver. Only if both the sender and receiver have support for these features enabled will they be used during data transmission.

A small TCP window size will be negotiated initially and over time, using internal algorithms, the window size will grow to the maximum specified size. To enable support for scaling Windows or improved time stamp (RTT) estimation, make the following changes to the registry.

Key: HKLM\SYSTEM \CurrentControlSet  
  \services\Tcpip\Parameters  
Value: TCP1323Opt  
Data type: REG\_DWORD  
Range: 0x0 - 0x3 (see Table 13-6)  
Default: 0x0  
Recommendation: 0x3  
Value exists by default: No, needs to be added.

For more information, see:

<http://support.microsoft.com/?kbid=224829>

**Note:** The registry entry for this value is a 2-bit bitmask. The lower-bit determines whether scaling is enabled, and the higher-bit determines whether timestamps are enabled.

Table 13-6 Possible entries for TCP1323Opt registry value

TCP1323Opt registry value	Result
0x0	Disable windows scale and time stamps
0x1	Windows scaling enabled only
0x2	Time stamp scaling enabled only
0x3 (recommended)	Windows scaling and time stamps enabled

After TCP1323Opts has been used to enable TCP window scaling, the registry value TCPWindowSize described in 13.15.1, “TCP window size” on page 395 can be increased to value from 64 K (65535 bytes) through to 1 GB (1,073,741,824 bytes). For best performance and throughput, the value set here should be a multiple of the maximum segment size (MSS).

Because the optimal value for TCPWindowSize with window scaling support enabled will be different for each implementation, no specific recommendation is made here. This should be determined through careful testing in your environment. Note that as the window size increases, so does the risk of data corruption and subsequent resends because fewer acknowledgements will be issued from the receiver. This might actually then have a negative impact on performance.

### 13.15.3 TCP connection retransmissions

The number of times that TCP will retransmit an unacknowledged connection request (SYN) before aborting is determined by the registry value TcpMaxConnectRetransmissions. For each given attempt, the retransmission time-out is doubled with each successive retransmission. For Windows Server 2003, the default number of time-outs is 2 and the default time-out period is 3 seconds (set by the TCPInitialRTT registry entry).

The parameter for connection retransmissions can be incremented to prevent a connection from timing out across slow WAN links. Because the optimal value will be different for each implementation, no specific recommendation is made. This should be determined through careful testing in your environment.

**Note:** This parameter should not be set so high that the connection will not time out at all.

Key:	HKLM\SYSTEM \CurrentControlSet \Services\Tcpip\Parameters
Value:	TCPMaxConnectRetransmissions
Data type:	REG_DWORD
Range:	0x0 - 0xFF
Default:	0x2
Recommendation:	None made - environment specific
Value exists by default:	No, needs to be added.

For more information, see:

<http://support.microsoft.com/?kbid=120642>

#### 13.15.4 TCP data retransmissions

The number of times that TCP will retransmit an unacknowledged data segment before aborting is specified by the registry value `TcpMaxDataRetransmissions`. The default value is 5 times.

TCP establishes an initial interval by measuring the round trip for a given connection. With each successive retransmission attempt, the interval doubles until responses resume or time-out altogether - at which time the interval is reset to the initially calculated value.

Because the optimal value will be different for each implementation, no specific recommendation is made here. This should be determined through careful testing in your environment.

Key:	<code>HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters</code>
Value:	<code>TCPMaxDataRetransmissions</code>
Data type:	<code>REG_DWORD</code>
Range:	<code>0x0 - 0xFFFFFFFF</code>
Default:	<code>0x5</code>
Recommendation:	None made, environment specific
Value exists by default:	No, needs to be added.

For more information, see:

<http://support.microsoft.com/?kbid=120642>

#### 13.15.5 TCP TIME-WAIT delay

By default, TCP will normally allocate a port with a value between 1024 and 5000 for a socket request for any available short-lived (ephemeral) user port. When communications over a given socket have been closed by TCP, it waits for a given time before releasing it. This is known as the TIME-WAIT delay. The default setting for Windows Server 2003 is two minutes, which is appropriate for most situations. However, some busy systems that perform many connections in a short time might exhaust all ports available, reducing throughput.

Windows has two registry settings that can be used to control this time-wait delay:

- ▶ `TCPTimedWaitDelay` adjusts the amount of time that TCP waits before completely releasing a socket connection for reuse.
- ▶ `MaxUserPort` sets the number of actual ports that are available for connections, by setting the highest port value available for use by TCP.

Reducing TCPTimedWaitDelay and increasing MaxUserPort can increase throughput for your system.

**Note:** These changes only optimize performance on exceptionally busy servers hosting thousands of simultaneous TCP connections, such as a heavily loaded LDAP, FTP or Web server.

Key: HKLM\SYSTEM \CurrentControlSet  
  \Services\Tcpip\Parameters

Value: TCPTimedWaitDelay

Data type: REG\_DWORD

Range: 0x0 - 0x12C (0 - 300 seconds)

Default: 0x78 (120 seconds)

Recommendation: 0x1E (30 seconds)

Value exists by default: No, needs to be added.

Key: HKLM\SYSTEM \CurrentControlSet  
  \Services\Tcpip\Parameters

Value: MaxUserPort

Data type: REG\_DWORD

Range: 0x1388 - 0xFFFF (5000 - 65534)

Default: 0x1388 (5000)

Recommendation: 0xFFFF

Value exists by default: No, needs to be added.

**Note:** The value name is MaxUserPort, not MaxUserPorts.

For more information, see *Microsoft Windows Server 2003 TCP/IP Implementation Details*, which is available from:

<http://www.microsoft.com/technet/prodtechnol/windowsserver2003/technologies/networking/tcpip03.mspx>

### 13.15.6 TCP Control Block (TCB) table

For each active TCP connection on a system, various control variables about the connection are stored in a memory block called a TCP Control Block (TCB). These TCB values are initialized when the connection is established, and then continually updated throughout the lifetime of the connection. Each of these TCBs are maintained in a hash table called the TCB table.

The size of the TCB table is controlled by the registry value MaxHashTableSize. On a large system with many active connections, having a larger table reduces the amount of time spent the system must spend to locate a particular TCB.

By partitioning the TCB table, contention for table access is minimized. TCP performance can be optimized by increasing the number of partitions. This is particularly the case on multi-processor systems. The registry value NumTcbTablePartitions controls the number of partitions. By default, the value is the square of the number of processors in the system.

The MaxHashTableSize value should always be of a power of two to correctly function. You might consider using the maximum value for large servers that might host a high number of connections. Keep in mind, however, that the table uses non-paged pool memory, so do not set too high a value if the system is constrained on non-paged pool memory or if the system simply will not support a high load of connections.

With a server that has more than one CPU, the NumTcbTablePartitions parameter should be four times the number of processors installed in the system. In most cases this will perform equal to or better than the default square of the number of CPUs in the system, especially on servers with 8 or 16 CPUs, where a too high a value of NumTcbTablePartitions can impact CPU performance.

Key: HKLM\SYSTEM \CurrentControlSet  
  \Services\Tcpip\Parameters  
Value: MaxHashTableSize  
Data type: REG\_DWORD  
Range: 0x40 - 0x10000 (1 - 65536), should be a power of 2 (2<sup>n</sup>)  
Default: 0x200 (512)  
Recommendation: 0x10000 (65536)  
Value exists by default: No, needs to be added.

Key: HKLM\SYSTEM \CurrentControlSet  
  \Services\Tcpip\Parameters  
Value: NumTcbTablePartitions  
Data type: REG\_DWORD  
Range: 0x1 - 0xFFFF (1 - 65535), should be a power of 2 (2<sup>n</sup>)  
Recommendation: 4 x the number of processors in the system  
Default: n<sup>2</sup>, where n is the number of CPUs installed

**Tip:** Because this value does not exist in the registry by default, be careful to ensure the value set is NumTcbTablePartitions, and *not* to NumTcpTablePartitions.

For more information, see *Microsoft Windows Server 2003 TCP/IP Implementation Details*, which is available from:

<http://www.microsoft.com/technet/prodtechnol/windowsserver2003/technologies/networking/tcpip03.mspx>

TCBs are normally preallocated in memory to avoid spending time allocating and deallocating them every time TCP connections are established and closed. The reuse or caching of TCBs improves memory management, but it also restricts how many active connections TCP can support at a given time.

The registry value MaxFreeTcbs configures the threshold number of connections required before TCBs in the TIME-WAIT state (that is, the connection has been closed but is not free for reuse yet), are reused. This value was often set higher than the default to optimize performance in older Windows NT implementations to ensure there were always sufficient preallocated TCBs.

Since Windows 2000, a feature was added to decrease the chance of running out of preallocated TCBs. If more than the number of TCBs specified in the new registry value MaxFreeTWTcbs are in the TIME-WAIT state, then all connections that have been in the TIME-WAIT state for longer than 60 seconds are forcibly closed and made available for use again.

With this feature now incorporated into Windows 2000 Server and now Windows Server 2003, modification of the MaxFreeTcbs registry value is no longer deemed valuable in optimizing TCP performance.

### 13.15.7 TCP acknowledgement frequency

TCP uses delayed acknowledgements to reduce the number of packets transmitted in the network, thereby improving performance. Normally, an acknowledgement (ACK) is sent every second TCP segment. This value can be increased to reduce the cost of processing of network traffic, especially in the case of large data uploads from the client to the server.

**Note:** This value is configured at the *interface* level for each network interface installed in the system, and it differs depending on the speed of the interface.

The default value is 2.

For Fast Ethernet (100 Mbps) network interfaces, use a value of 5 (0x5).

For Gigabit (1000 Mbps) network interfaces, use a value of 13 (0xD).

Key:	HKLM\System\CurrentControlSet\services\tcpip\parameters\interface\xx (xx depends on network interface)
Value:	TcpAckFrequency
Data type:	REG_DWORD
Range:	0x1 - 0xD (1-13)
Default:	2

Recommendation: 0x5 (5) for FastEthernet or 0xD (13) for Gigabit interfaces

Value exists by default: No, needs to be added.

For more information, see *Performance Tuning Guidelines for Windows Server 2003*, which is available from:

<http://www.microsoft.com/windowsserver2003/evaluation/performance/tuning.mspx>

### 13.15.8 Maximum transmission unit

The TCP/IP maximum transmission unit (MTU) defines the maximum size of an IP datagram that can be transferred in one frame over the network over a specific data link connection. The MTU might differ for network segments between the sender and receiver. Too small a packet size means data transmission is inefficient. Too large a packet size means that data might exceed the MTU of the links over which the packet is transmitted.

One method of determining the most efficient packet size allows routers to divide packets as they encounter a network segment with a smaller MTU than that of the packet being transmitted. This is referred to as *IP segmentation or fragmentation*. This method places extra overhead on routers that need to divide and reassemble packets.

A preferred and more common option is for a client to determine the maximum MTU that can be used on all segments between the sender and receiver. The client communicates with routers along the path as required to determine the smallest MTU that can be used in all segments from start to end. This process is known as “calculating the path maximum transmission unit (PMTU)”, and it will result in the most efficient packet size that will not need to be fragmented.

TCP/IP normally determines the optimal MTU dynamically, as described in 13.15.9, “Path Maximum Transmission Unit (PMTU) Discovery” on page 405. Windows does, however, allow the MTU to be statically configured. This is not normally recommended, but it might be suitable in some environments. Under most circumstances, allowing TCP/IP to dynamically determine the MTU is preferred, unless you can be certain of the MTU for every segment in your environment that a host might need to communicate over. By setting it statically, the MTU will not need to be negotiated and can offer a performance improvement.

The MTU for your environment can be determined by using the PING command on one of your servers and then issuing the following command:

```
PING -f -1 <MTUsize> <remote host IP address>
```

**Tip:** The **f** and **l** parameters must be in lower case for this command to work.

The **MTUSize** parameter is one you will use to determine the PMTU between the server and a remote host that it communicates with frequently. This might be a host on the same segment or one across multiple intercontinental WAN links. The command should be issued repeatedly using different values for **MTUSize** until the highest possible PMTU setting is achieved without receiving a packet needs to be fragmented response.

A useful value to start with for **MTUSize** is 1500, which is the Windows default. Work up or down from there (normally down) until the maximum setting is determined. Example 13-1 shows an optimal setting of a 1472-byte MTU before the packet begins to be fragmented at 1473 bytes.

*Example 13-1 Determining the MTU size*

---

```
C:\>ping -f -l 1472 w3.ibm.com
```

```
Pinging w3ibm.southbury.ibm.com [9.45.72.138] with 1472 bytes of data:
```

```
Reply from 9.45.72.138: bytes=1472 time=26ms TTL=245
```

```
Ping statistics for 9.45.72.138:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 26ms, Maximum = 26ms, Average = 26ms
```

```
C:\>ping -f -l 1473 w3.ibm.com
```

```
Pinging w3ibm.southbury.ibm.com [9.45.72.138] with 1473 bytes of data:
```

```
Packet needs to be fragmented but DF set.
```

```
Ping statistics for 9.45.72.138:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

---

**Tip:** This registry value is set at the interface level, not as an overall TCP/IP parameter.

After you determine this optimal MTU value, you set it in the registry as follows:

Key: HKLM\SYSTEM \CurrentControlSet  
  \services\tcpip\parameters\interface\xxxxxxx  
  (depends on network interface)

Value: MTU

Data type: REG\_DWORD

Range: 0x44 (68), determined dynamically MTU OR  
  0xFFFFFFF

Default: 0xFFFFFFFF (determine dynamically PMTU)

Recommendation: 0xFFFFFFFF

Value exists by default: No, needs to be added.

For more information, see

<http://www.microsoft.com/windows2000/techinfo/reskit/en-us/regentry/58792.asp>

**Important:** There is a close interaction between the MTU registry setting and the registry setting EnablePMTUDiscovery, which is described in the next section.

To use a statically determined value for MTU as described, EnablePMTUDiscovery should be disabled (that is, set to 0).

If EnablePMTUDiscovery is disabled and no value is set for MTU as described, then TCP/IP will configure a default MTU size of 576 bytes. This packet size usually avoids any packet fragmentation, but it is far from optimal for most environments. This means that in most instances, if EnablePMTUDiscovery is disabled, a value should be set for MTU as well.

If EnablePMTUDiscovery is enabled (the default), then the MTU registry value should be set to 0xFFFFFFFF, or it can be removed altogether.

### 13.15.9 Path Maximum Transmission Unit (PMTU) Discovery

Under most circumstances, the maximum transmission unit (MTU) of every network segment that a server might possibly communicate over will not be known. Remote networks will often have an entirely different MTU to that of local networks.

When enabled, the registry value EnablePMTUDiscovery lets TCP/IP automatically determine the MTU for all networks along the path to a remote

host. When the MTU has been determined for all network segments on a path, it will use the highest, and thus most efficient MTU value that can be used without packet fragmentation occurring.

PMTU detection is enabled by default in Windows. Because the `EnablePMTUDiscovery` does not ordinarily exist in the registry, it would normally only be created with the intention of disabling PMTU detection.

If you choose to disable PMTU detection, a default MTU of 576 bytes will be used for all communications, unless a value is also set for the MTU registry value as described in 13.15.8, “Maximum transmission unit” on page 403.

**Note:** This registry value applies to all network interfaces.

The following registry value controls the use of PMTU Discovery:

Key:	<code>HKLM\SYSTEM\CurrentControlSet\services\Tcpip\Parameters</code>
Value:	<code>EnablePMTUDiscovery</code>
Data type:	<code>REG_DWORD</code>
Range:	0 or 1
Default:	1
Recommendation:	1
Value exists by default:	No, needs to be added.

For more information, see

<http://www.microsoft.com/windows2000/techinfo/reskit/en-us/regentry/58752.asp>

## 13.16 Memory registry optimizations

For most purposes, Windows operates very well in its native self-tuning capacity. Nevertheless, there are many other registry changes relating to the memory subsystem that can be modified to improve system performance under specific circumstances. Some of those that have been noted to improve performance in production environments are listed here.

Keep in mind that several of the memory specific tuning parameters listed here hold relevance only for the 32-bit (x86) versions of the Windows Server 2003 operating system. They are no longer valid for the 64-bit (x64) editions, given the greatly expanded memory architecture.

As with all changes, ensure you have a working and tested backup of the registry and entire server before making the change. Changes should be made and

tested only one at a time. If system performance is negatively affected by making such a change, it should be reversed immediately.

### 13.16.1 Disable kernel paging

Servers with sufficient physical memory might benefit from disabling portions of the Windows operating system kernel and user-mode and kernel-mode drivers from being paged to disk. This registry setting forces Windows to keep all components of the kernel (or executive) and drivers in memory and, thus, allows much faster access to them when required.

Key: HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management  
Value: DisablePagingExecutive  
Data type: REG\_DWORD  
Range: 0x0 (default) or 0x1  
Recommendation: 0x1  
Value exists by default: No, needs to be added

For more information, see:

<http://support.microsoft.com/?kbid=184419>

### 13.16.2 Optimizing the Paged Pool Size (32-bit x86)

Windows allocates memory in *pools* for the operating system and its components, which processes access through the use of *kernel mode*. Two pools of kernel mode memory exist:

- ▶ The paged pool (which can be paged to the pagefile)
- ▶ The non-paged pool (which can never be paged)

Performance and system stability can be seriously impacted if Windows experiences memory resource constraints and is unable to assign memory to these pools.

The amount of physical memory assigned to these two pools is assigned dynamically at system boot time. The maximum default size on 32-bit (x86) editions of Windows Server 2003 for the paged memory pool is 491 MB, and 256 MB for the non-paged pool. In the 64-bit (x64) editions of Windows Server 2003, both the paged pool and non-paged pool have a limit of 128 GB. As a result the following values do not apply for the 64-bit (x64) editions.

Some applications and workloads can demand more pooled memory than the system has been allocated by default. Setting the PagedPoolSize registry value as listed in Table 13-7 can assist in ensuring sufficient pooled memory is available.

Changing this setting requires a restart of the operating system.

Table 13-7 PagedPoolSize values - 32-bit (x86) Editions of Windows Server 2003

PagedPoolSize value	Meaning
0x0 (default)	Requests that the system will dynamically calculate an optimal value at system startup for the paged pool based on the amount of physical memory in the computer. This value will change if more memory is installed in the computer. The system typically sets the size of the paged pool to approximately twice that of the nonpaged pool size.
Range: 0x1 - 0x20000000 (512 MB)	<p>Creates a paged pool of the specified size, in bytes. This takes precedence over the value that the system calculates, and it prevents the system from adjusting this value dynamically.</p> <p>Limiting the size of the paged pool to 192 MB (or smaller) lets the system expand the file system (or system pages) virtual address space up to 960 MB. This setting is intended for file servers and other systems that require an expanded file system address space (meaning slightly faster access) at the expense of being able to actually cache less data. This only makes sense if you know the files your server frequently accesses already fit easily into the cache.</p>
0xFFFFFFFF	With this value, Windows will calculate the maximum paged pool allowed for the system. For 32-bit systems, this is 491 MB. This setting is typically used for servers that are attempting to cache a very large number of frequently used small files, some number of very large size files, or both. In these cases, the file cache that relies on the paged pool to manage its caching is able to cache more files (and for longer periods of time) if more paged pool is available.

Setting this value to 0xB71B000 (192 MB) provides the system with a large virtual address space, expandable to up to 960 MB. Note that a corresponding entry of zero (0) is required in the SystemPages registry value for this to take optimal effect.

Key:	HKLM\SYSTEM \CurrentControlSet \Control Session Manager\Memory Management
Value:	PagedPoolSize
Data type:	REG_DWORD
Range:	0x0 (default) to 0xFFFFFFFF (4294967295)
Recommendation:	0xB71B000 (192000000) for native SMB shares. For NFS shares when using Services for UNIX, use a value of 0xFFFFFFFF (4294967295)
Value exists by default:	Yes
Key:	HKLM\SYSTEM \CurrentControlSet \Control Session Manager\Memory Management
Value:	SystemPages
Data type:	REG_DWORD
Range:	0x0 (default) to 0xFFFFFFFF
Recommendation:	0x0
Value exists by default:	Yes

For more information, see:

[http://www.microsoft.com/resources/documentation/windows/2000/server/rekit/en-us/core/fnec\\_evl\\_fhcj.asp](http://www.microsoft.com/resources/documentation/windows/2000/server/rekit/en-us/core/fnec_evl_fhcj.asp)

### 13.16.3 Increase memory available for I/O locking operations

By default, the 32-bit (x86) editions of Windows Server 2003 sets a limit on the amount of memory that can be set aside for I/O locking operations at 512 KB. Depending on the amount of physical memory in a server, this can be increased in various increments as listed in Table 13-8. You can insert the values listed in this table into the registry to increase the amount of memory available for locking operations on 32-bit (x86) systems. These values hold no validity for the 64-bit (x64) editions of Windows Server 2003.

Table 13-8 Maximum I/O lock limit values

Amount of physical RAM	Maximum lock limit (IoPageLockLimit value)
Less than 64 MB	Physical memory minus 7 MB
64 MB - 512 MB	Physical memory minus 16 MB
512 MB upwards	Physical memory minus 64 MB

These value ranges listed in Table 13-8 on page 409 equate to those calculated in Table 13-9, depending on the exact amount of physical RAM in the machine. Because almost all servers today have more than 512 MB RAM, the calculations in Table 13-9 take into account only 512 MB RAM and above.

The appropriate value should be determined from Table 13-8 on page 409 and then entered into the registry value `IoPageLockLimit`. This value then takes precedence over the system default of 512 KB and specifies the maximum number of bytes that can be locked for I/O operations:

Key:	<code>HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management</code>
Value:	<code>IoPageLockLimit</code>
Data type:	<code>REG_DWORD</code>
Range:	0 (default) to 0xFFFFFFFF (in bytes, <i>do not exceed this maximum!</i> )
Recommendation:	depends on RAM, see Table 13-9
Default:	0x80000 (512 KB)
Value exists by default:	No, needs to be added.

Table 13-9 Recommended settings for `IoPageLockLimit`

Amount of physical RAM	IoPageLockLimit Setting (hex)
512 MB	0x1C000000
1 GB (1024 MB)	0x3C000000
2 GB (2048 MB)	0x80000000
4 GB (4096 MB)	0xFC000000
8 GB (8096 MB) and above	0xFFFFFFFF

For more information, see:

<http://www.microsoft.com/windows2000/techinfo/reskit/en-us/regentry/29932.asp>

#### 13.16.4 Increasing available worker threads

At system startup, Windows creates several server threads that operate as part of the System process. These are called *system worker threads*. They exist for the sole purpose of performing work on behalf of other threads generated by the kernel, system device drivers, the system executive, and other components. When one of these components puts a work item in a queue, a thread is assigned to process it.

The number of system worker threads would ideally be high enough to accept work tasks as soon as they become assigned. The trade-off is that worker threads sitting idle are using system resources unnecessarily.

The DefaultNumberOfWorkerThreads value sets the default number of worker threads created for a given work queue. Having too many threads needlessly consumes system resources. Having too few threads slows the rate at which work items are serviced.

Key:	HKLM\SYSTEM \CurrentControlSet \services\rpcxdr\parameters
Value:	DefaultNumberOfWorkerThreads
Data type:	REG_DWORD
Range:	0x0 (default) to 0x40 (64)
Recommendation:	16 times the number of CPUs in the system
Value exists by default:	No, needs to be added

Delayed worker threads process work items that are not considered time-critical and can have their memory stack paged out while waiting for work items. The value for AdditionalDelayedWorkerThreads increases the number of delayed worker threads created for the specified work queue. An insufficient number of threads slows the rate at which work items are serviced; a value too high will consume system resources unnecessarily.

Key:	HKLM\SYSTEM \CurrentControlSet \Control \Session Manager\Executive
Value:	AdditionalDelayedWorkerThreads
Data type:	REG_DWORD
Range:	0x0 (default) to 0x10 (16)
Recommendation:	0x10 (16)
Value exists by default:	Yes

Critical worker threads process time-critical work items and have their stack present in physical memory at all times. The value for AdditionalCriticalWorkerThreads increases the number of critical worker threads created for a specified work queue. An insufficient number of threads slows the rate at which time-critical work items are serviced. A value that is too high consumes system resources unnecessarily.

Key:	HKLM\SYSTEM \CurrentControlSet \Control \Session Manager\Executive
Value:	AdditionalCriticalWorkerThreads
Data type:	REG_DWORD
Range:	0x0 (default) to 0x10 (16)
Recommendation:	0x10 (16)
Value exists by default:	Yes

### 13.16.5 Prevent the driver verifier from running randomly

The driver verifier, at random intervals, verifies drivers for debugging randomly. Disabling this functionality might improve system performance.

Key:	HKLM\SYSTEM \CurrentControlSet \Control
	\Session Manager\Memory Management
Value:	DontVerifyRandomDrivers
Data type:	REG_DWORD
Range:	0x0 (default) or 0x1
Recommendation:	0x1
Value exists by default:	No

For more information, see:

<http://www.microsoft.com/windowsserver2003/evaluation/performance/tuning.mspx>

## 13.17 File system optimizations

Several registry tuning parameters are available in Windows Server 2003. These will assist with performance in both 32-bit (x86) and 64-bit (x64) editions of the server operating system.

### 13.17.1 Increase work items and network control blocks

The maximum number of concurrent outstanding network requests between a Windows Server Message Block (SMB) client and server is determined when a session between the client and server is negotiated. The maximum value that is negotiated is determined by registry settings on both the client and server. If these values are set too low on the server, they can restrict the number of client sessions that can be established with the server. This is particularly a problem in a Terminal Server environment where clients are typically launching many simultaneous application instances on the server itself and using many local resources.

The three values that can be adjusted to improve system performance for work items exist in the LanmanServer and LanmanWorkstation registry keys and are:

- ▶ MaxWorkItems
- ▶ MaxMpxCt
- ▶ MaxCmds

None of these values exists, by default, in the registry. The default settings for the first two values are determined by the hardware configuration of the server

combined with the value of the Server dialog (File & Print Sharing setting discussed in 13.6, “File system cache” on page 365). MaxCmds has a default of 50.

The MaxWorkItems value specifies the maximum number of receive buffers, or work items, that the Server service is permitted to allocate at one time. If this limit is reached, then the transport must initiate flow control, which can significantly reduce performance.

The MaxMpxCt value sets the maximum number of simultaneous outstanding requests on a client to a particular server. During negotiation of the Server Message Block dialect, this value is passed to the client's redirector where the limit on outstanding requests is enforced. A higher value can increase server performance but requires more use of server work items (MaxWorkItems).

The MaxCmds value specifies the maximum number of network control blocks that the redirector can reserve. The value of this entry coincides with the number of execution threads that can be outstanding simultaneously. Increasing this value will improve network throughput, especially if you are running applications that perform more than 15 operations simultaneously.

Take care not to set any of these values too high. The more outstanding connections that exist, the more memory resources will be used by the server. If you set the values too high, the server could run out of resources such as paged pool memory.

**Tip:** The MaxWorkItems value must be at least *four times* as large as MaxMpxCt.

Key:	HKLM\SYSTEM\CCS\Services\LanmanServer\Parameters
Value:	MaxWorkItems
Data type:	REG_DWORD
Value:	1 - 65535
Default:	Configured dynamically based on system resources and server setting
Recommendation:	32768
Value exists by default:	No, needs to be added

Key:	HKLM\SYSTEM\CCS\Services\
	LanmanWorkstation\Parameters
Value:	MaxCmds
Data type:	REG_DWORD
Value:	50 - 65535
Default:	50
Recommendation:	4096
Value exists by default:	No, needs to be added
Key:	HKLM\SYSTEM\CCS\Services\LanmanServer\
	Parameters
Value:	MaxMpxCt
Data type:	REG_DWORD
Value:	8192
Default:	Configured dynamically based on system resources and server setting
Recommendation:	1
Value exists by default:	No, needs to be added

For more information, see:

<http://support.microsoft.com/?kbid=232476>

<http://support.microsoft.com/?kbid=271148>

### 13.17.2 Disable NTFS last access updates

Each file and folder on an NTFS volume includes an attribute called Last Access Time. This attribute shows when the file or folder was last accessed, such as when a user performs a folder listing, adds files to a folder, reads a file, or makes changes to a file.

Maintaining this information creates a performance overhead for the file system, especially in environments where a large number of files and directories are accessed quickly and in a short period of time, such as by a backup application. Apart from in highly secure environments, retaining this information might add a burden to a server that can be avoided by updating the following registry key:

Key:	HKLM\SYSTEM \CurrentControlSet \Control
	\FileSystem
Value:	NTFSDisableLastAccessUpdate
Data type:	REG_DWORD
Value:	0 or 1
Default:	0
Recommendation:	1
Value exists by default:	No, needs to be added

For more information, see:

<http://www.microsoft.com/resources/documentation/WindowsServ/2003/all/dependencyguide/en-us/46656.asp>

In Windows Server 2003, this parameter can also be set by using the command:

```
fsutil behavior set disablelastaccess 1
```

### 13.17.3 Disable short-file-name (8.3) generation

By default, for every long file name created in Windows, NTFS generates a corresponding short file name in the older 8.3 DOS file name convention for compatibility with older operating systems. In many instances this functionality can be disabled, thereby offering a performance improvement.

Note that before disabling short name generation, ensure that there is no DOS or 16-bit application running on the server that requires 8.3 file names, or users accessing the files on the server through 16-bit applications or older file systems or operating systems. Also be aware that even some recent applications have been known to exhibit problems at installation and run time if this setting is made.

Key:	HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem
Value:	NTFSDisable8dot3NameCreation
Data type:	REG_DWORD
Value:	0 or 1
Default:	0
Recommendation:	1
Value exists by default:	Yes

In Windows Server 2003, this parameter can also be set by using the command:

```
fsutil behavior set disable8dot3 1
```

### 13.17.4 Use NTFS on all volumes

Windows offers multiple file system types for formatting drives, including NTFS, FAT, and FAT32. NTFS offers considerable performance benefits over the FAT and FAT32 file systems, and should be used exclusively on Windows servers. In addition, NTFS offers many security, scalability, stability and recoverability benefits over FAT.

Under previous versions of Windows, FAT and FAT32 were often implemented for smaller volumes (say, <500 MB) because they were often faster in such situations. With disk storage relatively inexpensive today and operating systems and applications pushing drive capacity to a maximum, however, it is unlikely that

such small volumes will be warranted. FAT32 scales better than FAT on larger volumes, but it is still not an appropriate file system for Windows servers.

FAT and FAT32 have often been implemented in the past as they were seen as more easily recoverable and manageable with native DOS tools in the event of a problem with a volume. Today, with the various NTFS recoverability tools built both natively into the operating system and as third-party utilities available, there should no longer be a valid argument for not using NTFS for file systems.

### 13.17.5 Do not use NTFS file compression

Although it is an easy way to reduce space on volumes, NTFS file system compression is not appropriate for enterprise file servers. Implementing compression places an unnecessary overhead on the CPU for all disk operations and is best avoided. Think about options for adding additional disk, near-line storage, or archiving data before seriously considering file system compression.

### 13.17.6 Monitor drive space utilization

The less data a disk has on it, the faster it will operate. This is because on a well-defragmented drive, data is written as close to the outer edge of the disk as possible, because this is where the disk spins the fastest and yields the best performance.

Disk seek time is normally considerably longer than read or write activities. As noted, data is initially written to the outside edge of a disk. As demand for disk storage increases and free space reduces, data is written closer to the center of the disk. Disk seek time is increased when locating the data as the head moves away from the edge, and when found, it takes longer to read, hindering disk I/O performance.

This means that monitoring disk space utilization is important not just for capacity reasons, but also for performance. However, it is not practical or realistic to have disks with excessive free space.

**Tip:** As a rule of thumb, work towards a goal of keeping disk free space between 20% to 25% of total disk space.

See 11.6.3, “Active data set size” on page 271 for a further discussion about this topic and the performance benefits.

### 13.17.7 Use disk defragmentation tools regularly

Over time, files become fragmented in non-contiguous clusters across disks, and system performance suffers as the disk head jumps between tracks to seek and reassemble them when they are required.

Disk defragmentation tools work to ensure that all file fragments on a system are brought into contiguous areas on the disk, thereby improving disk I/O performance. Regularly running a disk defragmentation tool on a server is a relatively easy way to yield impressive system performance improvements.

**Tip:** We recommend you defragment your drives daily, if possible.

Most defragmentation tools work the fastest and achieve the best results when they have plenty of free disk space to work with, which is another important reason to monitor and manage disk space usage on production servers.

In addition, try to run defragmentation tools when the server is least busy and if possible, during scheduled system downtime. Defragmentation of the maximum number of files and directories will be achieved if carried out while applications and users that typically keep files open are not accessing the server.

Windows Server 2003 includes a basic disk defragmentation tool. While offering useful defragmentation features, it offers little in the way of automated running because each defragmentation process must be initiated manually or through external scripts or scheduling tools.

A number of high-quality third-party disk defragmentation tools exist for the Windows operating system, including tailorable scheduling, reporting, and central management functionality. The cost and performance benefit of using such tools is normally quickly realized when compared to the ongoing operational costs and degraded performance of defragmenting disks manually, or not at all.

### 13.17.8 Review disk controller stripe size and volume allocation units

When configuring drive arrays and logical drives within your hardware drive controller, ensure you match the controller stripe size with the allocation unit size that the volumes will be formatted with within Windows. This will ensure disk read and write performance is optimal and gain better overall server performance.

Having larger allocation unit (or *cluster* or *block*) sizes means the disk space is not used as efficiently, but it does ensure higher disk I/O performance because the disk head can read in more data in one read activity.

To determine the optimal setting to configure the controller and format the disks with, determine the average disk transfer size on the disk subsystem of a server with similar file system characteristics. Using the Windows System (Performance) Monitor tool to monitor the Logical Disk object counters of Avg. Disk Bytes/Read and Avg. Disk Bytes/Write over a period of normal activity will help determine the best value to use.

Some server purposes might warrant a smaller allocation unit size where the system will be accessing many small files or records. However, with file sizes, file systems, and disk storage increasing every day, our testing has found that setting an allocation unit size of 64 KB delivers sound performance and I/O throughput under most circumstances. Improvements in performance with tuned allocation unit sizes can be particularly noted when disk load increases.

Note that either the FORMAT command line tool or the Disk Management tool are needed to format volumes larger than 4096 bytes (4 KB). Windows Explorer will only format up to this threshold. Note also that CHKDSK can be used to confirm the current allocation unit size of a volume, but it needs to scan the entire volume before the desired information is displayed (shown as Bytes in each allocation unit).

**Tip:** The default allocation unit size that Windows Server 2003 will format volumes with is 4096 bytes (4 KB). It is definitely worth considering whether this size really is the most appropriate for your purposes for the volumes on your servers.

### 13.17.9 Use auditing and encryption judiciously

Auditing and file encryption are two valuable security features which, although very beneficial, can also add considerable overhead to system resources and performance. In particular, CPU and memory resources can be taxed by using auditing and encryption.

Auditing is often a required system function by security-conscious organizations. The extent to which it impacts server resources will be determined by at what level auditing servers are actually employed. To ensure that the performance impact of auditing is kept as low as possible, ensure that only system events and areas of file systems and the registry that actually require auditing are configured with this feature.

Similarly, file encryption is required in some instances to improve system security. As with auditing, ensure that only directories and files that actually require the security supplied by encryption are set up with this feature.

## **13.18 Other performance optimization techniques**

This section details various other methods that should be implemented on Windows servers to extract the best performance.

### **13.18.1 Dedicate server roles**

Where budget and operational resources permit, use dedicated domain-controller servers, and dedicated member, or stand-alone, servers. Do not combine Active Directory, DNS, WINS, DHCP, Certificate Services, or similar infrastructure services onto a member server that has a primary function for another purpose. Each of these services places additional overhead on the server, thereby taking away valuable resources that should be dedicated to the primary function that the member server is serving.

In the same manner, system resources are best dedicated to specific intensive functions such as file serving, line-of-business application servers, mail servers, database servers, or Web servers. The nature of each of these functions will affect the server subsystems in a different way; therefore, to improve performance and increase stability, dedicate servers for different server functions.

### **13.18.2 Run system-intensive operations outside peak times**

Applications and jobs that are intensive on any server subsystem should be scheduled to run outside peak server times. When a server needs to be running fastest, it does not make sense to slow it down with applications such as virus scanners, backup jobs, or disk fragmentation utilities.

### **13.18.3 Log off the server console**

A simple but important step in maximizing your server's performance is to keep local users logged off the console. A locally logged-on user unnecessarily consumes systems resources, thereby potentially impacting the performance of applications and services running on the system.

### **13.18.4 Remove CPU-intensive screen savers**

A server is not the place to run fancy 3D or OpenGL screen savers. Such screen savers are known to be CPU-intensive and they use important system resources when they are running. It is best to avoid installing these altogether as an option at server-build time, or to remove them if they have been installed. The basic "Windows Server 2003" or blank screen savers are the best choice.

Similarly, avoid using memory-wasting fancy desktop wallpapers images on your server.

### 13.18.5 Use the latest drivers, firmware, and service packs

Installing the latest version of a device driver, patch, BIOS update, microcode, or firmware revision for hardware is a very important part of routine server maintenance. Newer device drivers not only fix bugs and increase system stability, but can also increase the performance and efficiency of a device, improving overall system performance. Notable examples of this with System x servers are the various models of ServeRAID adapters. Revisions in the firmware and drivers for these RAID controllers has often added significant performance and functionality benefits over previous releases.

Microsoft periodically issues service packs and hot fixes for their operating systems. After a period of testing in your environment, these should be deployed to production systems. Service packs and hot fixes often introduce updated code to key kernel and subsystem components of the operating system, and can add extra performance and functionality benefits.

In addition to the performance benefits that can be offered by latest version of device drivers, patches, firmware and service packs, many hardware and software vendors will not offer support for their products until the latest revision is installed, so it is good maintenance practice to do so periodically.

### 13.18.6 Avoid the use of NET SERVER CONFIG commands

As discussed, Windows is for the most part a self-tuning operating system. If a value does not ordinarily exist in the registry, creating it and manually setting it normally prevents Windows from self-tuning the value automatically. In some circumstances this is what you hope to achieve by setting values that you believe are more optimal for your environment than Windows normally auto-configures.

The core Windows *server* service is controlled by a set of values defined in registry at:

`HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\LanmanServer\Parameters`

In a standard *clean* installation, many values that can exist in this registry location do not. This allows Windows to tune these values dynamically as the operating system sees fit to optimize system performance of the server service.

A command exists to set several of these parameters to desired values to avoid direct manipulation of the registry. These include the following commands:

```
net config server /autodisconnect:time  
net config server /srvcomment:"text"  
net config server /hidden:yes|no
```

As expected, these three commands create and manipulate the contents of the registry values autodisconnect, disc, srvcomment, and hidden in the LanmanServer\parameters key to the desired values.

There is, however, a most unfortunate outcome from using these commands. Not only do they create an necessary registry entry for the parameter requested, but they also create a whole series of other, unrelated registry values that are extremely important to the functioning of the server service. Creating all these values sets these values to static settings and subsequently prevents them from further self-tuning by Windows. This is most undesirable and should be avoided.

For example, administrators who want to hide Windows computers from the network browse list might issue the command: net config server /hidden:yes

Before issuing such a command, the registry parameters for the LanmanServer key on a typical Windows 2000 Server appear similar to Figure 13-18.

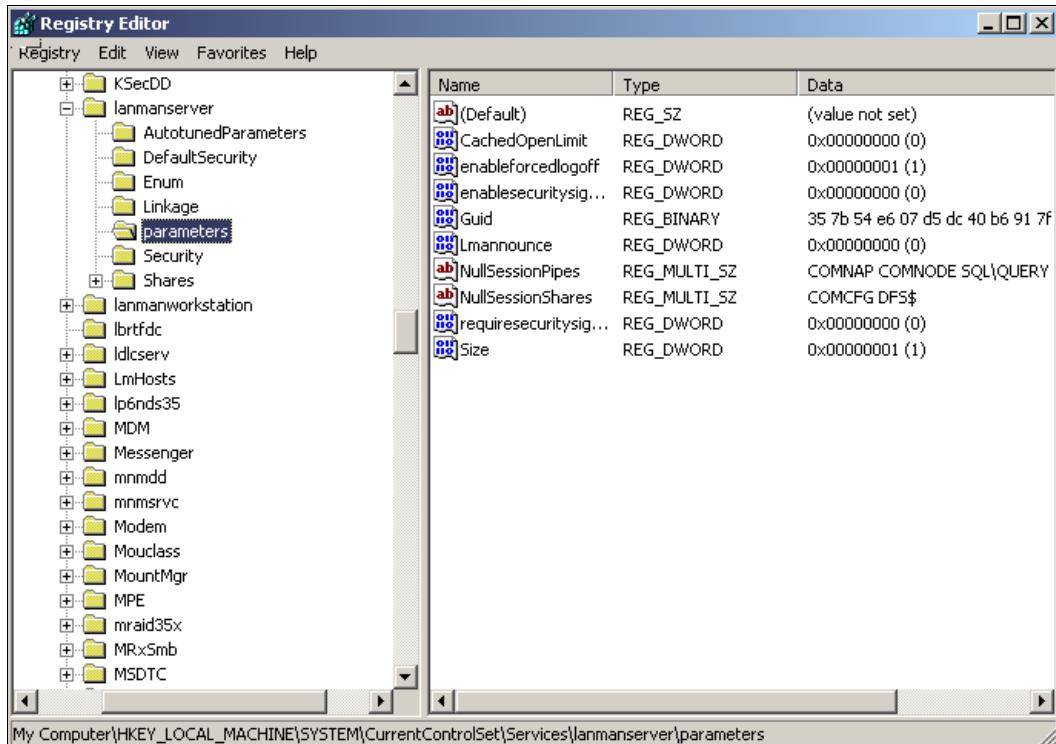


Figure 13-18 LanmanServer registry parameters - before using NET CONFIG command

After issuing this single, relatively simple command, however, Figure 13-19 shows the number of registry entries that have been inserted into the LanmanServer key. Each of these values is now statically set at the values displayed and is no longer able to be automatically tuned by Windows. Some of these values control important performance and scalability parameters of the Server service. They are now constrained to these static values and cannot be changed unless they are modified manually or removed altogether.

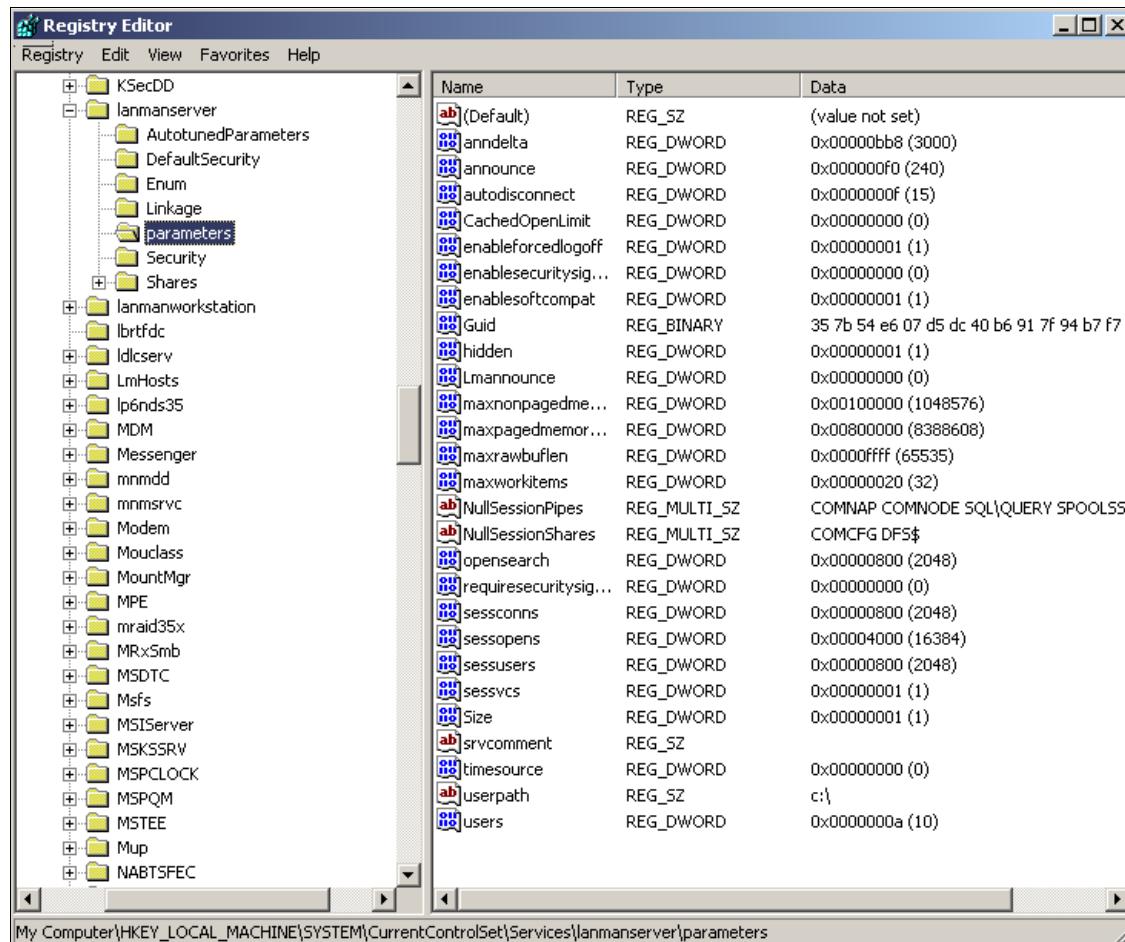


Figure 13-19 LanmanServer registry parameters - after using NET CONFIG command

These figures illustrate the undesirable impact of typing these commands. To avoid this issue, you need to create the values of autodisconnect, disc, srvcomment, and hidden manually as required when using the registry directly.

Windows Server 2003 does not populate the LanmanServer key with as many values as in Windows 2000 Server or Windows NT Server, but still introduces several parameters that are best left to auto-tune under most circumstances. This issue is clearly outlined in Microsoft KB 128167.

<http://support.microsoft.com/?kbid=128167>

### 13.18.7 Monitor system performance appropriately

Running the Windows System (Performance) Monitor directly on the server console of a server you are monitoring will impact the performance of the server and will potentially distort the results you are examining.

Wherever possible, use System Monitor from a remote system to avoid placing this extra load on the server itself. Similarly, do not use the remote control software of the server console for performance monitoring, and do not use a remote client session using thin-client sessions such as Terminal Services or Citrix to carry out the task. Also be aware, however, that monitoring a system remotely will affect the network and network interface counters, which might impact your measurements.

Keep in mind that the more counters monitored, the more overhead is required, so avoid monitoring unnecessary counters. This approach should be taken regardless of whether you carry out “live” charted monitoring of a system or you log system performance over longer periods. In the same way, do not publish System Monitor as a Citrix application.

In addition, reduce the monitoring interval to the minimum necessary to obtain the information you want to gather. When logging over an extended period, collecting data every five minutes or even longer (rather than using the system default of every second) might still provide acceptable levels of information. Monitoring more frequently will generate additional system overhead and create significantly larger log files than might be required.



# Microsoft Windows Server 2008

Microsoft's latest iteration of their flagship server operating system product, Windows Server 2008, was officially released in February 2008. It is based on the same kernel as the desktop operating system Windows Vista®. However, the adoption between the two products has been markedly different. Many firms have already successfully deployed many production instances of the new server operating system and enjoying many of the improvements over previous versions.

This chapter discusses the following topics:

- ▶ 14.2, “The Windows Server 2008 product family” on page 427
- ▶ 14.3, “New features of Windows Server 2008” on page 431
- ▶ 14.4, “Networking performance” on page 439
- ▶ 14.5, “Storage and file system performance” on page 443
- ▶ 14.6, “Other performance tuning measures” on page 447
- ▶ 14.7, “Windows Server 2008 R2” on page 450

## 14.1 Introduction to Microsoft Windows Server 2008

The feature set of Windows Server 2008 offers much to those concerned about server and operating system performance. This chapter explores the many new facets of Windows Server 2008 that have made a positive impact to system performance. Where appropriate, the chapter describes how it is differentiated from its predecessor, Windows Server 2003.

Windows Server 2003 was notably updated when Microsoft made available Release 2 (R2) in December 2005 (for more details, see 13.3, “Windows Server 2003, Release 2 (R2)” on page 358). In March 2007, Service Pack 2 for Windows Server 2003 was released, incorporating many important patches, hotfixes and reliability and performance enhancements.

Windows Server 2008 has built upon all these progressive improvements in Windows Server 2003 and several developments from its desktop counterpart, Windows Vista, to deliver a more scalable, reliable, secure, and highly performing Microsoft server platform.

### 14.1.1 Performance tuning for Windows Server 2008

As with all previous versions in the Windows Server product line, Microsoft has designed Windows Server 2008 to perform well “right out of the box.” Microsoft has incorporated directly into the operating system many of the discretionary performance tuning methods that existed with previous versions.

As a result, there is less emphasis in this chapter on individual system tuning techniques and “tweaks” than previous iterations of the chapter have included. Instead, the chapter explores opportunities to further enhance server performance by taking advantage of many of the new features of Windows Server 2008. It also explains how these can be implemented or tailored to offer the most optimized server platform for a given application or user base.

**Note:** Some of the tuning techniques identified in Chapter 13, “Microsoft Windows Server 2003” on page 351 are no longer relevant in Windows Server 2008. As a result, server engineers must determine whether performance tips described in any source on Windows server performance tuning remain applicable for Windows Server 2008. Where possible, this chapter describes which settings are no longer relevant with this latest edition of Windows Server.

However, this chapter is still intended to be used in conjunction with Chapter 13, “Microsoft Windows Server 2003” on page 351. Many of the best practice tips and techniques described in that chapter are still applicable to Windows Server 2008.

### 14.1.2 What is covered in this chapter

Windows Server 2008 offers a considerable number of feature additions and improvements over previous releases. It is beyond the scope of this chapter to detail all of these or the many applications or role types that a Windows Server 2008 deployment could host. Instead, this chapter focuses on the fundamental, core operating system changes and enhancements that improve system performance and differentiate Windows Server 2008 from its predecessors.

The focus is also primarily on the 64-bit (x64) edition of the operating system, because it offers better system performance over its 32-bit (x86) counterpart and is the ultimate stated direction for the Windows Server operating system from Microsoft.

Much of the material for this chapter has been referenced from the comprehensive document from Microsoft, *Performance Tuning Guidelines for Windows Server 2008*, available from:

[http://www.microsoft.com/whdc/system/sysperf/Perf\\_tun\\_srv.mspx](http://www.microsoft.com/whdc/system/sysperf/Perf_tun_srv.mspx)

This document has received several updates to ensure it remains current. Readers are encouraged to refer to this document in addition to the information presented here.

## 14.2 The Windows Server 2008 product family

As with previous versions, Windows Server 2008 has been released in several editions. These include:

- ▶ Windows Server 2008, Standard Edition (x86 and x64)

- ▶ Windows Server 2008, Enterprise Edition (x86 and x64)
- ▶ Windows Server 2008, Datacenter Edition (x86 and x64)
- ▶ Windows Server 2008, Itanium Edition (IA64)
- ▶ Windows Web Server 2008 (x86 and x64)
- ▶ Windows HPC Server 2008

The Standard, Enterprise, and Datacenter editions are also each available as versions “without Hyper-V”, removing the Microsoft hypervisor for customers who do not require virtualization functionality.

In addition, while not strictly “editions” of Window Server 2008, a number of other server products are available from Microsoft that are based on the same operating system codebase. These include:

- ▶ Windows Small Business Server 2008 (Standard and Premium) (x64 only)
- ▶ Windows Essential Business Server 2008 (Standard and Premium)
- ▶ Windows Storage Server 2008
- ▶ Windows Server 2008 Foundation (x64 only)

Although the primary focus for this chapter is around the Standard and Enterprise editions of Windows Server 2008, it is likely that many of the performance tuning facts and features discussed will be applicable to other editions as well.

The Microsoft server family also incorporates Windows Home Server. However, at time of writing, this version is based on Windows Server 2003 R2, not Windows Server 2008.

A more thorough overview of the feature difference between the various flavors now available for Windows 2008 can be found at the following reference:

<http://www.microsoft.com/windowsserver2008/en/us/editions-overview.aspx>

Table 14-1 shows a technical comparison of the number of processors and memory able to be addressed between versions of Windows Server 2008. It is worth noting that the 32-bit x86 CPUs are supported with up to 32 cores per processor, and the 64-bit x64 and IA-64 CPUs are supported with up to 64 cores.

*Table 14-1 Windows Server 2008 supported memory and CPU sockets comparison*

Edition	Web Edition	Standard Edition	Enterprise Edition	Datacenter Edition*	Itanium Edition (IA-64 only)	HPC Edition
Supported RAM 32-bit (x86)	1 to 4 GB	1 to 4 GB	1 to 64 GB	1 to 64 GB	Not applicable	Not applicable

Edition	Web Edition	Standard Edition	Enterprise Edition	Datacenter Edition*	Itanium Edition (IA-64 only)	HPC Edition
Supported RAM 64-bit (x64 & IA-64)	1 to 32 GB	1 to 32 GB	1 to 2 TB	1 to 2 TB	2 TB	128GB
Supported CPU sockets 32-bit (x86)	1 to 4	1 to 4	1 to 8	1 to 32	Not applicable	Not applicable
Supported CPU sockets 64-bit (x64 & IA-64)	1 to 4	1 to 4	1 to 8	1 to 64	1 to 64	1 to 4

A more thorough technical comparison of supported differences between the various editions of Windows Server 2008 can be found here:

<http://www.microsoft.com/windowsserver2008/en/us/compare-specs.aspx>

### 14.2.1 Service Pack 1 and Service Pack 2

When originally released, Windows Server 2008 included Service Pack 1 by default. Microsoft's reasoning behind this was to coincide the release of Windows Server 2008 and Microsoft Windows Vista, Service Pack 1. Because the core products are based on the same kernel and code, Windows Server 2008 Service Pack 1 included all the overlapping code between the two products.

Thus, there was no separate release of Service Pack 1 for Windows Server 2008 and it cannot be uninstalled or managed as a separate installation. To install Windows Server 2008 RTM is to install Windows Server 2008 with Service Pack 1.

Service Pack 2 for Windows Server 2008 was released in late May 2009. Because Service Pack 1 was included as part of the original release of the RTM product, Service Pack 2 is not a cumulative Service Pack dependent on previous Service Pack installations.

Service Pack 2 includes some important updates in the following area:

- ▶ Operating system update

Of particular note is the removal of 10 half-open outbound TCP connections. By default, SP2 has no limit on the number of connections.

- ▶ Improved power management

This includes the ability to shut down processors because they are no longer required during off-peak processing periods.

- ▶ Application compatibility enhancements

These enhancements are based on feedback from customers through the native Help Gathering process.

- ▶ Improved hardware support

Notably, this includes 64-bit (x64) processor support from VIA technologies and enhancements for Bluetooth and wireless network support (where appropriate).

- ▶ Setup and deployment improvements for the Service Pack itself

These include better error handling, logging, detection and blocking of incompatible drivers, and cleanup tools to recover drive space from files that are no longer required after Service Pack 2 is installed.

As with all Service Pack updates, we strongly recommend that all customers install Service Pack 2 on all Windows Server 2008 servers after appropriate testing. Service packs offer an accumulation of hotfixes and patches to the operating system that generally improve system stability and performance.

## 14.3 New features of Windows Server 2008

Windows Server 2008 introduces many new features to the Microsoft server product suite, some of which are enhancements to existing capabilities in previous versions of the product, and others which are entirely new functions. The most notable changes that have delivered performance improvements are detailed in the following sections.

### 14.3.1 Server roles

Windows Server 2008 seeks to improve performance and manageability by allowing engineers to configure the server with server “roles.” This is achieved by installing only the system components necessary to perform a specific function, such as a DHCP server, print server or Active Directory domain controller. A server can run with one or more server roles; however, each role will typically add to the system workload.

The following list highlights the various roles and services that can be deployed on an installation of Windows Server 2008:

- ▶ Active Directory Certificate Services
- ▶ Active Directory Domain Services
- ▶ Active Directory Federation Services
- ▶ Active Directory Lightweight Directory Services
- ▶ Active Directory Rights Management Services
- ▶ Application Server
- ▶ DHCP Server (typically used for hosting .NET applications)
- ▶ DNS Server
- ▶ Fax Server
- ▶ File Services
- ▶ Network Policy and Access Services
- ▶ Print Services
- ▶ Terminal Services
- ▶ Universal Description Discovery Integration (UDDI) Services
- ▶ Internet Information Services (IIS) - also known as Web Services
- ▶ Windows Deployment Services (WDS)
- ▶ Windows Sharepoint Services
- ▶ Windows Server Update Services (WSUS)

The new Server Manager administration tool in Windows Server 2008 is the utility used to manage server roles, including the common tasks of adding and removing roles, services, and features. Alternatively, the command line utility `servermanagercmd` can be used to achieve the same results using the graphical user interface. Figure 14-1 displays the server roles installation window.

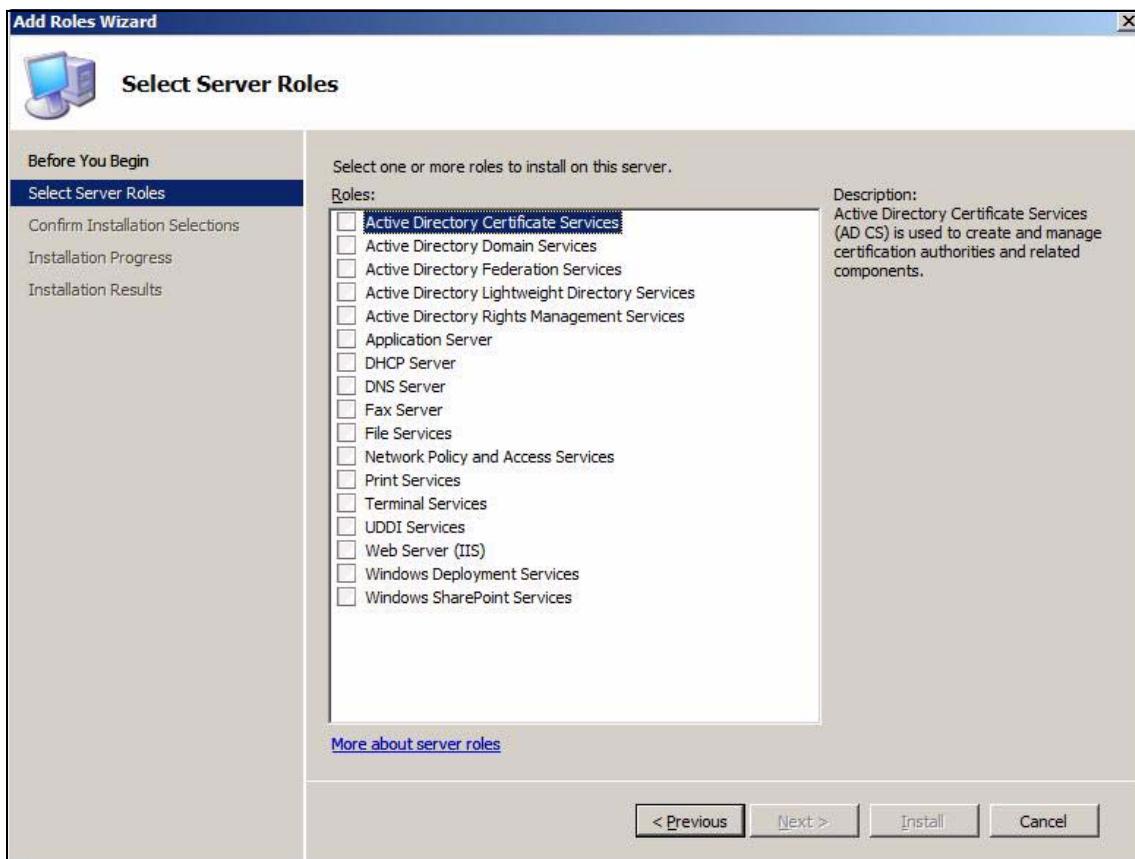


Figure 14-1 Windows Server 2008 Server Roles installation window

### 14.3.2 Server Core

Windows Server 2008 Server Core is one of the most interesting features of the new operating system. Server Core is essentially Windows Server 2008 without the graphical user interface. Stripping down Windows Server 2008 to a command-line only yields two primary benefits: improved security and increased performance.

The benefits of Server Core are discussed here in more detail:

- ▶ The Windows graphical user interface increased both performance load and security risks, and Server Core seeks to reduce both of these. By way of comparison, a standard installation of Windows Server 2008 requires approximately 6 GB of disk space, whereas Server Core only requires approximately 1 GB.

- ▶ With Server Core, regular maintenance tasks such as system patching are greatly reduced because a notable percentage of patches are only applicable to the graphical interface of Windows. This translates into less management overhead and greater system uptime.
- ▶ System resources are not as heavily taxed in a Windows system running Server Core because there is not as much going on in the background. Maintaining a rich graphical user interface like that of native Windows places a load on a server that Server Core is able to shed. Instead, Server Core-based systems function purely to service the applications and functions running on them, not on managing and presenting the overhead of an often rarely-used GUI.
- ▶ Reducing the number of services both installed and running by default on a Server Core installation has also improved performance. Only about 40 services are installed in Server Core, as opposed to about 75 services in a full installation of Windows Server 2008.
- ▶ Without a Web browser like Internet Explorer®, the chances of downloading a harmful application or virus are greatly decreased. In addition, many Windows-based viruses attack the core components that make up the Windows graphical interface, so the attack surface is considerably smaller for a Windows server running Windows Core.

However, learning to use Server Core may present a few challenges at first, because it is very different from using the Windows graphical interface and there is no Start menu. When a Windows Server 2008 running Server Core starts, the operating system presents only a command prompt window, as shown in Figure 14-2 on page 434.

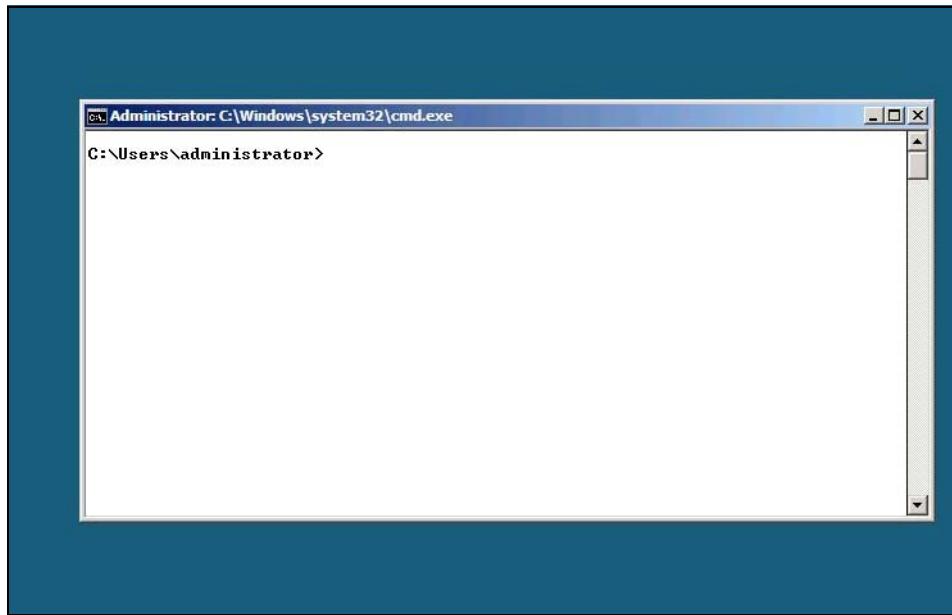


Figure 14-2 Windows Server 2008 Server Core interface

You can perform almost every administrative task, including changing an IP address, editing the registry, creating a volume, changing file system permissions and installing a new application, by using the Server Core command line interface, which is the only portal into the operating system.

With familiarity, executing such tasks using Server Core will become simple and easily repeatable. In many instances you will be able to perform them much faster than with the Windows graphical interface. This leads to higher productivity, and ultimately, higher performing systems. The trade-off for ease of use is flexibility and high performance.

Some of the GUI-based tools that are included with Server Core include Notepad, Task Manager, and Regedit.

Some of the components, utilities, and applications that are *not* included or able to be used with a Server Core installation include Windows Explorer, Internet Explorer, Wordpad, Paint, Media Player, Control Panel (although some individual control panels can be launched), MMC or any .NET-managed application or code. This list is expected to change as Server Core matures.

Despite its many benefits, Server Core does not suit every purpose. There is a defined set of roles that a Server Core system can run, and there are some that it cannot run. For example, Server Core is not normally considered an “application

server.” This is particularly the case where the command line does not have the necessary functionality to manage all features of a given application, such as Microsoft Exchange or SQL Server.

Note that you can still manage applications like DHCP, DNS, or Active Directory on a Windows Server Core system by using remote management MMC tools that are used for normally managing these applications. You can launch such tools from an administrator’s workstation or another server running the Windows GUI. Directing these tools to manage the remote Server Core installation will allow these applications to be administered in a very familiar manner.

Remote management of a Server Core system via RDP is faster than for traditional Windows servers. Instead of transmitting all screen refresh changes across the LAN or WAN that is associated with the Windows GUI, only command line entries need to be transmitted.

Server Core is not a separate product. It is an option chosen when installing Windows Server 2008. Server Core is installed as a server role and is available in the Standard, Enterprise, and Datacenter Editions of both the x86 and x64 versions of Windows Server 2008.

Server Core can be installed to support the following server roles (as outlined in 14.3.1, “Server roles” on page 431):

- ▶ Active Directory
- ▶ Active Directory Lightweight Directory Services (LDS)
- ▶ DHCP Server
- ▶ DNS Server
- ▶ File Server
- ▶ Streaming Media Services
- ▶ Print Services
- ▶ Windows Server Virtualization

Installations of Windows Server 2008 Server Core can include the following features:

- ▶ Bitlocker Drive Encryption
- ▶ Failover Clustering
- ▶ Multipath I/O
- ▶ Removable Storage Management
- ▶ SNMP Services
- ▶ Subsystems for UNIX-based Applications
- ▶ Telnet Client
- ▶ Windows Server Backup
- ▶ WINS Server

Server Core is an important new feature of Windows Server 2008 that helps you to optimize system performance. It delivers unique changes in the way the operating system is administered for servers having a graphical interface, and its streamlined build and functionality offers impressive performance advantages for servers running the services that Server Core accommodates.

### 14.3.3 Read-only Domain Controllers

The Read-Only Domain Controller (RODC) feature of Windows Server 2008 is a significant shift away from the typical model Active Directory's fully distributed directory service. Unlike typical Read-Write Domain Controllers (RWDC), a Read-Only Domain Controller hosts a read-only copy of the domain data in an Active Directory.

The RODC is typically aimed at installations in branch offices where the physical security of a server has more potential for compromise than those stored in large, central data centers. Because the RODC does not store user or computer credentials in their replicated copy of Active Directory, there is limited sensitive or valuable information held on the host server, thereby making it more secure. Only when a user authenticates against an RODC in an Active Directory site will the RODC communicate with the upstream RWDC to request a copy of the user's credentials.

As a result, any given RODC will typically only contain the credentials of anyone who has authenticated using that domain controller. As a result, the impact of a stolen server would be greatly reduced relative to a domain controller containing a full replica of the Active Directory domain.

Although primarily aimed at strengthened security, a side benefit of the RODC is improved performance. A lesser amount of data to replicate means that the domain controller is less occupied with replication activities and is instead able to focus on servicing client authentication requests.

Another key benefit is to the WAN and other systems accessed via the same link; less replicated traffic means less data on the network and as a result, higher-performing systems.

The client logon experience may also see performance improvements, particularly in corporations with complex directory implementation, because as Active Directory only has to traverse a small subset of organizational units and accounts to find that belonging to a requesting user.

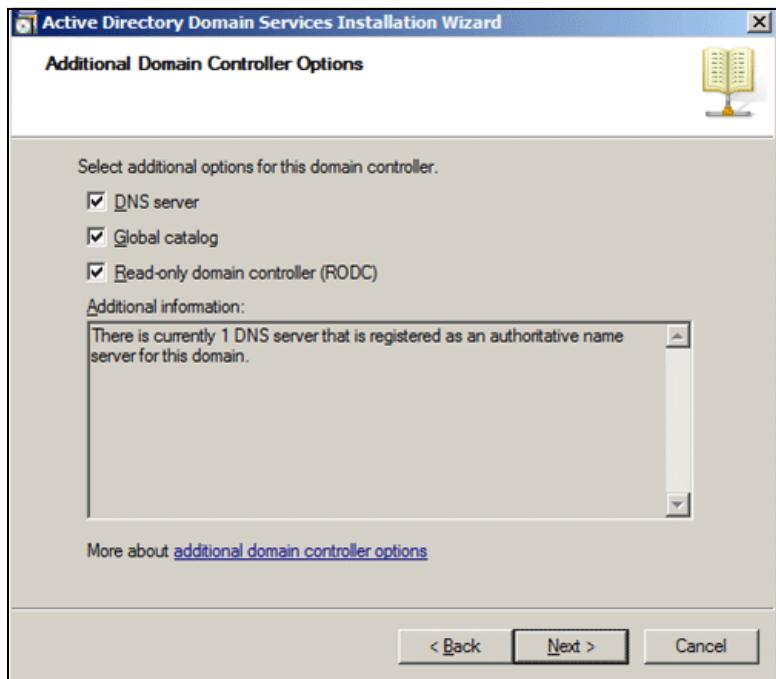


Figure 14-3 DCPromo installation wizard for Read-only Domain Controller

Read-only Domain Controller is installed when DCPromo.exe is run to install Active Directory services onto a Windows 2008 server, as shown in Figure 14-3. Note that there are a number of Active Directory prerequisite requirements for the successful installation of an RODC into a branch office:

1. The domain and forest are running at Windows Server 2003 functional mode or higher.
2. The PDC emulator function is hosted on a Windows 2008 RWDC.
3. There is only one AD site link separating the RODC and a RWDC.

#### 14.3.4 Hyper-V

Windows Server 2008 was delivered with the much anticipated virtualization hypervisor known as Hyper-V. Microsoft describes Hyper-V as its enterprise class virtualization offering, which surpasses the capabilities of its Virtual Server product, the latest version being Windows Virtual Server 2005 R2.

Hyper-V is delivered as a role of Windows Server 2008 Server Core. Windows Virtualization is the thin hypervisor layer that sits between the host hardware and

the guest virtual machines, and it manages the scheduling of system resources. The hypervisor also partitions the virtual machines from each other.

Hyper-V provides excellent integration with the Windows Server 2008 operating system environment and management tools.

Hyper-V is Microsoft's first serious step into the virtualization arena for virtualizing product server workloads. Using Hyper-V and virtualization is an excellent way to obtain improved hardware resource utilization.

Microsoft recommends Hyper-V for basic virtualization functionality, including:

- ▶ Test and development servers
- ▶ Basic server consolidation
- ▶ Branch server consolidation
- ▶ Hosted Desktop Virtualization (VDI).

Hyper-V provides 64-bit host and guest (virtual machine) support, which is the ability to run guest machines in a multi-processor environment,

The R2 release for Windows Server 2008, described in detail at the end of this chapter, brings considerable feature enhancements to Hyper-V including "Live Migration", which is the ability to move virtual machines from one Hyper-V host to another without causing any service outage.

With time Microsoft will include support for virtual machines running operating systems beyond just Windows, including some distributions of Linux such as SUSE Linux Enterprise Server.

#### **14.3.5 Windows System Resource Manager**

Windows System Resource Manager (WSRM) is a powerful feature of Windows Server 2008 that is applicable to server implementations where multiple users or applications are executing applications or processes locally on a given system. This is particularly applicable to Terminal Servers, IIS, and any application server running multiple (multi-purpose) applications locally.

WSRM seeks to balance system resources evenly among user sessions to ensure that no individual session can monopolize system resources and drastically impact the Terminal Server experience for another user.

This delivers key performance benefits in that more applications, processes, or services can run concurrently on a given server implementation. It also delivers a key management advantage, which is that server administrators can access a system even in times of maximum server load.

WSRM is used to manage how CPU and memory resources between session are shared through the use of policies. WSRM only becomes active when the combined processor load on a server exceeds 70%.

Five preconfigured policies come with a default installation of WSRM:

- ▶ Equal per process - each process is given equal treatment.
- ▶ Equal per user - processes are grouped according to the user account running them.
- ▶ Equal per session - resources are allocated on an equal basis for each session.
- ▶ Equal per IIS application pool - each running IIS application pool is treated equally.
- ▶ Weighted remote sessions - processes are grouped according to the priority associated with the user account; priorities include Basic, Standard and Premium.

WSRM is an optional feature that is installed as part of Terminal Services via the Server Manager utility. For multipurpose servers, despite requiring some performance overhead, its functionality far outweighs any minor system impact it creates.

## 14.4 Networking performance

Windows Server 2008 brings major performance improvements to the network subsystem. Some of the most notable are listed in the following sections.

### 14.4.1 Server Message Block version 2.0

Server Message Block (SMB) also goes by the term Common Internet File System (CIFS). SMB is the protocol for file sharing used by Windows. Largely untouched for the last 15 years, Windows Server 2008 and Windows Vista have brought about a rewrite of SMB to version 2.0 and delivered significant improvements in file sharing performance.

Key features of SMB 2.0 include:

- ▶ Reduced packets sent between a client and server through support for sending multiple SMB commands within the same packet
- ▶ Considerably larger buffer sizes
- ▶ Greater resiliency to network outages through the use of more persistent file handles

Although Windows Server 2008 and Windows Vista receive and transmit files with older operating systems, this will only occur over SMB 1.0. SMB 2.0 will only be negotiated when Windows Server 2008 is communicating with Windows Vista or other Windows Server 2008 servers, and vice versa.

#### **14.4.2 TCP/IP stack improvements**

Microsoft has delivered very measurable performance improvements to Windows Server 2008 through improvements in the core TCP/IP stack.

In addition, with Windows Server 2003, Microsoft released the Scalable Networking Pack as an out-of-band release. This included support for TCP Chimney Offload, Receive Side Scaling (RSS) and Network Direct Memory Access (NetDMA). These features are now included natively with Windows Server 2008 and can deliver considerable network performance benefits.

##### **TCP Chimney Offload**

TCP Chimney Offload is a stateful offload that enables TCP/IP processing to be offloaded to network adapters that can handle the processing in hardware. Network adapters need to be enabled to support TCP Chimney Offload, and there is a maximum number of connections that will be supported by hardware. Beyond this, the connections will still need to be handled by the operating system.

##### **Receive Side Scaling**

Multiple CPU Windows servers will typically limit network “receive” protocol processing by a network adapter to a single CPU. Receive Side Scaling (RSS) is a stateless offload that balances the packets received by a server across multiple CPUs.

##### **Network Direct Memory Access**

Network Direct Memory Access (NetDMA) is a stateless offload that allows for a Direct Memory Access (DMA) engine on the server PCI bus. The TCP/IP stack can use the DMA engine to copy data instead of using the system CPU to perform the process.

The following knowledge base articles describe TCP Chimney Offload, Receive Side Scaling (RSS) and Network Direct Memory Access, as well as the affected registry settings and their interactions with each other in considerably more detail than is outlined in this chapter:

<http://support.microsoft.com/kb/951037>  
<http://support.microsoft.com/kb/912222/>

## **Receive Window Auto Tuning**

The TCP receive window is the maximum number of bytes that a sender can transmit without receiving an acknowledgment from the receiver. The larger the window size, the fewer acknowledgements are sent back, and the more optimal the network communications are between the sender and receiver. Having a smaller window size reduces the possibility that the sender will time out while waiting for an acknowledgement, but it will increase network traffic and reduce throughput.

Previous versions of Windows Server have provided mechanisms for improving TCP receive window tuning through registry modifications, as described in 13.15.1, “TCP window size” on page 395 and 13.15.2, “Large TCP window scaling and RTT estimation (time stamps)” on page 396 including settings for Windows TCP receive window scaling.

Unfortunately, this means statically setting the TCP receive window, which does not allow for varying degrees of network bandwidth or reliability. This behavior is no longer required or supported in Windows Server 2008 and thus the registry values described for Windows Server 2003 do not apply for Windows Server 2008. The Receive Window Auto Tuning feature ensures that throughput between TCP nodes is constantly optimized given fluctuating network conditions.

## **Compound TCP**

TCP includes algorithms to prevent it from overwhelming a network with data, known as “slow start and congestion avoidance”. However, this means that bandwidth utilization is not optimized for quite some time after data transmission commences.

These algorithms work sufficiently well for LANs with smaller TCP window sizes. However, networks with high bandwidth and high delay, such as replicating data between two servers located across a high-speed WAN link, are hindered by these algorithms because they do not increase the send window fast enough to fully utilize the bandwidth of the connection. By way of example, a 1 Gbps WAN link with a 100 ms round trip time (RTT), can take up to an hour for the send window to initially increase to the large window size and obtain the highest throughput and bandwidth efficiency.

Compound TCP (CTCP), included with Windows Server 2008, more aggressively increases the send window for connections with large receive window sizes and large bandwidth-delay products. By monitoring delay variations and losses, CTCP attempts to maximize throughput on these types of connections. Connections with ample bandwidth can have even better performance. CTCP and Receive Window Auto-Tuning work together for increased link utilization, and they can result in substantial performance gains for large bandwidth-delay product connections.

Windows Server has CTCP enabled by default. In Vista, it is not enabled by default.

## **Explicit Congestion Notification support**

When a TCP segment is lost, the TCP protocol addresses this by quickly reducing the sender's rate of transmission. This may be due to a lost segment or simply due to congestion. This can take some time to recover from, thus hindering the performance of data transmission.

Explicit Congestion Notification (ECN) seeks to dynamically detect when network congestion is occurring and slow the rate of transmission before a TCP segment is lost and performance is seriously hindered. ECN is now included with Windows Server 2008 and Windows Vista (though disabled in the Vista by default).

## **Path Maximum Transmission Unit (PMTU) Discovery**

Under most circumstances, the maximum transmission unit (MTU) of every network segment that a server might possibly communicate over will not be known. Remote networks will often have an entirely different MTU to that of local networks.

When enabled in previous editions of Windows Server, the registry value `EnablePMTUDiscovery` lets TCP/IP automatically determine the MTU for all networks along the path to a remote host. After the MTU has been determined for all network segments on a path, it will use the highest, and thus most efficient, MTU value that can be used without packet fragmentation occurring.

In Windows Server 2008, this behavior is now enabled by default, ensuring that bandwidth performance for all segments of a network link is optimized.

## **Registry values no longer relevant**

Due to the improvements in the networking stack in Windows Servers 2008, the following registry values as described in Chapter 13, "Microsoft Windows Server 2003" on page 351, are no longer relevant.

### **Registry keys:**

- ▶ `HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\TcpWindowSize`
- ▶ `HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Tcp\NumTcbTablePartitions`
- ▶ `HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Tcp\MaxHashTableSize`

### **14.4.3 Tolly Group study**

Microsoft has delivered significant performance, throughput, and reliability improvements with Windows Server 2008. To obtain an independent testing and analysis of this, they commissioned the Tolly Group to perform a benchmarking exercise.

Tolly Group has found that file transfer improvements improved by up to four times for large file transfers between Windows Server 2008 and Windows Vista when compared to the same transfers between Windows Server 2003 and Windows XP. This was achieved through default settings and by relying only on the performance benefits delivered through the new operating systems. The series of tests also showed much more efficient use of the available network bandwidth, in some cases almost sending data at line speed.

The results of the Tolly Group study can be found at the following location:

<http://www.tolly.com/docdetail.aspx?docnumber=208306>

## **14.5 Storage and file system performance**

The storage subsystem has received some improvements in Windows Server 2008 that will be of interest to performance-oriented server engineers.

### **14.5.1 Self-healing NTFS**

Previous versions of the Windows server operating system have relied on the CHKDSK tool to repair corruptions with the NTFS file system. CHKDSK is a disk-intensive and intrusive tool. While CHKDSK is running, it typically requires an interruption (or impact) to the service that the hosting server is providing.

Windows Server 2008 introduces self-healing NTFS to the file system, which is largely enabled by default. Self-healing NTFS attempts to correct file system corruptions without the need to run CHKDSK. Improvements to the operating system kernel allow self-healing NTFS to function without any performance impact to the server. This is an important feature for server engineers who are concerned with obtaining optimal file system performance to the system.

### **14.5.2 Capacity limits**

Windows Server 2008 recognizes that demands on storage continue to grow at an exponential rate. With the latest version of the Microsoft server platform, the

maximum NTFS volume size is 256 TB through the use of 64KB clusters (allocation units). This drops to a maximum of 16TB when using 4 KB clusters.

The maximum individual file size now supported in Windows Server 2008 is 16 TB, and the maximum number of files that can be accommodated on a single volume is 4,294,967,294.

### 14.5.3 Hardware versus software RAID

As with previous versions, Windows Server 2008 accommodates operating system (software)-based RAID configurations. Supported RAID implementations include RAID 0 (striping), RAID-1 (mirroring), and RAID-5 (striping with parity). These are implemented using the native Disk Management tool.

Although RAID implemented with the operating system is straightforward and easy to set up, we do not normally recommend it where performance tuning is essential. Hardware-based RAID requires the purchase and installation of an additional RAID-controller. However, the considerable performance and redundancy benefits obtained by offloading this function from the operating system to an external device typically far outweigh any cost disadvantage.

External RAID controllers reduce the requirement for precious operating system processing power to be spent managing the disk subsystem, and typically introduce further RAID configurations and performance advantages such as disk caching, as described in the following section.

### 14.5.4 Disk write-caching

Disk write-caching offers performance advantages by indicating to the operating system that the data has been written to disk before it actually has been. It stores disk write requests together and caches them for writing to the disk either periodically or during system idle time to the disk. This can substantially improve storage performance.

Windows Server 2008 provides the ability to manage disk write caching through the Disk Management tool, where the underlying disk controller and disk support it. To open the applet to configure write caching, right-click the volume you want to manage in the Disk Management tool and select **Properties**. Then select the Policies tab, as shown in Figure 14-4 on page 445.

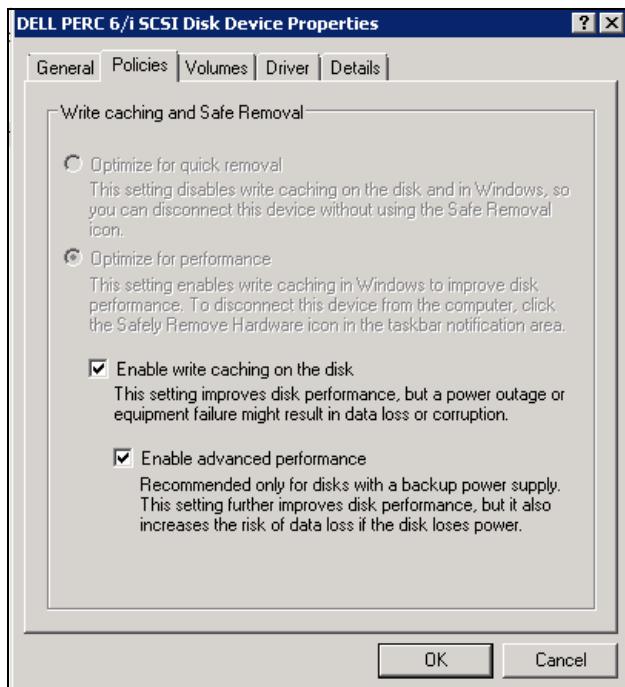


Figure 14-4 Disk Write-caching applet

From here, perform the following tasks:

- ▶ Select **Enable write caching on the disk** - this enables the disk-write caching feature of Windows Server 2008 for the selected disk.
- ▶ Select **Enable advanced performance** - this option is only available when the above setting is enabled. This setting ignores all write-through flags from disk requests and removes all flush-cache commands, offering a possible further performance advantage.

Windows Server 2008 disables disk write caching by default. The disadvantage of disk caching is the possibility of data being lost that is in cache at a given time and has not been written to disk in the event of a power outage.

We recommend the use of write caching providing that adequate power protection precautionary measures have been implemented, including uninterruptible power supplies (UPS) and disk RAID controllers that have battery-backed cache. This ensures that any data that is not written to disk in the event of a power outage is still stored and able to be written back to disk after the system power has been restored, minimizing the chance of a disk corruption.

#### **14.5.5 GPT and MBR disk partitions**

Windows Server 2008 has introduced support for both MBR (master boot record) and GPT (GUID partition table) partition formats for both x86 (32-bit)-based and x64 (64-bit)-based operating system versions. Although the MBR file system has been available for x86-based systems for a long time, the GPT file system format was previously only available for the x64 edition of Windows Server for Itanium-based computers.

The primary difference between the two partition styles is in regard to how the partition data is stored.

- ▶ MBR volumes have a single partition table and support up to four *primary partitions* or three primary partitions and one *extended* partition type, the latter which contains up to four *logical* drives.
- ▶ GPT volumes are generally considered more flexible than the MBR type. GPT volumes store partition data in each individual partition instead of a single partition table like MBR. There are also redundant primary and backup partition tables and enhanced error correction through checksum fields. GPT can also support up to 128 partitions per disk and a maximum raw partition size of 18 Exabytes.

#### **14.5.6 Partition offset**

When creating a volume, previous versions of Windows Server required the use of the DISKPART utility to align the partition boundary offset with the underlying array disk stripe unit boundaries. Without this modification, disk performance was not optimized.

In Windows Server 2008, partitions are now created with a 64 K offset for volumes under 4 GB, or a 1 MB offset for volumes larger than 4 GB. This removes the requirement to use DISKPART and ensures optimal performance. DISKPART can still be used if desired to force alternative alignments.

#### **14.5.7 Last-access time stamp**

Previous versions of Windows server updated the last-accessed time of a file every time the file was opened, read, or written to. This had some burden on performance, particularly on disk I/O-intensive servers. Windows Server 2008 recognizes that most applications do not depend on this file system feature and thus, with this version of the operating system, it is disabled by default.

The registry key that controls this setting is set as follows:

**Registry key:** HKLM\System\CurrentControlSet\Control\FileSystem\NTFSDisableLastAccessUpdate

**Values:** 0 or 1 (0 to disable, the default; 1 to enable)

## 14.6 Other performance tuning measures

This section describes further tuning tasks you can perform.

### 14.6.1 Visual effects

The visual effects that can have a pleasing cosmetic effect for users of desktop operating systems do not typically belong on server consoles. We recommend that you disable these to ensure that system performance is optimized, and then focus on servicing applications rather than on screen aesthetics.

To disable visual effects, follow these steps.

1. Open the **System** Control panel.
2. Click the **Advanced** tab. Under **Performance**, click **Settings**.
3. Click the **Visual Effects** tab, then select the radio button **Adjust for best performance**.
4. Press **OK** to apply these changes.

Figure 14-5 on page 448 shows the Visual Effects Performance Options control panel.

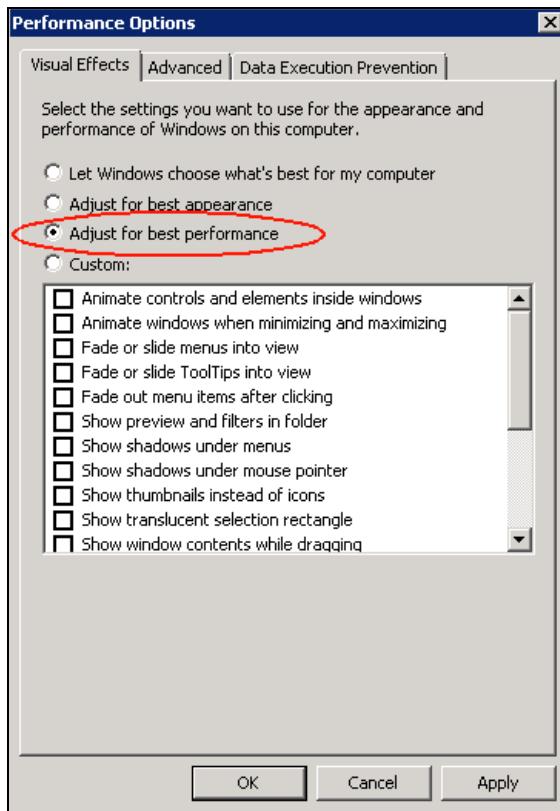


Figure 14-5 Visual Effects performance options control panel

## 14.6.2 System Configuration utility

Windows Server 2008 has improved the System Configuration utility that has been available in Windows for several versions. It has now removed references to the old SYSTEM.INI and WIN.INI configuration files.

This utility can be launched by executing `msconfig.exe` or, alternatively, from **Start → Administrative Tools → System Configuration** on the server. This is a powerful and simple tool you can use to save opening the Services Control Panel, Startup folders, or system registry.

Of particular note for performance tuning are the **Services** and **Startup** tabs. By viewing the applications, processes, and services that are starting with the system at boot time, there is an opportunity to remove unnecessary load on the server by disabling services or removing applications that are no longer required.

On the **Services** tab, selecting the **Hide All Microsoft Services** is a valuable feature to determine what non-operating system applications and services have been installed on the system.

The System Configuration utility is shown in Figure 14-6.

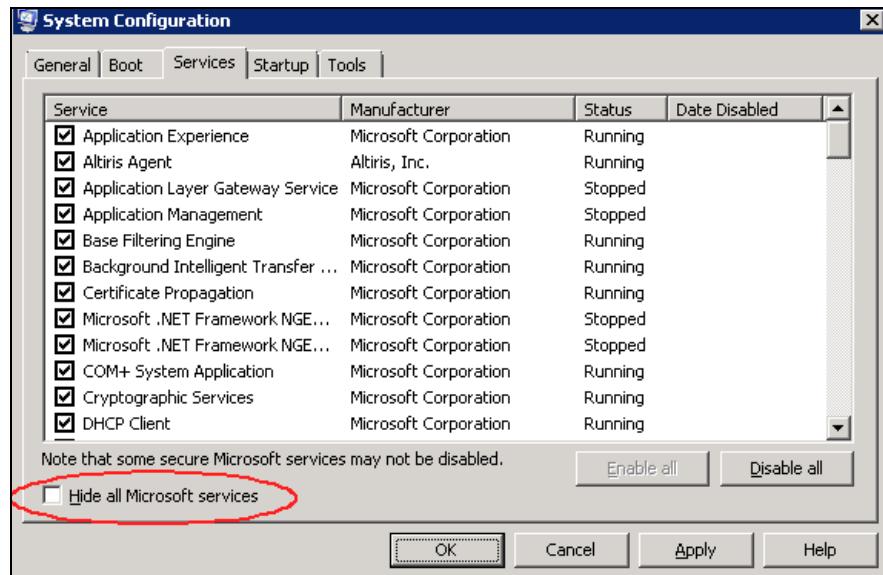


Figure 14-6 System Configuration Utility - Services tab

### 14.6.3 Windows Error Reporting - per process dump files

Windows Error Reporting (WER) within Windows Server 2008 can be configured to collect full dump files for user-mode processes (that is, applications) and store them locally after a user-mode process crashes, where they can be used for further debugging, analysis, and ultimately, performance tuning. This feature is configured through a series of registry values, as listed in Table 14-2 on page 450.

Table 14-2 Registry values used to configure per-process dump files

Value	Description	Type	Default Value
<b>DumpFolder</b>	The path where the dump files are to be stored. If you do not use the default path, then make sure that the folder contains ACLs that allow the crashing process to write data to the folder.	REG_EXPAND_SZ	%LOCALAPPDATA%\CrashDumps
DumpCount	The maximum number of dump files in the folder. When the maximum value is exceeded, the oldest dump file in the folder will be replaced with the new dump file.	REG_DWORD	10
DumpType	Specify one of the following dump types:  0 - Custom Dump 1 - Mini dump 2 - Full dump	REG_DWORD	1
CustomDumpFlags	The custom dump options to be used. This value is used only when <b>DumpType</b> is set to 0.	REG_DWORD	MiniDumpWithDataSegs MiniDumpWithUnloadedModules MiniDumpWithProcessThreadData

For further information about the configuration of these settings, consult the content found at the following page:

[http://msdn.microsoft.com/en-us/library/bb787181\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/bb787181(VS.85).aspx)

## 14.7 Windows Server 2008 R2

Windows Server 2008 Release 2 (R2) is scheduled for release in late October 2009. It expands the existing features of Windows Server 2008 and adds several new features to improve server reliability, performance and functionality. These product enhancements are listed in the following sections.

## **Scalability and reliability**

R2 delivers considerable scalability, reliability, and performance differences:

- ▶ Windows Server 2008 R2 introduces support for up to 256 logical processor cores per operating system instance.
- ▶ A key change for Windows Server 2008 R2 is that it is the first Windows server operating system that will *only* be available in 64-bit (x64) editions.
- ▶ Powershell scripting support will be introduced for Windows Server Core made possible by the support of the .NET framework.
- ▶ The R2 release brings with it support for new server roles and improvements to existing roles, such as IIS.
- ▶ Network Load Balancing (NLB) will support applications and services that require persistent connections to a given NLB node and improved health monitoring and awareness for applications and services running on NLB clusters.
- ▶ The R2 release offers considerably improved storage solutions including reduced processor overhead, increased multi-path throughput, and improved connection performance for iSCSI attached storage. Further enhancements will be delivered to storage fault tolerance and recoverability, as well as improved manageability and monitoring of the storage subsystem and settings.

## **Desktop and server virtualization**

Windows Server 2008 R2 introduces a new version of Hyper-V, including a Live Migration feature for moving virtual machines between virtual hosts with no interruption to service.

Hyper-V in Windows Server 2008 R2 permits up to 64 logical processors to be accessed in the host processor “pool”. Hyper-V also takes advantage of Second Level Address Translation (SLAT), which makes the most of features in today’s CPUs to improve VM performance while reducing load on the Hyper-V hypervisor.

Terminal Services has been renamed in Windows Server 2008 R2 as Remote Desktop Services (RDS), in line with Microsoft’s step towards desktop virtualization services. RDS introduces the RemoteApp capability, similar to the Citrix Published Application concept, in that the application is running on a back-end server while the application appears local on the user’s desktop. The integration is particularly tight with the new desktop offering, Windows 7.

## **Improved Web application platform**

Windows Server 2008 R2 introduces Internet Information Services (IIS) 7.5, including support for increased automation, new remote administration functionality, enhanced auditing of configuration changes to IIS, and performance support and troubleshooting through the Best Practices Analyzer (BPA). It further includes improved FTP compatibility, greater support for .NET, enhanced application pool security, and the ability to extend IIS functionality through extensions.

## **Power management**

Windows Server 2008 R2 introduces an exciting new feature called Core Parking to assist with reducing power consumption in servers with multi-core processors. Core Parking consolidates processing activities onto fewer cores and suspends inactive processors, thereby reducing power consumption.

## **Improved Powershell**

Powershell 2.0 is introduced with Windows Server 2008 R2, including improved remote server management functionality, portability of scripts between multiple computers, enhanced GUIs for creating and debugging scripts and constrained runspaces to improve security for management data, including state and configuration information.

## **Branch office performance**

Windows Server 2008 R2 includes BranchCache. BranchCache, in a completely transparent way, reduces WAN link utilization and provides end-to-end encryption and optimized HTTP, SMB, and BITS protocols.

BranchCache is supported in two modes:

- ▶ **Distributed mode**

This is a peer-to-peer mode where remotely accessed data is cached on local Windows 7 PCs in the branch office that can be used by other clients in the branch office to save traversing the WAN.

- ▶ **Hosted caching**

In this mode, a server at the branch office caches the remote content that clients are accessing and accordingly reduces WAN network traffic and speeds up performance of data access for users. The advantage of this over distributed mode is that the server is normally always available, unlike the Windows 7 PCs that will may or may not be in the office at any given time.



# Linux

By its nature and heritage, the Linux distributions and the Linux kernel offer a variety of parameters and settings to let the Linux administrator tweak the system to maximize performance. This chapter describes the general Linux kernel concepts needed to understand performance and the steps that you can take to tune Red Hat Enterprise Linux (RHEL)<sup>1</sup> or SUSE Linux Enterprise Server (SLES)<sup>2</sup>. It describes the parameters that give the most improvement in performance and provides a basic understanding of the tuning techniques that are used in Linux. This chapter covers the following topics:

- ▶ 15.1, “Linux kernel 2.6 overview” on page 454
- ▶ 15.2, “Working with daemons” on page 455
- ▶ 15.3, “Shutting down the GUI” on page 461
- ▶ 15.4, “Security Enhanced Linux” on page 464
- ▶ 15.5, “Changing kernel parameters” on page 466
- ▶ 15.6, “Kernel parameters” on page 469
- ▶ 15.7, “Tuning the processor subsystem” on page 473
- ▶ 15.8, “Tuning the memory subsystem” on page 476
- ▶ 15.9, “Tuning the file system” on page 480
- ▶ 15.10, “Tuning the network subsystem” on page 492
- ▶ 15.11, “Xen virtualization” on page 496

<sup>1</sup> Red Hat is a registered trademark of Red Hat, Inc. The information and screen captures in this chapter were used with permission.

<sup>2</sup> SUSE Linux is a registered trademark of SUSE Linux AG, a Novell, Inc. company. The information and screen captures in this chapter were used with permission from Novell, Inc.

## 15.1 Linux kernel 2.6 overview

The current Linux kernel main release, 2.6, has been implemented in all major Linux distributions. Significant improvements have been made since release 2.4, and we review them here to help you identify and solve potential performance issues.

As illustrated in Figure 15-1, the main advances of the new kernel are mainly in the areas of scalability, performance, common hot plug infrastructure, and security. In this book we discuss the elements relevant to performance in detail.

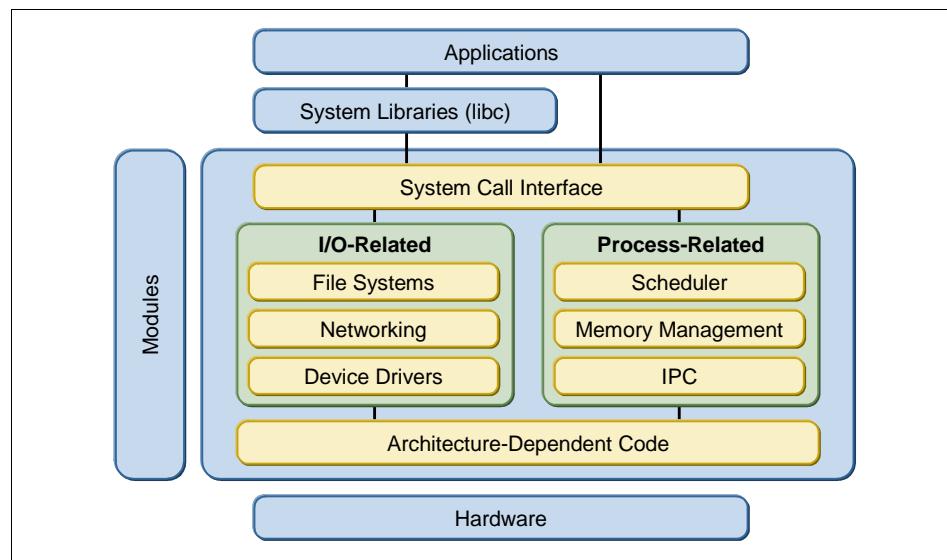


Figure 15-1 Linux 2.6 kernel high level description

Both SUSE Linux Enterprise Server (SLES) and Red Hat Enterprise Linux (RHEL) use the 2.6 kernel. The main new features of the 2.6 kernel are listed here.

- ▶ Scale up: Large SMP and NUMA
  - 32 CPU xSeries
  - 64 CPU pSeries/iSeries
  - 16 CPU zSeries®
- ▶ Major kernel internal overhauls for robustness, performance, scalability>
  - VM, Scheduler, NUMA topology
  - File system and block IO
  - Hyper-Threading and SMT support
  - Up to 256 GB memory support

- Max users/Groups from 64 K to 4 billion
- PIDs/processes from 32 K to 1 billion
- 16 TB file system, 1 million devices
- ▶ Common hot plug infrastructure for PCI devices, CPU, USB and Firewire
- ▶ Security: Policy-based security architecture and new security policies (SELinux and so on)
- ▶ Scalable APIs: futexes, epoll, Direct I/O & Async I/O, Large Page APIs, Native POSIX Thread Library (NPTL), NUMA APIs and topology, distributed file system support, and IRQ and scheduling affinity
- ▶ New networking protocols: Stream Control Transmission Protocol (SCTP), IPv6, Mobile IPv6 and DHCPv6
- ▶ Enhanced file system support: SCTP, IPv6, Mobile IPv6 and DHCPv6

## 15.2 Working with daemons

On every server, there are daemons (background services) running that are not always needed. Disabling these daemons frees memory, decreases startup time, and decreases the number of processes that the CPU must handle. A side benefit is increased server security, because fewer daemons mean fewer security risks in the system.

By default, several daemons that have been started can be stopped and disabled safely on most systems. Table 15-1 lists the daemons that are started in various Linux installations. Consider disabling these daemons in your environment, if appropriate.

Table 15-1 lists the respective daemons for several commercially available Linux distributions. However, the exact number of running daemons might differ from your specific Linux installation. For a more detailed explanation of Red Hat daemons, refer to the Red Hat Service Configuration detail shown in Figure 15-2 on page 458. For SUSE Linux daemons, refer to the YaST GUI shown in Figure 15-3 on page 459.

*Table 15-1 Tunable daemons started on a default installation*

Daemons	Description
apmd	Advanced power management daemon. apmd will usually not be used on a server.
arptables_jf	User space program for the arptables network filter. Unless you plan to use arptables, you can safely disable this daemon.

<b>Daemons</b>	<b>Description</b>
autofs	Automatically mounts file systems on demand (for example, mounts a CD-ROM automatically). On server systems, file systems rarely have to be mounted automatically.
cpuspeed	Daemon used to dynamically adjust the frequency of the CPU. In a server environment, this daemon is recommended off.
cups	Common UNIX Printing System. If you plan to provide print services with your server, do not disable this service.
gpm	Mouse server for the text console. Do not disable if you want mouse support for the local text console.
hpoj	HP OfficeJet support. Do not disable if you plan to use an HP OfficeJet printer with your server.
irqbalance	Balances interrupts between multiple processors. You may safely disable this daemon on a single CPU system or if you plan to balance IRQ statically.
isdn	ISDN modem support. Do not disable if you plan to use an ISDN modem with your server.
kudzu	Detects and configures new hardware. Should be run manually in case of a hardware change.
netfs	Used in support of exporting NFS shares. Do not disable if you plan to provide NFS shares with your server.
nfslock	Used for file locking with NFS. Do not disable if you plan to provide NFS shares with your server.
pcmcia	PCMCIA support on a server. Server systems rarely rely on a PCMCIA adapter so disabling this daemon is safe in most instances.
portmap	Dynamic port assignment for RPC services (such as NIS and NFS). If the system does not provide RPC-based services there is no need for this daemon.
rawdevices	Provides support for raw device bindings. If you do not intend to use raw devices you may safely turn it off.
rpc	Various remote procedure call daemons mainly used for NFS and Samba. If the system does not provide RPC-based services, there is no need for this daemon.
sendmail	Mail Transport Agent. Do not disable this daemon if you plan to provide mail services with the respective system.
smartd	Self Monitor and Reporting Technology daemon that watches S.M.A.R.T.-compatible devices for errors. Unless you use an IDE/SATA technology-based disk subsystem, there is no need for S.M.A.R.T. monitoring.

Daemons	Description
xfs	Font server for X Windows. If you will run in runlevel 5, do not disable this daemon if you are using the graphical interface.

**Attention:** Turning off the xfs daemon prevents X from starting on the server. This should be turned off only if the server will not be booting into the GUI. Simply starting the xfs daemon before issuing the `startx` command enables X to start normally.

On SLES and RHEL systems, the `/sbin/chkconfig` command provides the administrator with an easy-to-use interface to change start options for various daemons. One of the first commands that should be run when using chkconfig is a check for all running daemons:

```
/sbin/chkconfig --list | grep on
```

If you do not want the daemon to start the next time the machine boots, issue either one of the following commands as root. They accomplish the same results, but the second command disables a daemon on all run levels, whereas the `--level` flag can be used to specify exact run levels:

```
/sbin/chkconfig --levels 2345 sendmail off
/sbin/chkconfig sendmail off
```

**Tip:** Instead of wasting time waiting for a reboot to complete, simply change the run level to 1 and back to 3 or 5, respectively.

There is another useful system command, `/sbin/service`, that enables an administrator to immediately change the status of any registered service. In a first instance, an administrator should always choose to check the current status of a service (`sendmail`, in our example) by issuing this command:

```
/sbin/service sendmail status
```

To immediately stop the `sendmail` daemon in our example, use this command:

```
/sbin/service sendmail stop
```

The `service` command is especially useful because it lets you immediately verify whether or not a daemon is needed. Changes performed through `chkconfig` will not be active unless you change the system run level or perform a reboot. However, a daemon disabled by the `service` command will be re-enabled after a reboot. If the `service` command is not available with your Linux distribution, you

can start or stop a daemon through the init.d directory. Checking the status of the CUPS daemon, for example, could be performed like this:

```
/etc/init.d/cups status
```

Similarly, there are GUI-based programs for modifying which daemons are started, as shown in Figure 15-2. To run the service configuration GUI for Red Hat Enterprise Linux, click **Main Menu → System Settings → Server Settings → Services** or issue this command:

```
/usr/bin/redhat-config-services
```

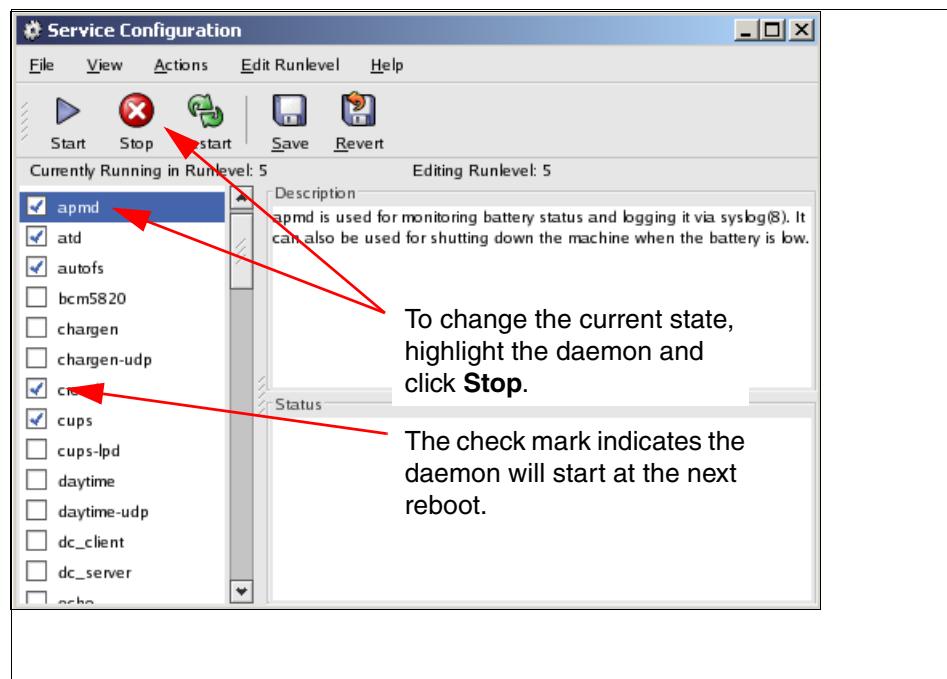


Figure 15-2 Red Hat Service Configuration interface

Novell SUSE systems offer the same features via the YaST utility. In YaST, the service configuration can be found under **System → System Services (Runlevel)**. After you are in the service configuration, we suggest that you use the expert mode to accurately set the status of the respective daemon. Running YaST in runlevel 3 would look as shown in Figure 15-3 on page 459.

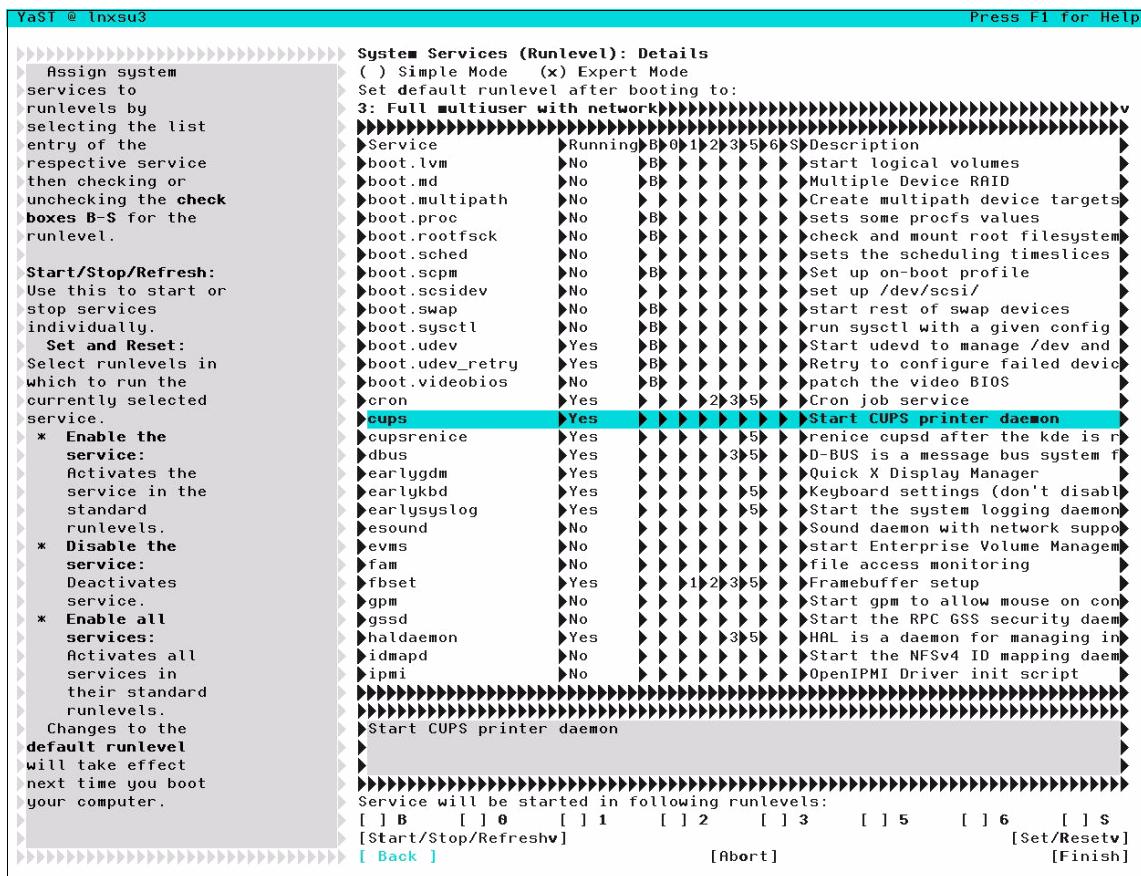


Figure 15-3 The System Services panel in YaST

In the YaST panel shown in Figure 15-3, various services can be enabled or disabled on a per-run level basis. However, this requires the utilization of the expert mode as displayed at the top of Figure 15-3.

You can stop a daemon immediately if required. For example, you can stop the sendmail daemon using the following commands:

- ▶ RHEL: `/sbin/service sendmail stop`
- ▶ SLES: `/etc/init.d/sendmail stop`

You can use the **status** parameter to see whether a daemon is running, and the **start** parameter to start the daemon.

You can also configure the daemon to not start the next time the server starts. Again, for the sendmail daemon, you can use the following commands:

- RHEL: `/sbin/chkconfig sendmail off`
- SLES: `/sbin/chkconfig -s sendmail off`

In addition, these distributions provide a graphical interface to managing daemons.

**Tip:** People often think that changes performed through `chkconfig` are not active until the next reboot. In reality, changing the run level has the same effect on the running daemons as rebooting does. So, instead of waiting for a reboot to complete, simply change the run level to 1 and back to 3 or 5, respectively.

To run the GUI, issue the following command:

- RHEL 3: `/usr/bin/redhat-config-services`
- RHEL 4: `/usr/bin/system-config-services`

Alternatively, you can click **Main Menu → System Settings → Server Settings → Services**. The Red Hat Service Configuration window opens, as shown in Figure 15-4.

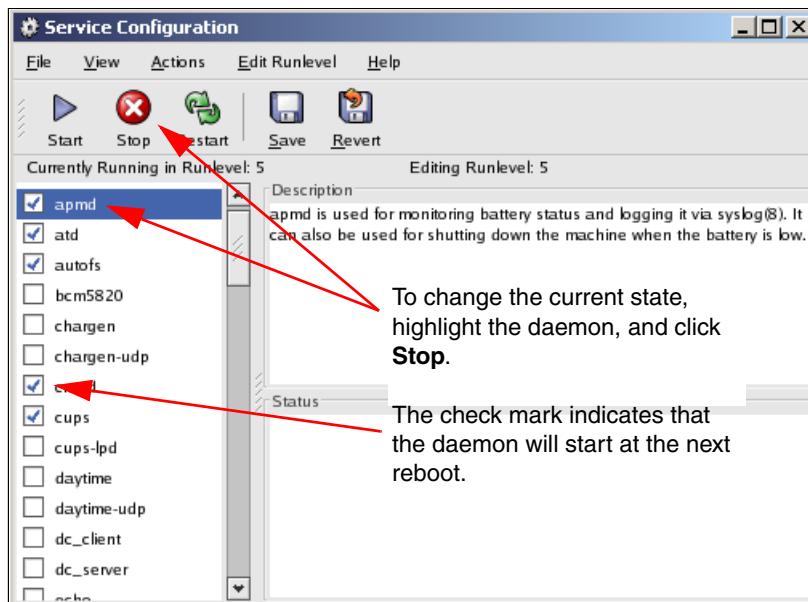


Figure 15-4 Red Hat Service Configuration interface

**Tip:** Not all daemons appear in the Red Hat Service Configuration window. To see a complete list, issue the following command:

```
/sbin/chkconfig --list
```

For SLES the graphical interface is YaST2, which you can start either with the command `/sbin/yast2 runlevel` or by clicking **Browse: YaST/ → YaST modules → System → Runlevel editor**, as shown in Figure 15-5.

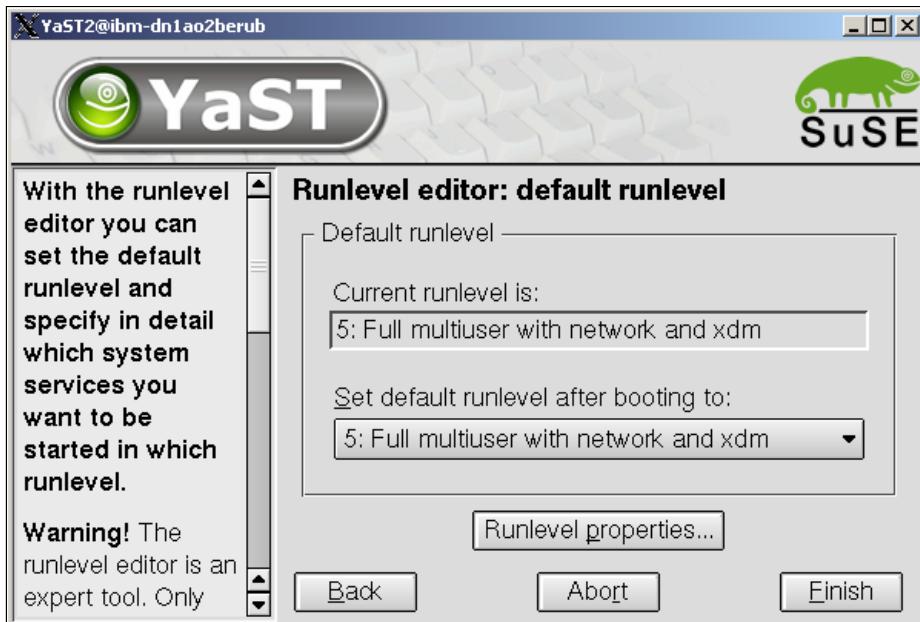


Figure 15-5 SUSE Linux YaST runlevel editor

### 15.3 Shutting down the GUI

Whenever possible, do not run the GUI on a Linux server. Normally, there is no need for a GUI on a Linux server. You can perform all administration tasks through the command line by redirecting the X display or through a Web browser interface. There are also several different useful Web-based tools (for example, webmin, Linuxconf, and SWAT) to perform administrative tasks.

If you must use a GUI, then start and stop it as needed rather than run it all the time. In most cases the server should be running at run level 3, which does not start the X Server when the machine boots up. If you want to restart the X Server, use the `startx` command from a command prompt. Follow these steps:

1. Determine at which run level the machine is running by using the `runlevel` command.

This command prints the previous and current run level. For example, N 5 means that there was no previous run level (N) and that the current run level is 5.

2. To switch between run levels, use the `init` command. (For example, to switch to run level 3, enter `init 3`.)

The run levels that are used in Linux are:

- 0** Halt (Do not set `initdefault` to this or the server will shut down immediately after finishing the boot process.)
- 1** Single user mode
- 2** Multiuser, without NFS (the same as 3, if you do not have networking)
- 3** Full multiuser mode
- 4** Unused
- 5** X11
- 6** Reboot (Do not set `initdefault` to this or the server machine will continuously reboot at startup.)

To set the initial run level of a machine at boot, modify the /etc/inittab file as shown in Figure 15-6 with the following line:

id:3:initdefault:

```
... (lines not displayed)                                     To start Linux without starting the  
# The default runlevel is defined here                    GUI, set the run level to 3.  
id:3:initdefault: ←  
  
# First script to be executed, if not booting in emergency (-b) mode  
si::bootwait:/etc/init.d/boot  
  
# /etc/init.d/rc takes care of runlevel handling  
#  
# runlevel 0 is System halt (Do not use this for initdefault!)  
# runlevel 1 is Single user mode  
# runlevel 2 is Local multiuser without remote network (e.g. NFS)  
# runlevel 3 is Full multiuser with network  
# runlevel 4 is Not used  
# runlevel 5 is Full multiuser with network and xdm  
# runlevel 6 is System reboot (Do not use this for initdefault!)  
#  
  
... (lines not displayed)  
  
# getty-programs for the normal runlevels  
# <id>:<runlevels>:<action>:<process>  
# The "id" field MUST be the same as the last  
# characters of the device (after "tty").  
1:2345:respawn:/sbin/mingetty --noclear tty1  
2:2345:respawn:/sbin/mingetty tty2  
3:2345:respawn:/sbin/mingetty tty3  
#4:2345:respawn:/sbin/mingetty tty4 ←  
#5:2345:respawn:/sbin/mingetty tty5  
#6:2345:respawn:/sbin/mingetty tty6  
#  
#S0:12345:respawn:/sbin/agetty -L 9600 ttyS0 vt102  
  
... (lines not displayed)
```

To only provide three consoles and thereby save memory, comment out the **mingetty** entries for 4, 5, and 6.

Figure 15-6 /etc/inittab, modified (only part of the file is displayed)

With SUSE Linux Enterprise Server, shutting down the GUI can also be accomplished by running the YaST **runlevel** command and changing the default run level (see Figure 15-5 on page 461).

By default, six consoles are saved. F1 through F6 are separate consoles. To regain some memory, you might want to limit the number of consoles to three from the original six. To do this, comment out each `mingetty ttyn` line that you want to disable. In Figure 15-6 on page 463, for example, we have limited the consoles to three.

**Tip:** Even if you have the GUI disabled locally on the server, you can still connect remotely and use the GUI, using the `-X` parameter on the `ssh` command.

## 15.4 Security Enhanced Linux

Red Hat Enterprise Linux 4 introduced a new security model, Security Enhanced Linux (SELinux), which is a significant step toward higher security. SELinux introduces a mandatory policy model that overcomes the limitations of the standard discretionary access model employed by Linux.

SELinux was developed by the National Security Agency of the US, and it has been included by Red Hat since RHEL release 4. The original code went to open source in 2000 and merged into the mainline kernel in release 2.6.0-test3. Other Linux distributions support this initiative but Red Hat is the only major commercial one that implements it by default.

SELinux enforces security on user and process levels. Thus, a security flaw of any given process affects only the resources that are allocated to this process and not the entire system. SELinux works similar to a virtual machine (see Figure 15-7 on page 465).

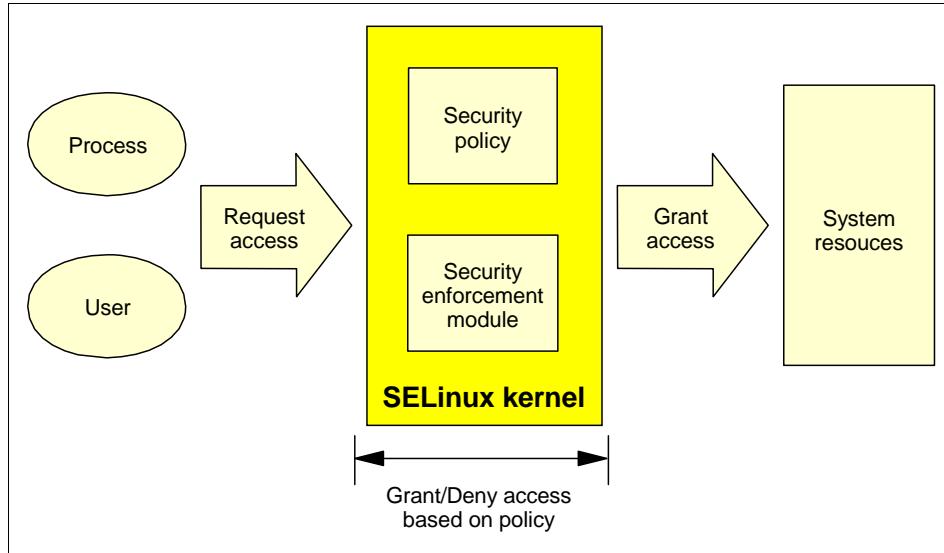


Figure 15-7 Schematic overview of SELinux

For example, if a malicious attacker uses a root exploit within Apache, only the resources that are allocated to the Apache daemon could be compromised. Novell introduced a similar feature in version 10 of its SUSE Linux Enterprise Server. Enforcing such a policy-based security model, however, comes at a price. Every access from a user or process to a system resource, such as an I/O device, must be controlled by SELinux. Thus, the process of checking permissions can cause an overhead of up to 10%. SELinux is of great value to any edge server such as a firewall or a Web server, but the added level of security on a back-end database server might not justify the loss in performance.

Often systems have been installed using default parameters and are unaware that SELinux affects performance. Generally, the easiest way to disable SELinux is to not install it in the first place. To disable SELinux after an installation, append the `selinux=0` entry to the line that includes the running kernel in the GRUB boot loader, as shown in Example 15-1.

*Example 15-1 Sample grub.conf file with disabled SELinux*

---

```
default=0
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux AS (2.6.9-5.ELsmp)
    root (hd0,0)
    kernel /vmlinuz-2.6.9-5.ELsmp ro root=LABEL=/ selinux=0
    initrd /initrd-2.6.9-5.ELsmp.img
```

---

If you decide to use SELinux with your Linux-based server, you can tune its settings to better accommodate your environment. On a running system, check whether the working set of the cached Linux Security Modules (LSM) permissions exceeds the default Access Vector Cache (AVC) size of 512 entries.

Check `/selinux/avc/hash_stats` for the length of the longest chain. Anything over 10 signals a likely bottleneck.

**Tip:** To check for usage statistics of the access vector cache, you can alternatively use the `avcstat` utility.

If the system experiences a bottleneck in the Access Vector Cache (for example, on a heavily loaded firewall), try to resize `/selinux/avc/cache_threshold` to a slightly higher value and recheck the hash statistics.

## 15.5 Changing kernel parameters

The Linux kernel is the core of the operating system, and it is common to all Linux distributions. You can make changes to the kernel by modifying parameters that control the operating system. You make these changes on the command line by using the `sysctl` command.

**Tip:** By default, the kernel includes the necessary module to enable you to make changes using `sysctl` without needing to reboot. However, If you choose to remove this support (during the operating system installation), then you have to reboot Linux before the change takes effect.

SUSE Linux offers a graphical method of modifying these `sysctl` parameters. To launch the `powertweak` tool, issue the following command:

```
/sbin/yast2 powertweak
```

For a text-based menu version, use the command:

```
/sbin/yast powertweak
```

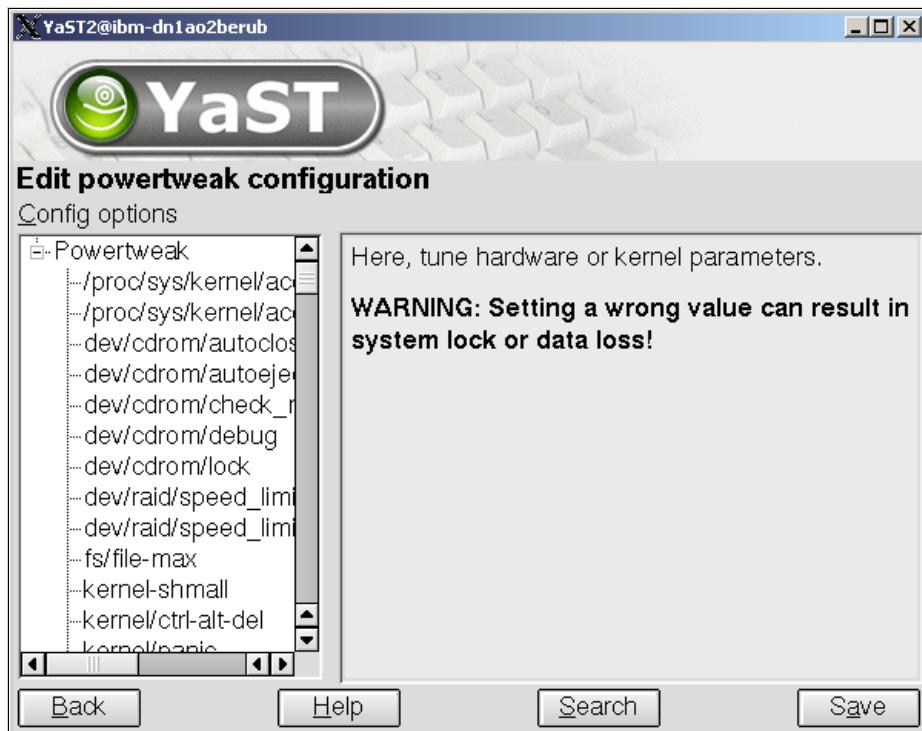


Figure 15-8 SUSE Linux powertweak

### 15.5.1 Parameter storage locations

The kernel parameters that control how the kernel behaves are stored in /proc (and in particular, /proc/sys). Reading the files in the /proc directory tree provides a simple way to view configuration parameters that are related to the kernel, processes, memory, network, and other components. Each process running in the system has a directory in /proc with the process ID (PID) as name.

Table 15-2 lists some of the files that include kernel information.

Table 15-2 Parameter files in /proc

Files/directory	Purpose
/proc/loadavg	Information about the load of the server in 1-minute, 5-minute, and 15-minute intervals. The <code>uptime</code> command gets information from this file.
/proc/kcore	(SUSE Linux Enterprise Server only) Includes data to generate a core dump at run time, for kernel debugging purposes. The command to create the core dump is <code>gdb</code> as in the <code>#gdb /usr/src/linux/vmlinux /proc/kcore</code> directory.

Files/directory	Purpose
/proc/stat	Kernel statistics as process, swap, and disk I/O.
/proc/cpuinfo	Information about the installed CPUs.
/proc/meminfo	Information about memory usage. The <b>free</b> command uses this information.
/proc/sys/abi/* (all files in this directory)	Used to provide support for binaries that are not native to Linux and that are compiled under other UNIX variants, such as SCO UnixWare 7, SCO OpenServer, and Sun Solaris™ 2. By default, this support is installed, although it can be removed during installation.
/proc/sys/fs/*	Used to increase the number of open files that the operating system allows and to handle quota.
/proc/sys/kernel/*	For tuning purposes, you can enable hotplug, manipulate shared memory, and specify the maximum number of PID files and level of debug in syslog.
/proc/sys/net/*	Tuning of network in general, IPV4 and IPV6.
/proc/sys/vm/*	Management of cache memory and buffer.

### 15.5.2 Using the **sysctl** commands

The **sysctl** commands use the names of files in the /proc/sys directory tree as parameters. For example, the manual way to modify the **shmmmax** kernel parameter is to display (using **cat**) and change (using **echo**) the file **/proc/sys/kernel/shmmmax** as in the following example:

```
#cat /proc/sys/kernel/shmmmax
33554432
#echo 33554430 > /proc/sys/kernel/shmmmax
#cat /proc/sys/kernel/shmmmax
33554430
```

In the previous example, we pushed the 33554430 value to **/proc/sys/kernel/shmmmax** by using the standard command line utilities.

However, using these commands can introduce errors easily. So, we recommend that you use the **sysctl** command because it checks the consistency of the data before it makes any change. For example:

```
#sysctl kernel.shmmmax
kernel.shmmmax = 33554432
#sysctl -w kernel.shmmmax=33554430
kernel.shmmmax = 33554430
#sysctl kernel.shmmmax
kernel.shmmmax = 33554430
```

This change to the kernel stays in effect only until the next reboot. To make the change permanent, edit the /etc/sysctl.conf or /etc/sysconfig/sysctl file and add the appropriate command. In our example:

```
kernel.shmmax = 33554439
```

The next time you reboot, the parameter file is read. You can do the same thing without rebooting by issuing the following command:

```
#sysctl -p
```

## 15.6 Kernel parameters

The Linux kernel has many parameters that can improve performance for your installation. Table 15-3 lists the SLES kernel parameters that are most relevant to performance.

*Table 15-3 SLES parameters that are most relevant to performance tuning*

Parameter	Description/example of use
kernel.shm-bigpages-per-file	Normally used for tuning database servers. The default is 32768. To calculate a suitable value, take the amount of SGA memory in GB and multiply by 1024. For example: sysctl -w kernel.shm-bigpages-per-file=16384
kernel.sched_yield_scale	Enables the dynamic resizing of time slices given to processes. When enabled, the kernel reserves more time slices for busy processes and fewer for idle processes. The parameters kernel.min-timeslice and kernel.max-timeslice are used to specify the range of time slices that the kernel can supply as needed. If disabled, the time slices given to each process are the same. The default is 0 (disabled). Applications such as ERP and Java can benefit from this being enabled. For real-time applications such as streaming audio and video, leave it disabled. For example: sysctl -w kernel.sched_yield_scale=1
kernel.shm-use-bigpages	Enables the use of bigpages (typically for databases). Default is 0 (disabled). For example: sysctl -w kernel.shm-use-bigpages=1
net.ipv4.conf.all.hidden	All interface addresses are hidden from ARP broadcasts and will be included in the ARP response of other addresses. Default is 0 (disabled). For example: sysctl -w net.ipv4.conf.all.hidden=1
net.ipv4.conf.default.hidden	Enables all interfaces as hidden by default. Default is 0 (disabled). sysctl -w net.ipv4.conf.default.hidden=1

Parameter	Description/example of use
net.ipv4.conf.eth0.hidden	Enables only interface eth0 as hidden. Uses the ID of your network card. Default is 0 (disabled). sysctl -w net.ipv4.conf.eth0.hidden=1
net.ipv4.ip_conntrack_max	This setting is the number of separate connections that can be tracked. Default is 65536. sysctl -w net.ipv4.ip_conntrack_max=32768
net.ipv6.conf.all.mtu	Default maximum for transfer unit on IPV6. Default is 1280. sysctl -w net.ipv6.conf.all.mtu=9000
net.ipv6.conf.all.router_solicitation_delay	Determines whether to wait after interface opens before sending router solicitations. Default is 1 (the kernel should wait). For example: sysctl -w net.ipv6.conf.all.router_solicitation_delay=0
net.ipv6.conf.all.router_solicitation_interval	Number of seconds to wait between router solicitations. Default is 4 seconds. For example: sysctl -w net.ipv6.conf.all.router_solicitation_interval=3
net.ipv6.conf.all.router_solicitations	Number of router solicitations to send until assuming no routers are present. Default is 3. sysctl -w net.ipv6.conf.all.router_solicitations=2
net.ipv6.conf.all.temp_prefered_lft	Lifetime preferred in seconds for temporary addresses. Default is 86400 (1 day). sysctl -w net.ipv6.conf.all.temp_prefered_lft=259200
net.ipv6.conf.all.temp_valid_lft	Lifetime valid in seconds for temporary address. Default is 604800 (1 week). sysctl -w net.ipv6.conf.all.temp_valid_lft=302400
net.ipv6.conf.default.accept_redirects	Accepts redirects sent by a router that works with IPV6, but it cannot set if forwarding is set to enable. Always one or other, it can never set together because it will cause problems in all IPV6 networks. Default is 1 (enabled). sysctl -w net.ipv6.conf.default.accept_redirects=0
net.ipv6.conf.default.autoconf	This automatically generates an address such as "ff81::221:21ff:ae44:2781" on an interface with an L2-MAC Address. Default is 1 (enabled). sysctl -w net.ipv6.conf.default.autoconf=0
net.ipv6.conf.default.dad_transmits	Determines whether Duplicate Address Detection (DAD) probes are sent. Default is 1 (enabled). sysctl -w net.ipv6.conf.default.dad_transmits=0
net.ipv6.conf.default.mtu	Sets the default value for Maximum Transmission Unit (MTU). Default is 1280. sysctl -w net.ipv6.conf.default.mtu=9000

Parameter	Description/example of use
net.ipv6.conf.default.regen_max_retry	Number of attempts to try to generate a valid temporary address. Default is 5. sysctl -w net.ipv6.conf.default.regen_max_retry=3
net.ipv6.conf.default.router_solicitation_delay	Number in seconds to wait, after interface is brought up, before sending router request. Default is 1 (enabled). sysctl -w net.ipv6.conf.default.router_solicitation_delay=0
vm.heap-stack-gap	Enables the heap of memory that is used to store information about status of process and local variables. You should disable this when you need to run a server with JDK™; otherwise, your software will crash. Default is 1 (enabled). sysctl -w vm.heap-stack-gap=0
vm.vm_anon_lru	Allows the VM to always have visibility of anonymous pages. Default is 1 (enabled). sysctl -w vm.vm_anon_lru=0
vm.vm_lru_balance_ratio	Balances active and inactive sections of memory. Define the amount of inactive memory that the kernel will rotate. Default is 2. sysctl -w vm.vm_lru_balance_ratio=3
vm.vm_mapped_ratio	Controls the pageout rate. Default is 100. sysctl -w vm.vm_mapped_ratio=90
vm.vm_passes	Number of attempts that the kernel will try to balance the active and inactive sections of memory. Default is 60. sysctl -w vm.vm_passes=30
vm.vm_shmem_swap	Improves performance of applications that use large amounts of non-locked shared memory (such as ERP and database applications) on a server with more than 8 GB of RAM. Default is 0 (disabled). sysctl -w vm.vm_shmem_swap=1
vm.vm_vfs_scan_ratio	Proportion of Virtual File System unused caches that will try to be in one VM freeing pass. Default is 6. sysctl -w vm.vm_vfs_scan_ratio=6

Table 15-4 lists the RHEL kernel parameters that are most relevant to performance.

*Table 15-4 Red Hat parameters that are most relevant to performance tuning*

Parameter	Description/example of use
net.ipv4.inet_peer_gc_maxtime	How often the garbage collector (gc) should pass over the inet peer storage memory pool during low or absent memory pressure. Default is 120, measured in jiffies. sysctl -w net.ipv4.inet_peer_gc_maxtime=240
net.ipv4.inet_peer_gc_mintime	Sets the minimum time that the garbage collector can pass cleaning memory. If your server is heavily loaded, you might want to increase this value. Default is 10, measured in jiffies. sysctl -w net.ipv4.inet_peer_gc_mintime=80
net.ipv4.inet_peer_maxttl	The maximum time-to-live for the inet peer entries. New entries will expire after this period of time. Default is 600, measured in jiffies. sysctl -w net.ipv4.inet_peer_maxttl=500
net.ipv4.inet_peer_minttl	The minimum time-to-live for inet peer entries. Set to a high enough value to cover fragment time to live in the reassembling side of fragmented packets. This minimum time must be smaller than net.ipv4.inet_peer_threshold. Default is 120, measured in jiffies. sysctl -w net.ipv4.inet_peer_minttl=80
net.ipv4.inet_peer_threshold	Set the size of inet peer storage. When this limit is reached, peer entries will be thrown away, using the inet_peer_gc_mintime timeout. Default is 65644. sysctl -w net.ipv4.inet_peer_threshold=65644
vm.hugetlb_pool	The hugetlb feature works in the same way as bigpages, but after hugetlb allocates memory, only the physical memory can be accessed by hugetlb or shm allocated with SHM_HUGETLB. It is normally used with databases such as Oracle or DB2. Default is 0. sysctl -w vm.hugetlb_pool=4608
vm.inactive_clean_percent	Designates the percent of inactive memory that should be cleaned. Default is 5%. sysctl -w vm.inactive_clean_percent=30
vm.pagecache	Designates how much memory should be used for page cache. This is important for databases such as Oracle and DB2. Default is 1 15 100. This parameter's three values are: <ul style="list-style-type: none"><li>▶ Minimum percent of memory used for page cache. Default is 1%.</li><li>▶ The initial amount of memory for cache. Default is 15%.</li><li>▶ Maximum percent of memory used for page cache. Default is 100%.</li></ul> sysctl -w vm.pagecache=1 50 100

## 15.7 Tuning the processor subsystem

The CPU is one of the most important hardware subsystems for servers with a primary role of application or database server. However, in these systems, the CPU is often the source of performance bottlenecks.

For information about the tweaking of processor tuning parameters, refer to 15.6, “Kernel parameters” on page 469.

Consider the performance impact of enabling or disabling Hyper-Threading, if Hyper-Threading is available on the processors in your server. Processors based on the Intel Nehalem architecture use a similar technology known as Simultaneous Multi-Threading (SMT). Hyper-Threading or SMT are ways of executing two separate code stream (threads) concurrently. The operating system recognizes these and can allocate work to them appropriately. Hyper-Threading and SMT are supported by both Red Hat Enterprise Linux AS and SUSE Linux Enterprise Server.

By virtualizing the processor, you can execute two threads or processes at a time (this is also known as *thread-level parallelism*). By having your operating system and software designed to take advantage of this technology, you can gain significant increases in performance without needing an increase in clock speed.

Modern CPUs have multiple cores. The multi-core technology is the evolution of the multithreading technology, and it enables a socket to have more than one CPU or core. This is actually like having more than one CPU per physical CPU (or socket). In this case, the Linux kernel will recognize each core as an individual CPU on an SMP/NUMA configuration.

For example, if you have Hyper-Threading enabled on an old 4-way server, or 4-socket dual-core processors, monitoring tools such as **top** will display eight processors (Example 15-2).

*Example 15-2 Output of top on a 4-way server with Hyper-Threading enabled*

---

```
10:22:45 up 23:40, 5 users, load average: 26.49, 12.03, 10.24
373 processes: 370 sleeping, 2 running, 1 zombie, 0 stopped
CPU states: cpu user nice system irq softirq iowait idle
      total 36.1% 0.1% 9.7% 0.3% 4.1% 1.6% 47.7%
      cpu00 17.0% 0.0% 5.9% 3.1% 20.8% 2.1% 50.7%
      cpu01 54.9% 0.0% 10.9% 0.0% 0.9% 1.3% 31.7%
      cpu02 33.4% 0.1% 8.5% 0.0% 2.5% 0.9% 54.2%
      cpu03 33.8% 0.7% 10.0% 0.0% 0.9% 2.1% 52.0%
      cpu04 31.4% 0.0% 9.3% 0.0% 2.9% 2.5% 53.6%
      cpu05 33.4% 0.0% 9.9% 0.0% 2.1% 0.7% 53.6%
      cpu06 30.5% 0.0% 11.1% 0.0% 1.7% 1.3% 55.1%
```

```
cpu07 54.5% 0.0% 12.1% 0.0% 0.5% 1.9% 30.7%
Mem: 8244772k av, 3197880k used, 5046892k free, 0k shrd, 91940k buff
      2458344k active, 34604k inactive
Swap: 2040244k av, 0k used, 2040244k free 1868016k cached
```

---

With respect to Hyper-Threading, note that:

- ▶ SMP-based kernels are required to support Hyper-Threading.
- ▶ The more CPUs that are installed in a server, the fewer benefits Hyper-Threading has on performance. On servers that are CPU-bound you can expect, at most, the following performance gains:
  - Two physical processors: 15% to 25% performance gain
  - Four physical processors: 15 to 13% gain
  - Eight physical processors: 0 to 5% gain

**Note:** These performance gains are true for specific workloads, software, and operating system combinations only.

For more information about Hyper-Threading, see:

<http://www.intel.com/business/bss/products/hyperthreading/server/>

### 15.7.1 Selecting the right kernel

Both Red Hat Enterprise Linux AS and SUSE Linux Enterprise Server include several kernel packages precompiled for the majority of the uses of Linux, as described in Table 15-5. It is important to select the most appropriate kernel for your system and how you will use it, because it can influence the final performance.

*Table 15-5 Available kernels within the distribution*

Kernel type	Description
SMP	Kernel with support for SMP and Hyper-Threaded machines.
Hugemem	(Red Hat Enterprise Linux AS only) Support for machines with greater than 12 GB of memory. Includes support for NUMA.
Standard	Single processor machines.

**Note:** These are the precompiled kernels that are supported by the main distributors and hardware vendors. However, you can recompile your own optimized version of the Linux kernel to optimize the use of your hardware. When doing so, however, verify that your compiled kernel will be supported by your hardware or Linux distribution vendor, because this is not always the case. IBM publishes a list of default supported Linux versions and hardware under the ServerProven® program; see:

<http://www-03.ibm.com/servers/eserver/serverproven/compat/us/>

## 15.7.2 Interrupt handling

One of the highest-priority tasks that a CPU has to handle is interrupts. Interrupts can be caused by subsystems, such as a network interface card. Hard interrupts cause a CPU to stop its current work and perform a *context switch*, which is undesirable because the processor has to flush its cache to make room for the new work. (Think of a processor's cache as a work bench that has to be cleaned up and supplied with new tools every time new work has to be done.)

Two principles have proven to be most efficient when it comes to interrupt handling:

- ▶ Bind processes that cause a significant amount of interrupts to a specific CPU.

CPU affinity enables the system administrator to bind interrupts to a group or a single physical processor (of course, this does not apply on a single-CPU system). To change the affinity of any given IRQ, go into `/proc/irq/%{number of respective irq}/` and change the CPU mask stored in the file `smp_affinity`. For example, to set the affinity of IRQ 19 to the third CPU in a system (without Hyper-Threading) use the following command:

```
echo 03 > /proc/irq/19/smp_affinity
```

- ▶ Let physical processors handle interrupts.

**Note:** For multi-core CPUs, each core is seen by the operating system as one processor. Each core has its own cache L1, meaning that it does not share its cache (at least in this layer) with another processor. Then, the physical processor already handles interrupts.

## Considerations for NUMA systems

Non-uniform memory access (NUMA) systems are gaining market share and are seen as the natural evolution of classic symmetric multiprocessor systems.

Although the CPU scheduler used by current Linux distributions is well suited for

NUMA systems, applications might not always be. Bottlenecks caused by a non-NUMA-aware application can cause performance degradations that are hard to identify. The recent **numastat** utility shipped in the **numactl** package helps to identify processes that have difficulties dealing with NUMA architectures.

To help with spotting bottlenecks, you can use the statistics that are provided by the **numastat** tool in the `/sys/devices/system/node/%{node number}/numastat` file. High values in `numa_miss` and the `other_node` field signal a likely NUMA issue. If you find that a process is allocated memory that does not reside on the local node for the process (the node that holds the processors that run the application), try to **renice** the process to the other node or work with NUMA affinity.

## 15.8 Tuning the memory subsystem

Tuning the memory subsystem is a difficult task that requires constant monitoring to ensure that changes do not negatively affect other subsystems in the server. If you do choose to modify the virtual memory parameters (in `/proc/sys/vm`), we recommend that you change only one parameter at a time and monitor how the server performs.

Remember that most applications under Linux do not write directly to the disk, but to the file system cache that is maintained by the virtual memory manager that will eventually flush out the data. When using an IBM ServeRAID controller or an IBM TotalStorage disk subsystem, you should try to decrease the number of flushes, effectively increasing the I/O stream that is caused by each flush. The high-performance disk controller can handle the larger I/O stream more efficiently than multiple small ones.

### 15.8.1 Configuring bdflush (kernel 2.4 only)

There is tuning in the virtual memory subsystem that can help improve overall file system performance. The **bdflush** kernel daemon is responsible for making sure that dirty buffers, any modified data that currently resides only in the volatile system memory, are committed to disk. Changes in the `/proc` system take effect immediately but will be reset at boot time. To make changes permanent, include the **echo** or **sysctl** command in the `/etc/rc.d/rc.local` file.

Configuring how the Linux kernel flushes dirty buffers to disk can tailor the flushing algorithm toward the specifications of the respective disk subsystem. Disk buffers are used to cache data that is stored on disks, which are very slow compared with RAM. So, if the server uses this kind of memory, it can create serious problems with performance.

By modifying the /proc/sys/vm/bdflush parameters, you can modify the writing-to-disk rate, possibly avoiding disk contention problems. To edit the parameters of the bdflush subsystem, you can use either the **echo** command as shown in Example 15-3 or **sysctl** as shown in Example 15-4, although we recommend that you use **sysctl**.

*Example 15-3 Modifying the bdflush parameters in the kernel using echo*

---

```
echo 30 500 0 0 500 30000 60 20 0 > /proc/sys/vm/bdflush
```

---

*Example 15-4 Using sysctl to change parameters of bdflush*

---

```
sysctl -w vm.bdflush="30 500 0 0 500 3000 60 20 0"
```

---

The parameters in the /proc/sys/vm/bdflush of 2.4 Linux kernels are:

nfract	Maximum percentage of dirty buffers in the buffer cache. The higher the value, the longer the write to the disk will be postponed. When available memory is in short supply, large amounts of I/O have to be processed. To spread I/O out evenly, keep this a low value.
ndirty	Maximum number of dirty buffers that the bdflush process can write to disk at one time. A large value results in I/O occurring in bursts, and a small value might lead to memory shortages if the bdflush daemon is not executed enough.
dummy2	Unused (formerly nrefill).
dummy3	Unused.
interval	Minimum rate at which kupdate will wake and flush. Default is 5 seconds, with a minimum value of zero (0) seconds and a maximum of 600 seconds. kupdate is a daemon that manages in-use memory.
age_buffer	Maximum time the operating system waits before writing buffer cache to disk. Default is 30 seconds, with a minimum of one second and a maximum of 6000 seconds.
nfract_sync	Percent of dirty buffers to activate bdflush synchronously. Default is 60%.
nfract_stop	Percent of dirty buffers to stop bdflush. Default is 20%.
dummy5	Unused.

## 15.8.2 Configuring kswapd (kernel 2.4 only)

Another pertinent VM subsystem is the kernel swap daemon (kswapd). This daemon takes care of writing the swap file and moving the data to the swap file. You can configure this daemon to specify how many pages of memory are paged out by Linux:

```
sysctl -w vm.kswapd="1024 32 64"
```

The three parameters are as follows:

- ▶ tries\_base is four times the number of pages that the kernel swaps in one pass. On a system with a significant amount of swapping, increasing the number might improve performance.
- ▶ tries\_min is the minimum number of pages that kswapd swaps out each time the daemon is called.
- ▶ swap\_cluster is the number of pages that kswapd writes at the same time. A smaller number increases the number of disk I/Os performed, but a larger number might also have a negative impact on the request queue.

If you do make changes, check their impact using tools such as **vmstat**.

Other relevant VM parameters that might improve performance include:

- ▶ buffermem
- ▶ freepages
- ▶ overcommit\_memory
- ▶ page-cluster
- ▶ pagecache
- ▶ pagetable\_cache

## 15.8.3 Setting kernel swap behavior (kernel 2.6 only)

With the introduction of the improved virtual memory subsystem in the Linux kernel 2.6, administrators now have a simple interface to fine-tune the swapping behavior of the kernel. You can use the parameters in /proc/sys/vm/swappiness to define how aggressively memory pages are swapped to disk.

Linux moves memory pages that have not been accessed for some time to the swap space even if there is enough free memory available. By changing the percentage in /proc/sys/vm/swappiness, you can control that behavior, depending on the system configuration. If swapping is not desired, /proc/sys/vm/swappiness should have low values. Systems with memory constraints that run batch jobs (processes that sleep for a long time) might benefit from an aggressive swapping behavior.

To change swapping behavior, use either **echo** or **sysctl** as shown in Example 15-5.

*Example 15-5 Changing swappiness behavior*

---

```
# sysctl -w vm.swappiness=100
```

---

#### 15.8.4 HugeTLBfs

The HugeTLBfs memory management feature is valuable for applications that use a large virtual address space. It is especially useful for database applications.

The CPU's Translation Lookaside Buffer (TLB) is a small cache used for storing virtual-to- physical mapping information. By using the TLB, a translation can be performed without referencing the in-memory page table entry that maps the virtual address. However, to keep translations as fast as possible, the TLB is typically quite small. It is not uncommon for large memory applications to exceed the mapping capacity of the TLB.

The HugeTLBfs feature permits an application to use a much larger page size than normal, so that a single TLB entry can map a correspondingly larger address space. A HugeTLB entry can vary in size. For example, in an Itanium 2 system, a huge page might be 1000 times larger than a normal page. This enables the TLB to map 1000 times the virtual address space of a normal process without incurring a TLB cache miss. For simplicity, this feature is exposed to applications by means of a file system interface.

**Important:** Although there are useful tools to tune the memory subsystem, swapping should be avoided as much as possible. The fact that the server swaps is almost never a good behavior. Before trying to improve the swap process, ensure that your server simply has enough memory or that there are no memory leaks.

## 15.9 Tuning the file system

Ultimately, all data must be retrieved from and stored to disk. Disk accesses are usually measured in milliseconds and are thousands of times slower than other components (such as memory or PCI operations, which are measured in nanoseconds or microseconds). The Linux file system is the method by which data is stored and managed on the disks.

Many different file systems are available for Linux that differ in performance and scalability. In addition to storing and managing data on the disks, file systems are also responsible for guaranteeing data integrity. The newer Linux distributions include *journaling* file systems as part of their default installation. Journaling or logging prevents data inconsistency in case of a system crash. All modifications to the file system metadata have been maintained in a separate journal or log and can be applied after a system crash to bring it back to its consistent state. Journaling also improves recovery time, because there is no need to perform file system checks at system reboot.

As with other aspects of computing, you will find that there is a trade-off between performance and integrity. However, as Linux servers make their way into corporate data centers and enterprise environments, requirements such as high availability can be addressed.

In this section, we discuss the default file systems that are available on Red Hat Enterprise Linux AS and SUSE Linux Enterprise Server and some simple ways to improve their performance.

### 15.9.1 Hardware considerations before installing Linux

Minimum requirements for CPU speed and memory are well documented for current Linux distributions. Those instructions also provide guidance for the minimum disk space that is required to complete the installation. However, they fall short when detailing how to set up the disk subsystem initially. Because Linux servers cover a vast assortment of work environments as server consolidation makes its impact in data centers, one of the first questions to answer is: What is the function of the server that is being installed?

A server's disk subsystems can be a major component of overall system performance. Understanding the function of the server is key to determining whether the I/O subsystem will have a direct impact on performance.

The following examples show where disk I/O is most important:

- ▶ A file and print server must move data quickly between users and disk subsystems. Because the purpose of a file server is to deliver files to the client, the server must initially read all data from a disk.
- ▶ A database server's ultimate goal is to search and retrieve data from a repository on the disk. Even with sufficient memory, most database servers will perform large amounts of disk I/O to bring data records into memory and flush modified data to disk.

The following examples show where disk I/O is not the most important subsystem:

- ▶ An e-mail server acts as a repository and router for electronic mail and tends to generate a heavy communication load. Networking is more important for this type of server.
- ▶ A Web server that is responsible for hosting Web pages (static, dynamic, or both) benefits from a well-tuned network and memory subsystem.

## Disk technology selection

In addition to understanding the function of the server, you must also understand the size of the deployment that the installation will have to serve. Current disk subsystem technologies were designed with size of deployment in mind.

See Table 11-1 on page 238 for a brief description of the disk technologies.

## Number of drives

The number of disk drives affects performance significantly because each drive contributes to total system throughput. Analysis of the effect on performance is discussed in 11.6.2, “Number of drives” on page 269.

Capacity requirements are often the only consideration that is used to determine the number of disk drives that are configured in a server. Throughput requirements are usually not well understood or are completely ignored. Good performance by the disk subsystem depends on maximizing the number of read-write heads that can service I/O requests.

With RAID (redundant array of independent disks) technology, you can spread the I/O over multiple spindles. There are two options for implementing RAID in a Linux environment: software RAID or hardware RAID. Many System x servers ship with hardware RAID support, but if not, you might want to start with the software RAID options that come with the Linux distributions.

Software RAID in the 2.4 Linux kernel distributions is implemented through the md device driver. This driver implementation is device-independent and is

therefore flexible in allowing many types of disk storage such as EIDE or SCSI to be configured as a RAID array. Supported software RAID levels are RAID-0 (striping), RAID-1 (mirroring), and RAID-5 (striping with parity). These can be implemented as part of the initial installation or through the `mdadm` tool set.

It is important to note that the choice of RAID level has a noticeable effect on performance. For more information about this topic, refer to 11.6.1, “RAID strategy” on page 268.

If it is necessary to implement a hardware RAID array, you need a RAID controller for your system. In this case, the disk subsystem consists of the physical hard disks and the controller.

**Tip:** In general, adding drives is one of the most effective changes that can be made to improve server performance.

For additional, in-depth coverage of the available IBM storage solutions, see Chapter 11, “Disk subsystem” on page 237 as well as the IBM Redbooks publication *IBM System Storage Solutions Handbook*, SG24-5250, which is available from:

<http://www.redbooks.ibm.com/abstracts/sg245250.html>

### 15.9.2 Ext3: the default Red Hat file system

Since the release of the Red Hat 7.2 distribution, the default file system at the time of installation has been Ext3. This file system is an updated version of the widely used Ext2 file system with the addition of journaling. Highlights of this file system include:

- ▶ Availability: Ext3 always writes data to the disks in a consistent way, so in the case of an unclean shutdown (unexpected power failure or system crash), the server does not have to spend time checking the consistency of the data, thereby reducing system recovery from hours to seconds.
- ▶ Data integrity: By specifying the journaling mode `data=journal` on the `mount` command, all data, both file data and metadata, is journaled.
- ▶ Speed: By specifying the journaling mode `data=writeback`, you can decide on speed versus integrity to meet the needs of your business requirements. Setting the parameter `data` to `writeback` will result in a performance increase that is notable in environments where there are heavy synchronous writes.
- ▶ Flexibility: Upgrading from existing Ext2 file systems is simple and no reformatting is necessary. By executing the `tune2fs` command and modifying the `/etc/fstab` file, you can easily update an Ext2 to an Ext3 file system. Also note that Ext3 file systems can be mounted as ext2 with journaling disabled.

Products from many third-party vendors have the capability of manipulating Ext3 file systems. For example, PartitionMagic can handle the modification of Ext3 partitions.

### 15.9.3 ReiserFS: the default SUSE Linux file system

The default file system on a SUSE Linux installation since SUSE Linux 7.1 has been ReiserFS, developed by Hans Reiser. From its initial design, key performance aspects have included:

- ▶ Journaling designed into the file system from the beginning to improve reliability and recovery.
- ▶ Faster access through the use of balanced tree data structures that allow for storing both content data and security metadata.
- ▶ Efficient use of disk space because, unlike other file systems, this file system does not rely on block sizes.

**Note:** ReiserFS is not supported by Red Hat Enterprise Linux AS.

### 15.9.4 File system tuning in the Linux kernel

Settings for the default file systems as it is shipped might be adequate for most environments. However, this section discusses a few pointers to help improve overall disk performance.

#### Accessing time updates

The Linux file system keeps records of when files are created, updated, and accessed. Default operations include updating the last-time-read attribute for files during reads and writes to files. Because writing is an expensive operation, eliminating unnecessary I/O can lead to overall improved performance.

Mounting file systems with the `noatime` option prevents the inode access times from being updated. If file update times are not critical to your implementation, as in a Web-serving environment, a user can choose to mount file systems with the `noatime` flag in the `/etc/fstab` file as follows:

```
/dev/sdb1 /mountlocation ext3 defaults,noatime 1 2
```

It is generally a good idea to have a separate `/var` partition and mount it with the `noatime` option.

## Tuning the elevator algorithm (kernel 2.4 only)

The disk I/O elevator algorithm was introduced as a feature in the V2.4 kernel. It enables the user to tune the algorithm that schedules block I/O by controlling the amount of time an I/O request remains on the queue before being serviced. This is accomplished by adjusting the read and write values of the elevator algorithm. By increasing latency times (that is, larger values for read, write, or both), I/O requests stay on the queue for a longer period of time, thereby giving the I/O scheduler the opportunity to coalesce these requests to perform more efficient I/O and increase throughput.

If your Linux server is in an environment with large amounts of disk I/O, finding the right balance between throughput and latency might be beneficial. Linux file systems are implemented as block devices, so improving how often those blocks are read and written can improve file system performance. As a guideline, heavy I/O servers benefit from smaller caches, prompt flushes, and a balanced high-latency read to write.

As with other system tuning, tuning the elevator algorithm is an iterative process. You want to baseline current performance, make changes, and then be able to measure the effect of those changes. Example 15-6 shows how to use the `/sbin/elvtune` command to first show the current settings and then change the values for the read and write queues.

**Tip:** Red Hat's recommendation is to tune the elevator algorithm so that the read latency (-r) is half the write latency (-w).

If any change is made, be sure that the `/sbin/elvtune` call is added to the `/etc/rc.d/rc.local` file (Red Hat) or `/etc/init.d/boot.local` file (SUSE Linux) to make it a persistent change between system boots.

*Example 15-6 Finding current defaults for your installation and changing them*

---

```
[root@x232 root]# elvtune /dev/sda
```

```
/dev/sda elevator ID          2
      read_latency:        2048
      write_latency:       8192
      max_bomb_segments:   6
```

```
[root@x232 root]# elvtune -r 1024 -w 2048 /dev/sda
```

```
/dev/sda elevator ID          2
      read_latency:        1024
      write_latency:       2048
      max_bomb_segments:   6
```

---

If you are using a 2.6 kernel, use the I/O scheduler instead of `elvtune`. Although `elvtune` is still available on kernel 2.6 systems, when launching the command under Red Hat, you get the following message:

`elvtune` is only useful on older kernels; for 2.6 use `IO scheduler sysfs tunables` instead.

## The I/O scheduler (kernel 2.6 only)

The Linux kernel, which is the core of the operating system, is responsible for controlling disk access by using kernel I/O scheduling. 2.4 kernel uses a single, robust, general purpose I/O elevator. The I/O schedulers that are provided in Red Hat Enterprise Linux 4 and SUSE Linux Enterprise 9, based on the 2.6 kernel, have advanced the I/O capabilities of Linux significantly.

The kernel I/O is selectable at boot time by choosing one of four different I/O schedulers to accommodate different I/O usage patterns. Add the elevator options to the boot loader configuration file (`/boot/grub/grub.conf`) to select one of the following:

- ▶ The Completely Fair Queuing (CFQ) scheduler, `cfq`, is the default algorithm in Red Hat Enterprise Linux 4. As the name implies, CFQ maintains a scalable per-process I/O queue and attempts to distribute the available I/O bandwidth equally among all I/O requests. CFQ is well suited for mid-to-large multi-processor systems and for systems that require balanced I/O performance over multiple LUNs and I/O controllers.
- ▶ The Deadline elevator, `deadline`, uses a deadline algorithm to minimize I/O latency for a given I/O request. The scheduler provides near real-time behavior and uses a round robin policy to attempt to be fair among multiple I/O requests and to avoid process starvation. Using five I/O queues, this scheduler will aggressively reorder requests to improve I/O performance.
- ▶ The NOOP scheduler, `noop`, is a simple FIFO queue and uses the minimal amount of CPU/instructions per I/O to accomplish the basic merging and sorting functionality to complete the I/O. It assumes performance of the I/O has been or will be optimized at the block device (memory-disk) or with an intelligent HBA or externally attached controller.
- ▶ The Anticipatory elevator, `as`, introduces a controlled delay before dispatching the I/O to attempt to aggregate and reorder requests, thereby improving locality and reducing disk seek operations. This algorithm is intended to optimize systems with small or slow disk subsystems. One artifact of using the `as` scheduler, however, can be higher I/O latency.

## Selecting the journaling mode of an Ext3 file system

Three different journaling options in the Ext3 file system can be set with the data option in the **mount** command:

- ▶ **data=journal**

This journaling option provides the highest form of data consistency by causing both file data and metadata to be journaled. It also has the higher performance overhead.

- ▶ **data=ordered (default)**

In this mode, only metadata is written. However, file data is guaranteed to be written first. This is the default setting.

- ▶ **data=writeback**

This journaling option provides the fastest access to the data at the expense of data consistency. The data is guaranteed to be consistent because the metadata is still being logged. However, no special handling of actual file data is done and this might lead to old data appearing in files after a system crash.

There are three ways to change the journaling mode on a file system:

- ▶ When executing the **mount** command:

```
mount -o data=writeback /dev/sdb1 /mnt/mountpoint
```

where /dev/sdb1 is the file system that is being mounted.

- ▶ By including it in the options section of the /etc/fstab file:

```
/dev/sdb1 /testfs ext3 defaults,journal=writeback 0 0
```

- ▶ If you want to modify the default **data=ordered** option on the root partition, make the change to the /etc/fstab file listed above, then execute the **mkintrd** command to scan the changes in the /etc/fstab file and create a new image. Update grub or lilo to point to the new image.

For more information about Ext3, see:

<http://www.redhat.com/support/wpapers/redhat/ext3/>

## Tuning ReiserFS

**Note:** ReiserFS is not supported by Red Hat Enterprise Linux AS.

One of the strengths of the ReiserFS is its support for a large number of small files. Instead of using the traditional block structure of other Linux file systems, ReiserFS uses a tree structure that has the capability of storing the actual contents of small files, or the *tails* of those that are larger, in the access tree

itself. This file system does not use fixed block sizes, so only the space that is needed to store a file is consumed, thus leading to less wasted space.

There is an option when mounting a ReiserFS file system that improves performance, but at the expense of space. When mounting a ReiserFS, you can disable this *tail packing* option by specifying `notail` so that the file system performs a little faster but uses more disk space, as shown in Example 15-7.

*Example 15-7 Example of mounting a ReiserFS file system with the notail option*

---

```
/dev/sdb1 /testfs resierfs notail 0 0
```

---

## Tagged command queuing for SCSI drives

Tagged command queuing (TCQ), first introduced in the SCSI-2 standard, is a method by which commands arriving at the SCSI drive are tagged and reordered while in the queue. This implementation can increase I/O performance in server environments that have a heavy, random workload by reordering the requests to optimize the position of the drive head. Recently, this method of queuing and reordering pending I/O requests has been extended to IDE drives and is referred to as ATA TCQ or existing TCQ and Native Command Queuing (NCQ) in the SATA II specification.

Some System x servers include the integrated Adaptec AIC-7xxx SCSI controller. By executing `cat /proc/scsi/aic7xxx/0`, you can check the current TCQ settings in effect. See `/usr/src/linux-2.4/drivers/scsi/README.aic7xxx` for a detailed description of how to change the default SCSI driver settings.

It is not necessary to recompile the kernel to try different settings. You can specify a parameter `aic7xxx=global_tag_depth:xx` by adding a line in `/etc/modules.conf`, as shown in Example 15-8.

*Example 15-8 Setting TCQ option on a server with an Adaptec aic7xxx SCSI card*

---

```
Edit the /etc/modules.conf file to include  
options aic7xxx aic7xxx=verbose.global_tag_depth:16
```

---

**Note:** If you make a change to `/etc/modules.conf` that involves a module in `initrd`, then it requires a new image through the execution of `mkinitrd`.

## Block sizes

The block size, the smallest amount of data that can be read or written to a drive, can have a direct impact on server performance. As a guideline, if your server is handling many small files, then a smaller block size is more efficient. If your server is dedicated to handling large files, then a larger block size might improve

performance. Block sizes cannot be changed dynamically on existing file systems, and only a reformat will modify the current block size.

When a hardware RAID solution is being used, careful consideration must be given to the *stripe size* of the array (or *segment* in the case of Fibre Channel). The *stripe-unit size* is the granularity at which data is stored on one drive of the array before subsequent data is stored on the next drive of the array. Selecting the correct stripe size is a matter of understanding the predominant request size performed by a particular application.

As a general rule, streaming or sequential content benefits from large stripe sizes by reducing disk head seek time and improving throughput. However, a more random type of activity, such as that found in databases, performs better with a stripe size that is equivalent to the record size.

The block sizes that are offered by Linux vary depending on the distribution:

- ▶ Red Hat Enterprise Linux AS with Ext3 allows block sizes of 1 KB, 2 KB, and 4 KB.
- ▶ SUSE Linux Enterprise Server with ReiserFS allows a block size of 4 KB only.

**Note:** Even though the file system is limited to a maximum block size of 4 KB, it is still best to set a large stripe size on the RAID controller, because the kernel merges the 4 KB reads and writes into larger requests to the disk subsystem. The maximum size of the request to the drives depends on the driver and the available buffer memory. We have seen better performance with a larger stripe size on the disk due to this request merging.

## Guidelines for setting up partitions

A *partition* is a contiguous set of blocks on a drive that are treated as though they were independent disks. The default Linux installation creates a very monolithic install with only three partitions:

- ▶ A swap partition (automatically set to 2x RAM or 2 GB, whichever is larger)
- ▶ A small boot partition, /boot (for example, 100 MB)
- ▶ All remaining space dedicated to /

There is a great deal of debate in Linux circles about the optimal disk partition. A single root partition method can lead to problems in the future if you decide to redefine the partitions because of new or updated requirements. Alternatively, too many partitions can lead to a file system management problem. During the installation process, Linux distributions allow you to create a multi-partition layout.

There are benefits to running Linux on a multi-partitioned disk:

- ▶ Improved security with finer granularity on file system attributes.

For example, the /var and /tmp partitions are created with attributes that permit very easy access for all users and processes on the system, and they are susceptible to malicious access. By isolating these partitions to separate disks, you can reduce the impact on system availability if these partitions need to be rebuilt or recovered.
  - ▶ Improved data integrity, because loss of data with a disk crash would be isolated to the affected partition.

For example, if there is no RAID implementation on the system (software or hardware) and the server suffers a disk crash, only partitions on that bad disk would have to be repaired or recovered.
  - ▶ New installation and upgrades can be performed without affecting other, more static, partitions.

For example, if the /home file system has not been separated to another partition, it is overwritten during an operating system upgrade and all user files that are stored on it are lost.
  - ▶ More efficient backup process
- Partition layouts must be designed with backup tools in mind. It is important to understand whether backup tools operate on partition boundaries or on a more granular level like file systems.

Table 15-6 lists some of the partitions that you might want to consider separating out from the root directory to provide more flexibility and better performance in your environment.

Table 15-6 Linux partitions and server environments

Partition	Contents and possible server environments
/home	A <i>file server environment</i> benefits from separating out /home to its own partition. This is the home directory for all users on the system if there are no disk quotas implemented, so separating this directory should isolate a user's runaway consumption of disk space.
/tmp	If you are running a <i>high-performance computing environment</i> , large amounts of temporary space are needed during compute time, then are released upon completion.
/usr	This is the directory where the <i>kernel source tree</i> and Linux <i>documentation</i> (as well as most executable binaries) are located. The /usr/local directory stores the executables that need to be accessed by all users on the system, and is a useful location to store custom scripts that are developed for your environment. If it is separated to its own partition, then files will not need to be reinstalled during an upgrade.

Partition	Contents and possible server environments
/var	The /var partition is important in <i>mail</i> , <i>Web</i> , and <i>print server environments</i> because it includes the log files for these environments as well as the overall system log. Chronic messages can flood and fill this partition. If this occurs and the partition is not separate from the root directory, service interruptions are possible. Depending on the environment, further separation of this partition is possible by separating out /var/spool/mail for a mail server or /var/log for system logs.
/opt	The installation of some third-party software products, such as Oracle's <i>database server</i> , default to this partition. If not separate, the installation will continue under / and, if there is not enough space allocated, might fail.

For a more detailed and in-depth understanding of how Linux distributions handle file system standards, see the File System Hierarchy project's home page:

<http://www.pathname.com/fhs>

### 15.9.5 The swap partition

The swap device is used when physical RAM is fully in use and the system needs additional memory. When there is no free memory available on the system, it begins paging the least-used data from memory to the swap areas on the disks.

The initial swap partition is created during the Linux installation process, with current guidelines stating that the size of the swap partition should be two times physical RAM. The maximum total size of swap that is supported is 64 GB for both kernel 2.4 and kernel 2.6. If you add more memory to the server after the initial installation, you must configure additional swap space.

There are two ways to configure additional swap after the initial install:

- ▶ You can create a free partition on the disk as a swap partition, which can be difficult if the disk subsystem has no free space available. In that case, you can create a swap file.
- ▶ If there is a choice, the preferred option is to create additional swap partitions. There is a performance benefit because I/O to the swap partitions bypasses the file system and all of the overhead involved in writing to a file.

Another way to improve the performance of swap partitions or files is to create multiple swap areas. Linux can take advantage of multiple swap partitions or files and perform the reads and writes in parallel to the disks. After creating the additional swap partitions or files, the /etc/fstab file includes entries such as those shown in Example 15-9.

*Example 15-9 /etc/fstab file*

/dev/sda2	swap	swap	sw	0 0
/dev/sdb2	swap	swap	sw	0 0
/dev/sdc2	swap	swap	sw	0 0
/dev/sdd2	swap	swap	sw	0 0

Under normal circumstances, Linux would use the /dev/sda2 swap partition first, then /dev/sdb2, and so on, until it had allocated enough swapping space. This means that perhaps only the first partition, /dev/sda2, would be used if there is no need for a large swap space. The maximum supported number of swapfiles is 32.

Spreading the data over all available swap partitions improves performance because all read/write requests are performed simultaneously to all selected partitions. If you change the file as shown in Example 15-10, you assign a higher priority level to the first three partitions.

*Example 15-10 Modified /etc/fstab to make parallel swap partitions*

/dev/sda2	swap	swap	sw,pri=3	0 0
/dev/sdb2	swap	swap	sw,pri=3	0 0
/dev/sdc2	swap	swap	sw,pri=3	0 0
/dev/sdd2	swap	swap	sw,pri=1	0 0

Swap partitions are used from the highest priority to the lowest (where 32767 is the highest and 0 is the lowest). Giving the same priority to the first three disks causes the data to be written to all three disks; the system does not wait until the first swap partition is full before it starts to write on the next partition. The system uses the first three partitions in parallel and performance generally improves.

The fourth partition is used if additional space is needed for swapping after the first three are completely filled up. It is also possible to give all partitions the same priority to stripe the data over all partitions, but if one drive is slower than the others, performance will decrease. A general rule is that the swap partitions should be on the fastest drives available.

**Note:** The swap space is not a replacement for RAM because it is stored on physical drives that have a significantly slower access time than memory.

## 15.10 Tuning the network subsystem

The network subsystem should be tuned when the operating system is first installed, as well as when there is a perceived bottleneck in the network subsystem. An issue here can affect other subsystems. For example, CPU utilization can be affected significantly, especially when block sizes are too small, and memory use can increase if there is an excessive number of TCP connections.

### 15.10.1 Preventing a decrease in performance

The following `sysctl` commands are used primarily to change security settings, but they also have the side effect of preventing a decrease in network performance. These commands are changes to the default values.

- ▶ Disabling the following parameters prevents a hacker from using a spoofing attack against the IP address of the server:

```
sysctl -w net.ipv4.conf.eth0.accept_source_route=0  
sysctl -w net.ipv4.conf.lo.accept_source_route=0  
sysctl -w net.ipv4.conf.default.accept_source_route=0  
sysctl -w net.ipv4.conf.all.accept_source_route=0
```

- ▶ (Red Hat Enterprise Linux AS only) This command enables TCP SYN cookies, which protect the server from syn-flood attacks, both denial-of-service (DoS) or distributed denial-of-service (DDoS):

```
sysctl -w net.ipv4.tcp_syncookies=1
```

**Note:** This command is valid only when the kernel is compiled with CONFIG\_SYNCOOKIES.

- ▶ These commands configure the server to ignore redirects from machines that are listed as gateways. Redirect can be used to perform attacks, so we only want to allow them from trusted sources:

```
sysctl -w net.ipv4.conf.eth0.secure_redirects=1  
sysctl -w net.ipv4.conf.lo.secure_redirects=1  
sysctl -w net.ipv4.conf.default.secure_redirects=1  
sysctl -w net.ipv4.conf.all.secure_redirects=1
```

In addition, you could allow the interface to accept or not accept any ICMP redirects. The ICMP redirect is a mechanism for routers to convey routing information to hosts. For example, the gateway can send a redirect message to a host when the gateway receives an Internet datagram from a host on a network to which the gateway is attached. The gateway checks the routing table to get the address of the next gateway, and the second gateway routes

the datagram's Internet to destination on network. Disable these redirects using the following commands:

```
sysctl -w net.ipv4.conf.eth0.accept_redirects=0  
sysctl -w net.ipv4.conf.lo.accept_redirects=0  
sysctl -w net.ipv4.conf.default.accept_redirects=0  
sysctl -w net.ipv4.conf.all.accept_redirects=0
```

- ▶ If this server does not act as a router, then it does not need to send redirects, so they can be disabled using the following commands:

```
sysctl -w net.ipv4.conf.eth0.send_redirects=0  
sysctl -w net.ipv4.conf.lo.send_redirects=0  
sysctl -w net.ipv4.conf.default.send_redirects=0  
sysctl -w net.ipv4.conf.all.send_redirects=0
```

- ▶ Configure the server to ignore broadcast pings or smurf attacks:

```
sysctl -w net.ipv4.icmp_echo_ignore_broadcasts=1
```

- ▶ Ignore all kinds of icmp packets or pings:

```
sysctl -w net.ipv4.icmp_echo_ignore_all=1
```

- ▶ Some routers send invalid responses to broadcast frames, and each one generates a warning that is logged by the kernel. These responses can be ignored using this command:

```
sysctl -w net.ipv4.icmp_ignore_bogus_error_responses=1
```

### 15.10.2 Tuning in TCP and UDP

You can use the following commands for tuning servers that support a large number of multiple connections:

- ▶ For servers that receive many connections at the same time, the TIME-WAIT sockets for new connections can be reused. This command is useful in Web servers, for example:

```
sysctl -w net.ipv4.tcp_tw_reuse=1
```

If you enable this command, you should also enable fast recycling of TIME-WAIT sockets status as follows:

```
sysctl -w net.ipv4.tcp_tw_recycle=1
```

Figure 15-9 shows that with these parameters enabled, the number of connections is reduced significantly. This reduction is good for performance because each TCP transaction maintains a cache of protocol information about each of the remote clients. In this cache, information such as round-trip time, maximum segment size, and congestion window are stored. For more details, review RFC 1644 from:

<http://www.ietf.org/rfc/rfc1644.txt>

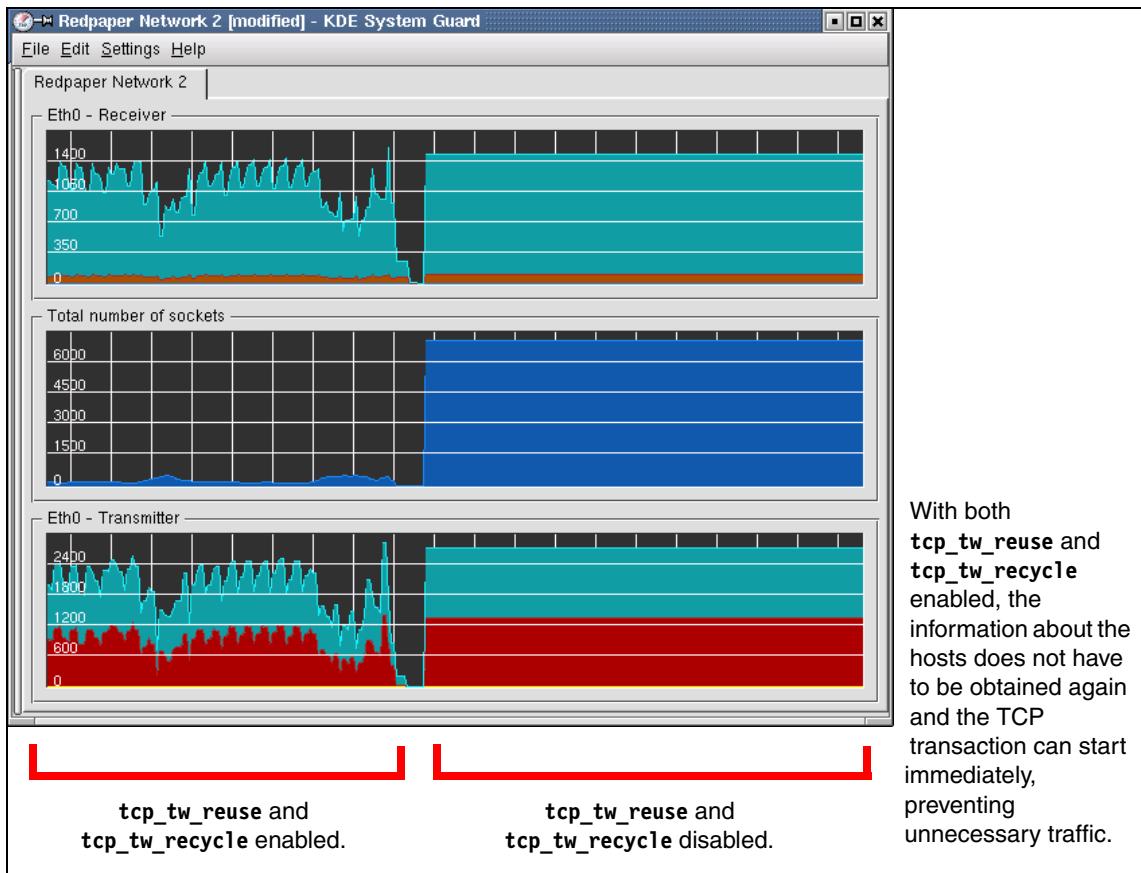


Figure 15-9 Parameters reuse and recycle enabled (left) and disabled (right)

- ▶ The parameter `tcp_fin_timeout` is the time to hold a socket in state FIN-WAIT-2 when the socket is closed at the server.

A TCP connection begins with a three-segment synchronization SYN sequence and ends with a three-segment FIN sequence, neither of which holds data. By changing the `tcp_fin_timeout` value, the time from the FIN sequence to when the memory can be freed for new connections can be reduced, thereby improving performance. You should change this value,

however, only after careful monitoring, because there is a risk of overflowing memory due to the number of dead sockets.

```
sysctl -w net.ipv4.tcp_fin_timeout=30
```

- ▶ One of the issues found in servers with many simultaneous TCP connections is the large number of connections that are open but unused. TCP has a keepalive function that probes these connections and, by default, drops them after 7200 seconds (2 hours). This length of time might be too large for your server and can result in excess memory usage and a decrease in server performance.

Setting keepalive to 1800 seconds (30 minutes), for example, might be more appropriate:

```
sysctl -w net.ipv4.tcp_keepalive_time=1800
```

- ▶ Set the max operating system send buffer size (`wmem`) and receive buffer size (`rmem`) to 8 MB for queues on all protocols as follows:

```
sysctl -w net.core.wmem_max=8388608
```

```
sysctl -w net.core.rmem_max=8388608
```

These commands specify the amount of memory that is allocated for each TCP socket when it is created.

In addition, you should also use the following commands for send and receive buffers. They specify three values: minimum size, initial size, and maximum size:

```
sysctl -w net.ipv4.tcp_rmem="4096 87380 8388608"
```

```
sysctl -w net.ipv4.tcp_wmem="4096 87380 8388608"
```

The third value must be the same as or less than the value of `wmem_max` and `rmem_max`.

- ▶ (SUSE Linux Enterprise Server only) Validate the source packets by reserved path. By default, routers route everything, even packets that obviously are not meant for this network. These packets can be dropped, by enabling the appropriate filter:

```
sysctl -w net.ipv4.conf.eth0.rp_filter=1
```

```
sysctl -w net.ipv4.conf.lo.rp_filter=1
```

```
sysctl -w net.ipv4.conf.default.rp_filter=1
```

```
sysctl -w net.ipv4.conf.all.rp_filter=1
```

- ▶ When the server is heavily loaded or has many clients with bad connections with high latency, it can result in an increase in half-open connections. This is very common for Web servers, especially when there are many dial-up users.

These half-open connections are stored in the *backlog connections* queue. You should set `tcp_max_syn_backlog` to at least 4096 (the default is 1024).

Setting this value is useful even if your server does not receive this kind of connection, because it can still be protected from a denial-of-service (syn-flood) attack.

```
sysctl -w net.ipv4.tcp_max_syn_backlog=4096
```

- ▶ We should set the `ipfrag` parameters particularly for NFS and Samba servers. Here, we can set the maximum and minimum memory used to reassemble IP fragments. When the value of `ipfrag_high_thresh` in bytes of memory is allocated for this purpose, the fragment handler drops packets until `ipfrag_low_thres` is reached.

Fragmentation occurs when there is an error during the transmission of TCP packets. Valid packets are stored in memory (as defined with these parameters) while corrupted packets are retransmitted.

For example, to set the range of available memory to between 256 MB and 384 MB, use the following commands:

```
sysctl -w net.ipv4.ipfrag_low_thresh=262144  
sysctl -w net.ipv4.ipfrag_high_thresh=393216
```

## 15.11 Xen virtualization

Virtualization has become a key requirement for the enterprise and results from a need to focus on reduced total cost of ownership (TCO) for enterprise computing infrastructure. Most servers today run at less than 15% utilization, meaning that most server capacity is wasted. Operating system virtualization allows multiple operating system and application images to share each server.

Because every physical server can host multiple virtual servers, the number of servers is reduced. However, today's virtualization offerings are also facing performance issues. To bypass these issues, a low level virtualization software layer known as a *hypervisor* has been introduced.

XenSource<sup>3</sup> founders created the Xen open source hypervisor, which is now developed collaboratively by over 20 major enterprises working on the Xen project. Xen hypervisor fully supports VT-x hardware virtualization from Intel and offers a software layer to this facility. Pacifica from AMD is not yet supported.

Both RHEL and SLES integrate the Xen hypervisor into their latest releases as part of the core Linux distribution. So, by default, the two major Linux distributions are virtualization-enabled.

---

<sup>3</sup> Portions of the material in this section are from XenSource, reprinted by permission.

### 15.11.1 What virtualization enables

Operating system virtualization is achieved by inserting a layer of software between the operating system and the underlying server hardware. This layer is responsible for allowing multiple operating system images (and their running applications) to share the resources of a single server. In this environment, each operating system believes that it has the resources of the entire machine under its control, but the virtualization layer, or hypervisor, transparently ensures that resources are properly shared between different operating images and their applications (Figure 15-10).

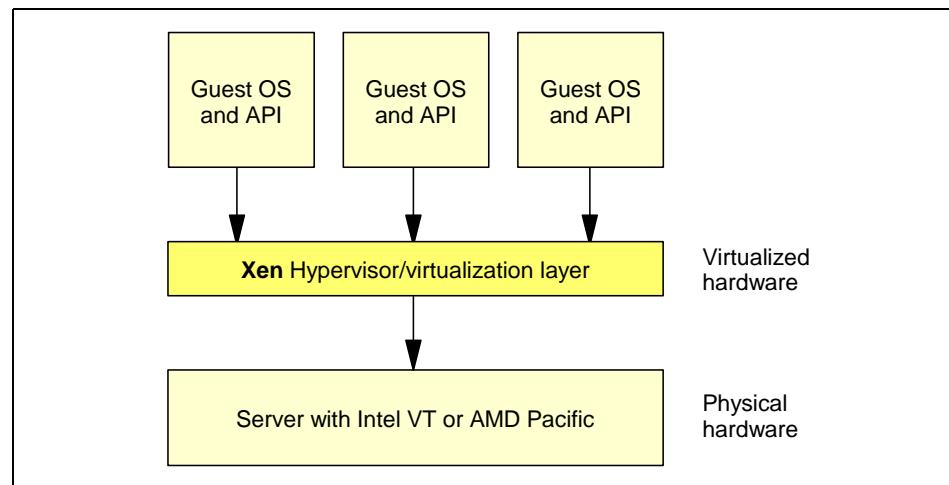


Figure 15-10 Virtualization concept

In operating system virtualization, the hypervisor must manage all hardware structures to ensure that each operating system, when running, has a consistent view of the underlying hardware.

Although there are several methods to manage hardware structures, the simplest method (with the worst performance) is to provide a software emulation layer of the underlying chipset. This method imposes severe performance overheads, restricts the technique to a single chipset architecture (such as x86), and involves patching the kernel of a running operating system dynamically to prevent it from modifying the state of the hardware. The additional overhead that is required to manage the hardware state for the operating system and to present to it an emulated chipset abstraction causes a significant performance overhead, frequently as much as 30% to 50% of the overall system resources.

### 15.11.2 Full virtualization versus paravirtualization

Xen is a virtualization product that uses *paravirtualization*.

Full virtualization is the concept of creating a virtual layer, typically a hypervisor, that fully simulates a standard x86 system. In this environment, the guest operating system does not need to be modified to be aware of the virtualization layer and can run natively on the virtualization layer as thought it were on a standard x86 system.

Both VMware ESX Server and Microsoft Virtual Server implement a full virtualization technology, as shown in Figure 15-11.

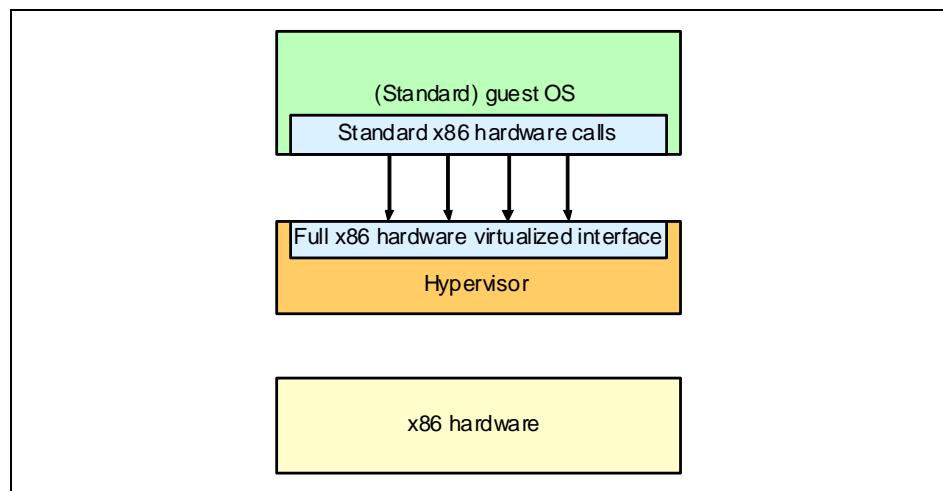


Figure 15-11 Full virtualization architecture

With paravirtualization, the guest operating system is modified to be virtualization-aware so that it can call the hypervisor directly to perform low-level functions, as illustrated in Figure 15-12 on page 499.

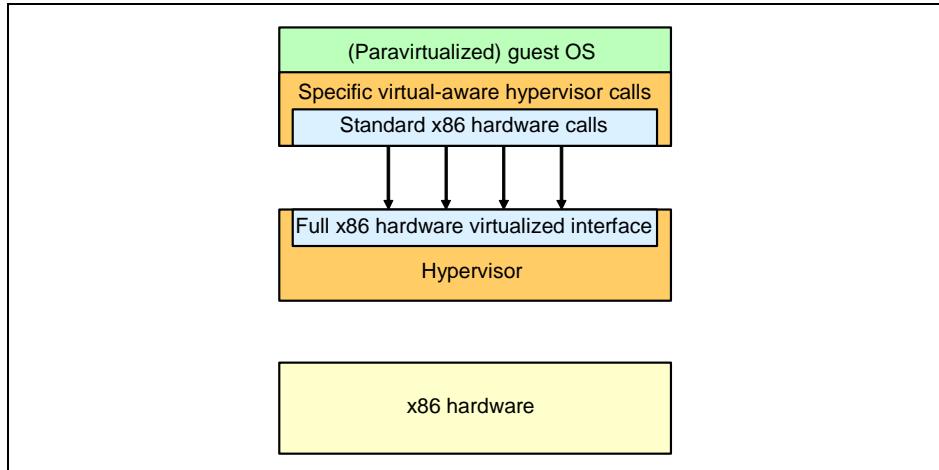


Figure 15-12 Paravirtualization architecture

There are at least two reasons for doing paravirtualization:

- ▶ Reduced complexity of the virtualization software layer

Because x86 historically does not support virtualization, the full virtualization approach must implement mechanisms for which it traps certain privileged guest operating system calls and then translate them into instructions that suit the virtualized environment. This process is accomplished using a technique called *binary translation*.

With this process, some current virtualization products trap and translate certain instructions that are directed at the hardware, thereby allowing guest operating systems to operate as though they were in full control of the hardware.

This is where paravirtualization differs from full virtualization. Instead of the hypervisor trapping low-level instructions and transforming those, it is the virtual-aware guest operating system that behaves differently and becomes aware that it is not the only operating system on the server. This in turn means that the hypervisor does not need to provide the complexity of entirely simulating an x86 computer. The result is that the entire system can be streamlined to run more efficiently.

- ▶ Performance

The second reason to implement paravirtualization is performance. The full virtualization approach often suffers in terms of performance because there are many overheads in running a standard, unmodified guest operating system on a virtualization layer. For example, a standard unmodified guest operating system typically checks everything before passing the information

to the hardware (CPU, memory, and I/O). So does the virtualization layer (which it needs to do because it is closest to the hardware).

### 15.11.3 CPU and memory virtualization

In Xen paravirtualization, virtualization of CPU, memory, and low-level hardware interrupts are provided by a low-level hypervisor layer. When the operating system updates hardware data structures, it collaborates with the hypervisor by making calls into an API that is provided by the hypervisor. This collaboration allows the hypervisor to keep track of all the changes that the operating system makes and to decide optimally how to manage the state of hardware data structures on context switches. The hypervisor is mapped into the address space of each guest operating system so that there is no context switch overhead between any operating system and the hypervisor. Finally, by cooperatively working with the guest operating systems, the hypervisor determines the I/O requirements of the guest operating system and can make the operating system aware that it is being virtualized.

From a kernel perspective, it means that parts of the core kernel code have to be modified to better perform over the Xen hypervisor. This does not affect the final virtual machines and guest operating systems, and increases the overall performance compared to the other virtualization methods.

### 15.11.4 I/O virtualization

Paravirtualization provides significant benefits in terms of I/O virtualization. In the Xen product, I/O is virtualized using only a single set of drivers for the entire system (across all guests and the hypervisor), unlike emulated virtualization in which each guest has its own drivers and the hypervisor has yet another set of drivers.

In each Xen hypervisor guest, simple paravirtualizing device drivers replace hardware-specific drivers for the physical platform. Paravirtualizing drivers are independent of all physical hardware, but represent each type of device (for example, block I/O, Ethernet, and USB). Moreover, in the Xen architecture the drivers run outside the base hypervisor, at a lower level of protection than the core of the hypervisor itself. In this way the hypervisor can be protected from bugs and crashes in device drivers and can make use of any device drivers that are available on the market. Also, the virtualized operating system image is much more portable across hardware, because the low levels of the driver and hardware management are modules that run under control of the hypervisor.

For more information about Xen, visit:

<http://www.xensource.com/products/xen/index.html>



## VMware ESX 3.5

VMware ESX<sup>1</sup> is currently the most popular virtualization software product for the Intel processor-based server market. Compared to hosted virtualization solutions such as VMware Server and Microsoft Virtual Server, ESX Server offers the advantage of eliminating one layer of overhead, namely, the host operating system. Because the VMware ESX kernel runs directly on the hardware by incorporating a hypervisor virtualization solution, the system has improved performance and stability.

VMware ESX is also more suitable for enterprise-level solutions because it features important redundancy features. VMware ESX is capable of simultaneously hosting many different operating systems and applications.

This chapter refers mostly to VMware ESX 3.5 with some comments about VMware vSphere 4.0

---

<sup>1</sup> Portions of the material in this section are from VMware, Inc., reprinted by permission.

## 16.1 Introduction to VMware ESX 3.5

Performance tuning for large VMware ESX environments is a challenging task. VMware ESX can be a heavy load on your server hardware, and depending on the workload and the number of virtual machines, your server might experience a bottleneck in one of the server's subsystems. It is, therefore, important that you design and configure the hardware so that you will not run into a bottleneck that could limit system performance.

Fine-tuning the virtual guest on the VMware ESX is also as important as the tuning of the VMware ESX itself. Recommended actions for some Microsoft Server versions are provided in 16.3.2, "Tuning the virtual machines" on page 520.

### 16.1.1 An approach to VMware ESX performance and tuning

Before discussing the tuning options that VMware ESX provides, it is important to understand the impact of virtualization on performance. VMware ESX virtualizes the hardware and provides an environment for multiple operating systems to run on one physical machine.

It is critical that you understand the core VMware ESX, hardware, and networking technologies. It is also very beneficial to understand your application environment when you are working toward achieving optimal performance (that is, knowing your memory requirements and if your application is I/O-intensive or CPU-intensive, and so on).

## 16.2 Hardware considerations

No amount of tuning can help if your hardware is underconfigured for the type of guest operating systems or applications you plan to run; it must align with your hardware requirements. Therefore, you must understand your hardware requirements, including the number and type of hardware components such as network cards, internal and external disk subsystems, CPUs, memory and system bus speed, and so on. These all play a vital role in determining how well your overall system performs and in how many virtual machines (VMs) you may be able to effectively use. In a sense, your hardware is your first line of defence against performance degradation.

As a best practice, use a hardware system that is scalable and fault tolerant, including multiple CPU cores, memory subsystems, I/O cards, RAID cards, disks, network cards, power supplies, and fans. It is also a good idea to choose

hardware that allows you to replace such components while the server is operational, because downtime is extremely costly and extremely undesirable.

There are four key focus areas for VMware ESX:

- ▶ Network
- ▶ Disk I/O
- ▶ CPU
- ▶ Memory

### 16.2.1 VMware ESX network concepts

You need basic knowledge of the networking features of VMware ESX to be able to tune network throughput. As of VMware ESX 3.5, support for the following enhancements has been enabled.

- ▶ Jumbo Frames and TCP Segmentation Offload (TSO) for vmxnet devices allows for the virtual adapter to be configured with large Ethernet frames of up to 9000 bytes, which are much bigger than standard 1500 bytes maximum transfer units.

This allows guest operating systems to:

- Transfer a smaller number of packets with a larger payload, thus reducing the internal ESX network traffic (this means less overhead on the transmit path).
- Use fewer CPU cycles due to the smaller number of transmit TCP packets.
- ▶ VLAN and checksum offloading
- ▶ Multiple (NetQueue) receive queues - these allow a single NIC card to have multiple receive queues for processing incoming packets from multiple CPUs. Thus, a device can use multiple CPUs for processing packets from multiple virtual machines on the host, thereby boosting network performance.
- ▶ 10 Gbps Ethernet NIC adapters

Other network-related tips:

- ▶ CPU plays a key role in network throughput; insufficient CPU resource will degrade network throughput, regardless of virtualized or dedicated environments.
- ▶ Internally within the ESX server, VMkernel, Console and the virtual machines contend for resources, Therefore, as a generally accepted good practice,

VMs that will be running a heavy workload and I/O-intensive applications should be assigned their own dedicated one or more physical NICs.

**Note:** The VMkernel network device driver, by default, is set to auto-negotiate (speed and duplex mode). Reconfigure this setting to your network setting.

- ▶ If you have an environment where multiple VMs will be communicating with each other, and if your network policy allows, it, then connect these machines to the same virtual switch. This will ensure that the network traffic between these VMs will be internal to the ESX host.

Connecting them to different virtual switches within the same host would mean that the traffic would go out of the VMware ESX host to the physical LAN, thus involving more CPU cycles.

- ▶ Remove or disable any virtual network adapters and switches that are not in use.
- ▶ Configure NIC to use auto-negotiation
- ▶ Set duplex mode of the NICs to full-duplex mode.
- ▶ Ensure you have selected the appropriate virtual switch driver for your environment.

A virtual switch in VMware ESX is a build-to-order technology. In other words, at runtime a set of components come together to enable this functionality. Each time ESX needs to build a virtual switch, it installs and executes the required components to enable communication between virtual adapters and physical adapters. This installation and execution at run time takes CPU cycles away from the system. Also remember each virtual switch maintains its own forwarding table. The enablement and creation of the virtual switches can be achieved via the Virtual Center.

The NIC teaming technique allows you to connect a single virtual switch to multiple physical NICs, which lets you share the network load on multiple physical NICs. Furthermore, this guards against one or more physical NIC failures.

VMware ESX network load balancing also allows you to spread network traffic from one virtual machine to multiple physical NIC cards, thereby providing higher throughput than a single physical NIC card. The ESX server provides the following options, which can be particularly useful in troubleshooting, isolating network-related issues, and eliminating poor network performance at a system level.

Note the following points:

- ▶ A virtual machine cannot use more than one physical NIC unless it uses multiple source MAC addresses for traffic.
- ▶ The physical NIC configuration (speed, duplex, and so on) is not important when configuring a virtual network.
- ▶ The VMKernel network interface is used by VMWare VMotion® and IP Storage, such as iSCSI, and NFS.

By default, NIC teaming applies a fail-back policy. In another words, if a NIC failed and then came back online, then ESX will place it in an active state.

*Table 16-1 Routing policies*

Policy description	Behavior	Comments
Route based on IP hash	Uplink is based upon source and destination IP address for each packet.	Provides a way to create fast network pipe, by grouping multiple physical NICs to a single virtual NIC.
Route based on the originating virtual switch port ID (default configuration)	Traffic from virtual adapter is consistently sent to same physical adapter, unless there is a failure - in that case, it is sent to a failover physical adapter.	Provides even network traffic distribution across number of physical adapters.
Route based on source MAC hash	Uplink is based upon source MAC address.	Provides a method whereby network traffic from a virtual NIC is sent to the same physical NIC.

### **16.2.2 VMware ESX Virtualized storage and I/O concepts**

The disk I/O component requires ongoing examination and monitoring to ensure optimal system performance. If all other system components, including CPUs, memory, network, and so on are performing at an optimum level and this component continues to be constrained, then the overall system health will be poor.

VMware ESX can use local and networked storage devices to create and manage VMs. ESX formats the storage devices under its control and creates datastores; essentially, ESX hides the physical storage systems from VMs.

Basically, the storage can be:

- ▶ Local storage - all disk subsystems are local to the ESX host, internal disks, or directly connected storage.
- ▶ Network storage - this is external storage connected typically via Fiber Channel (FC), Internal SCSI (iSCSI), and Network Attached Storage (NAS).
- ▶ With ESX 3.5, the BusLogic and LSI Logic storage adapters are available for VMs, and the VMs use these full virtualized adapters to obtain access to virtual storage.

However, ESX 4.x includes an additional paravirtualized storage adapter, called PVSCSI. PVSCSI provides performance enhancements over the previous versions of the storage adapters.

Additionally, ESX 4.x added the following features to further help improve I/O performance:

- I/O concurrency for enhanced storage performance
- Interrupt coalescing and delivery mechanisms for better I/O throughput

VMware ESX gives you the ability to configure the maximum I/O request size to be 32 MB. This means that you can directly pass up to 32 MB chunks of data to storage devices. Any requests larger than this are broken into smaller blocks. To determine whether your system requires tuning of this parameter, run the `esxtop` utility, then gather the data and look at the latency statistics. If your system is reporting high latency, then adjusting this parameter might help.

**Note:** Make sure your storage devices are able to handle larger I/O requests. Otherwise, system performance may decrease as a result.

Multipathing is another technique that can help, because it allows the ESX host and the storage system to communicate via multiple physical paths, thus redistributing I/O loads over multiple physical paths and inherently providing redundancy.

This technique is extremely handy in eliminating I/O bottlenecks. VMWare provides an extensible multipathing module known as Native Multipathing Plugin (NMP). The NMP further contains two sub-plugins known as Storage Array Type Plugin (SATP) and Path Selection Plugin (PSP).

**Note:** To achieve this, ESX enables Pluggable Storage Architecture (PSA) which runs at the VMKernel layer.

VMware ESX supports the three virtual disk modes listed and described in Table 16-2.

*Table 16-2 Virtual disk modes*

Description	Behavior	Impact
Independent persistent	Changes are immediately written to the disk.	Considered best for performance.
Independent nonpersistent	Disk writes are appended to a redo log. ESX first checks the redo log before going to the base disk.	This operation may impact performance.
Snapshot™	Captures the entire state of the VM, including memory, disk states, other settings.	This operation uses redo logs, which may impact performance.

ESX also support multiple disk types, as listed and described in Table 16-3.

*Table 16-3 Disk types*

Description	Behavior	Impact
Thick – Eager-zeroed	All space is zeroed out at the time of creation. Thus, it takes time to create the disk.	Provides best performance.
Thick – Lazy-zeroed	All space is allocated at time of creation. However, each block is only zeroed on first write.	Provides short creation time. However, there is reduced performance at the first write, but good performance thereafter.
Thin	Space is allocated and provisioned on demand.	High performance the first time, but the good performance thereafter.

By default:

- ▶ Active/Passive storage arrays use the “Most Recently Used” path policy. Do not use the “Fixed” path policy for these arrays, because this could result in LUN thrashing.
- ▶ Active/Active storage arrays use the “Fixed” path policy. In this case, you should also set the “preferred path” for each LUN.

The SCSI protocol allows you to execute multiple commands against a LUN simultaneously. SCSI device drivers have a configurable parameter known as LUN Queue Depth. This parameter determines how many commands can be simultaneously active for a given LUN. The VMware ESX default is set to 32 commands. If more than 32 commands are stored in the ESX kernel, this may increase I/O latency.

The host bus adapter (iSCSI initiator) can support many more than 32 commands. Thus, if your environment is experiencing increased I/O latency, then you can increase the LUN Queue Depth parameter. However, make sure that you fully understand which VMs are sharing the iSCSI initiator, and how they are sharing. When multiple VMs are sharing the same LUN (VMFS volume), then the combined outstanding commands permitted from all VMs is governed by the Disk.SchdNumReqOutstanding parameter, which can be set via Virtual Center. Note once again that, if all commands from all VMs exceed this parameter, then they are stored in the ESX kernel.

Typically, you would want to set the LUN Queue Depth parameter to be the same as the Disk.SchdNumReqOutstanding parameter. If this parameter is set lower than LUN Queue Depth, then only that number of commands are issued from the VM kernel to the LUN. If this parameter is set higher, it basically has no effect.

**Note:** The Disk.SchdNumReqOutstanding parameter has no effect if only a single VM is using the LUN.

Furthermore, these parameters must be regularly monitored and adjusted to assess the overcommit rate. This rate is really a balance between I/O latency versus additional CPU cycles required by VM kernel to manage excess. This can be monitored by the `esxtop` utility with QSTAT enabled. To be able to tune the storage component correctly, it is important to understand all of the layers that the ESX host imposes and bottlenecks at each layer.

**Note:** VMFS file systems can suffer a performance hit if the partitions are not aligned.

General tips for storage:

- ▶ Make sure that all storage firmware is current, as recommended by the storage vendor.
- ▶ If using SAN, ensure SAN is connected to the SAN switch at the same speed as ESX HBA.

- If multiple I/O-intensive VMs are running on the same ESX host, then ensure you have proper links, multiple paths, and I/O bandwidth to be used by the ESX host, especially when using external storage.

ESX also:

- Allows the disk I/O bandwidth to be set differently for each VM on the same ESX host. This should allow you to balance the VMs appropriately.
- Allows Raw Device Mapping (RDM) in the form of VMFS files. For certain situations, this feature maybe very useful as well.
- VMs that are I/O-intensive should not share Ethernet links to storage devices. Instead, multiple Ethernet links should be configured to avoid bottlenecks.
- The software-initiated iSCSI processing is done on the ESX host. This will require additional CPU cycles.

## General disk I/O performance/tuning approach

A General disk I/O performance/tuning approach is as follows:

1. Ensure that you have a correct hardware configuration with the appropriate level of disk speeds to meet your business requirements. Visit the following link to verify ESX support for your hardware:  
<http://www.vmware.com/resources/compatibility/>
2. Review the ESX hosts log files (/var/log/vmkernel or /var/log/dmesg), or, for SAN/NAS shared storage, the (/proc/vmware/scsi/vmhba) files.
3. Gather the data using **esxtop** and **resxtop** and other tools.
  - The data should be gathered over a period of time. If this is not possible due to already highly constrained systems, perform the following tasks:
    - Collect data over period of time running at different intervals of the day. One benefit of this approach is that you collect small amounts of data over different time intervals (for example, every 80 minutes over two days). Analyzing this data can help you to decide quickly whether the I/O is more constrained or less constrained at certain periods of time, or if the constraint is continuous. This in turn may lead you to look at what was happening in terms of how the system was being used and by which application, and so on.
    - Collect data for individual VMs using vendor-provided or operating system-provided tools. Compare these results to the ESX host's overall results. This could reveal an issue with a VM being constrained versus the ESX host. Generally, this is an easier fix, because adding additional resources to the VM in question could be the solution.

- Two key measurements for any I/O subsystems are:
    - Throughput, which refers to the amount of data (read/write) transferred over time, generally measured in bytes/sec. The goal here is to increase throughput by reducing the number of blocks of read/writes.
    - Latency is the time duration (in milliseconds) needed to complete the disk I/O operation. The goal here is to reduce the time for each I/O operation.
  - There are many parameters that are collected by esxtop and resxtop, but a quick starting point could be to assess the QUED and KAVG/cmd monitors. For an example, a large I/O queue (QUED counter), coupled with higher VMkernel latency, would indicate that queuing depth may need to be adjusted.
4. Identify the most critical I/O component in the data flow path, and assess its bandwidth usage and requirements. This might be your weakest link.
    - Configure (or reconfigure) the system to address the weakest link or required workload, by adding additional virtual resources, CPU, memory, or additional paths to the storage and network.
    - For VMs running with applications that are able to “cache data”, by increasing the amount of memory, the overall I/O can be reduced.
    - If possible, reduce software I/O with dedicated hardware

### 16.2.3 Virtualized CPU concepts

VMware ESX allocates a share of the physical resources to each virtual machine. VMware ESX time-slices the physical CPUs across all of its VMs, along with any resources required to run itself and the console. Each VM within the same ESX host receives an equal share of CPU per virtual CPU. Additionally, the CPU hardware vendors for some time have enabled Virtualization Acceleration features. These allow the CPU to be efficiently virtualized at a hardware level. These features also help with performance.

The host masks the hardware from VMs. However, VMs do detect and are aware of the underlying specific processor model, which could be different in terms of functionality they offer, for example, Intel versus AMD.

The CPU virtualization does add some amount of overhead and this overhead increases as additional VMs are added. The amount of overhead varies depending upon the need for VMs to use CPU, memory, and other physical resources.

**Note:** CPU virtualization overhead usually impacts overall performance.

A key factor in optimizing for performance is understanding your application environment and then planning accordingly. For example, the workload generated on the virtual CPUs and then on the physical CPUs by virtual machines running I/O versus running CPU-intensive applications would be different.

Therefore, their resource requirements would also be different. The CPU-intensive applications, which require more CPU time to execute additional instructions, would need VMs with multiple virtual CPUs. I/O-intensive applications which require more I/O bandwidth. Therefore, in this case VMs with multipaths to virtual storage be more appropriate for the task.

It is also important to understand if the application is single-threaded or multi-threaded. For single-threaded applications, uniprocessor VMs provide better performance as opposed to SMP-based VMs. The single-threaded applications cannot take advantage of multiple virtual VMs; therefore, these additional VMs would consume system resources.

Additionally, if applications can leverage multiple CPUs, it is advantageous to pin VMs processes to virtual CPUs of the guest VM. This is because, when an ESX host migrates these processes to or from another virtual CPU, it consumes resource, and the higher the migration, more resource is consumed.

The ESX host allows the VMs to be assigned to a specific processor in a multiprocessor system. Depending upon your application needs, this can enhance performance or degrade performance. If you have enabled the feature in your environment and are experiencing slow system response, reexamine the applications running in your VMs and make sure that your VMs are correctly configured. A simple test is to remove the assignment and assess the results. Generally speaking, this feature carries more risk than reward and should be used only sparingly.

If at all possible, disable any VMs that are sitting idle, because they consume system resources as well as additional resources required for the ESX host to manage them. The ESX host will try to halt or deschedule the VM that is trying to idle. The detection method, that ESX uses is controlled by the “idle loop spin parameter.

The idle VMs may halt or execute idle loop. If the VMs halt, then no additional system resources are consumed. If, however, they do not halt and continue to execute idle loop, this actively executes instructions on the physical CPU. As a result, these VMs are using real CPU cycles and, when you run performance-gathering tools, they pick up this usage.

Similarly, having VMs with multiple virtual CPUs that are not all used also consumes system resources and impacts overall system performance. This

parameter should be monitored regularly to reduce overall system resource consumption and enhance performance.

If your hardware provides support for hyper-threading, this feature can help boost performance because it allows a single processor to essentially act like two processors. That is, instead of running a single thread at a time, you now can run two threads at once. This option should be configured as “Any” as opposed to “Internal” or “None”.

The time a virtual machine must wait in a ready-to-run mode before it can be scheduled on a CPU is another key parameter that must be monitored, because it can indicate over-constrained physical CPU resources. If this parameter is reporting a high value, then you should consider taking remedial action, such as enabling or adding additional physical resources, reconfiguring existing VMs, migrating some of the VMs to another ESX host, and so on.

**Note:** This parameter must be passed along with other parameters (such as CPU utilization, response time, and so on), and not by itself.

Notice the %RDY column displayed in Figure 16-1.

ID	GID	NAME	NWLD	%USED	%RUN	%SYS	%WAIT	%RDY
1	1	idle	4	400.00	400.00	0.00	0.00	23.94
2	2	system	6	0.00	0.00	0.00	600.00	0.00
6	6	helper	22	0.05	0.05	0.00	2200.00	0.06
7	7	drivers	14	0.01	0.01	0.00	1400.00	0.00
8	8	vmotion	1	0.00	0.00	0.00	100.00	0.00
9	9	console	1	3.81	3.93	0.00	100.00	13.68
17	17	vmware-vmkauthd	1	0.00	0.00	0.00	100.00	0.00
21	21	BLDTACDATA	6	15.91	15.94	0.04	600.00	0.22
40	40	BLDTACWEB1	7	4.07	4.07	0.00	700.00	0.17

Figure 16-1 Output from esxtop

As a best practice, software components that do not provide direct business service or provide no business service should be scheduled to run at off-peak times or disabled; these include:

- ▶ Backups.
- ▶ Anti-virus updates and scanning.
- ▶ Screen savers and other type of animation.
- ▶ Disable unused hardware.
- ▶ Disable CDs and DVDs in VMs, because some applications frequently poll these devices, which takes resource away from production work.

- ▶ Unless you have a real need, do not enable the ESX host to manage CPU power.

Additionally, there are other performance enhancement features enabled with VMware ESX 4.x, such as:

- ▶ Support for 8 virtual CPUs per VM as opposed to 4 virtual VMs with ESX 3.5.
- ▶ Improved CPU scheduler for better support of larger I/O-intensive VMs.

#### 16.2.4 ESX virtualized memory concepts

Both the VMware ESX host and the virtual machines consume system resource, including memory. To effectively tune poor system performance that is related to memory, you need to understand how ESX uses memory, both physical and virtual. In fact, it is the VMkernel that manages memory (this does not apply to ESX Console).

Basically, the overhead is due to:

1. VMKernel
2. Device drivers
3. Console
4. Associated with each VM (space for frame buffers, paging tables, and so on)

Physical and virtual memory are divided into pages (blocks), typically 4 KB, 8 KB, and up to 2 MB (large pages). When system memory is fully consumed, the system will offload these pages onto disk, called as page/swap space.

Virtual memory is independent from the physical memory. Each VM's configuration specifies the amount of virtual memory it has, which is maintained by the ESX host's virtualization layers. Virtual memory is independent of physical memory, so ESX is able to allocate more virtual memory (overcommit) to VMs than the physical memory present in the system.

The VM's memory is limited to what its configuration says. If the applications running on the VM require more or additional memory than what is configured, then additional pages are stored on the disk. The following configurations control, to a great extent, how the VMKernel should handle the memory needs of virtual machines:

- ▶ Shares

This specifies the priority of the VM to use more memory than is configured relative to other VMs. The number of shares factors into how the ESX host determines memory allocation and working set size. The work set size is an estimate based upon memory monitoring activity. By default, the monitoring

interval is 60 seconds. In extreme cases, you may want to change this value (Mem.SamplePeriod).

- ▶ Limit

This defines an upper limit for the ESX host regarding how much physical memory a VM can use.

- ▶ Reservation

This defines a lower limit of physical memory that the ESX host will reserve for the VM. This condition would apply even if the ESX host is over-committing memory. However, if the VM underuses its allocated memory, the VMkernel can allocate the unused portion of the configured portion to the other VMs, until it reaches its reserved amount.

**Note:** Memory virtualization overhead is dependent upon workload. As suggested previously, idle VMs should be monitored and halted because they consume both CPU cycles and memory. In fact, ESX uses more resources for idle memory than for memory in use, so the higher the idle memory, the more resource is taken.

The VMKernel continues to load-balance memory across all VMs. It allocates more to VMs that require more, and reclaims (based upon VM configuration) memory from VMs that are underutilizing. Therefore, carefully plan how you configure a virtual machine, including which options to use and how much memory to reserve. Also consider how you are going to use the VM, including which applications are going to run on it and how they consume resource.

These basic configuration steps can have a profound effect on how ESX behaves and how well it performs. Plan your ESX host deployments well in advance to ensure your starting point is well thought out. When you do run into problems, you have an opportunity to take advantage of additional features within ESX to optimize your performance, such as Memory Sharing and Software-Based/Hardware-Assisted memory virtualization.

When virtual memory requirements are greater than the physical memory present in the system, ESX uses techniques called “ballooning” or “swapping” to handle the situation. ESX loads a ballooning driver (vmmemctl) into the guest operating system of each VM. Ballooning reclaims memory that can be configured. VMWare recommends that you use ballooning to help performance.

In certain instances, you will not want the ESX host to take away too much memory from the guest operating system, but still want to benefit from ballooning. It is also very important to configure the swap space adequately for each guest operating system. Generally, you should configure swap space on a local drive as opposed to a remote or networked drive. By default, the swap

location is also where the virtual machine's configuration file is. So, allocate adequate swap space, because this can impact not only the guest operating system, but the entire ESX host system.

For additional information about these concepts, refer to:

- ▶ Performance Counters  
[http://www.vmware.com/files/pdf/technote\\_PerformanceCounters.pdf](http://www.vmware.com/files/pdf/technote_PerformanceCounters.pdf)
- ▶ Performance Best Practices and Benchmarking Guidelines  
[http://www.vmware.com/pdf/VI3.5\\_Performance.pdf](http://www.vmware.com/pdf/VI3.5_Performance.pdf)
- ▶ Performance Tuning Best Practices for ESX  
[http://www.vmware.com/pdf/vi\\_performance\\_tuning.pdf](http://www.vmware.com/pdf/vi_performance_tuning.pdf)
- ▶ vSphere Resource Management Guide  
[http://www.vmware.com/pdf/vsphere4/r40/vsp\\_40\\_resource\\_mgmt.pdf](http://www.vmware.com/pdf/vsphere4/r40/vsp_40_resource_mgmt.pdf)
- ▶ ESX Configuration Guide  
[http://www.vmware.com/pdf/vsphere4/r40/vsp\\_40\\_esx\\_server\\_config.pdf](http://www.vmware.com/pdf/vsphere4/r40/vsp_40_esx_server_config.pdf)
- ▶ VMWare ESX product documentation  
<http://www.vmware.com/support/pubs/>

### 16.2.5 VMware disk partitioning

In VMware ESX, you should differentiate between two types of disk storage:

- ▶ Disk storage for the virtual machines
- ▶ Disk storage for the VMware ESX kernel, the SWAP file, the various log files, and the Console operating system

You must use VMFS for the partition on which your virtual machines reside. Generally, the default settings proposed by the VMware ESX installer are well suited and should not need further optimization. Table 16-4 illustrates a partition table.

*Table 16-4 Typical partition table for a VMware ESX server on a RAID-1 74 GB volume*

Partition	Recommended size	Format
Boot	100 MB	Ext3
/	5120 MB	Ext3
Swap	2048 MB	Swap
Vmkcore	100	Vmkcore

Partition	Recommended size	Format
/home	2048	Ext3
/opt	2048	Ext3
/opt/Tivoli	2048 MB	Ext3
/tmp	2048 MB	Ext3
/var	2048 MB	Ext3
Vmimages	Rest of space if any	Ext3

Note that the size of the Console operating system swap partition is set to at least twice the amount of maximum suggested Console operating system memory. This setting allows you to add more virtual machines to your server. If you know the size of the Console operating system memory requirements exactly, you can also set this partition to just twice the amount of memory you effectively are using.

On your external storage, configure more than one VM file system if the external storage is very large. Although you might lose some efficiency in storage usage, you will have added resiliency in case one VM file system is corrupted.

However, having multiple VM file system partitions on a relatively small disk array only decreases overall performance because the disk heads have to move back and forth every time another VM file system partition is accessed.

### 16.2.6 Firmware and BIOS settings

We recommend that you use the latest UpdateXpress CD to update your BIOS and firmware to the latest levels. You can download UpdateXpress the latest from:

<http://www.ibm.com/support/docview.wss?uid=psg1MIGR-53046>

The following examples are a typical BIOS configuration of a x3850 M2 suggested for an ESX VMware host.

---

*Example 16-1 BIOS configuration*

---

Date and Time

Enter the date and time

Start Options

Startup Sequence Options

Primary Startup Sequence:

First Startup Device: Diskette Drive 0

```
Second Startup Device: CD ROM
Third Startup Device: Hard Disk 0
Fourth Startup Device: Disabled
Wake on LAN: Enabled
Wake on LAN Startup Sequence:
    First Startup Device: Diskette Drive 0
    Second Startup Device: CD ROM
    Third Startup Device: Hard Disk 0
    Fourth Startup Device: Disabled
Advanced Setup
Memory Settings
    Memory Array Setting: High Performance Memory Array (HPMA)
CPU Options
    PowerExecutive Power Capping: Enabled
    Processor Adjacent Sector Prefetch: Disabled
    Processor Hardware Prefetcher: Disabled
    Processor Execute Disable Bit: Enabled
    Intel Virtualization Technology: Enabled
    Processor IP Prefetcher: Enabled
    Processor DCU Prefetcher: Disabled
RSA II Settings
    DHCP Control: Use Static IP
    Enter the Static IP Settings
    OS USB Selection: Linux OS
    Ensure Periodic SMI Generation: Disabled
    Save Values and Reboot RSA II
```

---

*Example 16-2 HBA configuration in BIOS (for each HBA installed)*

---

```
Configuration Settings
    Host Adapter Settings
        Host Adapter BIOS: Disabled
    Advanced Adapter Settings
        Enable LIP Reset: No
        Enable LIP Full Login: Yes
        Enable Target Reset: Yes
        Port Down Retry Count: 15
```

---

*Example 16-3 LSI Logic Corp. MPT IM BIOS RAID configuration*

---

```
Press <CTRL-C> to invoke SCSI configuration
Server should be factory configured with a RAID-1 mirror
```

---

---

*Example 16-4 RSA II configuration*

---

ASM Control  
Login Profiles  
    Create an additional Login ID with Supervisor Authority Level  
Network Interfaces  
    Interface: Enabled  
    DHCP: Disabled - Use Static IP configuration  
    Hostname: servernamersa  
    Enter Static IP Configuration  
Advanced Ethernet Setup  
    Data rate: 100 Mb (Ethernet)  
    Duplex: Full  
Network Protocols  
    Telnet Protocol  
        Telnet connection count: Disable  
Security  
    SSL Server Configuration for Web Server  
        SSL Server: Enabled  
        SSL Certificate Management  
            Generate a New Key and a Self signed Certificate  
            Enter the appropriate Certificate Data  
            <Generate Certificate>  
Restart ASM

---

## 16.3 Tuning activities

This section describes the various tuning tasks you can perform.

### 16.3.1 Tuning the VMware kernel

The VMware kernel has several tuning options that can impact your overall system performance significantly. In this section, we explore some of the important tuning parameters of the VMware ESX kernel.

#### Page sharing

VMware ESX features an algorithm to share equal memory pages across multiple virtual machines and thus reduce the total amount of used memory on the ESX Server system. Page sharing has little to no performance impact (about 2%) and might even speed up page lookups. The benefits of page sharing are largely workload-dependent.

## Setting network speeds

It is generally better to change all network interface cards that are used in the VMware ESX system from auto-negotiate to full duplex. The same applies to all involved hubs or switches because the auto-negotiate setting often results in less than optimal network speeds.

There are situations where the Broadcom Console OS driver set to auto-negotiate in many cases does not negotiate properly when connected to a 100 Mbps network. The network is generally still functional, however, the performance is only 1/100th of the expected bandwidth. Setting the correct network speed is important.

You can set the network speed of the network interface cards that are associated to the Console operating system in the /etc/modules.conf file as shown in Example 16-5.

*Example 16-5 Setting the network adapter speed in /etc/modules.conf*

---

```
alias parport_lowlevel parport_pc
alias scsi_hostadapter aic7xxx
alias eth0 e100 e100_speed_duplex=4
alias scsi_hostadapter ips
#alias eth1 eepro100
alias scsi_hostadapter1 aic7xxx
alias scsi_hostadapter2 aic7xxx
#alias usb-controller usb-ohci
alias scsi_hostadapter ips
alias scsi_hostadapter ips
```

---

The correct settings for your particular network interface card are displayed in the examples section at the end of the /etc/modules.conf file or in the readme file of your network interface cards.

You can also set the speed and duplex of the network cards that are associated to virtual machines in the VMware management interface. Log in to the VMware management interface as an Administrator in VI3 client and go to the configuration of the ESX server under Hardware & Networking, select properties of the specific virtual adapter and edit propertiesNetwork Connections menu in the options section. Figure 16-2 shows an example of setting the speed of the network interfaces.

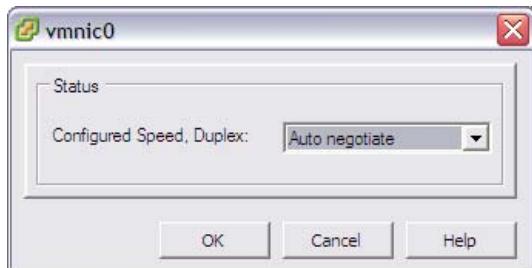


Figure 16-2 Setting the network speed in the VI3 management interface

### VMware kernel swapping

The swap mechanism of the VMware kernel allows you to run significantly more virtual machines on a single server. However, you will see an increase in disk I/O when the system starts to swap data out to the disk.

For optimal performance, monitor the swap file of the VMware kernel closely, and reduce the number of active virtual machines or install more memory as soon as the VMware kernel starts to swap data out to disk. The VMware swap file should ideally not be used by the system during normal operation. To minimize the impact if it is used, we suggest that you create a VMFS partition on your local disks and put the swap file there.

**Tip:** Monitor the data in the /proc/vmware/swap/stats file and make sure that the used swap value does not constantly exceed zero (0).

#### 16.3.2 Tuning the virtual machines

This section applies to most ESX versions. It lists the most common recommendations you should follow to help the ESX system avoid wasting key resources that could be useful at peak utilization.

- ▶ Use supported guests.
- ▶ Run one application per virtual machine.
- ▶ Install the latest version of VMware Tools in the guest operating system and ensure it stays updated after each VMware ESX upgrade.
- ▶ Installing VMware Tools updates the BusLogic driver within the guest operating system to the VMware supplied driver. The VMware driver has certain optimizations that guest-supplied drivers do not.
- ▶ The balloon driver used for memory reclamation on ESX is part of VMware Tools.
- ▶ Ballooning will not work if VMware Tools is not installed.

- ▶ Always use VMware Tools, the VI Client, or **esxtop** to measure resource utilization.
- ▶ CPU and memory usage reported in the guest can be different from what ESX reports.
- ▶ Disable screen savers and Window animations in the virtual machine. On Linux, if using an X server is not required, disable it.
- ▶ Screen savers, animations, and X servers all consume extra CPU, which can affect performance of other virtual machines and consolidation ratios. The non-trivial consumption of CPU by idling virtual machines can also have an adverse impact on Dynamic Resource Scheduling (DRS) decisions.
- ▶ Disconnect unused, unnecessary devices on both the guest and on the host:
  - COM ports
  - LPT ports
  - Floppy drives
  - CD-ROM drives
  - USB adapters
- ▶ Disabling USB and other devices on the host frees IRQ resources and eliminates IRQ sharing conflicts that can cause performance problems. Some devices, such as USB, also consume extra CPU because they operate on a polling scheme.
- ▶ Windows guests poll CD devices quite frequently. When multiple guests try to access the same physical CD drive, performance suffers. Disabling CD devices in virtual machines when they are not needed alleviates this problem.
- ▶ Schedule backups and anti-virus programs in the virtual machines to run at off-peak hours. In general, it is good practice to distribute CPU consumption evenly, not just across CPUs but also across time. For workloads such as backups and antivirus where the load is predictable, this is easily achieved by scheduling the jobs appropriately.
- ▶ Defragment the hard drives that run the guest operating systems.
- ▶ Do not run X Windows Systems on Linux virtual machines, if possible.
- ▶ Allocate sufficient key resources for a virtual machine's application. For example, dedicate a LUN to a database application.
- ▶ Configure the minimum CPU settings high enough to guarantee CPU resource
- ▶ Allocate for low-latency applications (as opposed to share-based CPU allocation). The trade-off is a lower consolidation ratio (that is, the number of virtual machines you can run on a server).

### **16.3.3 Tuning the VM memory allocation**

When you create a new virtual machine, you are required to select the amount of allocated memory. Similar to the physical memory that is installed in a dedicated server, an operating system running in a virtual machine cannot use memory that was previously allocated. Therefore, if you allocate a certain amount of memory to your virtual machine but the operating system and application running within demand more memory, swapping will occur.

As always, swapping caused by a lack of memory is mostly undesired because fast memory access times are replaced with relatively slow disk access times. Thus, we suggest that you size the memory available to the virtual machine according to the total needs of both the operating system and the application that is installed in the virtual machine.

VMware ESX offers advantages in sizing the memory allocation:

- ▶ You can always allocate more memory to a virtual machine than it will effectively use. Unused memory is shared among all other virtual machines.
- ▶ You can resize the allocated memory of a virtual machine easily. However, always keep in mind that a change in memory of a running virtual machine requires a reboot of that virtual machine.

### **16.3.4 Selecting the right SCSI driver**

You can create virtual machines with two different types of (virtual) disk controllers:

- ▶ The BusLogic driver is the default driver for Windows NT4 and Windows 2000. The LSI driver is the default for Windows 2003, which features compatibility with a broader range of operating systems.

The BusLogic driver that is shipped with all supported guest operating systems performs adequately with small files of up to 1 KB in size. If you have an application that makes use of such small files, the default BusLogic driver might be the best choice for you. Using the BusLogic driver that is supplied by the guest operating systems does not, however, yield the best possible performance.

VMware strongly recommends upgrading the operating system-provided driver with the version that is available for download from the VMware support site.

- ▶ The LSI Logic controller driver shows a significant performance improvement for larger files. However, this driver might not be available for all supported guest operating systems.

VMware provides disk images with the drivers for Linux and Windows operating systems so that you can install the virtual machine with the diskette drive.

### 16.3.5 Time synchronization

It is important that VMware ESX keep accurate time. To synchronize VMware ESX with an NTP server, follow the directions as outlined in VMware KB Answer ID# 1339:

[http://www.vmware.com/support/kb/enduser/std\\_adp.php?p\\_faqid=1339](http://www.vmware.com/support/kb/enduser/std_adp.php?p_faqid=1339)

VMware also recommends that you synchronize your virtual machine's time with the VMware ESX's time. This synchronization is a function of the VMware Tools that are installed in the virtual machines. For more detailed information about timekeeping, see the VMware white paper, *Timekeeping in VMware Virtual Machines*, which is available from:

[http://www.vmware.com/pdf/vmware\\_timekeeping.pdf](http://www.vmware.com/pdf/vmware_timekeeping.pdf)

## 16.4 VMware ESX 3.5 features and design

This chapter is not designed to replace the documentation already available from VMware and other sources. For detailed information about how to install and use VMware ESX 3.5, see the documentation that is provided by VMware at:

[http://www.vmware.com/support/pubs/esx\\_pubs.html](http://www.vmware.com/support/pubs/esx_pubs.html)

In discussing architecture and design, we assume that the environment consists of a minimum of two VMware ESX systems, shared SAN storage, VirtualCenter, and VMotion.

### 16.4.1 Overview of VMware ESX 3.5

VMware ESX 3.5 or later has the following specifications:

Physical VMware ESX:

- ▶ 32 logical processors per system
- ▶ 128 virtual CPUs in all virtual machines per VMware ESX system
- ▶ 256 GB of RAM per VMware ESX system
- ▶ 64 adapters of all types per system
- ▶ Up to 8 Gigabit Ethernet or 16 10/100 Ethernet ports per system
- ▶ Up to 32 virtual machines per virtual switch
- ▶ 16 host bus adapters per VMware ESX system
- ▶ 128 logical unit numbers (LUNs) per storage array

- ▶ 128 LUNs per VMware ESX system

VMware ESX 2.5 or later virtual machines:

- ▶ Up to 4 virtual CPUs per virtual machine
- ▶ Up to 64GB of RAM per virtual machine
- ▶ Up to four virtual SCSI adapters and up to 15 SCSI disks
- ▶ SCSI Virtual disk sizes up to 2 TB
- ▶ Up to 4 virtual Ethernet network adapters
- ▶ 800 MB RAM allocated to service console

For the latest list of supported guest operating systems and qualified hardware see the Systems Compatibility Guide, which is available at:

[http://www.vmware.com/vmtn/resources/esx\\_resources.html](http://www.vmware.com/vmtn/resources/esx_resources.html)

#### **16.4.2 Virtual Infrastructure 2.5 with VMware ESX 3.5 Update 4**

With VMware ESX 3.5 or later and VirtualCenter 2.5, virtual infrastructure consists of the following components:

- ▶ VMware ESX 3.5
- ▶ VirtualCenter 2.5
- ▶ vSMP
- ▶ VMware VMotion
- ▶ VMware HA (High Availability)
- ▶ VMware Storage VMotion
- ▶ VMware vCenter Update Manager
- ▶ VMware DRS (Disaster Recovery Services)

VMware ESX runs on a physical server. VirtualCenter can either run on a separate physical server or in a virtual machine. One thing to consider if you choose to run VirtualCenter in a VM is that if the parent VMware ESX system goes offline, you will not have access to VirtualCenter until the server is back online or until you restart the virtual machine on another host. vSMP and VMotion are features already installed and are unlocked with a license key.

#### **16.4.3 Number of servers and server sizing**

The number of servers that suit your needs depends on several factors, including:

- ▶ Scope of current project
- ▶ Future growth estimates
- ▶ High availability and disaster recovery plans
- ▶ Budgetary constraints

There are a number of different methods to use to calculate the number of ESX systems you will need. Here are two of the more popular methods.

## **Method 1**

The easiest rule of thumb is 4 to 5 virtual CPUs per physical CPU. This would result in 16 to 20 virtual machines per 4-way host or 32 to 40 per 8-way host, assuming that all were one vCPU virtual machine and had low-to-moderate workloads.

For memory, if you assume 1 GB per virtual machine, that should provide enough memory in most cases for virtual machines, the service console, and virtualization overhead. If you plan on running multiple, memory-intensive workloads, consider increasing this number.

From these calculations we can arrive at an 8-way (two-node) x3950 with 32 GB of RAM which could support 32 virtual machines, and a 16-way (4-node) x3950 with 64 GB of RAM, which could support 64 virtual machines.

These calculations assume single-core CPUs. Because a dual-core CPU will not provide 100% the performance of 2 single-core CPUs, we recommend that you count a dual-core CPU as 1.5 physical CPUs, resulting in 6 to 7 virtual machines per CPU socket.

## **Method 2**

If you are consolidating a number of existing physical servers, then another method that can be used is to record the average peak CPU utilization of the physical machines and convert this into a total of MHz used.

For example, if you have a physical server with two 500 MHz CPUs that have an average peak utilization of 50%, then your total would be 500 MHz of CPU for this system.

To get an average peak utilization, you must record CPU utilization for at least a week during the normal hours when the application is being used. A month is recommended for the most accurate information. If you already use an enterprise monitoring tool such as IBM Tivoli, HP OpenView, NetIQ, and so on, then you might already have all the data you need.

The next step is to add up the total CPU clock speeds of your VMware ESX system. For example, a two-node 8-way x3950 with 3 GHz CPUs would have a total of 24000 MHz.

From this total, subtract 10% for the Console operating system, which gives us 21600 MHz.

Subtract a certain amount for additional peak utilization and overhead; 20% is a safe number to use. This gives us 17280 MHz to work with for our virtual machines.

Divide that number against the 500 MHz of average peak utilization that we first determined. The yield is about 34 virtual machines ( $17\ 280/500 = 34.5$ ).

You can perform similar calculations to determine how much memory you need, as well. Take the average peak memory utilization of your physical servers, add 54 MB per system for virtualization overhead, and add 32 MB for any systems whose average peak is over 512 MB. This total is the amount of RAM that you need for your VMs.

Then, add the amount of RAM that is assigned to the Service Console (512 MB would be an appropriate starting number on an 8-way VMware ESX system), add 24 MB for the VMkernel, and this is the total amount of RAM needed.

For example, if you had 10 physical systems to virtualize and each had an average peak memory utilization of 512 MB, then that would equal 5120 MB. Add 54 MB each for virtualization overhead ( $5120+540 = 5660$  MB). This amount is the total amount of RAM for the VMs. Add 512 MB for the Service Console ( $5660+512 = 6172$  MB) and 24 MB for the VMkernel ( $6172+24 = 6196$ ) and this amount is the total amount of RAM that is needed to run these 10 VMs: 6 GB of RAM.

Both methods provide very similar results in the number of virtual machines you could support on an 8-way x3950 server. Our experience is that in most organizations, these two methods usually result in a similar number of virtual machines per host. Therefore, to save yourself some time, we recommend you use the first method for initial sizing of your ESX servers. The exact mix of virtual machines and applications running will affect how many virtual machines you can run. Unfortunately there is no single formula that will calculate exactly how many virtual machines you can run. The low end of the recommendations that we illustrated here should provide a realistic and conservative target for most organizations. In reality, you could end up supporting more or fewer virtual machines.

Future growth is harder to determine. The cycle that occurs in many organizations that implement VMware's Virtual Infrastructure model unfolds like this:

- ▶ At first, the organization is resistant to the idea of virtual machines.
- ▶ After seeing all the benefits of the virtual infrastructure and no loss of performance, the number of requests for new virtual machines can grow rapidly.

- ▶ The result can be an overcommitment of processor, memory, and I/O resources and a subsequent loss in overall performance.

To avoid this cycle, one recommendation is that you follow the same purchasing, approval, and change management procedures for virtual machines as you do for physical systems. Although the process can usually be streamlined and shortened for virtual machines, by having a formal process in place to request virtual machines, as well as a way to associate costs to each new virtual machine, you will have much better control over your virtual machine growth and a better idea of future growth.

For more information about best practices, see *VMware ESX Server: Scale Up or Scale Out?*, REDP-3953, which is available from:

<http://www.redbooks.ibm.com/abstracts/redp3953.html>

#### 16.4.4 VMotion considerations

When designing your virtual infrastructure, an important consideration is VMotion. VMotion is the feature that allows the migration of a virtual machine from one physical VMware ESX system to another while the virtual machine is running.

Because VMotion transfers the running architecture state of a virtual machine between two physical hosts, the CPUs of both physical hosts must be able to execute the same instructions.

At a bare minimum, for VMotion to work, your servers' CPUs must be:

- ▶ The same vendor class (Intel or AMD)
- ▶ The same processor family (Pentium III, Pentium 4, Opteron, and so forth)

Sometimes there are significant changes to processors in the same family that have different extended features, such as 64-bit extensions and SSE3. In these cases VMotion might not work, even though the CPUs are in the same processor family. CPU speed and cache level are not an issue, but the extended features will cause problems or VMotion failure if they are different on the target and host servers.

For example, because the x366 and x260 use the same processors as the x3950, these servers would be suitable candidates for joining some x3950s in a VMotion configuration. However, other System x servers with different processors will not.

Another important requirement for VMotion is shared storage. The VMware ESX systems across which you are going to run VMotion need to be zoned so that all LUNs are visible to all hosts.

## 16.4.5 Planning your server farm

With VirtualCenter 2.5 and later, a *farm* is a group of VMware ESX systems that can be used to organize your virtual infrastructure. A farm is also a VMotion boundary, meaning that all servers in a VMotion configuration must be defined in one farm. In your planning, you need to think about how many hosts are going to be in each farm.

VMware recommends the following guidelines:

- ▶ No more than 16 VMware ESX systems should be connected to a single VMFS volume.
- ▶ No more than 32 I/O-intensive virtual machines per LUN and no more than 100 low-I/O virtual machines per LUN.
- ▶ No more than 255 files per VMFS volume.
- ▶ Up to 2 TB limit on storage.

Because VMotion requires shared storage, then the upper limit per farm would be 16 VMware ESX systems per farm. You might want to create smaller farms for a number of reasons. The lower limit is two servers, assuming that you are using VMotion.

## 16.4.6 Storage sizing

Similar to server sizing, for storage sizing there is no single, universal answer that can be applied to every organization. There should not be more than 32 I/O-intensive virtual machines per VM file system volume, and staying within this limit should reduce any resource contention or SCSI locking issues.

There are a number of ways to determine the most appropriate size of your VM file system volume. Here is one of the easier ways. Assume that you have decided that two 8-way x3950 servers with 32 virtual machines on each server will meet your processing requirements. Using the 32 virtual machines per LUN guideline, you would need two LUNs for this configuration. If you create new virtual machines, you can estimate the average size of the virtual disks.

If we use 20 GB of disk per VM, this would give us 640 GB per LUN. Consider adding a little additional space for growth (10% is a good rule of thumb), which brings us to 720 GB. If you are planning on using redo logs, you might want to add additional space for that, as well.

## 16.4.7 Planning for networking

There are various options when it comes to designing the networking portion of your server farm. The options chosen are often based on the characteristics of your physical network and on the networking and security policies of your company. One important factor is whether a Gigabit Ethernet network is available. Although not absolutely required for VMware ESX, using a Gigabit network is highly recommended.

In VMware ESX 3.5, there are three basic components you should consider:

- ▶ Service console

It is recommended that the service console have its own dedicated NIC for performance and security reasons. If you have a separate management network in your data center, then this is where you want to locate the service console NIC.

In a default configuration, a 100 Mbps Ethernet controller is sufficient bandwidth for the service console. If you are planning on also using the service console for backups or other high bandwidth functions, then a Gigabit NIC is recommended.

- ▶ Virtual machines

The virtual machines use a separate network from the service console. A Gigabit network is not required but is highly recommended, because 32 virtual machines can generate significant network traffic.

A good rule of thumb is 10 to 20 virtual machines per Gigabit Ethernet controller. This means that we need a minimum of 2 Gigabit Ethernet NICs for an 8-way x3950 running 32 VMs. Remember that this is the minimum recommendation. Adding one or two more virtual machines should guarantee enough network bandwidth available for all virtual machines.

Another important consideration is if you have multiple VLANs in your data center that you want to make available to the virtual machines. When using multiple VLANs with VMware ESX 3.5 you have two options:

- Install a physically separate NIC for every network that you want available. If you only have a few networks that you want to use, then this is a viable option. However, if you have 10 different networks, then this is obviously not a practical solution. Remember that VMware ESX 2.5 or later only supports a maximum of 8 GB network cards.
- Use VMware ESX's support for VLAN tagging (802.1q). Using this option means that you can create a virtual switch with a separate port group for each VLAN that you want to use. If your physical switches support this, then this is the recommended option.

Another consideration is redundancy for your virtual machine's networking. With VMware ESX 3.5 or later, you can have multiple NICs that are connected to one virtual switch not only to combine bandwidth but also to provide redundancy in case of a failure of one of the NICs or a physical cable.

- ▶ VMotion

VMware lists a separate Gigabit Ethernet network as a requirement for VMotion. It is possible to use VMotion with a 100 Mbps network. However, performance might not be acceptable and it is not recommended. You should have a separate physical gigabit NIC for your VMotion network and a separate subnet created for VMotion to use.

If you only have two systems running VMware ESX, then it is possible to use a crossover cable between the two servers for a VMotion network. This is also useful for troubleshooting VMotion problems.

#### 16.4.8 Network load balancing

VMware ESX 3.5 provides two methods for network load balancing for the virtual machines:

- ▶ MAC Out is the default method. Using this method requires no additional configuration in VMware ESX. Simply connect two physical NICs to the virtual switch. No additional configuration on the physical switches is necessary. The only disadvantage of this method is that it is not very efficient. Often, most virtual machines end up using the same physical NIC and there is no method to select manually what physical NIC each virtual machine uses.
- ▶ IP Out is an optional way to configure your networking for better load balancing. The disadvantage to this method is that there are additional configuration steps required in VMware ESX, as well as your physical switches. You must configure your physical switches for 802.3ad (or EtherChannel in the case of Cisco switches). This is the recommended method for highest performance.

This is a brief overview of networking with VMware ESX 3.5. Advanced topics such as backup networks, DMZ networks, traffic shaping, and detailed configuration steps are beyond the scope of this book.

For in-depth information about networking and configuration steps, see the documentation that is available on the VMware Web site:

[http://www.vmware.com/vmtn/resources/esx\\_resources.html](http://www.vmware.com/vmtn/resources/esx_resources.html)

# Monitoring tools

In this part, we introduce the performance monitoring tools that are available to users of System x servers. We describe the tools that are specific to each of the three operating systems as well as Capacity Manager, which is a component of IBM Director. We also provide detailed instructions that show how to use these tools.

This part includes the following chapters:

- ▶ Chapter 17, “Windows tools” on page 533
- ▶ Chapter 18, “Linux tools” on page 607
- ▶ Chapter 19, “VMware ESX tools” on page 649





# Windows tools

This chapter introduces some of the tools that are available to an administrator of a Windows Server 2008<sup>1</sup> server for performance tuning. The tools that we discuss in this chapter include:

- ▶ 17.1, “Reliability and Performance Monitor console” on page 534
- ▶ 17.2, “Task Manager” on page 573
- ▶ 17.3, “Network Monitor” on page 580
- ▶ 17.4, “Other Windows tools” on page 588
- ▶ 17.5, “Windows Management Instrumentation” on page 593
- ▶ 17.6, “VTune” on page 600

How you use these tools depends on your performance tuning needs. Each tool provides its own unique capabilities, and each tool has its advantages and disadvantages. In Chapter 21, “Analyzing bottlenecks for servers running Windows” on page 691, we use the information in this chapter to explain how to detect bottlenecks.

---

<sup>1</sup> Product screen captures and content reprinted with permission from Microsoft Corporation.

## 17.1 Reliability and Performance Monitor console

The Performance console is a valuable monitoring tool that Windows administrators commonly use to monitor server performance and to isolate bottlenecks. The tool provides real-time information about server subsystem performance, and the data collection interval can be adjusted based on your requirements. The logging feature of the Performance console makes it possible to store, append, chart, export, and analyze data captured over time. Products such as SQL Server provide additional monitors that allow the Performance console to extend its usefulness beyond the operating system level.

The home page of Windows Reliability and Performance Monitor is the new Resource View screen, which provides a real-time graphical overview of CPU, disk, network, and memory usage as shown in Figure 17-1.

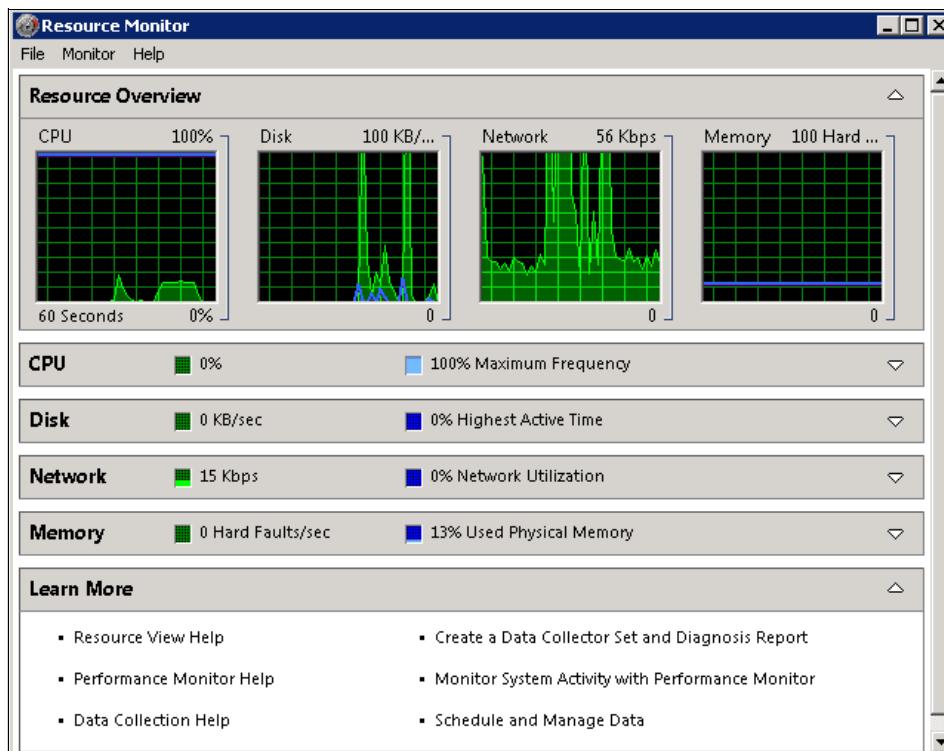


Figure 17-1 Resource Monitor Console

By expanding each of these monitored elements, system administrators can identify which processes are using which resources. In previous versions of Windows, this real-time process-specific data was only available in limited form

in Task Manager. Refer to 17.2, “Task Manager” on page 573 for more information about Task Manager.

Windows Server 2008 introduces also a Reliability Monitor console. It provides a system stability overview and details about events that impact reliability. It calculates the Stability Index shown in the System Stability Chart over the lifetime of the system. In this chapter, we will only describe the performance tools.

If you want to learn more about the reliability tools, review *Performance and Reliability Monitoring Step-by-Step Guide for Windows Server 2008*, available from:

<http://technet.microsoft.com/en-us/library/cc771692.aspx>

### 17.1.1 Overview of the Performance console window

The Performance console includes three tools:

- ▶ Monitoring tools
- ▶ Data Collector Sets
- ▶ Reports

Figure 17-2 on page 536 shows the main Performance console window.

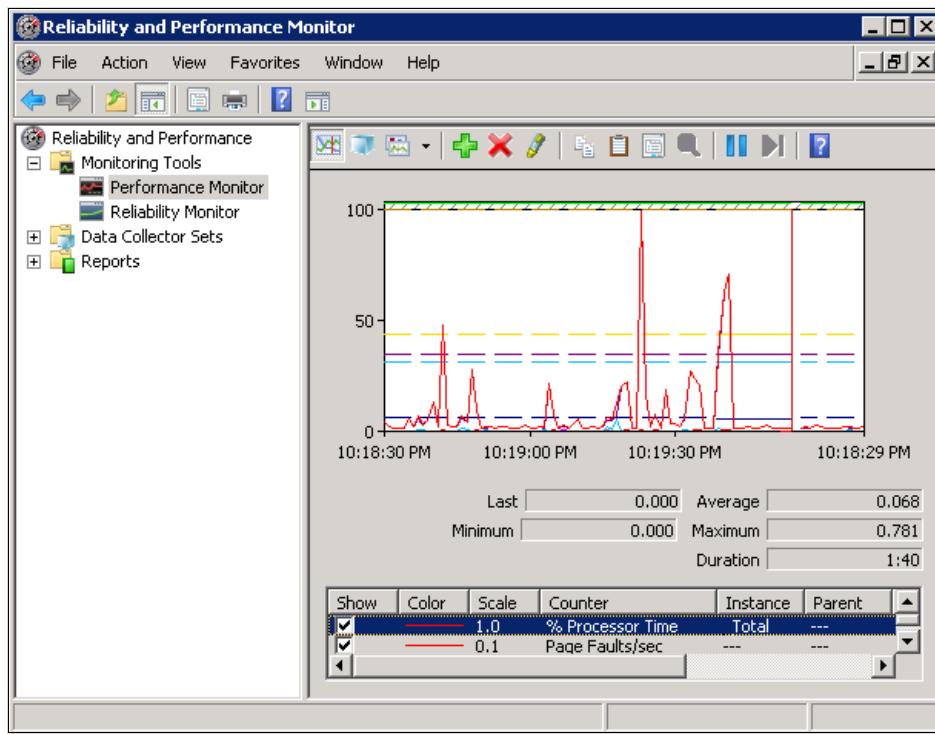


Figure 17-2 The Windows Server 2008 Performance console

The Performance console is a snap-in for Microsoft Management Console (MMC). You can use the Performance console to access the Monitoring tools, Data Collector Sets, and Reports tools.

You open the Performance console by clicking **Start** → **Administrative Tools** → **Reliability and Performance Monitor** or by typing **PERFMON** on the command line. Then, you need to select **Performance Monitor** in the left column.

**Tip:** If there is no Administrative Tools folder, you can display it as follows:

1. Right-click **Start** and click **Properties**.
2. At the Taskbar and Start Menu properties dialog box, click **Customize**.
3. Scroll down the Start menu items list until you find the System Administrative Tools section.
4. Select the Display on the All Programs menu and the Start menu option.
5. Click **OK** to close the Customize Start Menu dialog box.
6. Click **OK** to close the Taskbar and Start Menu properties dialog box.

When starting the Performance console on a Windows Server 2008, the Performance Monitor runs automatically. The default monitor is

- ▶ Processor: % Processor Time

You can use the Performance Monitor to view real-time or logged data of objects and counters. You can use Data Collector Set to log object and counters and to create alerts.

Displaying the real-time data of objects and counters is sometimes not enough to identify server performance. Logged data can provide a better understanding of the server performance.

You can configure alerts to notify the user or to write the condition to the system event log based on thresholds.

## Monitor Tools

Figure 17-2 on page 536 shows the Windows Server 2008 Performance Monitor interface.

There are three ways to view the real-time or logged data counters:

- ▶ Line

This view displays performance counters in response to real-time changes, or processes logged data to build a performance graph.

- ▶ Histogram bar

This view displays bar graphics for performance counters in response to real-time changes or logged performance data. It is useful for displaying peak values of the counters.

- ▶ Report

This view displays only numeric values of objects or counters. You can use it to display real-time activity or logged data results. It is useful for displaying many counters.

To edit the view, right-click in the main window of the Performance Monitor and select **Properties**. On the Graph tab, you can change the view.

We discuss Performance Monitor in detail in 17.1.2, “Using Performance Monitor” on page 541.

## Data Collector Sets

The data Collection Sets window (Figure 17-3 on page 538) lets you collect performance data manually or automatically from local or remote systems. It includes default Data Collector Set templates to help system administrators

begin collecting performance data specific to a server role or monitoring scenario immediately. You can display saved data in Performance Monitor or export data to a spreadsheet or database. It can be associated with rules of scheduling for data collection at specific times.

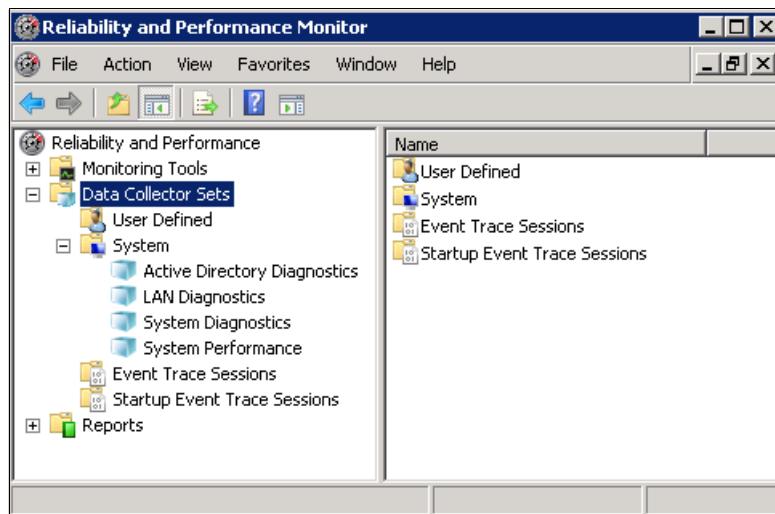


Figure 17-3 The Performance console: Data Collector Sets

Data Collector Sets can contain the following types of data collectors:

► User Defined

This function lets you create your own Data Collector using a wizard. There are two choices:

- You can use several included templates that focus on general system diagnosis information or collect performance data specific to server roles or applications.
- You can build a custom combination of Data Collectors. These Data Collectors can include Performance Counters, Configuration data, or data from Trace Providers.

► System

This function lets you use predefined templates already configured:

- Active Directory Diagnostics
- Lan Diagnostics
- System Diagnostics
- System Performance

- ▶ Event trace data

Event trace data is collected from trace providers, which are components of the operating system or of individual applications that report actions or events.

- Event Trace Sessions shows the trace sessions running currently on the system
- Startup Event Trace Sessions allows you to start or stop trace sessions. From this place, you can also create new trace sessions.

See 17.1.3, “Using Data Collector Sets” on page 546 for more information.

## Objects, counters, and instances

An *object* in Performance Monitor is any component that generates performance data. There are many objects built into Windows Server 2008. Each hardware component in your system is an object: processor, memory, hard drives, network cards, and other components in your machine. Objects are not only hardware components but also software components. Terminal services, Routing and Remote Access services, database server, and e-mail server applications that are installed on your system can have objects in Performance Monitor.

Each object has one or more *counters*. For example, the processor object has, among others, the following counters:

- ▶ %Processor Time
- ▶ %User Time
- ▶ Interrupts/sec
- ▶ %Interrupt Time

Each counter can have multiple *instances*, which means there can be more than one of the same counter for that object. For example, in a multi-homed server, there will be multiple instances of network adapters, as illustrated in Figure 17-4 on page 540.

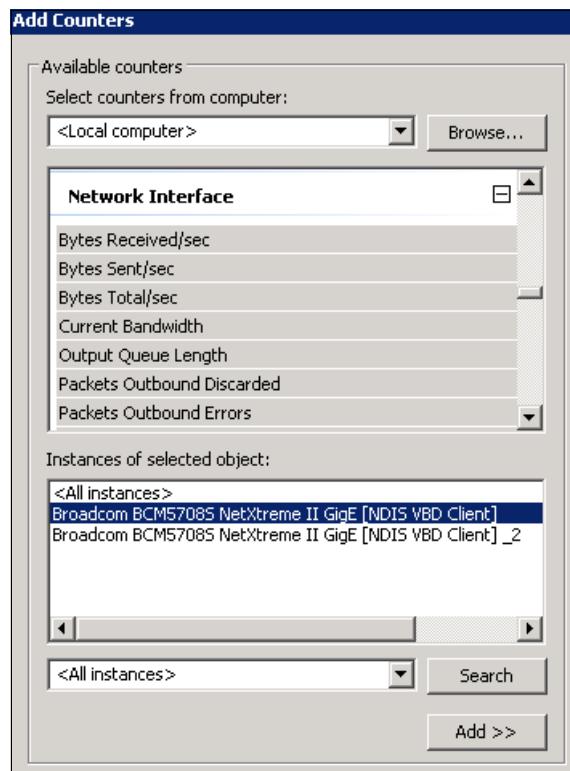


Figure 17-4 Objects, counters and instances

In summary, the object is the hardware or software component in your machine, the counter is the value of a specified object that can be measured by Performance Monitor, and the instance identifies all or each of the members of the object.

## 17.1.2 Using Performance Monitor

To create a chart in Performance Monitor, you select the performance objects and configure the view. When you select a system object for display, the values of the specified counter are put into a chart in graphical format, as shown in Figure 17-5.

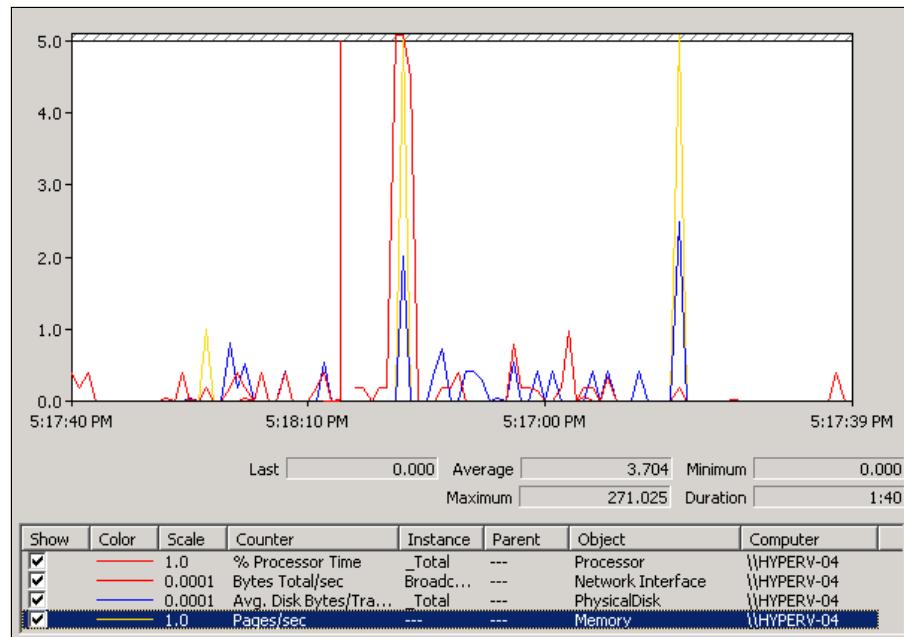


Figure 17-5 Multi-instance object chart view

Table 17-1 explains the values that are included in this chart view.

Table 17-1 Chart view values

Value	Description
Last	The latest value of the selected counter
Average	The average value of the selected counter
Minimum	The minimum value of the selected counter
Maximum	The maximum value of the selected counter
Duration	The period of time that you measure
Color	The selected color for the counter
Scale	The multiplier that you use to calculate the graphical value from the actual value
Counter	The performance values of the selected object

Value	Description
Instance	The member of the selected object
Parent	The upper level object of the selected object
Object	Hardware or software component
Computer	The name of the computer where you get the object

Figure 17-6 shows the Performance Monitor toolbar.



Figure 17-6 Performance Monitor toolbar

Table 17-2 describes the options that are available from the Performance Monitor toolbar.

Table 17-2 Performance Monitor toolbar icons

Button	Function	Description
	View Current Activity	View the current activity
	View Log Data	Display activity from log file or database
	Change Graph type	Select to display performance data using line graphic, histogram bar, or report
	Add counter	Add a new counter to Performance Monitor
	Remove counter	Remove selected counter from the counter list
	Highlight	Highlight selected counter
	Copy properties	Copy all graph properties and counter list to new windows
	Paste counter list	Paste all of the copied counter list to a new window
	Properties	Change Performance Monitor properties
	Zoom	Zoom on a specific period (available for log files)
	Freeze display	Pause display in Performance Monitor

Button	Function	Description
	Unfreeze display	Active display in Performance Monitor
	Update data	Update data instantly
	Help	Help for active performance function

The difference between the chart view and the histogram view (shown in Figure 17-7) is that the chart view shows the current and historical values in a line graph. The histogram view shows only the current values in a bar chart.

The histogram view is useful for displaying or tracking values of the selected counters when you are only interested in the current value (that is, when you are not interested in what value the counter showed the last time it was sampled).

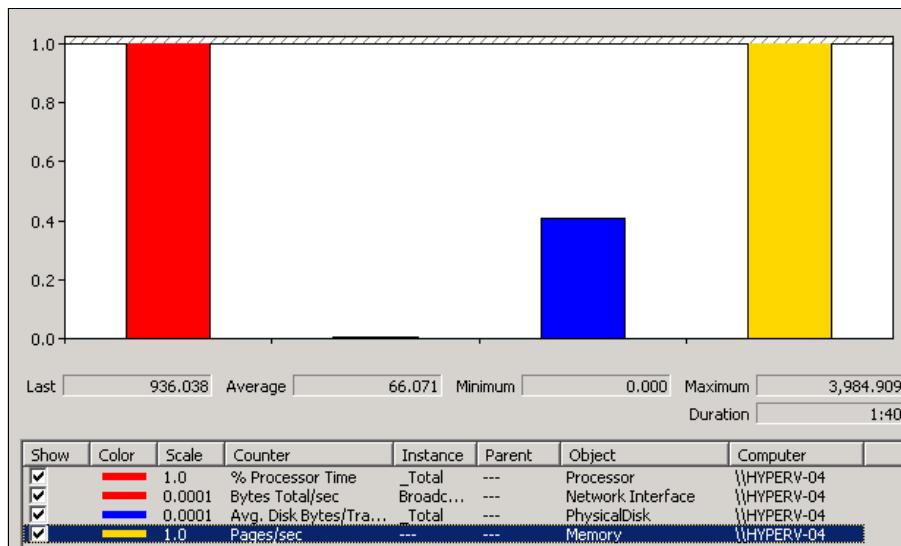


Figure 17-7 Performance Monitor histogram view

The difference between the chart view and the report view is that the report view uses only text and numbers. If you are tracking many values and you need to reduce system overhead, use the report view.

**Tip:** The report view gives a quick overview of the status of a server. This view is especially useful when you are performing an initial analysis of a poorly performing system and you want to determine quickly which counters are outside the range of good values. For more information, see Chapter 20, “Spotting a bottleneck” on page 663.

You can change the information that the report displays using the General tab of the Performance Monitor properties. You can choose how each counter is displayed (for example, real time, average, minimum, or maximum).

\\\HYPERV-04	
Memory	
Pages/sec	0.000
Network Interface	Broadcom BCM5708S NetXtreme II GigE [NDIS VBD Client]
Bytes Total/sec	0.000
PhysicalDisk	
Avg. Disk Bytes/Transfer	_Total 0.000
Processor	
% Processor Time	_Total 0.195

Figure 17-8 Performance Monitor, report view

## Adding counters

To create a chart that includes the objects that you want to monitor, follow these steps:

1. Click the **Add counter** icon (  ) on the toolbar or right-click the **Performance Monitor** and select **Add Counters** to open an Add Counters dialog box, as shown in Figure 17-9 on page 545. From here, you can select the performance object, counters, and instances that you want to monitor.

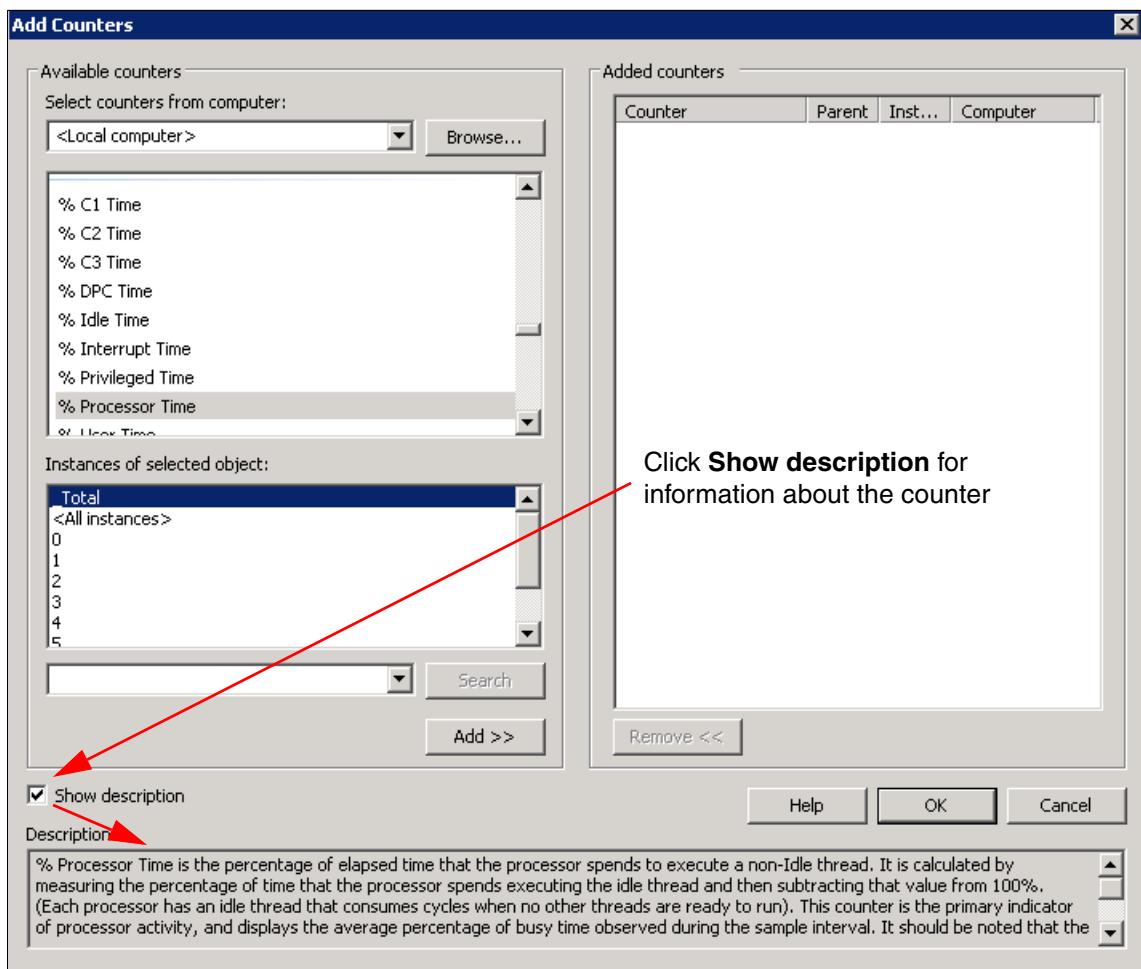


Figure 17-9 Add Counters window with the Explain window

2. Select the computer name that you want to monitor (local or remote).
3. Select the Performance object that you want to add.
4. Select the specific counters that you want to monitor or click **All counters** to select all the counters for the selected object.
5. Select the specific instances that you want to monitor or click **All instances**. Selecting **Total** shows the total amount of activity that is generated by all instances.
6. Click **Add**.

7. Repeat steps 3 on page 545 to 6 on page 545 until you have selected all the performance objects, counters, and instances in which you are interested, and then click **Close**.

## Deleting objects

If you no longer want to monitor a counter, you can delete it as follows:

1. Select the counter from the counter list at the bottom of the main menu. You can only select the counter while you are in the chart or histogram view.
2. Click the **Delete** icon (  ) in the toolbar, or press Delete on your keyboard.

**Note:** To clear all the counters from the counter list, right-click the **Performance Monitor** and select **Remove all Counters**.

To clear the Performance Monitor chart samples, right-click the **Performance Monitor** and select **Clear**.

## Saving counter settings

You can save the chart as a binary log file (BLG), a comma-delimited file (CSV) that you can export to a spreadsheet, and a report (.TSV) that you can export to a spreadsheet or word processor.

To save the report to file, right-click the window and click **Save Data As...**. Specify the file type, location, and file name.

## Highlighting an object counter

If you are working with multiple objects and counters in a graph, sometimes it is hard to differentiate or focus on a particular counter, especially if you are using the chart view. To highlight a particular counter:

1. Select the counter from the counter legend list.
2. Click the **Highlight** icon (  ) in the toolbar or press Ctrl+H.

### 17.1.3 Using Data Collector Sets

Data Collector Sets are useful for capturing and storing data to disk for analysis at a later time. You can also collect data from multiple systems into a single Data Collector Set. You can collect different counters, or counters of the same type, from multiple machines. You can use the data for comparative analysis for different machines with the same counters, or for analysis of only one machine with its counters. Creating a log consists of selecting performance counters, event trace data and system configuration information and starting and

scheduling the data logs capture. You can load this data back into Performance Monitor for analysis.

If you are using many counters and the sample frequency is too small, the log file requires a large disk space. If you are collecting multiple counters from a remote machine, this time this process might affect your network performance.

You can collect data from remote machines in two ways:

- ▶ Collect all the data from remote machines using one workstation (with Windows Vista at least). This is the easiest way to collect the remote data, but it can affect network performance.
- ▶ Configure all the remote machines to store data on their own local disks and collect it through batch files or scripts.

Performance counter alerts let you track counters to ensure that they are within a specified range. If the counter's value is below or above the specified value, an alert is issued. Actions from an alert include:

- ▶ Sending the alert to another machine
- ▶ Logging the alert in the event log
- ▶ Starting a new counter log
- ▶ Running a command from the command line

## Counter logs

There are three types of logs:

- ▶ Performance Counters

You can create a log file with specific counters and their instances. Log files can be saved in different formats (file name + file number or file name + file creation date) for use in Performance Monitor or for exporting to database or spreadsheet applications. You can schedule the logging of data, or you can start the counter log manually using program shortcuts. You can also save counter logs settings in HTML format for use in a browser, either locally or remotely through TCP/IP.

- ▶ Event trace sessions

You can create event trace logs that include trace data provider objects. Event trace data differs from performance counters in that data is measured continuously rather than at specific intervals. We discuss Event trace sessions in “Event trace sessions” on page 564.

- ▶ System configuration information

It allows you to record the state of, and changes to, registry keys.

## Toolbar

Figure 17-10 illustrates the toolbar icons for working with logs.

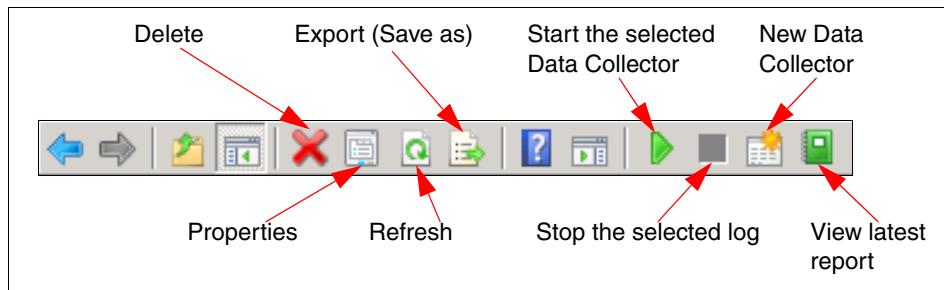


Figure 17-10 Data Collector Set: counter logs toolbar

## Creating a new User Defined Data Collector set

To create a new user defined data collector set:

1. From the Reliability and Performance console, select **Data Collector Set**.
2. From Data Collector Set, select **User Defined**.
3. Click the **Create a new Data Collection Set** icon (  ) from the toolbar, as shown in Figure 17-11.

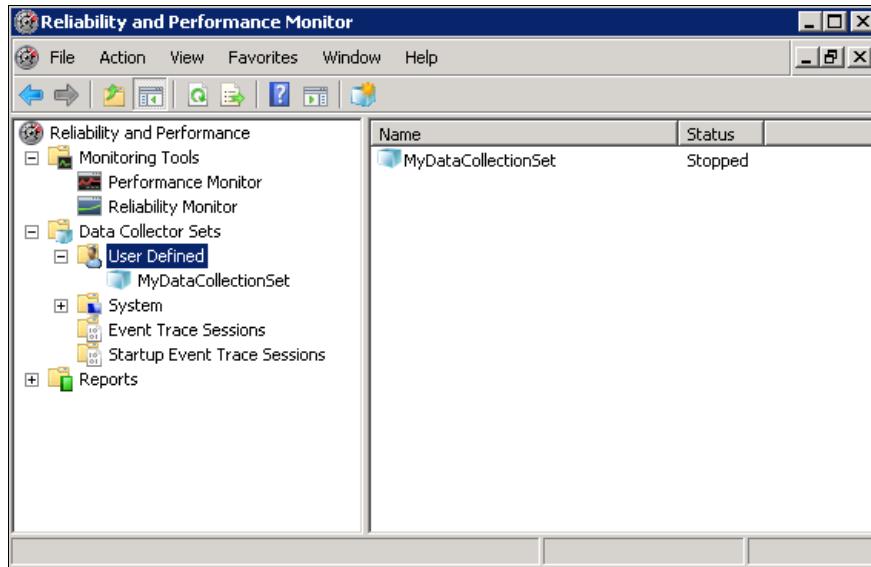


Figure 17-11 User Defined console

4. The New Data Collection Set wizard window opens, as shown in Figure 17-12.
5. Enter a new name for your User Defined Data Collector Set.

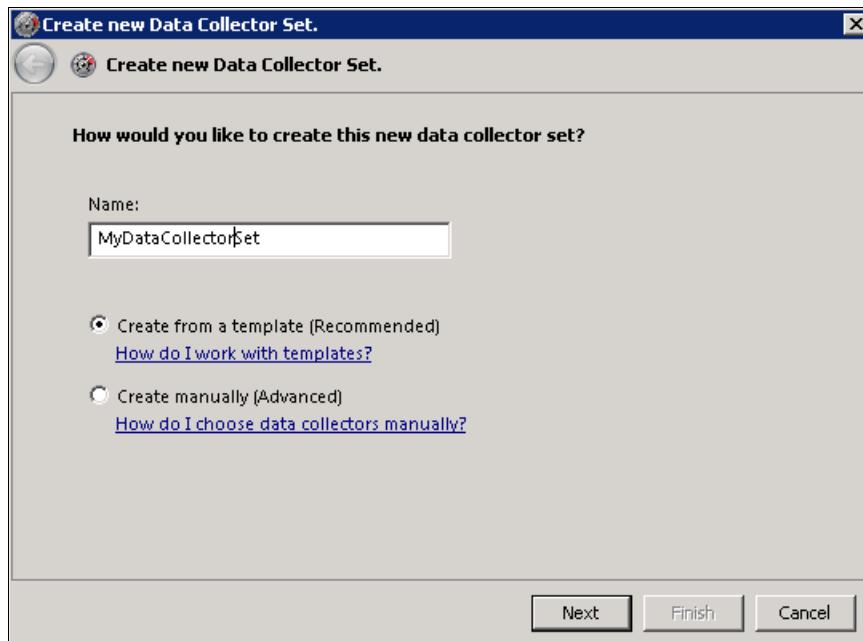


Figure 17-12 New Data Collection Set wizard, First choice

6. Now, you have the choice of creating the Data Collector set using two ways:
  - The simplest way to create a new Data Collector Set is by using the wizard in Windows Reliability and Performance Monitor.

Windows Server 2008 includes several templates that focus on general system diagnosis information or collect performance data specific to server roles or applications as shown on Figure 17-13 on page 550. In addition, you can import templates created on other computers and save the Data Collector Set you create for use elsewhere.

After you have selected your template, the wizard will be closed. If you choose this option, you can directly go to step 10 on page 554.

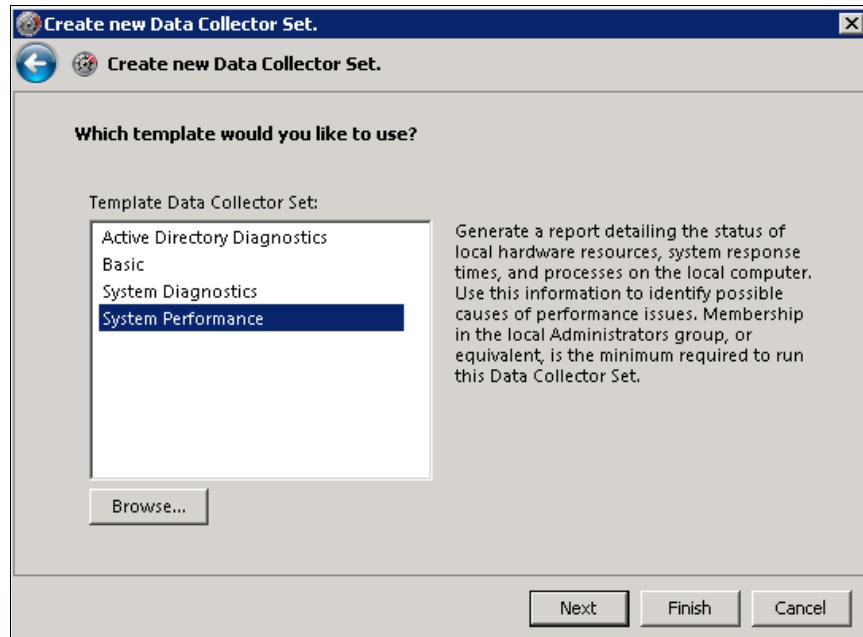


Figure 17-13 New Data Collection Set wizard: using template

- The second way is to build a Data Collector Set from a custom combination of Data Collectors as shown on Figure 17-14 on page 551. These Data Collectors can include Performance Counters, Configuration data, or data from Trace Providers. Select the ones you want to include in your Data Collector Set.

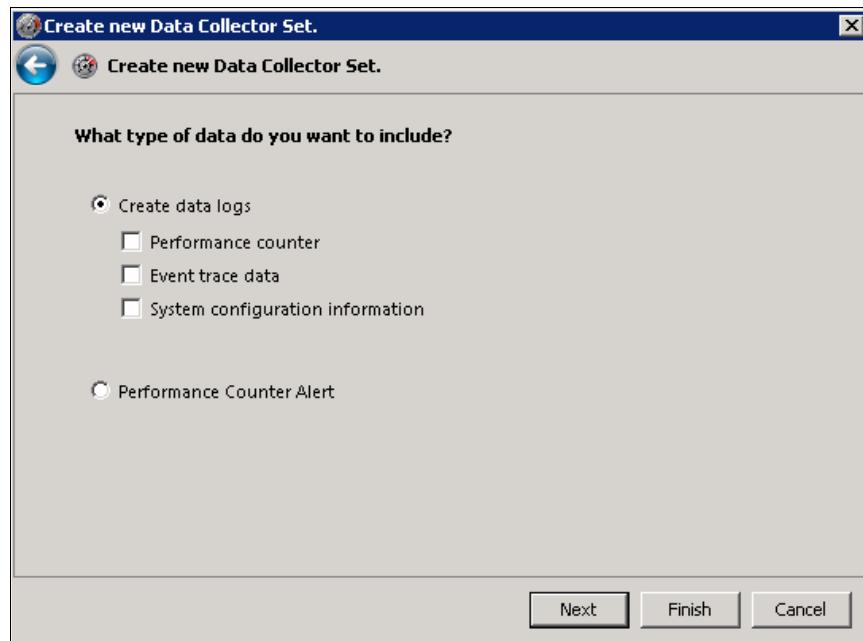


Figure 17-14 New Data Collection Set wizard: Using custom combination

7. If you have selected **Performance counter**, you have now the choice of adding providers to the set. Click **Add...** to add individual providers. You can select the computer to monitor and then select the relevant counters that you want to capture. You also must select the Sample interval for those counters.

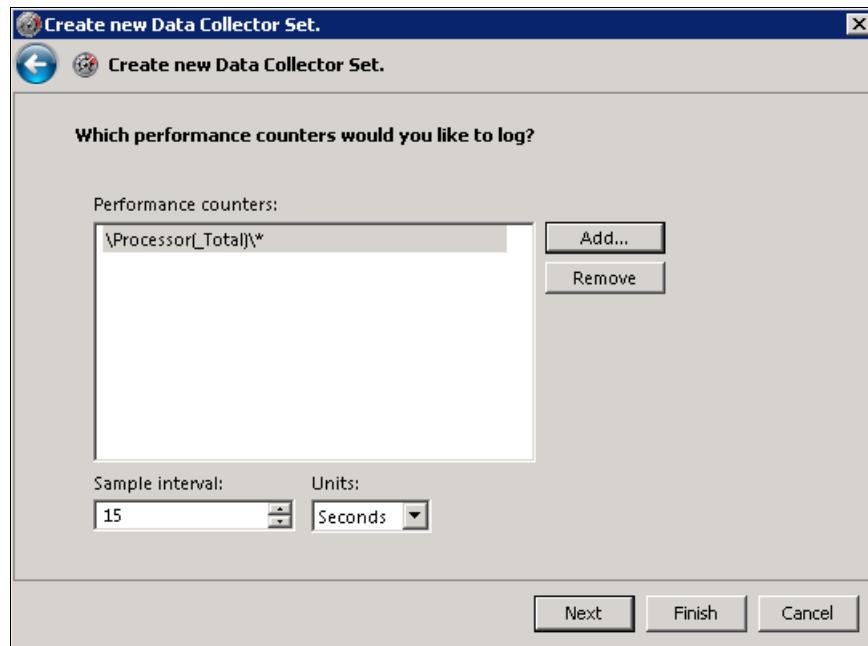


Figure 17-15 New Data Collection Set wizard: Performance counter

8. If you selected **Event trace providers**, you may now add traces to the set. Click **Add...** to add individual counters as shown on Figure 17-16 on page 553. You can also edit the options for each of those properties.

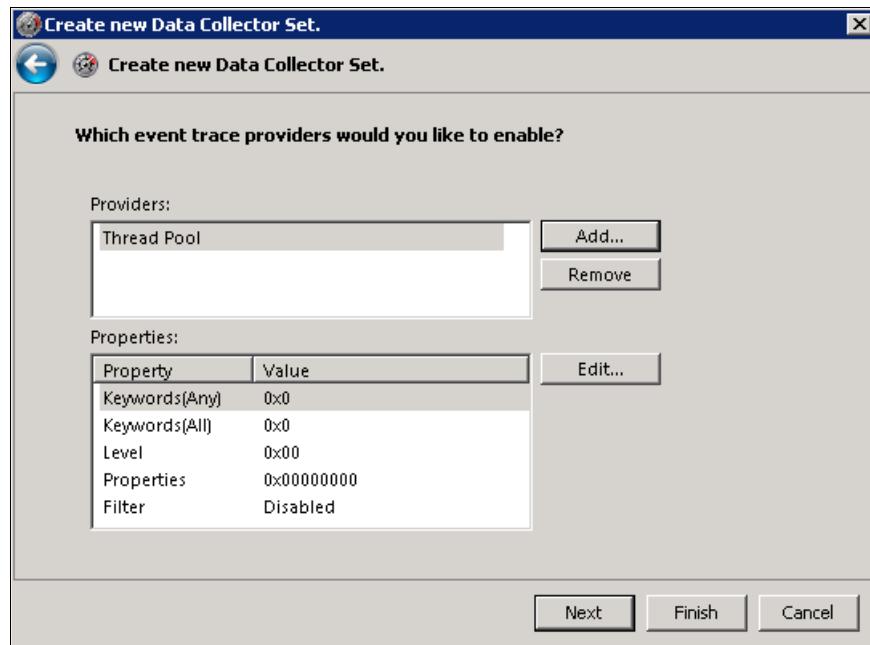


Figure 17-16 New Data Collection Set wizard: Event trace data

9. If you selected **System configuration information**, you may now add registry keys to the set as shown on Figure 17-17 on page 554. Click **Add...** to add a specific key.

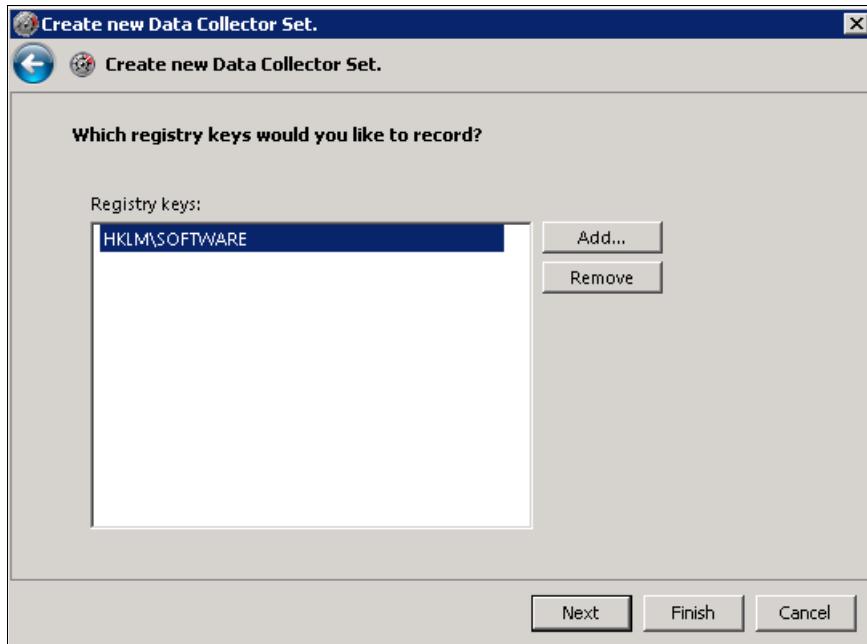


Figure 17-17 New Data Collection Set wizard: System configuration information

10. On the next window, you must choose where you want to save the data. Select a Root directory using the **Browse...** button. A good option is to use the default directory.
11. On the last window of the wizard, you must specify the account with sufficient rights to collect the information about the server to be monitored.  
You must also select whether you want to **open properties for this data collector set** (see “Data Collector Set properties” for more information on properties; or **start this data collector set now**; or **save and close it**).

### **Data Collector Set properties**

To access the Properties dialog of a given data collector set, right-click it and select **Properties**. The tabs in the dialog are as follows:

- ▶ The General tab allows you to write a description for this data collector set and add keywords or the account with sufficient rights to collect the information about the server to be monitored. If you created the Data Collector Set from a template that had a description and keywords, they will be included here.
- ▶ The Directory tab allows you to choose the directory and its name where the data will be collected.

- ▶ The Security tab allows you to change the permissions of groups or users for the Data Collector Set by selecting the group or user name and then selecting the allow or deny check boxes for each permission type. To add, remove, or change permission types, click the **Advanced** button.
- ▶ The Schedule tab allows you to specify when this log is started and an expiration date for it. You can also select the days you want this data collector set to run. You can also launch this set manually, see “Starting and stopping a Data Collector Set manually”.
- ▶ The Stop Condition tab allows you to specify when you want to stop the data collection set. You can specify an overall duration (in seconds, minutes, hours, days or weeks) and a limit. When a limit is reached, it will restart the data collector set.
- ▶ The Task tab allows you to run a Windows Management Instrumentation (WMI) task upon completion of the Data Collector Set collection by entering the command in the box “Run this task when the data collector set stops.”

### ***Starting and stopping a Data Collector Set manually***

When creating a Data Collector Set, you can schedule the start and stop time, or you can specify whether to start or stop the log manually. To start and stop the counter log manually, do the following:

1. Select the counter log that you want to start.
2. Click the **Start the Data Collector Set** icon (  ) on the toolbar.
3. To stop the counter log, click the **Stop the Data Collector Set** icon (  ) on the toolbar.

**Tip:** We recommend that you configure the Schedule tab to both start and stop any counter logs so that you can eliminate the problem of filling up a hard drive if you forget to stop the log manually.

You can also use the menu to start and stop the logs, as shown in Figure 17-18.

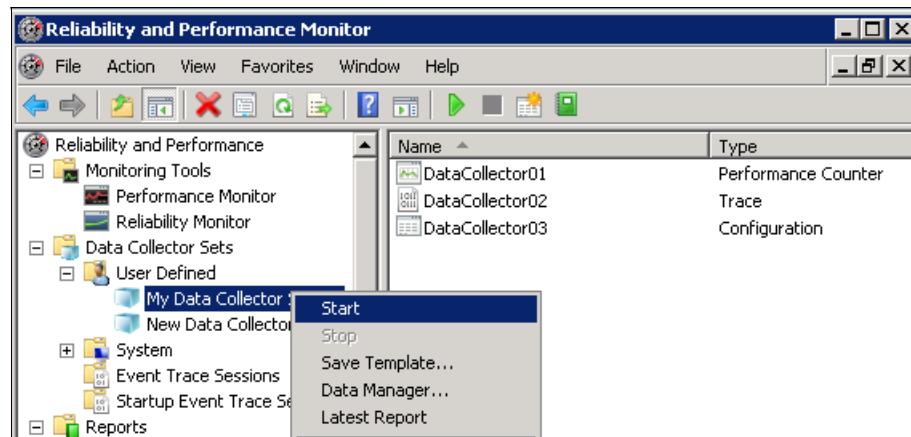


Figure 17-18 Data collector set menu

### **Saving the Data Collector Set settings**

You can save the Data Collector Set settings to use them later. To save log settings, do the following:

1. Select the Data Collector Set in which you want to save settings.
2. Right-click the set. The window shown in Figure 17-18 opens.
3. Click **Save Template...**
4. Select a location and enter a file name, and then click **Save** (saving to an XML file is the only option).

You can then open this log settings file using Internet Explorer or other XML reader.

### **Deleting a Data Collector Set**

If you no longer want to use a Data Collector log, you can delete it as follows:

1. Select the set that you want to delete.
2. Click the **Delete** icon ( ) in the toolbar or press Delete on your keyboard.

### **Importing Data Collector Set properties**

You can import Data Collector Set settings from saved files. To import settings, do the following:

1. Right-click the right-hand window.
2. Select **New Data Collector Set**.

3. The Open dialog window opens. Choose a name and select to create from a template, press next. Then select **Browse...** (see Figure 17-13 on page 550), choose the location and select a file name, and then click **Open**.
4. If you want to change the data collection set setting, you can do so. Otherwise, click **Finish**.

### ***Retrieving data from a counter log file***

After you have stopped a Data Collector Set, you can retrieve that data in the Reports portion of the Reliability and Performance Monitor console. Data Collector Set templates can also include report templates to help you interpret the performance information collected when they run.

Report generation time has improved since Windows Server 2003 and reports can be created from data collected with any Data Collector Set. This enables system administrators to repeat reports and assess how changes have affected performance or the report recommendations. To see a report for a specific Data Collection set, go to **Reports** then select **User Defined** or **System** as shown on Figure 17-19.

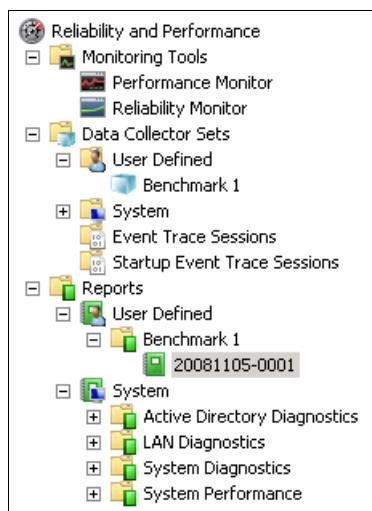


Figure 17-19 Reliability and performance monitor: Tree view

Figure 17-20 on page 558 shows you a typical report. This report provides useful information about the performance of your system.

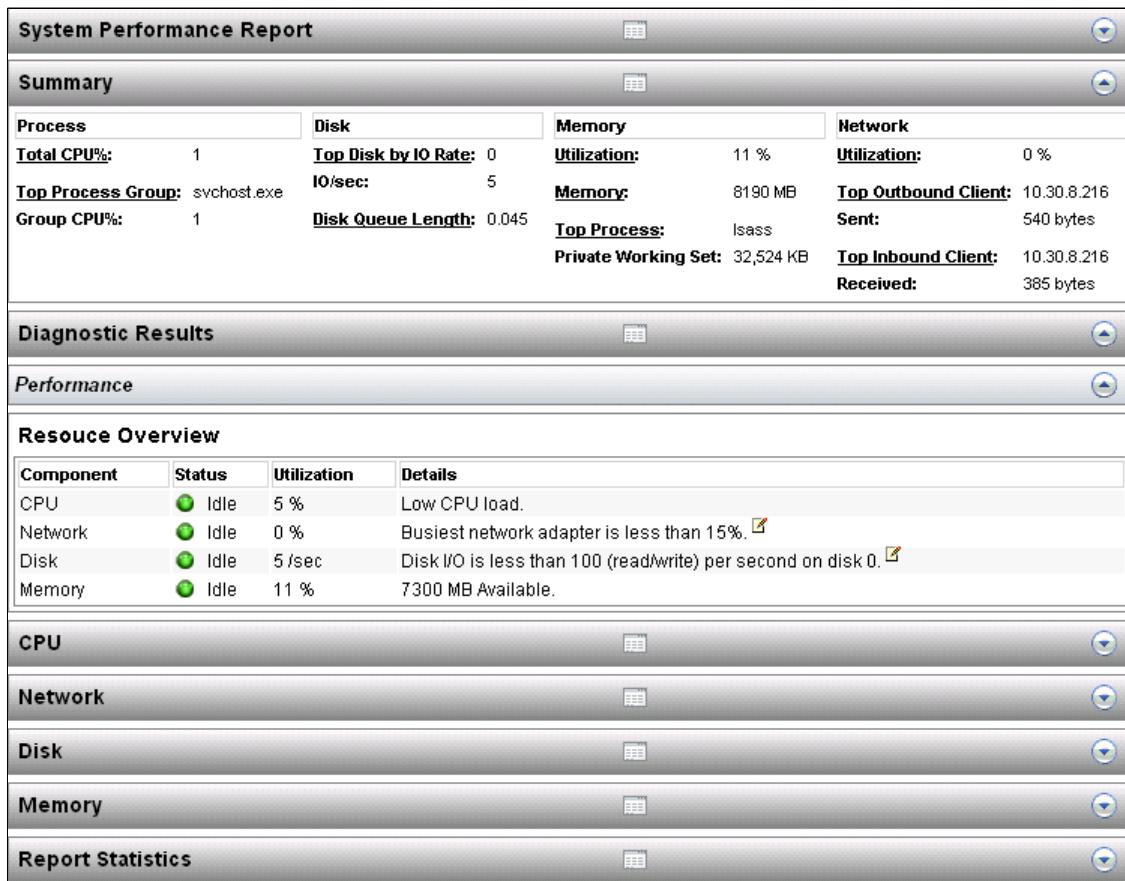


Figure 17-20 Example report

To view the results of your Data Collector Set in the Performance Monitor windows:

1. Right-click the report you want to see.
2. Select **View**, then **Performance Monitor**.

**Selecting a time frame:** Depending on how long the counter log file was running, there can be a significant amount of data to observe. If you are interested in looking at a certain time frame when the log file was recording data, complete these steps:

1. Click the **Properties** icon (  ) on the Performance Monitor toolbar.
2. The Performance Monitor Properties box opens. Click the Source tab.
3. Select the time frame that you want to view (Figure 17-21).

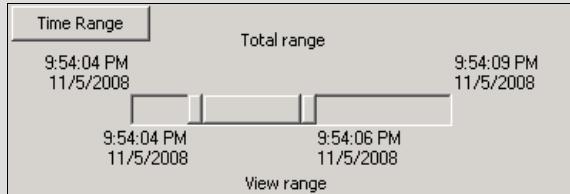


Figure 17-21 Selecting a time frame

4. Click **Ok**.

## Alerts

This function lets you track Performance Counters to ensure that they are within a specified range. If the counter's value is below or above the specified value, an alert is issued.

### *Creating an alert*

To create an alert that includes the counters that you want to track, follow these steps:

1. From the Reliability and Performance console, select **Data Collector Set**.
2. From Data Collector Set, select **User Defined**.
3. Click the **Create a new Data Collection Set** icon (  ) from the toolbar, as shown in Figure 17-22 on page 560.

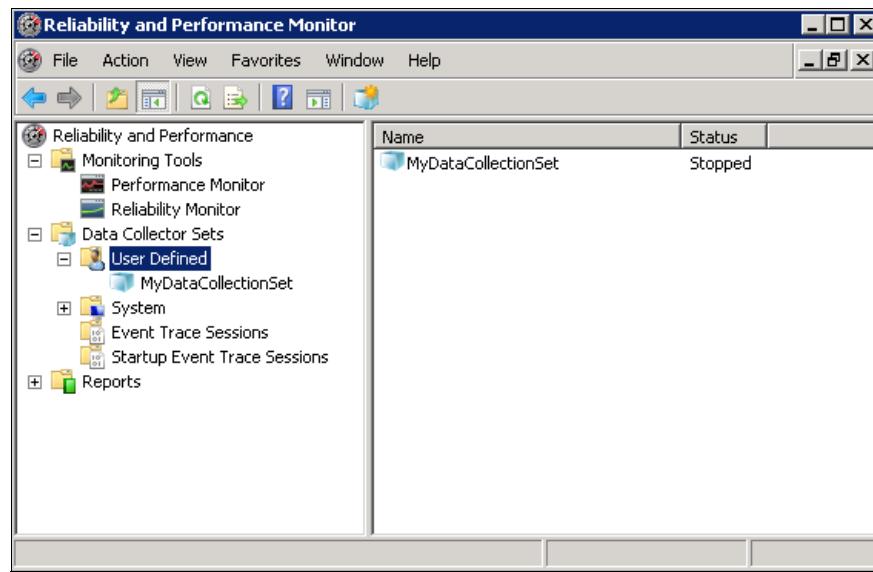


Figure 17-22 User Defined console

4. The New Data Collection Set wizard window opens, as shown in Figure 17-23 on page 561.
5. Enter a new name for your User Defined Data Collector Set.

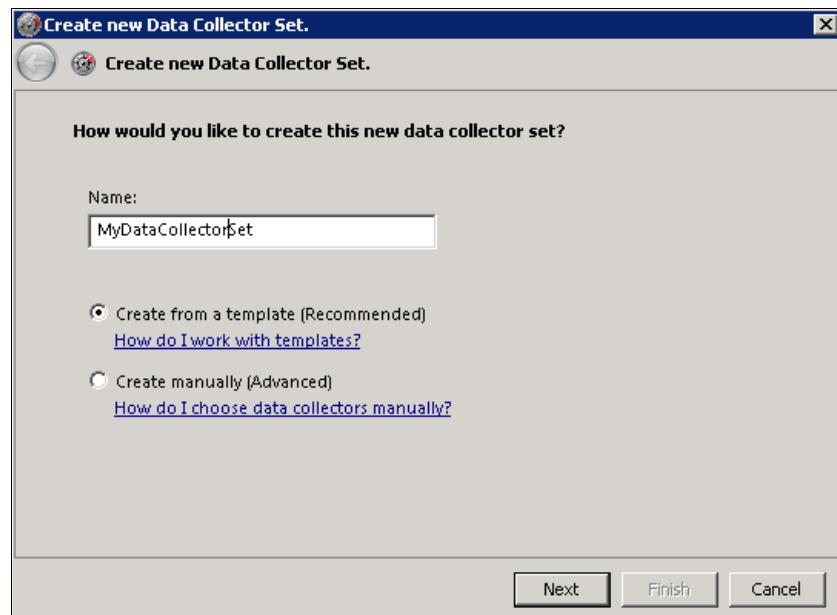


Figure 17-23 New Data Collection Set wizard, enter the alert name

6. Select **Performance Counter Alert**, as shown in Figure 17-24 on page 562.

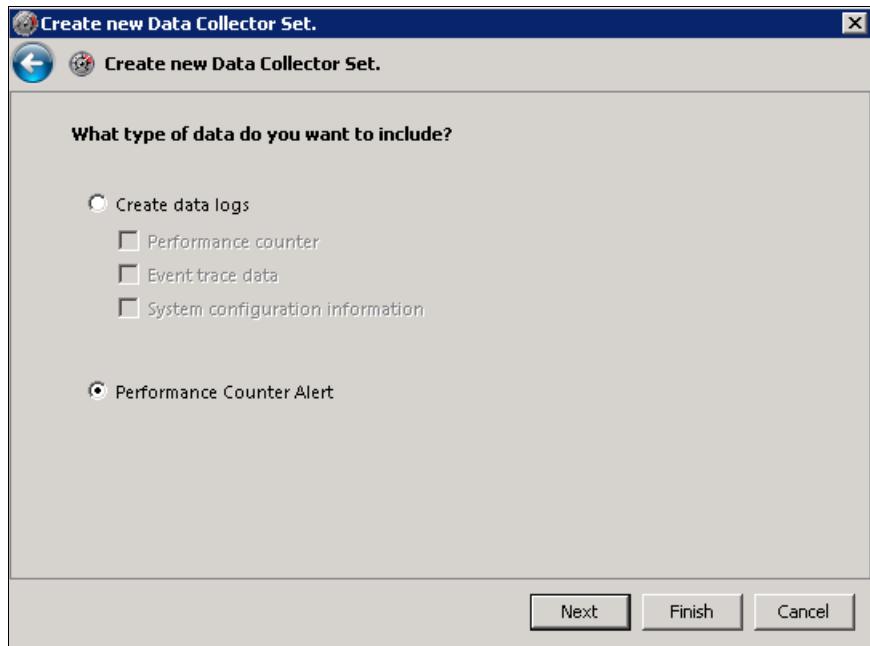


Figure 17-24 New Data Collection Set Wizard: Performance Counter Alert Choice

7. As shown in Figure 17-25 on page 563, after having added counters, you must define a limit (not above or not below a given value) for each of those performance counters.

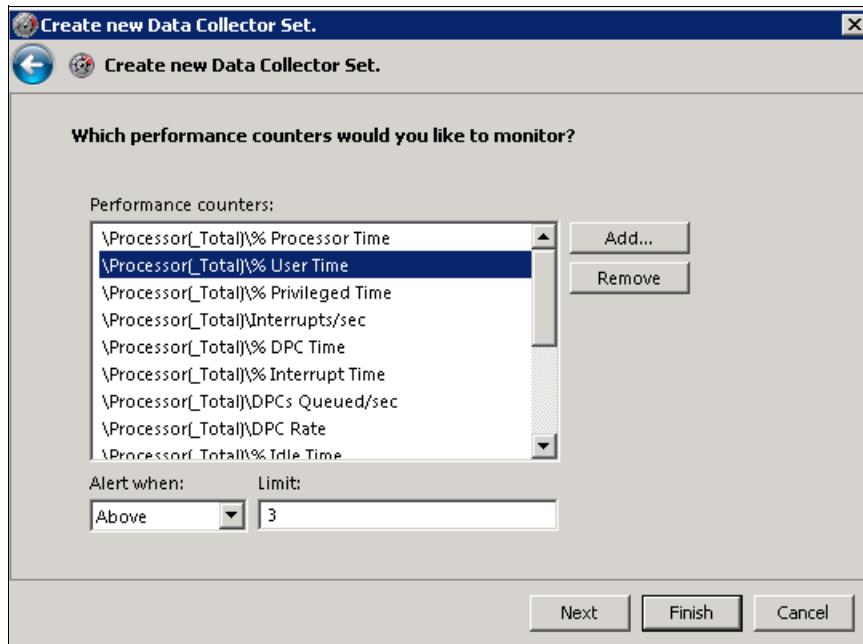


Figure 17-25 New Data Collection Set Wizard: select the counters and limit

8. On the next window, you must choose where you want to save the data. Select a Root directory using the **Browse** button. A useful option is to use the default directory.
9. On the last window of this wizard, choose the account with sufficient rights to collect the information about the server to be monitored. You must also select whether you want to **open properties for this data collector set** (for more information on properties go to “Data Collector Set properties” on page 554); **start this data collector set now**; or **save and close it**.

### Saving alert settings

You can save the alert settings to use them later. To save log settings, do the following:

1. Select the Data Collector Set in which you want to save settings.
2. Right-click the set. The window shown in Figure 17-18 on page 556 opens.
3. Click **Save Template...**
4. Select a location and enter a file name, and then click **Save** (saving to an XML file is the only option).

You can then open this log settings file using Internet Explorer or other XML reader.

### ***Starting and stopping an alert***

When creating an alert, you can schedule the start and stop time, or you can specify whether to start or stop the log manually. To start and stop the counter log manually, do the following:

1. Select the alert that you want to start.
2. Click the **Start the Data Collector Set** icon (  ) on the toolbar.
3. To stop the counter log, click the **Stop the Data Collector Set** icon (  ) on the toolbar.

### ***Importing alert settings***

You can import data collector set settings from saved files. To import settings, do the following:

1. Right-click the right-side window.
2. Select **New Data Collector Set**.
3. The Open dialog window opens. Choose a name and select to create from a template, then press **Next**. Then select **Browse** (see Figure 17-13 on page 550), choose the location and select a file name, and then click **Open**.
4. If you want to change the data collection set setting, you can do so. Otherwise, click **Finish**.

### ***Deleting alert settings***

If you no longer want to use a Data Collector log, you can delete it as follows:

1. Select the set that you want to delete.

Click the **Delete** icon (  ) in the toolbar or press Delete on your keyboard.

### **Event trace sessions**

Event Tracing sessions allows you to start and stop event tracing sessions. It records events from given providers.

### ***Creating an event trace session***

To create a new event trace log, perform the following steps:

1. From the Performance console, expand Data Collector Set and click **Startup Event trace Sessions**.

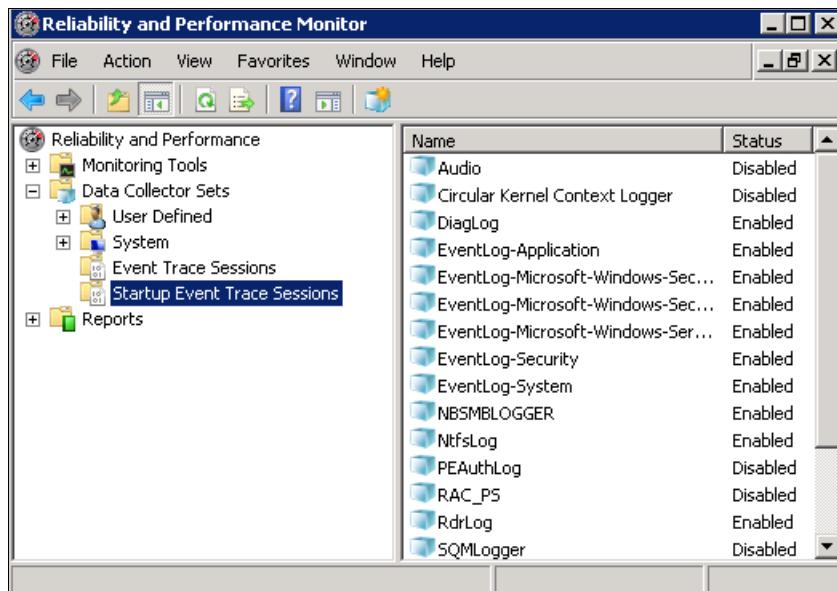


Figure 17-26 The Performance console showing trace logs

2. Click the **Create a new Data Collection Set** icon (  ) from the toolbar, as shown on Figure 17-26.
3. The New Data Collection Set wizard window opens, as shown in Figure 17-27 on page 566.
4. Enter a new name for your User Defined Data Collector Set, then select **Create manually (Advanced)** and click **Next**.

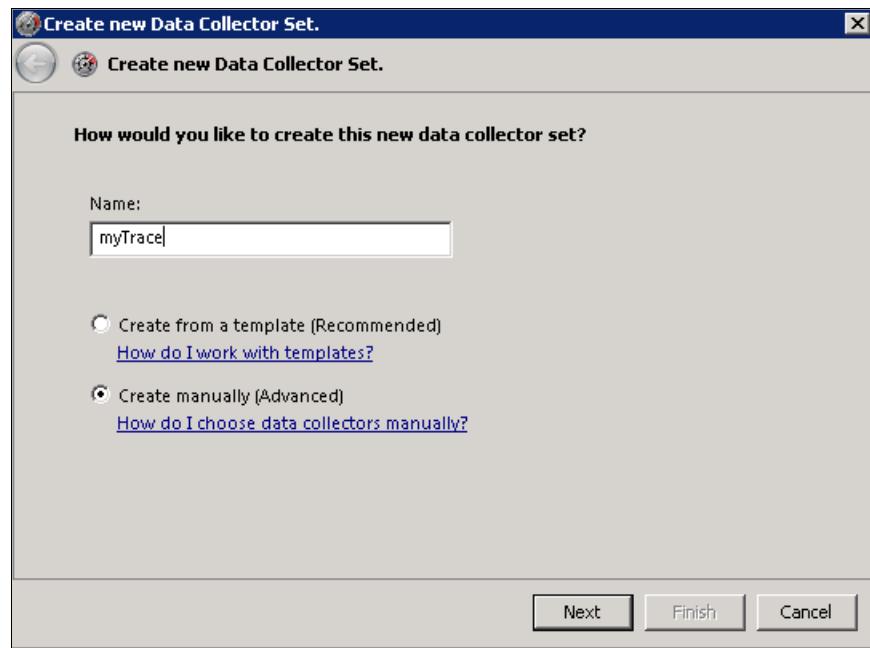


Figure 17-27 New Data Collection Set wizard, First choice

5. You may now add traces to the set. Click **Add...** to add individual counters as shown on Figure 17-28 on page 567. You can also edit the options for each of those properties.

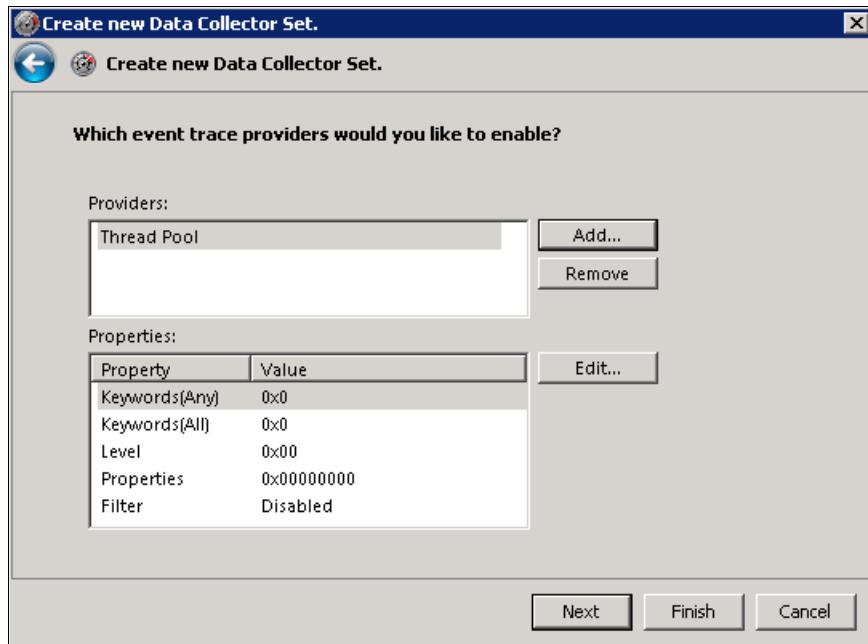


Figure 17-28 New Data Collection Set wizard: Event trace data

6. On the next window, you must choose where you want to save the data. Select a Root directory using the **Browse...** button. A useful option is to use the default directory.
7. On the last window of this wizard, choose the account with sufficient rights to collect the information about the server to be monitored. You must also select whether you want to **open properties for this data collector set** (See “Data Collector Set properties” for more information on properties) or **to save and close it**.

### **Trace properties**

To display the Properties dialog, right-click a given trace and then select **Properties**:

- ▶ The File tab allows you to specify the log file name and the log mode.
- ▶ The Directory tab allows you to choose the directory and its name where the data will be collected.
- ▶ The Security tab allows you to change the permissions of groups or users for the Data Collector Set by selecting the group or user name and then selecting the allow or deny check boxes for each permission type. To add, remove, or change permission types, click the Advanced button.

- ▶ The Stop Condition tab allows you to specify when you want to stop the Data Collection Set. You can specify the maximum size for the trace.
- ▶ The Trace Providers tab allows you to manage the event trace providers and edit their properties.
- ▶ The Trace Buffer tab allows you to set the buffer settings. Buffer settings are important when you are storing data in trace logs, because data is stored in memory buffers and then transferred to a trace log file. Here are the important parameters to configure:
  - Buffer size: size of the buffer (in KB) when you use trace data.
  - Minimum buffers: the smallest number of buffers when you use trace data.
  - Maximum buffers: the largest number of buffers when you use trace data.
  - To flush the buffers periodically, select **Flush timer** and specify the transfer interval.
- ▶ The Trace Session tab allows you to configure the clock type and the stream mode.

**Tip:** To check the installed providers and their status, select **Provider Status** in the General tab.

## Key objects and counters

The key counters in Windows Server 2008 are:

- ▶ Memory
- ▶ Processor
- ▶ Disk
- ▶ Network

These counters form the basic set of Windows Server 2008 performance objects. You might need to monitor other objects. Some counters are disabled by default because of system overhead. Refer to Chapter 21, “Analyzing bottlenecks for servers running Windows” on page 691.

### ***Monitoring disk counters***

In Windows Server 2008, there are two kinds of disk counters:

- ▶ Physical counters monitor single disks and hardware RAID arrays.
- ▶ Logical drive counters monitor software RAID arrays.

Use the DISKPERF command to enable or disable these counters. Enter DISKPERF -? for help with this command.

**Note:** You should use physical drive counters if the system is using hardware RAID such as an IBM ServeRAID adapter.

### ***Monitoring network counters***

Counters for monitoring network activity are activated by default on Windows Server 2008.

### **Using the Performance console with other tools**

You can use the following applications with a performance chart and logs to provide better reports or analysis data:

- ▶ Spreadsheet tools
- ▶ Word processing tools
- ▶ Database servers

You can export log file data to spreadsheets or databases to provide better data management. You can also add Performance Monitor Control to word processing applications. You can monitor data with this added control, as shown in Figure 17-29 on page 570.

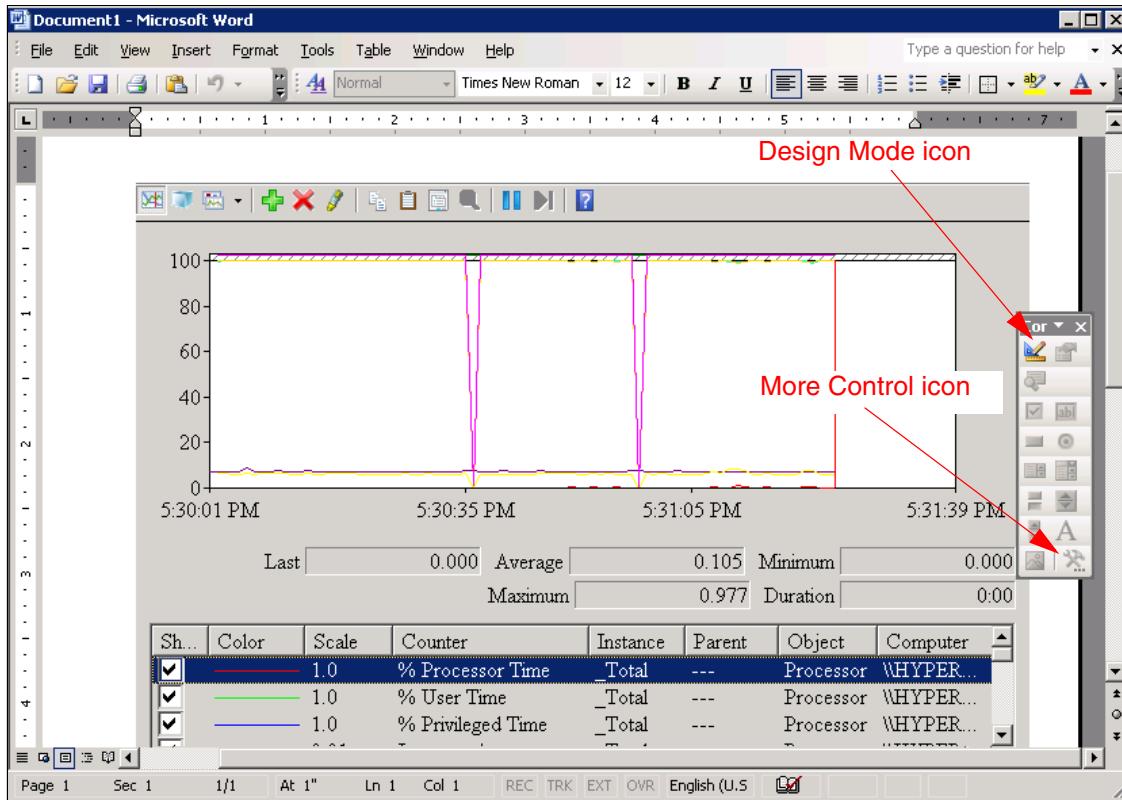


Figure 17-29 Implementing Performance Monitor in Microsoft Word

### **Microsoft Word**

You can send an active performance chart using Microsoft Word. Provided that the Microsoft Word user has permission (and authority) to access the server where the chart was created, real-time data can be displayed. This process requires at least Microsoft Word 97. To add this control:

1. Open a new Microsoft Word document and place the cursor where you want to insert this control.
2. On the toolbar, select **View → Toolbars → Control Toolbox**.
3. Click the **More Controls** icon and then select **Performance Monitor Control**, as shown in Figure 17-29.

**Note:** Control is added in design mode. In design mode, you can use this control as a Visual Basic® object. If you click the **Design Mode** icon in the control toolbox, you can use this control as Performance Monitor.

## ***Internet Explorer***

You can easily monitor servers from a remote location using Internet Explorer. To monitor a remote location, do the following:

1. Prepare chart settings as described in 17.1.2, “Using Performance Monitor” on page 541.
2. When you are monitoring performance, right-click the chart and then select **Save Setting As...**, as described in “Saving counter settings” on page 546.
3. Copy this file to a remote location.
4. Open this file using Internet Explorer.
5. The snapshot window opens. Click the **Freeze Display** icon (  ) in the toolbar to restart the chart view.

The chart view should now be running in the browser window.

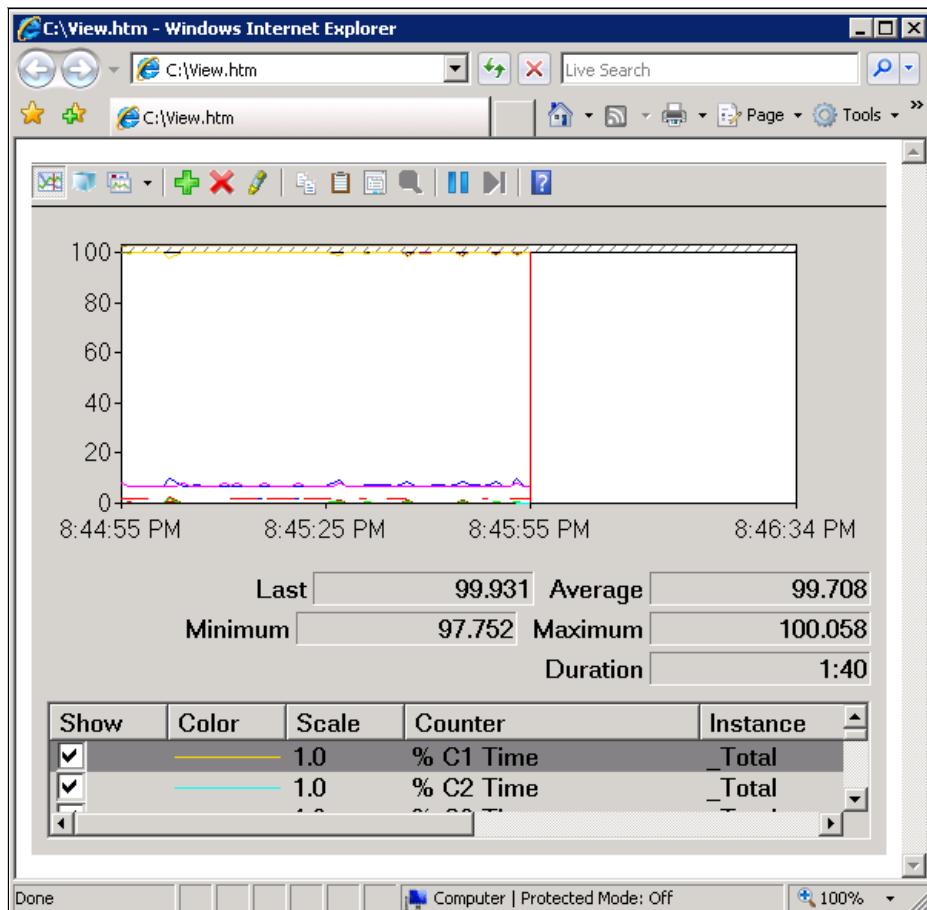


Figure 17-30 Running System Monitor in Internet Explorer

### Missing performance objects

If you cannot find the object that you want to add, refer to “Key objects and counters” on page 568.

If you still have a problem, check to be sure that disabled objects and counters do not appear in the Add Counters dialog box when you want to use them. If they do, follow the procedures in described in “Key objects and counters” on page 568.

If you still have a problem with counters, follow these steps:

1. Open the Registry Editor.
2. Change the appropriate value under HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\Service\_name\Performance\DisablePerformanceCounters from 1 to 0.

**Important:** Back up the registry before making any changes. If you edit the registry incorrectly, you can cause severe damage to your system.

## 17.2 Task Manager

In addition to the Performance console, Windows Server 2008 also includes Task Manager, which is a utility that allows you to view the status of processes and applications and gives you some real-time information about memory usage.

### 17.2.1 Starting Task Manager

You can run Task Manager using any one of the following methods:

- ▶ Right-click a blank area of the task bar and select **Task Manager**.
- ▶ Press Ctrl+Alt+Del and click **Task Manager**.
- ▶ Click **Start** → **Run** and type taskmgr.

Task Manager has six views:

- ▶ Applications
- ▶ Processes
- ▶ Services (new for Windows Server 2008)
- ▶ Performance
- ▶ Networking
- ▶ Users

This discussion focuses on the Processes, Performance, and Networking tabs.

## 17.2.2 Processes tab

Figure 17-31 shows the Processes tab.

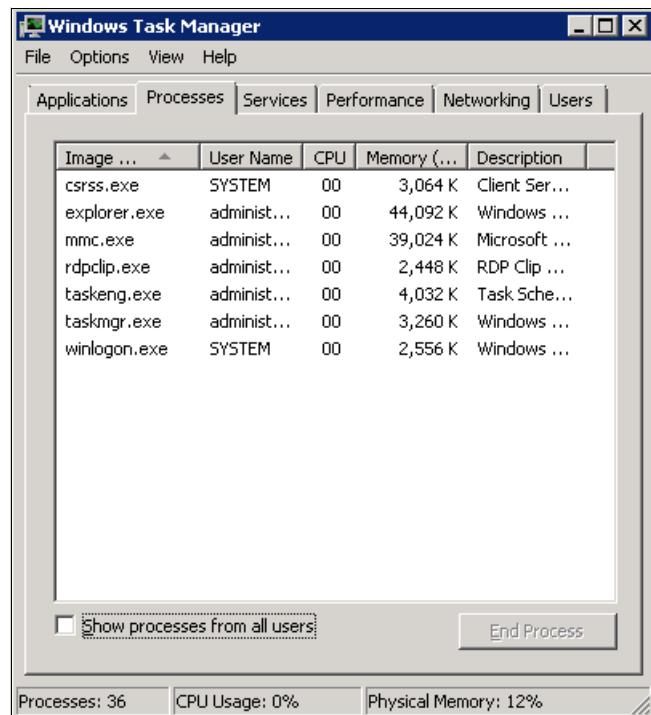


Figure 17-31 Task Manager: Processes

In this view, you can see the resources that are consumed by each of the processes currently running. You can click the column headings to change the sort order on which that column is based.

Click **View → Select Columns** to display the window shown in Figure 17-32. From this window, you can select additional data to display for each process.

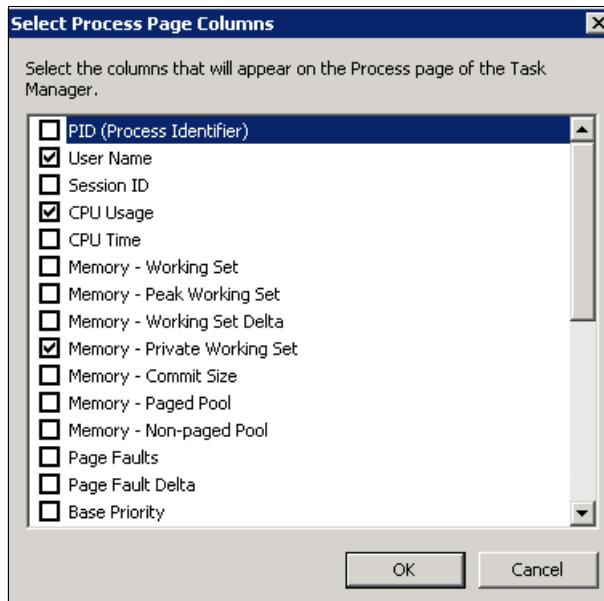


Figure 17-32 Task Manager: select columns for the Processes view

Table 17-3 shows the columns that are available in the Windows Server 2008 operating system.

Table 17-3 Columns in the Windows Server 2008 Processes view

Column	Description
PID (Process Identifier)	Process Identification Number, an internally assigned number.
User Name	The name of the user running the process.
Session ID	The ID of the session running the process.
CPU Usage	Current CPU utilization. When the system is not doing any work, the process "System Idle Process" will be near 100%.
CPU Time	Total amount of time this process has used since it was started, in seconds.
Memory - Working Set	The total amount of memory used by the process, in KB. It includes both the paged and nonpaged pool memory used.
Memory - Peak Working Set	The peak amount of memory, in KB, used by the process since it was started.

Column	Description
Memory - Working Set Delta	The change in memory usage since the last Task Manager update.
Memory - Private Working Set	The amount of memory a process is using that cannot be shared by other processes.
Memory - Commit Size	The amount of virtual memory that is reserved for use by a process.
Memory - Page Pool	The paged pool (user memory) usage of each process. The paged pool is virtual memory available to be paged to disk. It includes all of the user memory and a portion of the system memory.
Memory - Non-Paged Pool	The amount of memory reserved as system memory and not pageable for this process.
Page Faults	The number of times data had to be retrieved from disk for this process because it was not found in memory. This is a total since the process was started.
Page Faults Delta	The change in the number of page faults since the last update.
Base Priority	The process's base priority level (low/normal/high). You can change the process's base priority by right-clicking it and selecting <b>Set Priority</b> . This remains in effect until the process stops.
Handle	The number of handles used by the process.
Thread	The number of threads this process is running.
USER Objects	The number of Window Manager (USER) objects currently used by the process.
GDI Objects	The number of Graphics Device Interface (GDI) objects used by the process.
I/O Reads	The number of read input/output (file, network, and device) operations generated by the process.
I/O Writes	The number of write input/output operations (file, network, and device) generated by the process.
I/O Other	The number of input/output operations generated by the process that are not reads or writes (for example, a control type of operation).
I/O Read Bytes	The number of bytes read in input/output (file, network, and device) operations generated by the process.
I/O Write Bytes	The number of bytes written in input/output operations (file, network, and device) generated by the process.

Column	Description
I/O Other Bytes	The number of bytes transferred in input/output operations generated by the process that are not reads or writes (for example, a control type of operation).
Image path Name	The path where the process is installed.
Command Line	The Command used to launch the process.
Virtualization	Give a status on the UAC file and registry virtualization for the process.
Description	Give a description of a running process.
Data Execution prevention	See if the system prevents the process from executing code from a non-executable memory region.

By adding the relevant columns, it is very easy to determine if a particular application is behaving improperly. For example, it is possible to see if a particular application has a memory leak simply by adding the Virtual Memory Size column. After the column has been added, note the value in the Virtual Memory Size column for the relevant application. You can log off or lock the server console at this point. After a while, depending on how frequently your server is running out of RAM, you can recheck this value to see if it has grown. If this Virtual Memory Size always rises and never comes down, then it is possible that the relevant application has a memory leak.

### 17.2.3 Performance tab

The Performance tab displays performance indicators, as shown in Figure 17-33 on page 578.

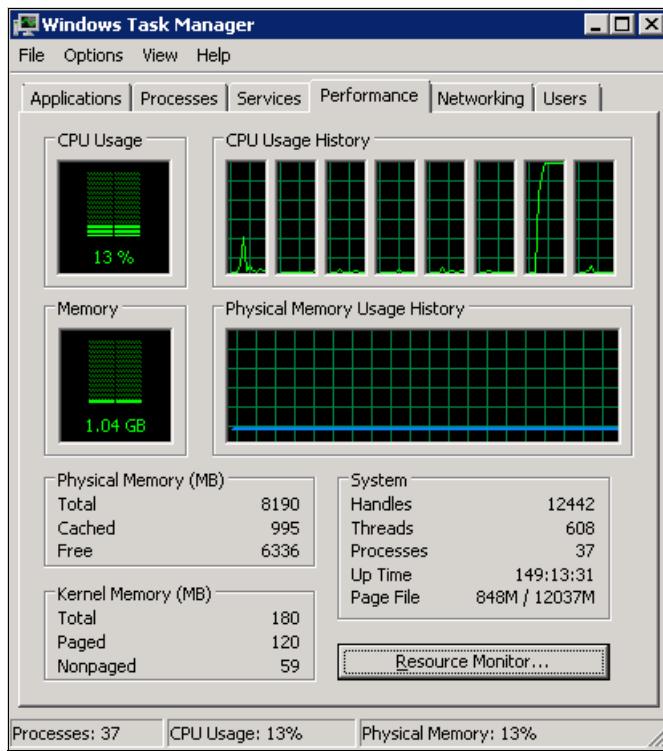


Figure 17-33 Task Manager: Performance

The charts show you the CPU and memory usage of the system as a whole. The bar charts on the left show the instantaneous values and the line graphs on the right show the history since Task Manager was started.

The three sets of numbers under the charts are as follows:

- ▶ **Physical Memory**
  - Total: total RAM installed (in MB).
  - Cached: total RAM released to the file cache on demand (in MB).
  - Free: total RAM available to processes (in MB).
- ▶ **Kernel Memory**
  - Total: sum of paged and nonpaged kernel memory (in MB).
  - Paged: size of paged pool that is allocated to the operating system (in MB).
  - Nonpaged: size of nonpaged pool that is allocated to the operating system (in MB).

- ▶ System
  - Handles: current number of unique object identifiers in use by processes.
  - Threads: current number of objects or processes running within larger processes or programs.
  - Processes: current number of individual processes running on the server.
  - Uptime: Amount of time that has passed since the computer was restarted.
  - Page File: A description of virtual memory use. The first number is the amount of RAM and virtual memory currently in use, and the second number is the amount of RAM and virtual memory available on your Server.

To view advanced information about how much CPU, Disk, Network and Memory resources are being used, click the **Resource Monitor** button. Resource Monitor shows graphical summaries like those in Task Manager, but in greater detail as shown on Figure 17-34.

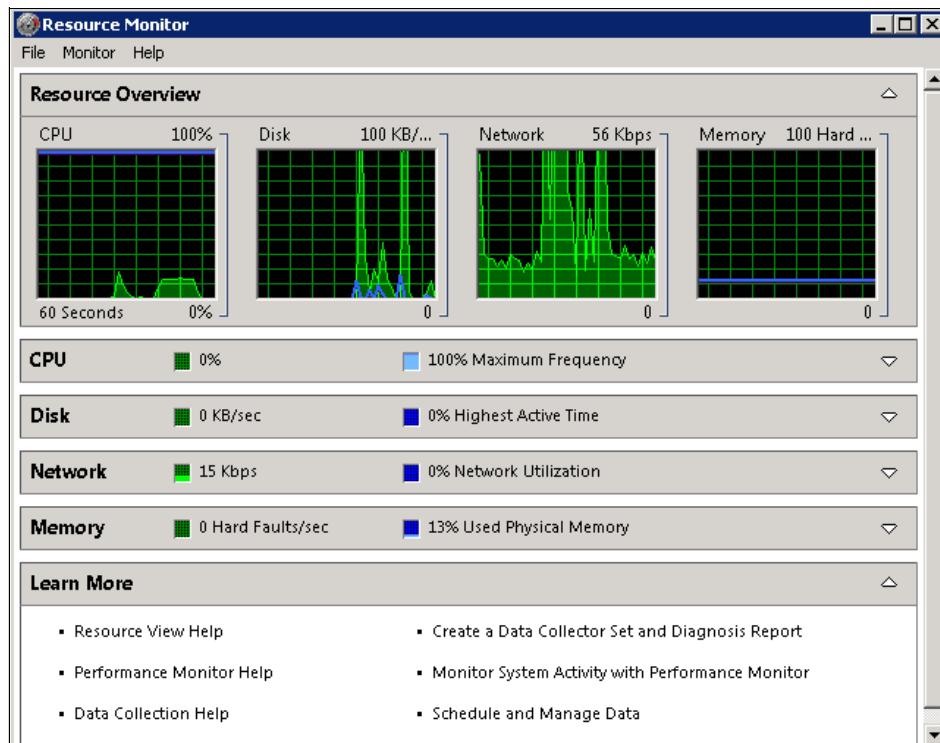


Figure 17-34 Resource Monitor Console

**Information:** If you have multicore processors, it shows as CPUs on the Performance tab of Windows Task Manager.

## Networking tab

The Networking tab is available if you have network issues. It provides a simple method of viewing network activity graphically, as shown in Figure 17-35.

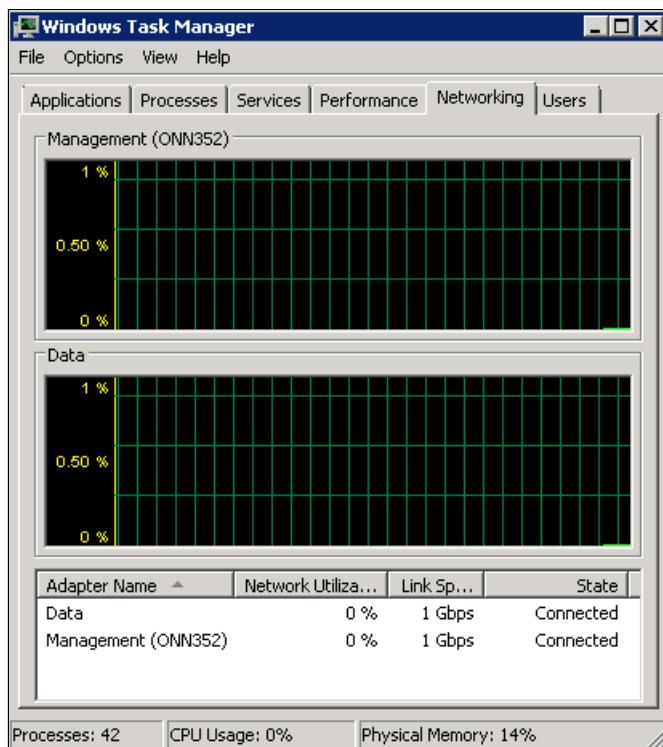


Figure 17-35 Task Manager: Networking

## 17.3 Network Monitor

Network Monitor is a useful tool that is shipped with Windows Server 2008. It captures network traffic for display and analysis. This tool makes troubleshooting complex network problems easier and more economical. With Network Monitor, you can capture network traffic from a local NIC. You can also attach remote stations, over the network or through dial-up, that are running the agent software.

Network Monitor works by placing the Ethernet controller in *promiscuous mode* so that it passes every frame on the wire to the tracing tool. The tool supports capture filters so that only specific frames are saved for analysis.

Network Monitor is a useful troubleshooting tool, and it also helps administrators in their daily work. It shows what types of protocols are flowing across the network, delivers a better understanding of how bandwidth is used, and detects security risks (viruses, unapproved client applications, and so forth).

The version of Network Monitor that ships with Windows Server 2008 allows the capture of traffic only on the local server. However, the version included with Windows Server 2008 System Management Server (SMS) allows you to monitor any machine on your network.

### 17.3.1 Installing Network Monitor

To install the latest Network Monitor on Windows Server 2003 or Windows Server 2008 version, first download the tool from the Microsoft Web site and install it.

The current version is 3.2 and you can download it from the following URL:

<http://www.microsoft.com/downloads/details.aspx?FamilyID=f4db40af-1e08-4a21-a26b-ec2f4dc4190d&DisplayLang=en>

### 17.3.2 Using Network Monitor

You can start Network Monitor application by clicking **Start** → **All Programs** → **Network Monitor** → **Microsoft Network Monitor 3.2** → **Microsoft Network Monitor 3.2**.

When you start Network Monitor the first time, you are presented with a Start Page Tab similar to that shown in Figure 17-36 on page 582, where you select which adapter you want to monitor. The bottom left part of the window displays the available adapter to monitor, and the right panel displays the information on the current release.

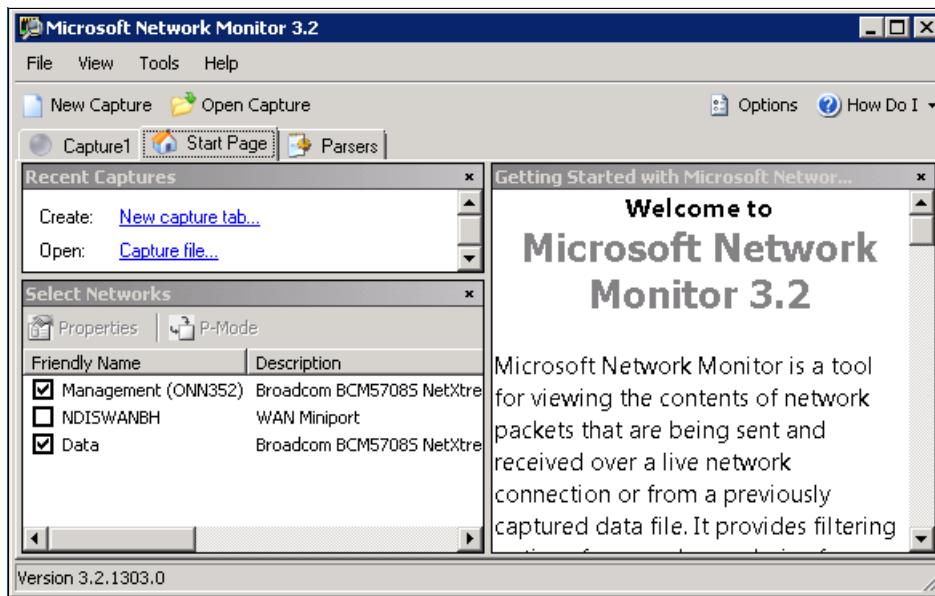


Figure 17-36 Select network connection

Select one of the connections and click the **New Capture** button ( ) to open the main window, as shown in Figure 17-37 on page 583.

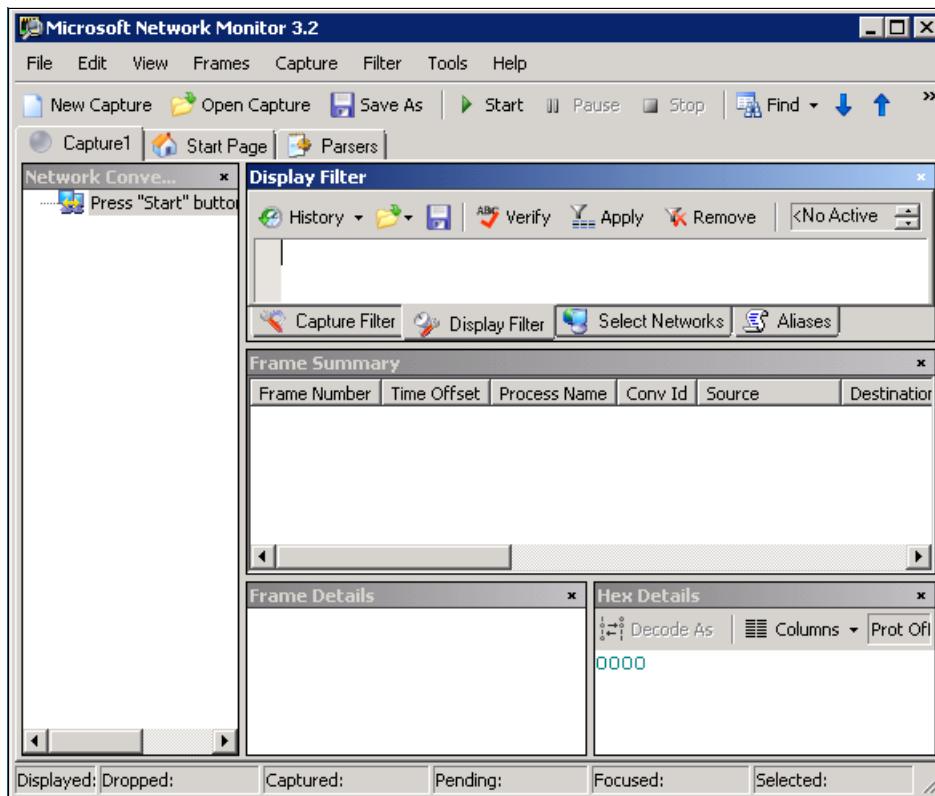


Figure 17-37 Network Monitor main window

When the window first opens, the data capture process is stopped. Normally, you set filters to limit the data that the tool captures.

There are two types of filters:

- ▶ Capture filters enable you to specify which types of packets are captured (for example, all HTTP traffic).
- ▶ Display filters capture all the frames that are running over the monitored NIC and the packets get filtered at the time of analysis.

Using display filters enables the administrator to have all the data available at the time of analysis, but it generates a large log file. So, the server might run out of available space to save the information. Any protocol, protocol element, or property can be filtered on.

## Configuring filters

You can configure filters by clicking **Filter** then **select Capture Filter or Display Filter**. From this menu, you can load predefined filters or add your own filters. The window that is shown in Figure 17-38 opens.

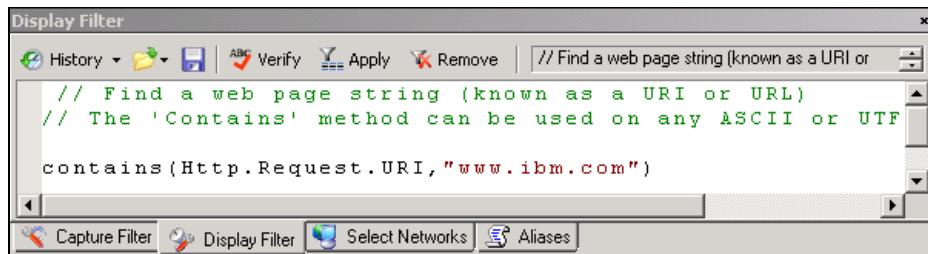


Figure 17-38 Display Filter main window

### Creating a new filter or loading a filter

In this window, you can create a new filter or load a filter that you previously saved. The current filter is displayed in the display filter frame as shown in Figure 17-38.

To verify the current filter, click **Verify** ( ).

To save the current filter to a file, click **Save** ( ).

To retrieve a previously saved filter, click **Load** ( ).

When you have completed the filter, click **Apply** ( ) to activate it in the main window.

### Starting the capture process

There are three ways to start capturing network traffic:

- ▶ Press F5.
- ▶ Click **Capture → Start**.
- ▶ Click the **Play** button ( ) in the toolbar.

Figure 17-39 illustrates the capture process.

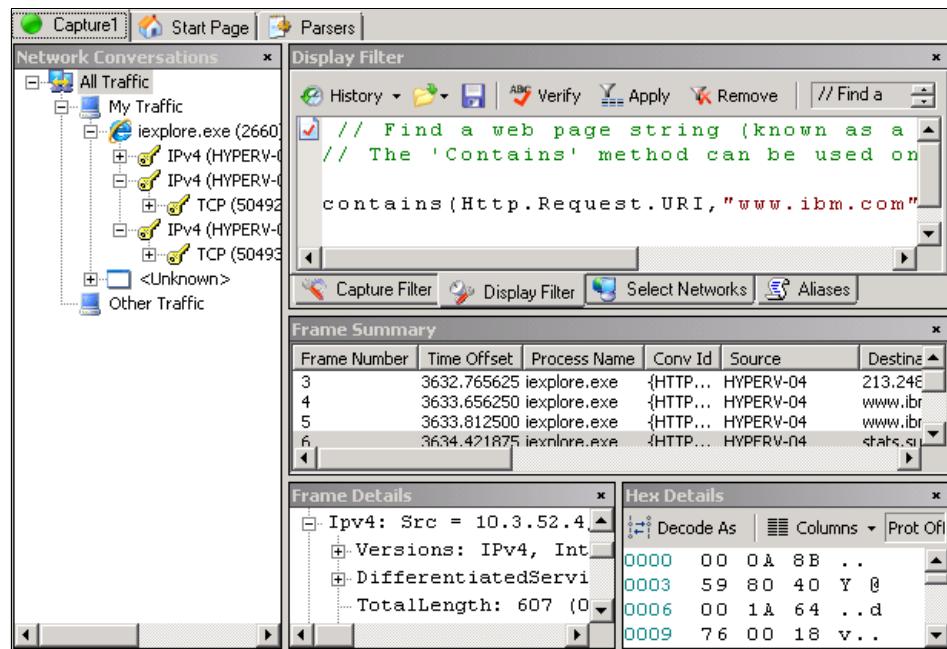


Figure 17-39 Starting the capture

## Stopping the capture process

After you have captured the data that you want, stop the capture process. You can stop the capture process and view the result by performing any of the following tasks:

- ▶ Press F7.
- ▶ Click **Capture → Stop**
- ▶ Click the **Stop** icon ( ).

**Note:** You can visualize the frame summary without stopping the capture.

## Viewing a capture summary

The window that shows the captured data looks similar to that shown in Figure 17-40 on page 586.

Frame Number	Time Offset	Process ...	Conn Id	Source	Destination	Protocol ...	Description
3	3632.765625	lexplore.exe	{HTTP...}	HYPERV...	213.248.111.83	HTTP	HTTP:Request, G
4	3633.656250	lexplore.exe	{HTTP...}	HYPERV...	www.ibm.co...	HTTP	HTTP:Request, G
5	3633.812500	lexplore.exe	{HTTP...}	HYPERV...	www.ibm.co...	HTTP	HTTP:Request, G
6	3634.421875	lexplore.exe	{HTTP...}	HYPERV...	stats.surfaid...	HTTP	HTTP:Request, G
7	3634.453125	lexplore.exe	{HTTP...}	HYPERV...	data.coremem...	HTTP	HTTP:Request, G
8	3634.562500	lexplore.exe	{HTTP...}	HYPERV...	stats.surfaid...	HTTP	HTTP:Request, G

*Figure 17-40 Frame summary*

After you have located a packet that is important for your analysis, click it. You get a more detailed view of the packet, as shown in Figure 17-41.

**Frame Summary**

Frame Number	Time Offset	Process ...	Conv Id	Source	Destination	Protocol ...	Description
3	3632.765625	iexplore.exe	{HTTP...}	HYPERV...	213.248.111.83	HTTP	HTTP:Request, GET
4	3633.656250	iexplore.exe	{HTTP...}	HYPERV...	www.ibm.co...	HTTP	HTTP:Request, GET
5	3633.812500	iexplore.exe	{HTTP...}	HYPERV...	www.ibm.co...	HTTP	HTTP:Request, GET
6	3634.421875	iexplore.exe	{HTTP...}	HYPERV...	stats.surfaid...	HTTP	HTTP:Request, GET
7	3634.453125	iexplore.exe	{HTTP...}	HYPERV...	data.coremet...	HTTP	HTTP:Request, GET
8	3634.562500	iexplore.exe	{HTTP...}	HYPERV...	stats.surfaid...	HTTP	HTTP:Request, GET
9	3634.765625	iexplore.exe	{HTTP...}	HYPERV...	data.coremet...	HTTP	HTTP:Request, GET

**Frame Details**

```

+ Ethernet: Etype = Internet IP (1)
+ Ipv4: Src = 10.3.52.4, Dest = 12
- Tcp: Flags=...AP..., SrcPort=504
  SrcPort: 50491
  DstPort: HTTP(80)
  SequenceNumber: 3677044047 (0)
  AcknowledgementNumber: 243343
+ DataOffset: 80 (0x50)
+ Flags: ...AP...
  Window: 63679 (scale factor 0)
  Checksum: 0xFD1C, Disregarded
  UrgentPointer: 0 (0x0)

```

**Hex Details**

	Decode As	Columns	Prot Off: 2 (0x02)
0000	00 0A 8B 59 80 .. Y		
0005	40 00 1A 64 76 0 .. dv		
000A	00 18 08 00 45 .... E		
000F	00 03 21 4A 14 .. J.		
0014	40 00 80 06 00 0 .. .		
0019	00 0A 03 34 04 .. 4.		
001E	81 2A 3A D8 C5 *:@		
0023	3B 00 50 DB 2B ;.PU+		
0028	3D 4F 91 0B 3C =O .<		
002D	1F 50 18 F8 BF .P..G		
0032	FD 1C 00 00 47 y...G		
0037	45 54 20 2F 69 ET / i		
003C	2F 63 2E 67 69 /c.cgi		

*Figure 17-41 Selected captured packet*

This window is divided into three sections:

- ▶ The first section is a window with the selected data point highlighted. If you want to view another packet in the detailed view you can select it here.
  - ▶ The bottom left section includes the packet's content in a decoded tree format. Our example is an HTTP request to the <http://www.ibm.com> Web site. Figure 17-42 on page 587 shows more detail about the packet's contents.

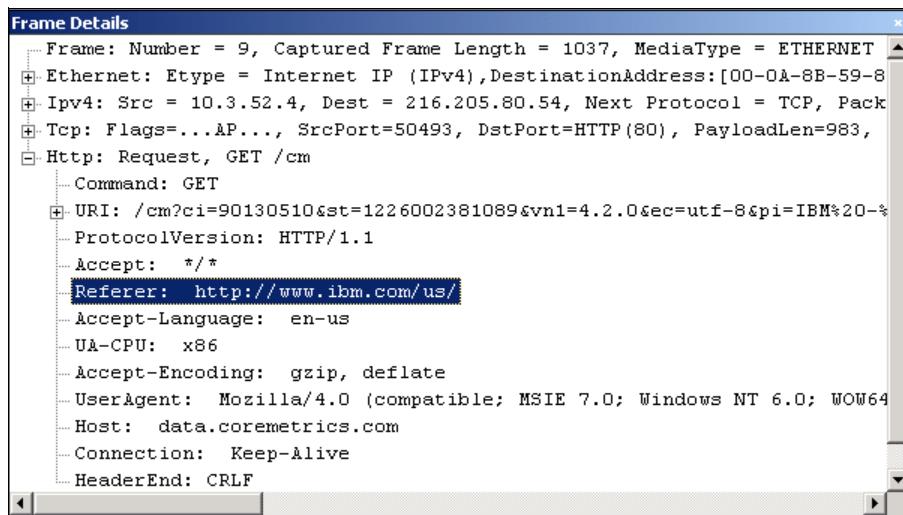


Figure 17-42 Expanded HTTP:Request packet

- ▶ The third section includes the raw data of the packet in hexadecimal format. The column to the far right includes the reprint of the data in ASCII text. So, for example, if a password or an e-mail is delivered over your network and it is not encrypted, you can read it here. In the following example, you can see the IBM Web site in hex detail, as shown in Figure 17-43.

	Hex Details		Decode As	Columns	Prot Off: 697 (0x2B9)	Frame Off: 751 (0x2EF)
02E8	65 66 65 72 65 72 3A 20	eferer:				
02F0	68 74 74 70 3A 2F 2F 77	http://w				
02F8	77 77 2E 69 62 6D 2E 63	ww.ibm.c				
0300	6F 6D 2F 75 73 2F OD 0A	om/us/..				
0308	41 63 63 65 70 74 2D 4C	Accept-L				
0310	61 6E 67 75 61 67 65 3A	nguage:				
0318	20 65 6E 2D 75 73 OD 0A	en-us..				
0320	55 41 2D 43 50 55 3A 20	UA-CPU:				
0328	78 38 36 OD 0A 41 63 63	x86..Acc				
0330	65 70 74 2D 45 6E 63 6F	ept-Enco				
0338	64 69 6E 67 3A 20 67 7A	ding: gz				
0340	69 70 2C 20 64 65 66 6C	ip, defl				
0348	61 74 65 OD 0A 55 73 65	ate..Use				
0350	72 2D 41 67 65 6E 74 3A	r-Agent:				
0358	20 4D 6F 7A 69 6C 6C 61	Mozilla				
0360	2F 34 2E 30 20 28 63 6F	/4.0 (co				
0368	6D 70 61 74 69 62 6C 65	mpatible				
0370	3B 20 4D 53 49 45 20 37	; MSIE 7				

Figure 17-43 Hex details in Network Monitor

## **Network Monitor tips**

In this book, we show the main steps so that you can become familiar with this tool. If an administrator plans the integration of Network Monitor in the infrastructure and customizes the right filters, it will prove to be a powerful tool that helps to keep the network running smoothly. The network administrator can use Network Monitor to identify patterns to prevent or solve network problems.

Tips for running Network Monitor:

- ▶ Capture only the minimum amount of network statistics necessary.  
Use capture filters to capture only the needed packets so as to keep the log file small. This will make a quick diagnostic easier.
- ▶ Use display filters.  
When viewing the captured data, use display filters even if you have used filters while capturing the data.
- ▶ Run Network Monitor during low usage time.  
If you run the Network Monitor, use it during times of low usage or only for short periods of time to make the analysis clearer and to decrease the effect that it has on your production workload.

## **17.4 Other Windows tools**

Since Windows 2000, Microsoft has made available a set of support tools with its server operating systems. These support tools are not installed by default, but you can download them from the Internet.

### **17.4.1 Microsoft Windows Performance Toolkit**

The Windows Performance Toolkit contains performance analysis tools. These tools are designed for measuring and analyzing system and application performance on Windows Server 2008 and later. It is available for download at no charge from:

<http://msdn.microsoft.com/en-us/library/cc305187.aspx>

The tools currently include:

- ▶ An xperf trace capture tool: Command line
- ▶ An xperfview visualization tool also known as Windows Performance Analyzer
- ▶ An xbootmgr boot trace capture tool.

The tools are designed for the analysis of a wide range of performance problems including application start times, boot issues, deferred procedure calls and

interrupt activity (DPCs and ISRs), system responsiveness issues, application resource usage, and interrupt storms. Some of the most important and useful kernel events that are available for capture and analysis are context switches, interrupts, DPCs, process and thread creation and destruction, disk I/Os, hard faults, processor P-state transitions (for more information, go to “Manage power consumption on CPUs” on page 56), registry operations, and many others. Figure 17-44 shows a view of the xperfview visualization tool.

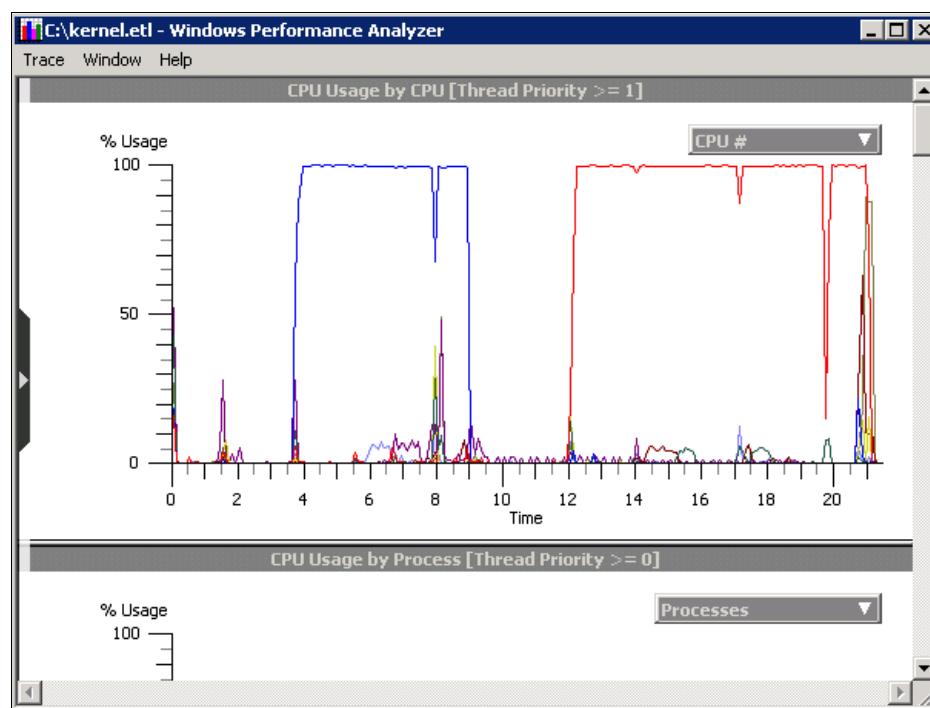


Figure 17-44 Windows Performance Analyzer

### 17.4.2 Microsoft Windows Sysinternals tools

Windows Sysinternals utilities will help you to manage, troubleshoot, and diagnose your Windows 2008 Servers and applications. Those tools are available from:

<http://technet.microsoft.com/en-us/sysinternals>

Those tools are classified into different categories. The following categories are the relevant ones in terms of performance:

- ▶ File and disk utilities

- Utilities for viewing and monitoring file and disk access and usage

- ▶ Networking
 

Networking tools that range from connection monitors to resource security analyzers.
- ▶ Process utilities
 

Utilities for looking “under the hood” to see what processes are doing and what resources they are consuming.
- ▶ System information
 

Utilities for looking at system resource usage and configuration.

Process Monitor and Process Explorer are very interesting tools and we will explain them in detail.

Process Monitor is an advanced monitoring tool for Windows that shows real-time file system, Registry, and process/thread activity as shown in Figure 17-45.

Time ...	Process Name	PID	Operation	Path	Result
10:35...	Explorer.EXE	756	RegOpenKey	HKLM\Software\Microsoft\Windows\C...	SUCCESS
10:35...	Procmon[1]64.e...	3000	QueryStandardInfo...	C:\Windows\System32\msctf.dll	SUCCESS
10:35...	Explorer.EXE	756	RegQueryValue	HKLM\SOFTWARE\Microsoft\Window...	SUCCESS
10:35...	Explorer.EXE	756	RegCloseKey	HKLM\SOFTWARE\Microsoft\Window...	SUCCESS
10:35...	Procmon[1]64.e...	3000	QueryStandardInfo...	C:\Windows\System32\drivers\mountm...	SUCCESS
10:35...	Explorer.EXE	756	RegOpenKey	HKLM\Software\Microsoft\Windows\C...	SUCCESS
10:35...	Procmon[1]64.e...	3000	QueryStandardInfo...	C:\Windows\System32\drivers\mountm...	SUCCESS
10:35...	Procmon[1]64.e...	3000	ReadFile	C:\Windows\System32\msctf.dll	SUCCESS
10:35...	Explorer.EXE	756	RegQueryValue	HKLM\SOFTWARE\Microsoft\Window...	SUCCESS
10:35...	Procmon[1]64.e...	3000	QueryStandardInfo...	C:\Windows\System32\drivers\mountm...	SUCCESS
10:35...	Explorer.EXE	756	RegCloseKey	HKLM\SOFTWARE\Microsoft\Window...	SUCCESS
10:35...	Procmon[1]64.e...	3000	ReadFile	C:\Windows\System32\drivers\mountm...	SUCCESS
10:35...	Explorer.EXE	756	RegOpenKey	HKLM\Software\Microsoft\Windows\...	SUCCESS
10:35...	Explorer.EXE	756	RegQueryValue	HKLM\SOFTWARE\Microsoft\Window...	SUCCESS
10:35...	Explorer.EXE	756	RegCloseKey	HKLM\SOFTWARE\Microsoft\Window...	SUCCESS
10:35...	Procmon[1]64.e...	756	RegOpenKey	HKLM\Software\Microsoft\Windows\...	SUCCESS

Showing 47,023 of 72,119 events (65%)      Backed by page file

Figure 17-45 Process Monitor main windows

To download Process Monitor directly to your system, use the following link:

<http://live.sysinternals.com/Procmon.exe>

Process Explorer shows you information about which handles and DLLs processes have opened or loaded as shown on Figure 17-46 on page 591. You

can also see in detail the system idle processes including the hardware interrupts and deferred procedure calls (DPCs).

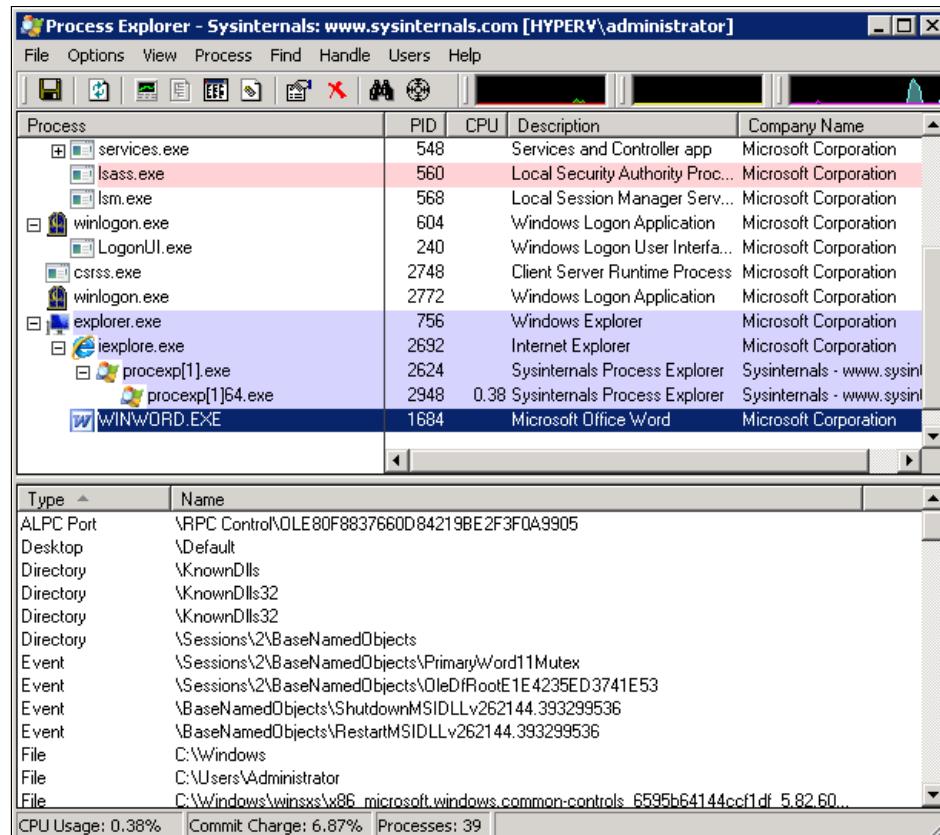


Figure 17-46 Process Explorer - Main Windows

For example, with this tool you can determine whether an application scales on all the CPUs or is running only one CPU, as shown in Figure 17-47 on page 592.

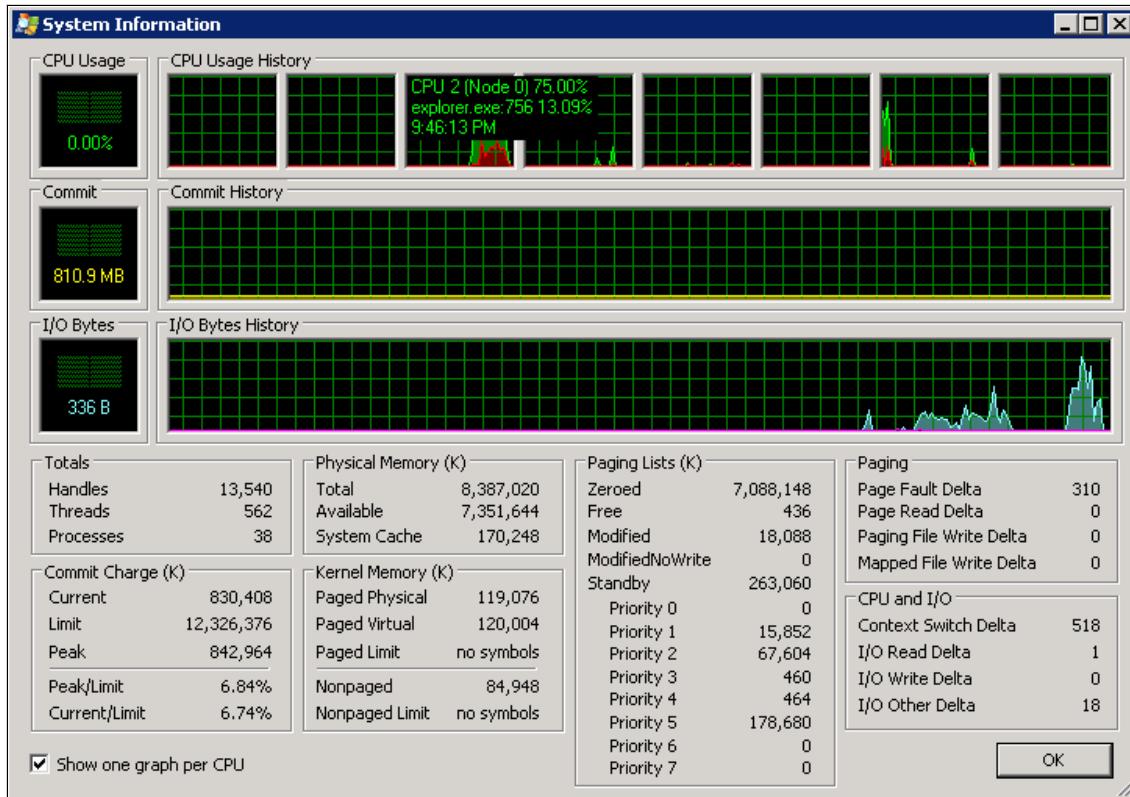


Figure 17-47 Detailed CPU view in Process Explorer

Here is the URL to install Process Explorer directly on your server:

<http://live.sysinternals.com/procexp.exe>

### 17.4.3 Others tools

Many of the tools that were in the Windows 2000 support tools or Resource Kit are included in the standard Windows Server 2008 build. For example, the **typeperf** command that was part of the Windows 2000 resource kit is now included as standard in Windows Server 2008. Table 17-4 lists a number of these tools and provides the executable, as well as a brief description.

Table 17-4 Windows Server 2008 performance tools

Name	Executable	Description
Defrag	defrag.exe	Used to defragment hard disks.

Name	Executable	Description
Logman	logman.exe	Command-line tool to manager performance monitor.
Typeperf	typeperf.exe	Displays performance counter data at a console.
Tracerpt	tracerpt.exe	Used to process trace logs.

**Tip:** With Windows Server 2008, you can schedule the defragmentation of your local hard disks using a combination of the task scheduler and defrag.exe. However, you might want to invest in a third-party tool for the enhanced features that they offer. See 13.17.7, “Use disk defragmentation tools regularly” on page 417 for details.

## 17.5 Windows Management Instrumentation

Up to this point, we have focused our discussion of performance tools largely on the Performance console. The Performance console is a fantastic tool for monitoring one or possibly a few servers. However, when you want to monitor perhaps 10 or more servers, it become laborious to set up all those servers.

One alternative is to write a shell script using some of the command line tools listed in Table 17-4 on page 592 to obtain the data. A better alternative is to access the data directly using Windows Management Instrumentation (WMI). WMI uses industry standards to retrieve management and event monitoring data from systems in the network. WMI effectively provides a large collection of data available to applications such as those written in VBScript or C++ (some functions are not available in VBScript).

In this section, we describe a script that we created in VBScript that calls WMI to retrieve and display data from multiple systems.

**Tip:** WMI was introduced in Windows 2000 Server. However, Microsoft has enhanced WMI significantly between Windows 2000 Server and Windows Server 2008, and some objects that are available in Windows Server 2008 might not be available in Windows 2000 Server.

For our purposes, we are interested in two particular classes in the WMI repository:

- ▶ **Win32\_PerfRawData**

The Win32\_PerfRawData class and its subclasses were introduced with Windows 2000. They provide access to the raw data that is made available to the Performance console.

- ▶ **Win32\_PerfFormattedData**

The Win32\_PerfFormattedData class and subclasses were introduced with Windows Server 2003. They include processed data. This data is the same as displayed by the Performance console.

For our purposes, the Win32\_PerfRawData class suffices because it runs on Windows 2000, Windows Server 2003, and Windows Server 2008 computers. Win32\_PerfRawData has a number of subclasses, and these hold the data in which we are interested. Table 17-5 shows some of the more commonly used Performance console objects and their equivalent classes in WMI. Note that these are not the only performance classes that are available in WMI.

*Table 17-5 Performance console objects and their equivalent WMI classes*

Performance console object	Class
Cache	Win32_PerfRawData_PerfOS_Cache
LogicalDisk	Win32_PerfRawData_PerfDisk_LogicalDisk
Memory	Win32_PerfRawData_PerfOS_Memory
Paging File	Win32_PerfRawData_PerfOS_PagingFile
PhysicalDisk	Win32_PerfRawData_PerfDisk_PhysicalDisk
Process	Win32_PerfRawData_PerfProc_Process
Processor	Win32_PerfRawData_PerfOS_Processor
Server	Win32_PerfRawData_PerfNet_Server

Example 17-1 on page 595 illustrates our example scenario. We have a large number of servers and want to monitor the memory usage of each. We want to retrieve this data and record it in a file for review. Example 17-1 on page 595 shows a VBScript that collects data from multiple servers and then outputs it to the console.

**Note:** You can download this script by going to the following address and clicking **Additional Materials**:

<http://www.redbooks.ibm.com/abstracts/sg245287.html>

*Example 17-1 Perfdta.vbs: collecting performance data using WMI*

```
• **** IBM ITSO ****  
• Script Name = perdata.vbs  
• Version = 1.0  
• Author = Brian Jeffery  
• Description = Collect performance data from multiple servers  
•  
  
• **** History ****  
• Date          Version      Description  
• ~~~~~~  
• 19/07/2004    1.0          Initial Release  
• ****
```

## Option Explicit

```
‘ **** Define Global Constants
Const FORREADING = 1

‘ **** Declare Objects
Dim oArgs          ‘Wscript.Arguments object
Dim oFS            ‘Scripting.FileSystemObject
Dim oInStream      ‘Stream with text input from file

‘ **** Declare Variables
Dim sServerListFile   ‘name of server file
Dim sServer          ‘individual server to connect to

‘ **** Initialise Objects
Set oFS = CreateObject("Scripting.FileSystemObject")
Set oArgs = Wscript.Arguments
```

```

' **** Ensure Cscript.exe used to run script
If (LCASE(RIGHT(WSCRIPT.FULLNAME, 11)) = "wscript.exe") Then
    Usage("Script must be run using cscript.exe")
End If

' **** populate variables with values from argument
If oArgs.Count <> 1 Then
    'If the number of arguments not equal to 1 exit gracefully
    Usage("Invalid number of arguments supplied")
Else
    'Arguments = 1 then set the sServerListFile variable to its value
    sServerListFile = oArgs(0)
End If

' **** open server list
If oFS.FileExists(sServerListFile) Then
    ' file found. Now open it
    Set oInstream = oFS.OpenTextFile(sServerListFile, FORREADING)
Else
    'text file not found, display usage and exit
    Usage("Unable to open " & sServerListFile)
End If

' **** loop through each line of the text file.  Each line should
'      correspond to a text file.
Do Until oInstream.AtEndOfStream
    sServer = Trim(oInstream.ReadLine)
    'Remove any leading backslashes
    If InStr(sServer, "\\") = 1 Then
        sServer = Mid(sServer, 3)
    End If
    'run sub routine collectdata, supplying name of server
    CollectData(sServer)
Loop

' ****
' End of Main script
' ****

' ****
' Sub CollectData
' ****

Sub CollectData(p_sComputerName)

```

```

' **** Declare objects
Dim oWMI      'Ref to WinMgmts
Dim colItems   'Collection of objects returned by the query
Dim oItem      'Individual object from the collection

' **** Declare Variables
Dim sQuery
Dim sConnect

' **** connect to WinMgmts on remote server
Set oWMI = GetObject("WinMgmts:{authenticationLevel=pkt}!\"_
& p_sComputerName)

' **** set WMI query
sQuery = "SELECT * FROM Win32_PerfRawData_PerfOS_Memory"

' **** Execute WMI Query
Set colItems = oWMI.ExecQuery(sQuery)

' **** Now Display the results
For Each oItem in colItems
    Wscript.Echo p_sComputerName
    Wscript.Echo vbTab & "Available Memory = " &
        oItem.AvailableMBytes & " MB"

    Wscript.Echo vbTab & "Commit Limit      = " &
        Round(oItem.CommitLimit/1024^2, 2) & " MB"

    Wscript.Echo vbTab & "Committed MB      = " &
        Round(oItem.CommittedBytes/1024^2, 2) & " MB"

    Wscript.Echo
Next
End Sub

' ****
' End Sub CollectData
' ****

' ****
' Sub Usage
' ****

Sub Usage(p_sError)

```

```

Dim sText

sText = "Perfdata.vbs failed to run because:" & vbCrLf & vbCrLf
sText = sText & vbTab & p_sError & vbCrLf & vbCrLf
sText = sText & "Usage:" & vbCrLf & vbCrLf
sText = sText & vbTab & "cscript /nologo perfdata.vbs <text file>"
sText = sText & vbCrLf & vbCrLf & "Where:" & vbCrLf & vbCrLf
sText = sText & vbTab & "<text file> refers to the file with list of "
sText = sText & vbCrLf & "& servers"
Wscript.Echo sText
' exit the script
Wscript.Quit
End Sub

' *****
' End Sub Usage
' *****

```

---

The steps to use this script are as follows:

1. Download the zip file from the **Additional Materials** link at the following address, and unzip the file.  
<http://www.redbooks.ibm.com/abstracts/sg245287.html>
2. Create a text file and enter the names of each of your servers (see Example 17-2) for a sample. Save this file in the same folder as the script file.

**Note:** The script is simple and assumes the file name of this text file has no spaces in it.

*Example 17-2 A possible list of servers - saved as servers.txt*

---

```

chaumes
paws
gbrhmx097

```

---

3. Run the script. Assuming that you saved the list of servers as servers.txt, you would run the script as follows:  
cscript.exe /nologo perfdata.vbs servers.txt
4. Assuming that everything runs correctly, you should get an output similar to that shown in Example 17-3.

*Example 17-3 Example output of perfdata.vbs*

---

chaumes

```
Available Memory = 48 MB
Commit Limit     = 495.23 MB
Committed MB     = 68.02 MB
```

paws

```
Available Memory = 69 MB
Commit Limit     = 558.77 MB
Committed MB     = 126.07 MB
```

gbrhmx097

```
Available Memory = 105 MB
Commit Limit     = 1694.14 MB
Committed MB     = 820.14 MB
```

---

Each server is listed in turn with three pieces of data:

- ▶ Available Memory: lists the available physical RAM
- ▶ Commit Limit: the size of the paging file
- ▶ Committed MB: the actual amount of the paging file that is used

To output this data to a file, simply pipe the output to a text file through the command line, as in:

```
cscript /nologo perfdata.vbs servers.txt >output.txt
```

If you want the script to append existing data rather than overwrite then simply replace the `>` with a `>>`.

**Tip:** To save any possible errors to the output text file along with the data, then add `2>&1` to the end of the command string:

```
cscript /nologo perfdata.vbs servers.txt >output.txt 2>&1
```

For more information about WMI and performance data, see:

[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wmisdk/wmi/performance\\_counter\\_classes.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wmisdk/wmi/performance_counter_classes.asp)

**Note:** We have not included a detailed description of perfdta.vbs, because the comments in the script should be sufficient for anyone who is familiar with VBScript and WMI.

However, if you are not familiar with either WMI or VBScript and want to know more, we recommend *Windows Management Instrumentation (WMI)*, New Riders, by Matthew Lavy and Ashley Meggitt, ISBN 1578702607.

## 17.6 VTune

VTune™ is a software tool from Intel that helps you analyze your system and applications. It works across the range of Intel architectures to detect hotspots, which are areas in your code that take long time to execute.

VTune collects performance data on your system and displays the result in a graphical surface to help you determine the hotspots and what is causing them. It also helps you decide how to eliminate them. VTune allows you to track areas such as where the processor is spending its time, where misaligned memory references occur, or where branch mispredictions happen. The VTune utility is available for Windows or for Linux operating systems.

You can obtain a 30-day trial version of the latest VTune software from Intel at:

<https://registrationcenter.intel.com/EvalCenter/EvalForm.aspx?ProductID=585>

The hardware and software requirements for using this tool are listed at:

<http://www.intel.com/cd/software/products/asmo-na/eng/220001.htm>

You can start the application going to **Start → All Programs → Intel Software Development Tools → Intel VTune Performance Analyzer → Intel VTune Performance Analyzer**.

When you run VTune, it collects data and displays it in a graphical interface. Figure 17-48 on page 601 shows the standard collection view. Each line in the chart represents data in a specific performance counter.

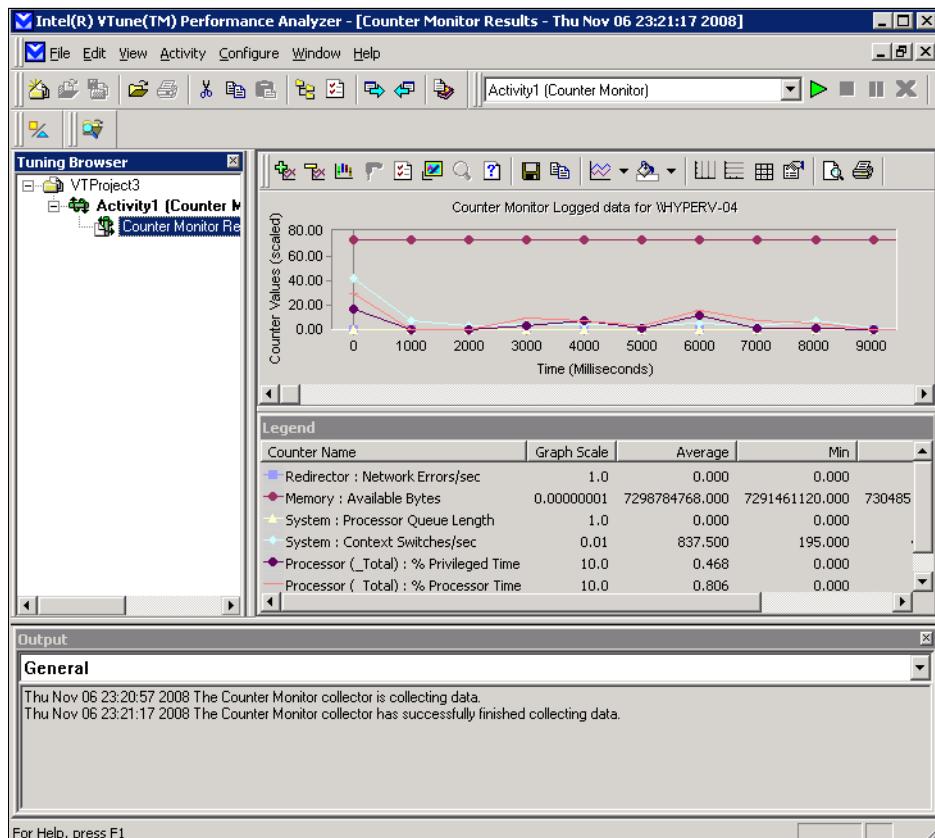


Figure 17-48 VTune sample collection

The counters are listed in the lower pane. You can highlight a counter in the chart by double-clicking it, and you can get an explanation of the purpose of the counter by right-clicking, as shown in Figure 17-49.

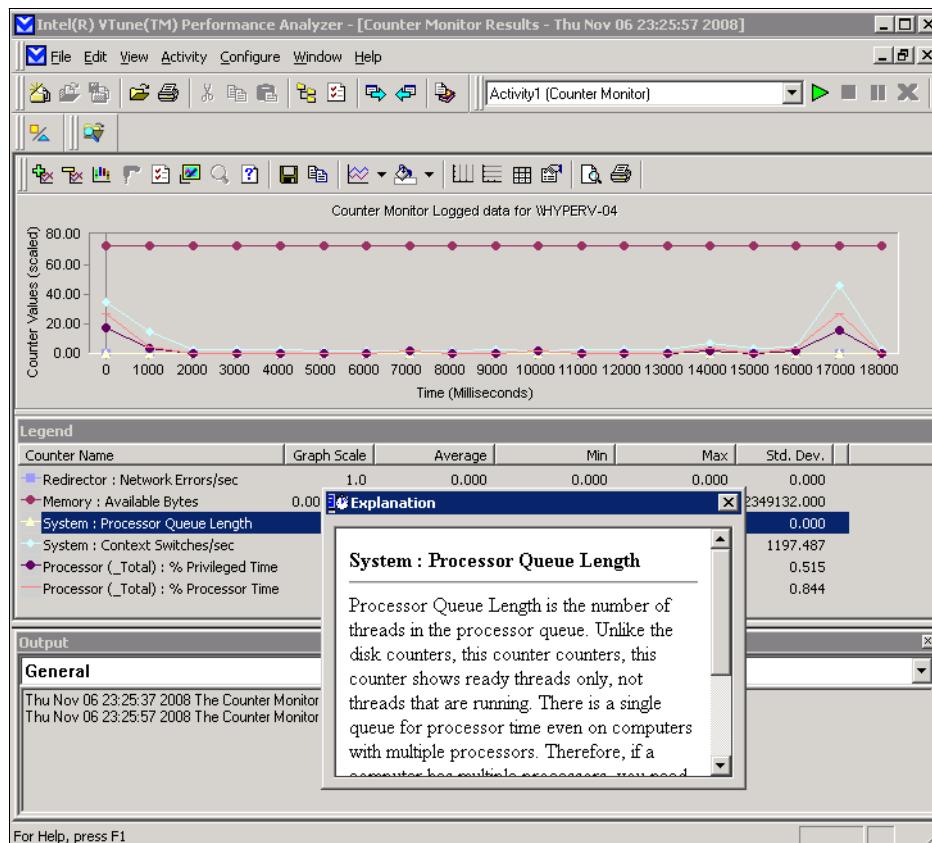


Figure 17-49 Counter explanation

You can also see an analysis of the data, as shown in Figure 17-50.

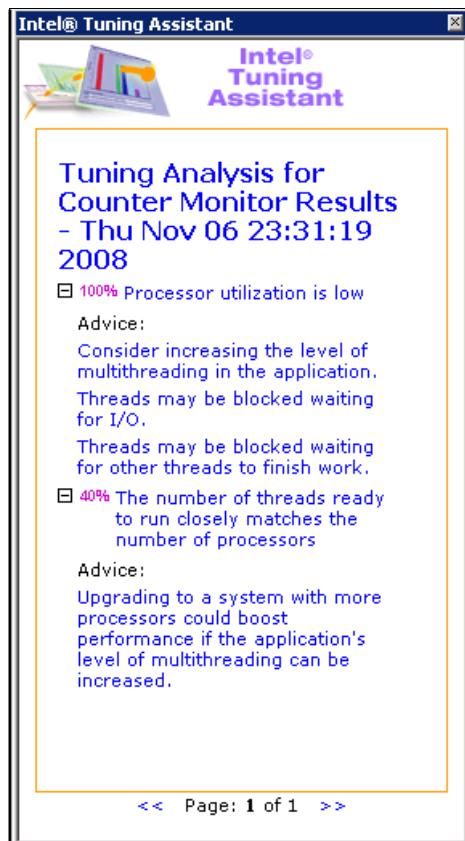


Figure 17-50 Tuning analysis

Other views are also available. A summary view shows the data for each counter represented as a bar diagram. The summary view shows the minimum, maximum, and average value of your counter, as shown in Figure 17-51.

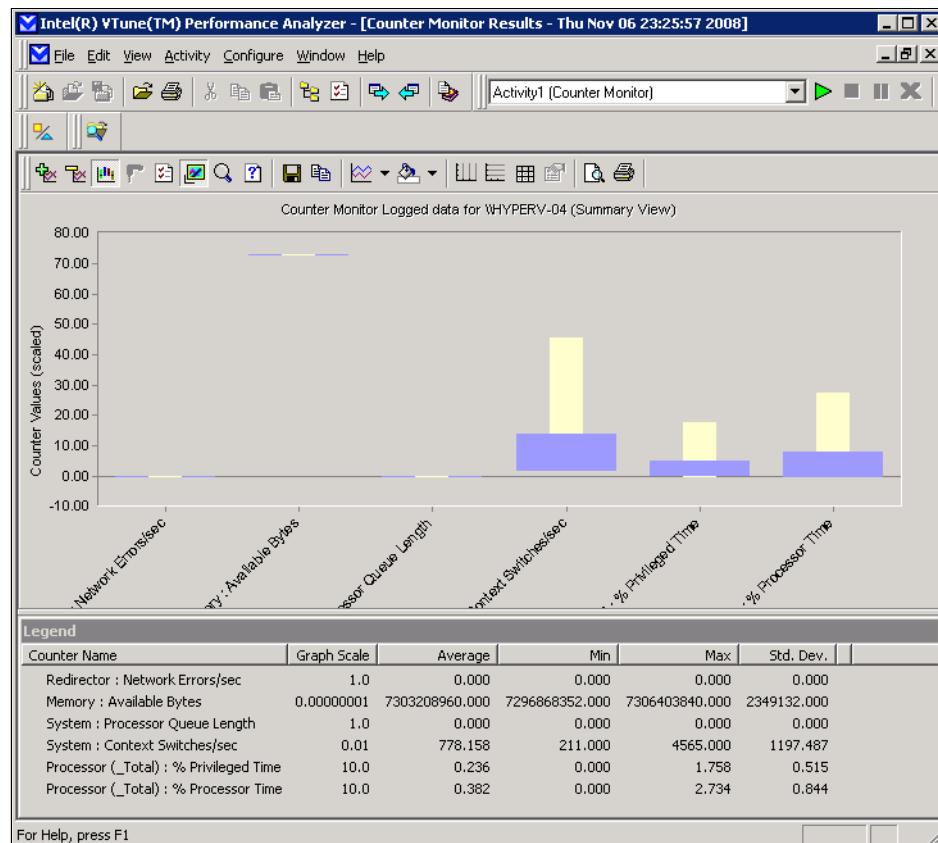


Figure 17-51 Summary view

To analyze hotspots and bottlenecks with this tool, it is important that you understand what each counter means. Read the explanation of each counter, like the one shown in Figure 17-49 on page 602, if you are unsure of its meaning. In addition, VTune help includes animated examples to better illustrate how to analyze the data.

You can use VTune to record server behavior over time under real-life conditions. For example, you can take samples at the same time each week, at different times of the day, or before and after making changes. Doing so, you can quickly recognize trends, identify workload peaks, and avoid performance problems.

If you need further information, we recommend that you download the evaluation version of VTune and read the documentation that is available with that version.

You can obtain a 30-day trial version of the latest VTune software from Intel at:

<https://registrationcenter.intel.com/EvalCenter/EvalForm.aspx?ProductID=585>

The evaluation version includes animated examples that provide an overview of how to use this tool.





## Linux tools

The open and flexible nature of the Linux operating system has led to a significant number of performance monitoring tools. Some of them are Linux versions of well-known UNIX utilities, and others were specifically designed for Linux. The fundamental support for most Linux performance monitoring tools is with the virtual proc file system. To measure performance, you also have to use appropriate benchmark tools.

This chapter outlines a selection of Linux performance monitoring tools and discusses useful commands. Useful benchmark tools are also introduced.

Most of the monitoring tools discussed here ship with Enterprise Linux distributions.

## 18.1 Introduction to Linux tools

The Enterprise Linux distributions are shipped with many monitoring tools. Some of the tools deal with several metrics all in the one tool and provide well-formatted output for easy understanding of system activities. Other tools are specific to certain performance metrics (such as disk I/O) and provide detailed information on just that part of the system.

Being familiar with these tools and understanding the data they provide helps you to enhance your understanding of the system and to find the root causes of a performance problem.

Table 18-1 lists the tools that we discuss in this chapter.

*Table 18-1 Linux performance monitoring tools*

Tool	Most useful tool function	Page
uptime	Average system load	609
dmesg	Hardware and system information	610
top	Processor activity	611
iostat	Average CPU load and disk activity	613
vmstat	System activity	615
sar	Collect and report system activity	615
numastat	NUMA monitoring tool	617
KDE System Guard	Real time systems reporting and graphing	617
free	Memory usage	625
traffic-vis	Network monitoring (SUSE Linux Enterprise Server only)	625
pmap	Process memory usage	628
strace	Programs	629
ulimit	System limits	630
mpstat	Multiprocessor usage	631
xPL	The System x Performance Logger for Linux (also known as PLPerf)	632
nmon	IBM-developed tool showing data from the /proc file system	642

## 18.2 The uptime command

You can use the **uptime** command to see how long the server has been running and how many users are logged on. You can also use it to obtain a quick overview of the average load of the server.

The system load average is displayed for the last 1-, 5-, and 15-minute intervals. Note that the load average is not a percentage. Instead, it is the number of processes in the queue that are waiting to be processed. If the processes that request CPU time are blocked (which means that the CPU has no time to process them), the load average increases. Alternatively, if each process gets immediate access to CPU time and no CPU cycles are lost, the load decreases. The optimal value of the load is 1 in single-core systems, which means that each process has immediate access to the CPU and there are no CPU cycles lost.

The typical load can vary from system to system. For a uniprocessor workstation, 1 or 2 might be acceptable, while you might see values of 8 to 10 on multiprocessor servers. For SMP systems, the optimal value would be the number of cores divided by the number of threads. Example 18-1 displays sample uptime output.

You can use the **uptime** command to pinpoint an issue with your server or with the network. For example, if a network application is running poorly, you can run **uptime**, and you will see whether the system load is high. If the system load is not high, the problem might be related to your network rather than to your server.

**Tip:** You can also use **w** instead of **uptime**, which also provides information about who is logged on to the machine and what the user is doing.

*Example 18-1 Sample output of uptime*

---

```
1:57am  up 4 days 17:05,  2 users,  load average: 0.00, 0.00, 0.00
```

---

## 18.3 The dmesg command

The main purpose of using the **dmesg** command is to display kernel messages. This command can provide helpful information in case of hardware issues or issues with loading a module into the kernel. Example 18-2 shows partial output from the **dmesg** command.

In addition, with **dmesg**, you can determine what hardware is installed in your server. During every boot, Linux checks your hardware and logs information about it. You can view these logs using the command **/bin/dmesg**.

*Example 18-2 Partial output from dmesg*

---

```
EXT3 FS 2.4-0.9.19, 19 August 2002 on sd(8,1), internal journal
EXT3-fs: mounted filesystem with ordered data mode.
IA-32 Microcode Update Driver: v1.11 <tigran@veritas.com>
ip_tables: (C) 2000-2002 Netfilter core team
3c59x: Donald Becker and others. www.scyld.com/network/vortex.html
See Documentation/networking/vortex.txt
01:02.0: 3Com PCI 3c980C Python-T at 0x2080. Vers LK1.1.18-ac
    00:01:02:75:99:60, IRQ 15
        product code 4550 rev 00.14 date 07-23-00
        Internal config register is 3800000, transceivers 0xa.
        8K byte-wide RAM 5:3 Rx:Tx split, autoselect/Autonegotiate interface.
        MII transceiver found at address 24, status 782d.
        Enabling bus-master transmits and whole-frame receives.
01:02.0: scatter/gather enabled. h/w checksums enabled
divert: allocating divert_blk for eth0
ip_tables: (C) 2000-2002 Netfilter core team
Intel(R) PRO/100 Network Driver - version 2.3.30-k1
Copyright (c) 2003 Intel Corporation

divert: allocating divert_blk for eth1
e100: selftest OK.
e100: eth1: Intel(R) PRO/100 Network Connection
    Hardware receive checksums enabled
    cpu cycle saver enabled

ide-floppy driver 0.99.newide
hda: attached ide-cdrom driver.
hda: ATAPI 48X CD-ROM drive, 120kB Cache, (U)DMA
Uniform CD-ROM driver Revision: 3.12
Attached scsi generic sg4 at scsil1, channel 0, id 8, lun 0, type 3
```

---

## 18.4 The top command

The **top** command shows actual processor activity. By default, it displays the most CPU-intensive tasks that are running on the server, and it updates the list every five seconds. You can sort the processes by processor ID (numerically), age (newest first), time (cumulative time), and resident memory usage and time (time the process has occupied the CPU since startup).

*Example 18-3 Example output from top command*

```
top - 02:06:59 up 4 days, 17:14, 2 users, load average: 0.00, 0.00, 0.00
Tasks: 62 total, 1 running, 61 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.2% us, 0.3% sy, 0.0% ni, 97.8% id, 1.7% wa, 0.0% hi, 0.0% si
Mem: 515144k total, 317624k used, 197520k free, 66068k buffers
Swap: 1048120k total, 12k used, 1048108k free, 179632k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
13737	root	17	0	1760	896	1540	R	0.7	0.2	0:00.05	top
238	root	5	-10	0	0	0	S	0.3	0.0	0:01.56	reiserfs/0
1	root	16	0	588	240	444	S	0.0	0.0	0:05.70	init
2	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
3	root	34	19	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/0
4	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/1
5	root	34	19	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/1
6	root	5	-10	0	0	0	S	0.0	0.0	0:00.02	events/0
7	root	5	-10	0	0	0	S	0.0	0.0	0:00.00	events/1
8	root	5	-10	0	0	0	S	0.0	0.0	0:00.09	kblockd/0
9	root	5	-10	0	0	0	S	0.0	0.0	0:00.01	kblockd/1
10	root	15	0	0	0	0	S	0.0	0.0	0:00.00	kirqd
13	root	5	-10	0	0	0	S	0.0	0.0	0:00.02	khelper/0
14	root	16	0	0	0	0	S	0.0	0.0	0:00.45	pdflush
16	root	15	0	0	0	0	S	0.0	0.0	0:00.61	kswapd0
17	root	13	-10	0	0	0	S	0.0	0.0	0:00.00	aio/0
18	root	13	-10	0	0	0	S	0.0	0.0	0:00.00	aio/1

You can further modify the processes using **renice** to give a new priority to each process. If a process hangs or occupies too much CPU, you can kill the process (using the **kill** command).

The columns in the output are as follows:

- PID      Process identification.
- USER     Name of the user who owns (and perhaps started) the process.
- PRI      Priority of the process (see 18.4.1, “Process priority and nice levels” on page 612 for details).

NI	Niceness level (that is, whether the process tries to be nice by adjusting the priority by the number given) See 18.4.1, “Process priority and nice levels” on page 612 for more information.
SIZE	Amount of memory (code+data+stack) in KB that are being used by the process.
RSS	Amount of physical RAM used in KB.
SHARE	Amount of memory shared with other processes in KB.
STAT	State of the process: S=sleeping, R=running, T=stopped or traced, D=interruptible sleep, Z=zombie. Zombie processes are discussed further in 18.4.2, “Zombie processes” on page 613.
%CPU	Share of the CPU usage (since the last screen update).
%MEM	Share of physical memory.
TIME	Total CPU time that is used by the process since it was started.
COMMAND	Command line used to start the task (including parameters).

**Tip:** The `/bin/ps` command gives a snapshot view of the current processes.

### 18.4.1 Process priority and nice levels

Process priority is a number that determines the order in which the process is handled by the CPU. The kernel adjusts this number up and down as needed. The *nice* value is a limit on the priority. The priority number is not allowed to go below the nice value (a lower nice value is a more favored priority).

It is not possible to change the priority of a process. This is only indirectly possible through the use of the nice level of the process. Note that it might not always be possible to change the priority of a process using the nice level. If a process is running too slowly, you can assign more CPU to it by giving it a lower nice level. Of course, doing so means that all other programs will have fewer processor cycles and will run more slowly.

Linux supports nice levels from 19 (lowest priority) to -20 (highest priority). The default value is zero (0). To change the nice level of a program to a negative number (which makes it a higher priority process), it is necessary to log on or `su` to root.

For example, to start the program xyz with a nice level of -5, issue the command:  
`nice -n -5 xyz`

To change the nice level of a program already running, issue the command:

```
renice level pid
```

To change the priority of the xyz program that has a PID of 2500 to a nice level of 10, issue the following command:

```
renice 10 2500
```

### 18.4.2 Zombie processes

When a process terminates, having received a signal to do so, it normally takes some time to finish all tasks (such as closing open files) before the process terminates itself. In that normally very short time frame, the process is called a *zombie* process. After the process has completed all the shutdown tasks, it reports to the parent process that it is about to terminate. Sometimes, a zombie process is unable to terminate itself, in which case you will see that it has a status of Z (zombie).

It is not possible to kill such a process with the **kill** command, because it is already considered *dead*. If you cannot kill a zombie process, you can kill the parent process and then the zombie disappears as well. However, if the parent process is the init process, you should not kill it. The init process is a very important process. Therefore, you might need to reboot to get rid of the zombie process.

## 18.5 The iostat command

The **iostat** command is part of the Sysstat set of utilities, which come with most of the main Linux distributions. It is also available from its creator Web site:

<http://perso.wanadoo.fr/sebastien.godard/>

The **iostat** command lets you see the average CPU times since the system was started, in a way similar to **uptime**. In addition, however, **iostat** creates a report about the activities of the disk subsystem of the server. The report has two parts: CPU utilization and device (disk) utilization. To learn how to use **iostat** to perform detailed I/O bottleneck and performance tuning, refer to 22.4.1, “Finding bottlenecks in the disk subsystem” on page 734. Example 18-4 on page 614 shows sample output for the **iostat** command.

*Example 18-4 Sample output of iostat*

---

Linux 2.4.21-9.0.3.EL (x232) 05/11/2004

avg-cpu:	%user	%nice	%sys	%idle	
	0.03	0.00	0.02	99.95	
Device:	tps	Blk_read/s	Blk_wrtn/s	Blk_read	Blk_wrtn
dev2-0	0.00	0.00	0.04	203	2880
dev8-0	0.45	2.18	2.21	166464	168268
dev8-1	0.00	0.00	0.00	16	0
dev8-2	0.00	0.00	0.00	8	0
dev8-3	0.00	0.00	0.00	344	0

---

The CPU utilization report has four sections:

- %user Shows the percentage of CPU utilization that was taken up while executing at the user level (applications).
- %nice Shows the percentage of CPU utilization that was taken up while executing at the user level with a nice priority. (For more information about priority and nice levels, see 18.4.1, “Process priority and nice levels” on page 612.)
- %sys Shows the percentage of CPU utilization that was taken up while executing at the system level (kernel).
- %idle Shows the percentage of time the CPU was idle.

The device utilization report is split into the following sections:

- ▶ Device: The name of the block device.
- ▶ tps: The number of transfers per second (I/O requests per second) to the device. Multiple single I/O requests can be combined in a transfer request, because a transfer request can have different sizes.
- ▶ Blk\_read/s, Blk\_wrtn/s: Blocks read and written per second indicate data read/written from/to the device in seconds. Blocks can also have different sizes. Typical sizes are 1024, 2048 or 4096 bytes, depending on the partition size. For example, the block size of /dev/sda1 can be found with:

```
dumpe2fs -h /dev/sda1 |grep -F "Block size"
```

This gives an output similar to:

```
dumpe2fs 1.34 (25-Jul-2003)
Block size: 1024
```

- ▶ Blk\_read, Blk\_wrtn: This indicates the total number of blocks read/written since the boot.

## 18.6 The vmstat command

The `vmstat` command provides information about processes, memory, paging, block I/O, traps, and CPU activity.

*Example 18-5 Example output from vmstat*

procs	-----memory-----				---swap--		-----io-----		--system--		----cpu----				
r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id	wa
2	0	0	154804	77328	910900	0	0	4	6	103	19	0	0	100	0

The columns in the output are as follows:

- ▶ Process
  - r The number of processes waiting for runtime.
  - b The number of processes in uninterruptable sleep.
- ▶ Memory
  - swpd The amount of virtual memory used (KB).
  - free The amount of idle memory (KB).
  - buff The amount of memory used as buffers (KB).
- ▶ Swap
  - si Amount of memory swapped from the disk (KBps).
  - so Amount of memory swapped to the disk (KBps).
- ▶ I/O
  - bi Blocks sent to a block device (blocks/s).
  - bo Blocks received from a block device (blocks/s).
- ▶ System
  - in The number of interrupts per second, including the clock.
  - cs The number of context switches per second.
- ▶ CPU (these are percentages of total CPU time)
  - us Time spent running non-kernel code (user time, including nice time).
  - sy Time spent running kernel code (system time).
  - id Time spent idle. Prior to Linux 2.5.41, this included IO-wait time.
  - wa Time spent waiting for IO. Prior to Linux 2.5.41, this appeared as zero.

## 18.7 The sar command

The `sar` command is also part of the Sysstat set of utilities, as are some of the tools reviewed in this chapter. The `sar` command is used to collect, report, or save system activity information. The `sar` command consists of three

applications: **sar** displays the data, and **sa1** and **sa2** are used for collecting and storing the data.

By using **sa1** and **sa2**, you can configure the system to obtain the information and log it for later analysis. To do this, you must configure a cron job by adding the lines shown in Example 18-6 to the /etc/crontab file.

---

*Example 18-6 Example of starting automatic log reporting with cron*

---

```
# 8am-7pm activity reports every 10 minutes during weekdays.  
*/10 8-18 * * 1-5 /usr/lib/sa/sa1 600 6 &  
# 7pm-8am activity reports every an hour during weekdays.  
0 19-7 * * 1-5 /usr/lib/sa/sa1 &  
# Activity reports every an hour on Saturday and Sunday.  
0 * * * 0,6 /usr/lib/sa/sa1 &  
# Daily summary prepared at 19:05  
5 19 * * * /usr/lib/sa/sa2 -A &
```

---

Alternatively, you can use **sar** to run almost real-time reporting from the command line, as shown in Example 18-7.

From the collected data, you get a detailed overview of your CPU utilization (%user, %nice, %system, %idle), memory paging, network I/O, and transfer statistics, process creation activity, activity for block devices, and interrupts/second over time.

---

*Example 18-7 Ad hoc CPU monitoring*

---

```
[root@x232 root]# sar -u 3 10  
Linux 2.4.21-9.0.3.EL (x232)      05/22/2004  
  
02:10:40 PM      CPU      %user      %nice      %system      %idle  
02:10:43 PM      all       0.00       0.00       0.00     100.00  
02:10:46 PM      all       0.33       0.00       0.00      99.67  
02:10:49 PM      all       0.00       0.00       0.00     100.00  
02:10:52 PM      all       7.14       0.00      18.57      74.29  
02:10:55 PM      all      71.43       0.00      28.57       0.00  
02:10:58 PM      all       0.00       0.00     100.00       0.00  
02:11:01 PM      all       0.00       0.00       0.00       0.00  
02:11:04 PM      all       0.00       0.00     100.00       0.00  
02:11:07 PM      all      50.00       0.00      50.00       0.00  
02:11:10 PM      all       0.00       0.00     100.00       0.00  
Average:         all      1.62       0.00      3.33      95.06
```

---

## 18.8 numastat

The **numastat** utility is a recent addition to the standard monitoring tools for Linux systems. It is provided with the numactl package and is especially useful on IBM System x hardware that features a Non-Uniform Memory Architecture (NUMA). This is the kind of memory architecture that can be found on IBM high-end servers, like the x3950 M2 server, where up to four servers can be connected together to form one large single-image complex.

This command will provide information about the system. In the output, three categories should be analyzed: numa\_miss, numa\_foreign, and other\_node, compared to the total amount of physical RAM installed in order to identify potential bottlenecks. In NUMA machines, other tools may not be accurate enough, because they will consider the architecture as non-NUMA, not considering the specifics of the system.

**Note:** In the output of the command, the values to watch have the following meanings:

- ▶ **numa\_miss**: a process wanted to allocate memory for its node, but ended up with memory of another node.
- ▶ **numa\_foreign**: a process wanted to allocate memory on other node, but ended up with memory from its node.
- ▶ **other\_node**: a process ran from its node, but got memory from another node.

The **numastat** tool can also be useful if you want to run a process in a specific NUMA node (or CPU core). However, this use is generally not recommended because it increases the complexity of operation in production, and the Linux kernel scheduler is typically more efficient than a manual CPU affinity selection.

## 18.9 KDE System Guard

KDE System Guard (KSysguard) is the KDE task manager and performance monitor. It features a client/server architecture that enables monitoring of local hosts and remote hosts.

The graphical front end (shown in Figure 18-1 on page 618) uses sensors to retrieve the information that it displays. A sensor can return simple values or more complex information such as tables. For each type of information, one or more displays are provided. Displays are organized into work sheets that can be saved and loaded independently of each other.

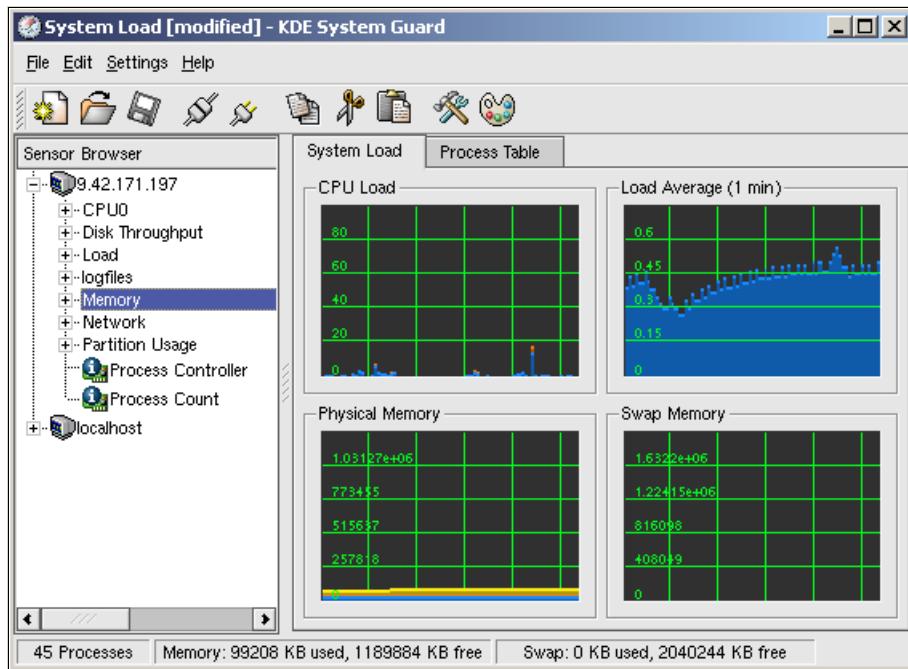


Figure 18-1 Default KDE System Guard window

The KSysguard main window (Figure 18-1) consists of a menu bar, an optional tool bar and status bar, the sensor browser, and the work space. When first started, you see your local machine listed as `localhost` in the sensor browser, and two tabs in the work space area. This is the default setup.

Each sensor monitors a certain system value. You can drag and drop all of the displayed sensors in the work space. There are three options:

- ▶ You can delete and replace sensors in the actual work space.
- ▶ You can edit work sheet properties and increase the number of rows and columns.
- ▶ You can create a new work sheet and drop new sensors to meet your needs.

### 18.9.1 The KSysguard work space

Looking at the work space in Figure 18-2 on page 619, notice that there are two tabs:

- ▶ System Load, the default view when first starting up KSysguard
- ▶ Process Table

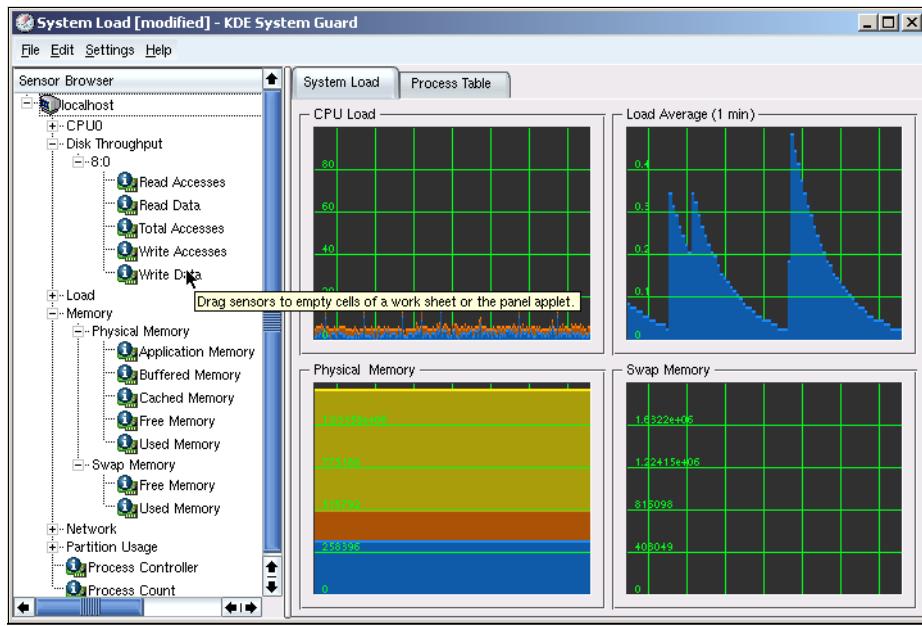


Figure 18-2 KDE System Guard sensor browser

## System Load

The System Load work sheet consists of four sensor windows:

- ▶ CPU Load
- ▶ Load Average (1 Minute)
- ▶ Physical Memory
- ▶ Swap Memory

You will note from the Physical Memory window that it is possible to have multiple sensors displayed within one window. To determine which sensors are being monitored in a given window, mouse over the graph and some descriptive text appears. Another way to do this is to right-click the graph and click **Properties**, then go to the Sensors tab, as shown in Figure 18-3 on page 620. The Sensors tab also shows a key of what each color represents on the graph.

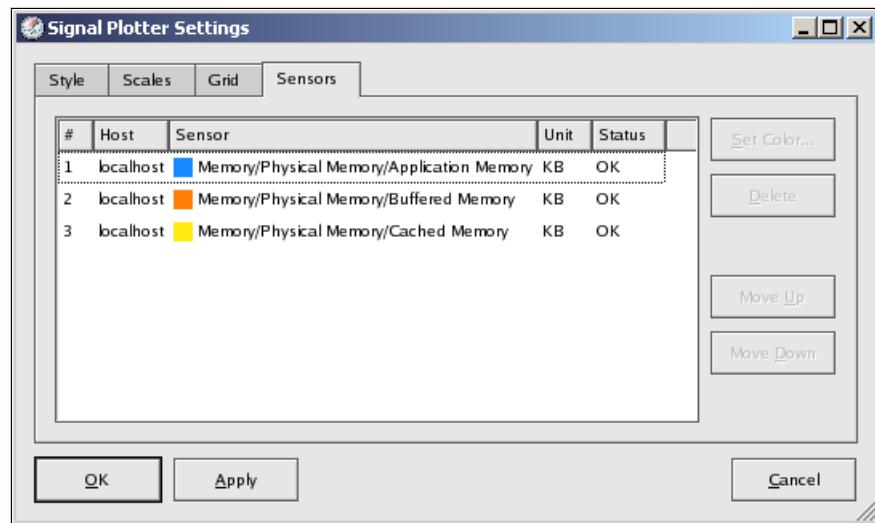


Figure 18-3 Sensor Information, Physical Memory Signal Plotter

## Process Table

The Process Table tab displays information about all the running processes on the server (Figure 18-4). The table, by default, is sorted by System CPU utilization. You can change the way the table is sorted by clicking the heading by which you want to sort.

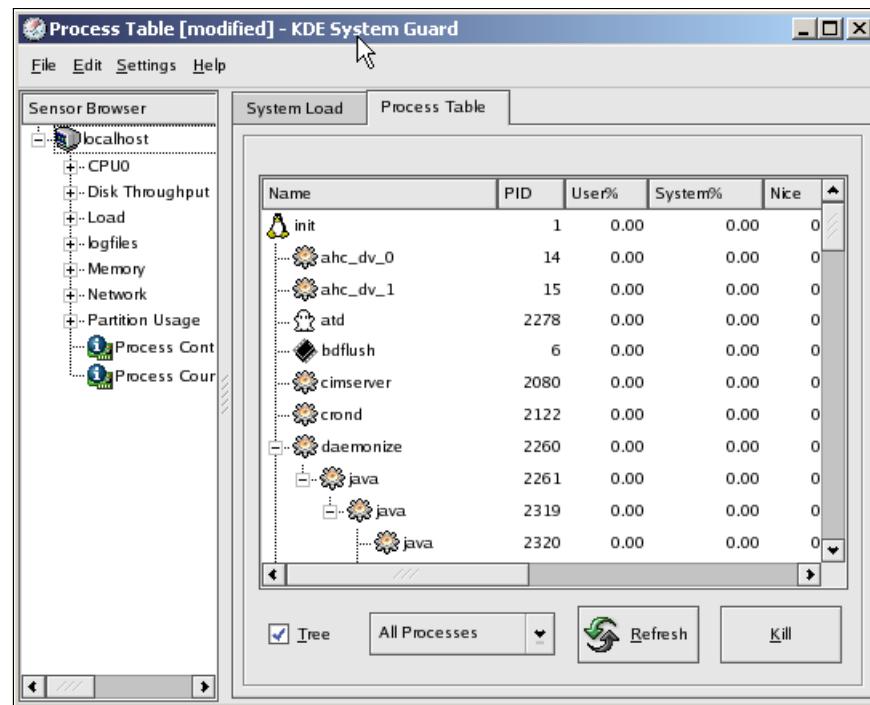


Figure 18-4 Process Table view

## Configuring a work sheet

For your environment or the particular area that you want to monitor, it might be necessary to use different sensors for monitoring. The best way to do this is to create a custom work sheet. In this section, we guide you through the steps that are required to create the work sheet that is shown in Figure 18-7 on page 624.

The steps to create a work sheet are as follows:

1. Create a blank work sheet by clicking **File** → **New** to open the window that is shown in Figure 18-5.

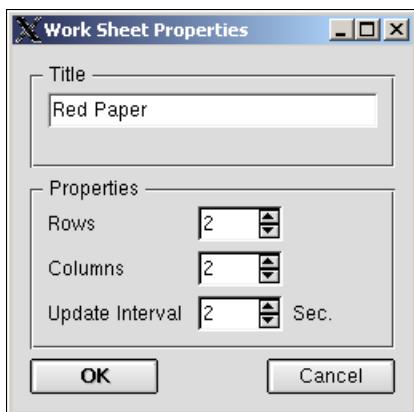


Figure 18-5 Properties for new work sheet

2. Enter a title and a number of rows and columns. This gives you the maximum number of monitor windows, which in our case is four. When the information is complete, click **OK** to create the blank work sheet, as shown in Figure 18-6 on page 623.

**Note:** The fastest update interval that can be defined is two seconds.

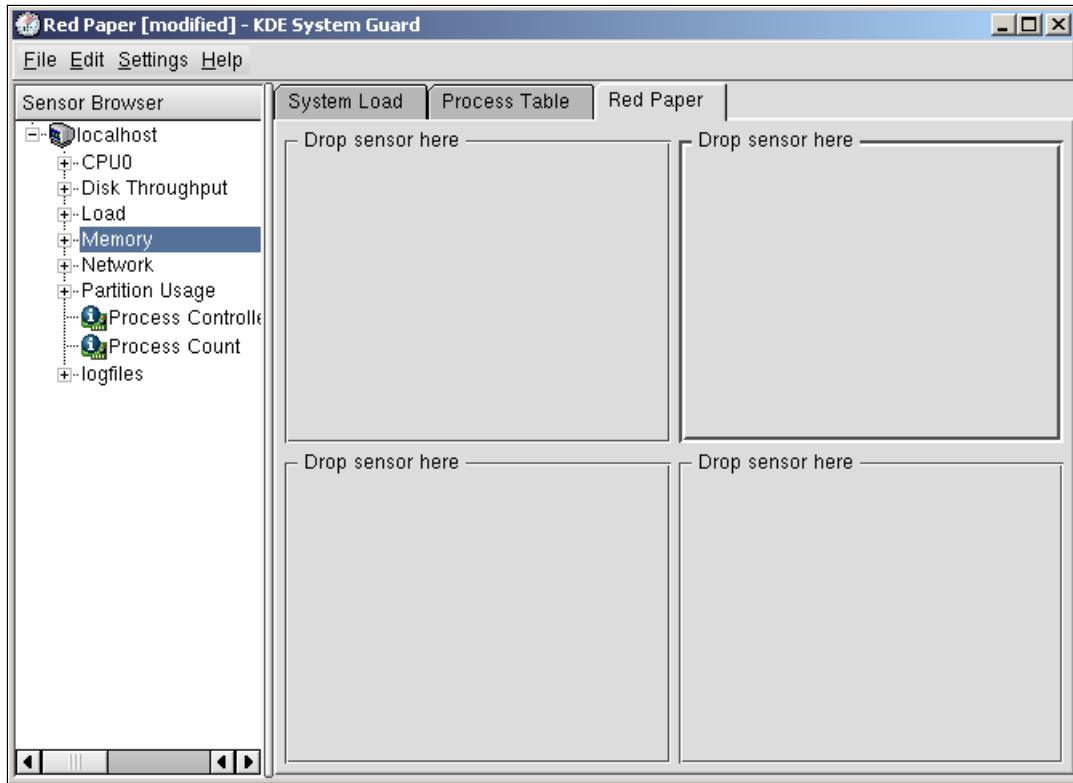


Figure 18-6 Empty work sheet

3. Now, you can complete the sensor boxes by simply dragging the sensors on the left side of the window to the desired box on the right.

The display choices are:

- **Signal Plotter.** This sensor style displays samples of one or more sensors over time. If several sensors are displayed, the values are layered in different colors. If the display is large enough, a grid is displayed to show the range of the plotted samples.

By default, the automatic range mode is active, so the minimum and maximum values are set automatically. If you want fixed minimum and maximum values, you can deactivate the automatic range mode and set the values in the Scales tab from the Properties dialog window (which you access by right-clicking the graph).

- **Multimeter.** The Multimeter displays the sensor values as a digital meter. In the properties dialog, you can specify a lower and upper limit. If the range is exceeded, the display is colored in the alarm color.

- **BarGraph**: The BarGraph displays the sensor value as dancing bars. In the properties dialog, you can also specify the minimum and maximum values of the range and a lower and upper limit. If the range is exceeded, the display is colored in the alarm color.
- **Sensor Logger**: The Sensor Logger does not display any values, but logs them in a file with additional date and time information.

For each sensor, you have to define a target log file, the time interval the sensor will be logged and whether alarms are enabled.

4. Click **File → Save** to save the changes to the work sheet.

**Note:** When you save a work sheet, it is saved in your home directory, which might prevent other administrators from using your custom work sheet.

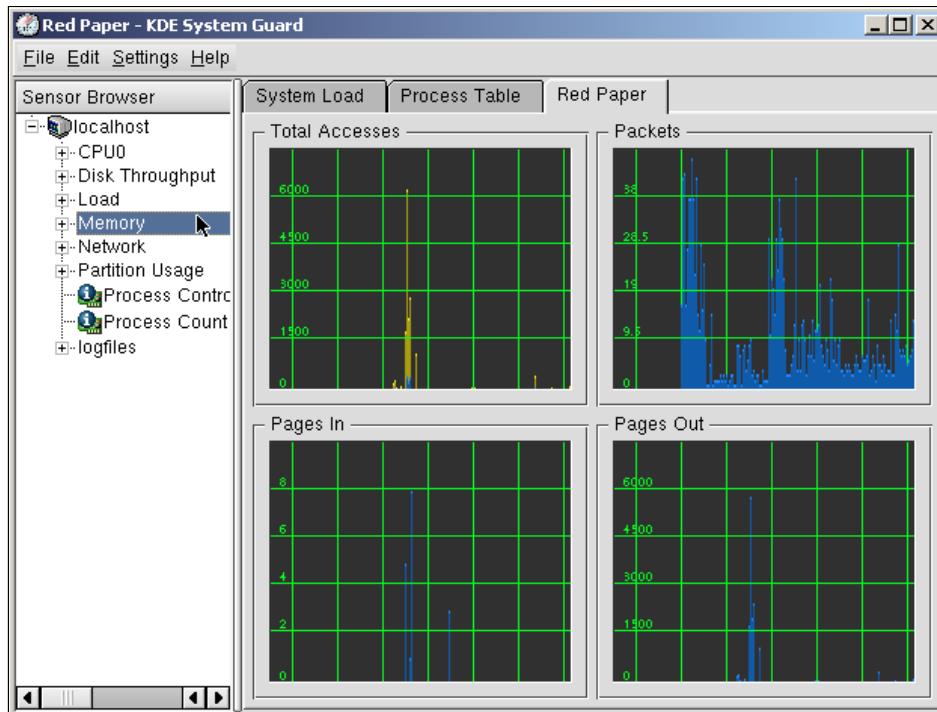


Figure 18-7 Example work sheet

You can find more information about KDE System Guard online at:

<http://docs.kde.org/en/3.2/kdebase/ksysgaurd>

## 18.10 The free command

The `/bin/free` command displays information about the total amounts of free and used memory (including swap) on the system, as shown in Example 18-8. This command also includes information about the buffers and cache used by the kernel.

*Example 18-8 Example output from the free command*

	total	used	free	shared	buffers
cached					
Mem:	1291980	998940	293040	0	89356
772016					
-/+ buffers/cache:	137568	1154412			
Swap:	2040244	0	2040244		

## 18.11 Traffic-vis

Traffic-vis is a suite of tools that determine which hosts have been communicating on an IP network, with whom they have been communicating, and the volume of communication that has taken place. The final report can be generated in plain text, HTML, or GIF.

**Note:** Traffic-vis is for SUSE Linux Enterprise Server only.

Start the program to collect data on interface eth0, for example:

```
traffic-collector -i eth0 -s /root/output_traffic-collector
```

After the program starts, it is detached from the terminal and begins the collection of data. You can control the program by using the `killall` command to send a signal to the process. For example, to write the report to disk, issue the following command:

```
killall -SIGUSR1 traffic-collector
```

To stop the collection of data, issue this command:

```
killall -SIGTERM traffic-collector
```

**Important:** Be sure to run this last command. Otherwise, your system's performance will degrade due to a lack of memory.

You can sort the output by packets, bytes, TCP connections, the total of each one, or the number of sent or received of each one. For example, to sort total packets sent and received on hosts, use this command:

```
traffic-sort -i output_traffic-collector -o output_traffic-sort -Hp
```

To generate a report in HTML format that displays the total bytes transferred, total packets recorded, total TCP connections requests, and other information about each server in the network, run, use this command:

```
traffic-tohtml -i output_traffic-sort -o output_traffic-tohtml.html
```

This output file can be displayed in a browser, as shown in Figure 18-8.

The screenshot shows a Microsoft Internet Explorer window titled "traffic-vis report - Microsoft Internet Explorer". The address bar shows the URL "C:\Documents and Settings\luciano\Desktop\SUSE24\output\_traffic-tohtml.html". The page content is as follows:

## traffic-vis report

**Collection started at Thu, 13 May 2004 03:27:32 PDT**  
**Collection concluded at Thu, 13 May 2004 03:41:06 PDT**

**Number of active hosts:** 105  
**Total bytes transferred:** 676151777  
**Total packets recorded:** 7075966  
**Total TCP connection requests:** 707650

Unknown (9.42.171.197)			
Source host	Destination host	Bytes sent	packets sent
Unknown (9.42.171.197)	Unknown (9.42.171.111)	526504	1616
Unknown (9.42.171.111)	Unknown (9.42.171.197)	45352	1034
Unknown	Unknown	421634472	3535676

Figure 18-8 Report generated by traffic-vis

To generate a report in GIF format with a width of 600 pixels and a height of 600 pixels, use the following command:

```
traffic-togif -i output_traffic-sort -o output_traffic-togif.gif -x 600 -y 600
```

Figure 18-9 shows the communication between systems in the network. You can also see that some hosts talk to others, but there are also servers that never talk to each other. This output is used typically to find broadcasts in the network. To see which servers are using IPX/SPX protocol in a TCP network and to separate both networks, remember that IPX™ is based on broadcast packets.

To pinpoint others types of issues, such as damaged network cards or duplicated IPs on networks, you need to use more specific tools, such as Ethereal, which is installed by default on SUSE Linux Enterprise Server.

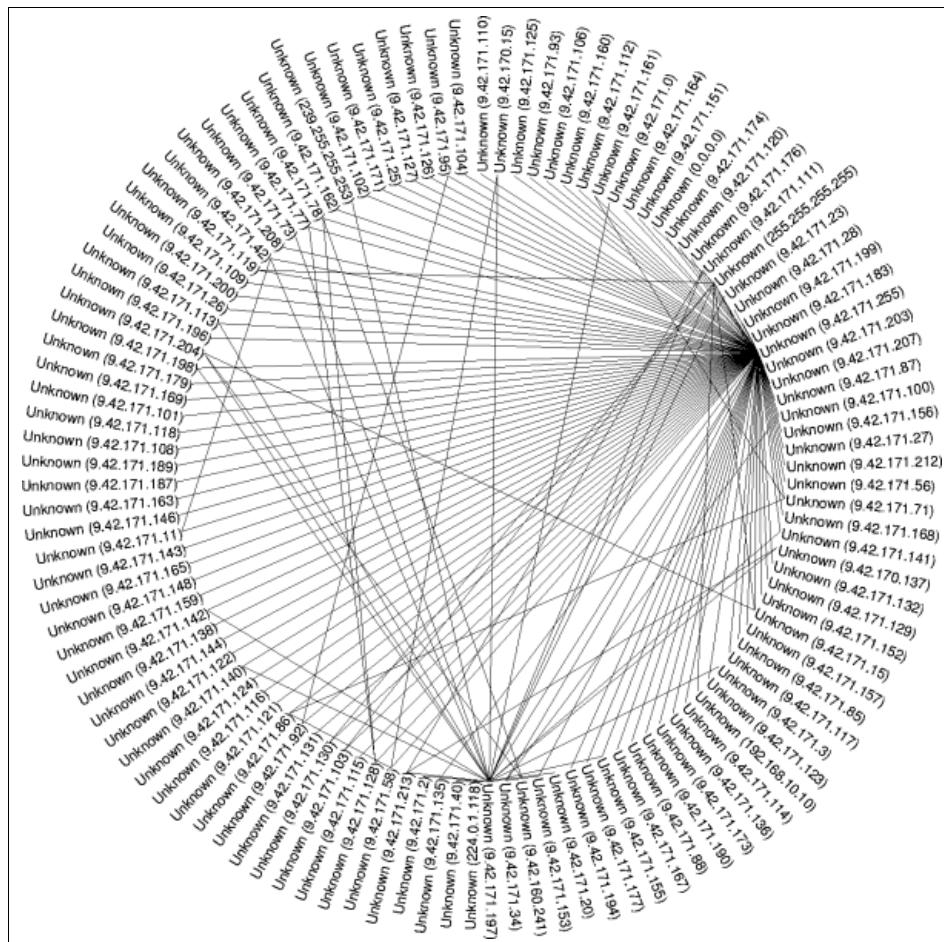


Figure 18-9 Report generated by traffic-vis

**Tip:** Using pipes, it is possible to produce output in one command. For example, to generate a report in HTML, run the following command:

```
cat output_traffic-collector | traffic-sort -Hp | traffic-tohtml -o  
output_traffic-tohtml.html
```

To generate a report as a GIF file, run:

```
cat output_traffic-collector | traffic-sort -Hp | traffic-togif -o  
output_traffic-togif.gif -x 600 -y 600
```

## 18.12 The pmap command

The **pmap** command reports the amount of memory that one or more processes are using. You can use this tool to determine which processes on the server are being allocated memory, and whether this amount of memory is a cause of memory bottlenecks.

Example 18-9 shows the result of the following command in SUSE Linux Enterprise Server.

```
pmap -x <pid>
```

*Example 18-9 Total amount of memory cupsd process is using (SLES)*

---

```
linux:~ # pmap -x 1796
1796: /usr/sbin/cupsd
Address      Kbytes      RSS      Anon  Locked Mode    Mapping
08048000       244        -        -      - r-x--  cupsd
ffffe000         4        -        -      - -----  [ anon ]
-----
total kB     6364        -        -      -
```

---

Example 18-10 shows the result of the following command in Red Hat Enterprise Linux AS.

```
pmap <pid>
```

*Example 18-10 Total amount of memory the smbd process is using*

---

```
[root@x232 root]# pmap 8359
smbd[8359]
b723c000 (1224 KB)      r-xp (08:02 1368219)  /lib/tls/libc-2.3.2.so
b736e000 (16 KB)        rw-p (08:02 1368219)  /lib/tls/libc-2.3.2.so
mapped: 9808 KB          writable/private: 1740 KB      shared: 64 KB
```

---

For the complete syntax of the **pmap** command, issue:

```
pmap -?
```

## 18.13 The strace command

The **strace** command intercepts and records the system calls that are called by a process, and the signals that are received by a process. This is a useful diagnostic, instructional, and debugging tool. System administrators will find it valuable for solving problems with programs.

To use the command, specify the process ID (PID) to be monitored as follows:

```
strace -p <pid>
```

Example 18-11 shows an example of the output of **strace**.

*Example 18-11 Output of strace monitoring httpd process*

---

```
[root@x232 html]# strace -p 815
Process 815 attached - interrupt to quit
semop(360449, 0xb73146b8, 1)          = 0
poll([{fd=4, events=POLLIN}, {fd=3, events=POLLIN, revents=POLLIN}], 2, -1) = 1
accept(3, {sa_family=AF_INET, sin_port=htons(52534),
           sin_addr=inet_addr("9.42.171.197")}, [16]) = 13
semop(360449, 0xb73146be, 1)          = 0
getsockname(13, {sa_family=AF_INET, sin_port=htons(80),
                 sin_addr=inet_addr("9.42.171.198")}, [16]) = 0
fcntl64(13, F_GETFL)                  = 0x2 (flags O_RDWR)
fcntl64(13, F_SETFL, O_RDWR|O_NONBLOCK) = 0
read(13, 0x8259bc8, 8000)            = -1 EAGAIN (Resource temporarily
unavailable)
poll([{fd=13, events=POLLIN, revents=POLLIN}], 1, 300000) = 1
read(13, "GET /index.html HTTP/1.0\r\nUser-Agent: ...", 8000) = 91
gettimeofday({1084564126, 750439}, NULL) = 0
stat64("/var/www/html/index.html", {st_mode=S_IFREG|0644, st_size=152, ...}) = 0
open("/var/www/html/index.html", O_RDONLY) = 14
mmap2(NULL, 152, PROT_READ, MAP_SHARED, 14, 0) = 0xb7052000
writev(13, [{"HTTP/1.1 200 OK\r\nDate: Fri, 14 M..."}, {"<html>\n<title>\nRedPaper Per"..., 152}], 2) = 416
munmap(0xb7052000, 152)              = 0
socket(PF_UNIX, SOCK_STREAM, 0)        = 15
connect(15, {sa_family=AF_UNIX, path="/var/run/.nscd_socket"}, 110) = -1 ENOENT (No
such file or directory)
close(15)                           = 0
```

---

For the complete syntax of the **strace** command, issue:

```
strace -?
```

## 18.14 The ulimit command

The **ulimit** command is built into the bash shell. It is used to provide control over the resources that are available to the shell and to the processes that are started by it on systems that allow such control.

You can use the **-a** option to list all parameters that you can set:

```
ulimit -a
```

*Example 18-12 Output of ulimit*

---

```
[root@x232 html]# ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
file size               (blocks, -f) unlimited
max locked memory       (kbytes, -l) 4
max memory size         (kbytes, -m) unlimited
open files              (-n) 1024
pipe size               (512 bytes, -p) 8
stack size              (kbytes, -s) 10240
cpu time                (seconds, -t) unlimited
max user processes      (-u) 7168
virtual memory           (kbytes, -v) unlimited
```

---

The **-H** and **-S** options specify the hard and soft limits that can be set for the given resource. If the soft limit is passed, the system administrator receives a warning. The hard limit is the maximum value that can be reached before the user gets the error message *Out of file handles*. For example, you can set a hard limit for the number of file handles and open files (**-n**) as follows:

```
ulimit -Hn 4096
```

For the soft limit of number of file handles and open files, use:

```
ulimit -Sn 1024
```

To see the hard and soft values, issue the command with a new value as follows:

```
ulimit -Hn
ulimit -Sn
```

You can use this command, for example, to limit Oracle users. To set it on startup, enter the follow lines, in /etc/security/limits.conf:

```
soft    nofile 4096  
hard    nofile 10240
```

In addition, for Red Hat Enterprise Linux AS, make sure that the file /etc/pam.d/system-auth has the following entry:

```
session    required    /lib/security/$ISA/pam_limits.so
```

For SUSE Linux Enterprise Server, make sure that the files /etc/pam.d/login and /etc/pam.d/sshd have the following entry:

```
session    required    pam_limits.so
```

This entry is required so that the system can enforce these limits.

For the complete syntax of the **ulimit** command, issue:

```
ulimit -?
```

## 18.15 The mpstat command

The **mpstat** command is part of the Sysstat set of utilities, which are available from:

<http://perso.wanadoo.fr/sebastien.godard/>

The **mpstat** command is used to report the activities of each the CPUs that are available on a multiprocessor server. Global average activities among all CPUs are also reported. Example 18-13 on page 632 shows example output for the **mpstat** command.

For example, use the following command to display three entries of global statistics among all processors at two-second intervals:

```
mpstat 2 3
```

**Tip:** You can use this command on non-SMP machines, as well.

---

*Example 18-13 Output of mpstat command on uniprocessor machine (xSeries 342)*

---

```
x342rsa:~ # mpstat 2 3
Linux 2.4.21-215-default (x342rsa)        05/20/04
```

07:12:16	CPU	%user	%nice	%system	%idle	intr/s
07:12:34	all	1.00	0.00	1.50	97.50	104.00
07:12:36	all	1.00	0.00	1.50	97.50	104.50
07:12:38	all	1.00	0.00	1.50	97.50	104.00
Average:	all	1.10	0.00	1.55	97.35	103.80

---

To display three entries of statistics for all processors of a multiprocessor server at one second intervals, use the following command (Example 18-14):

```
mpstat -P ALL 1 3
```

---

*Example 18-14 Output of mpstat command on four-way machine (xSeries 232)*

---

```
[root@x232 root]# mpstat -P ALL 1 10
Linux 2.4.21-9.0.3.EL (x232)        05/20/2004
```

02:10:49 PM	CPU	%user	%nice	%system	%idle	intr/s
02:10:50 PM	all	0.00	0.00	0.00	100.00	102.00
02:10:51 PM	all	0.00	0.00	0.00	100.00	102.00
02:10:52 PM	0	0.00	0.00	0.00	100.00	102.00
Average:	all	0.00	0.00	0.00	100.00	103.70
Average:	0	0.00	0.00	0.00	100.00	103.70

---

For the complete syntax of the **mpstat** command, issue:

```
mpstat -?
```

## 18.16 System x Performance Logger for Linux

IBM System x Performance Logger for Linux (xPL, formerly known as PLPerf) is a parameter-driven command line tool that collects performance counters from /proc on Linux into a CSV file, which is then readable by Windows Performance Monitor. It allows you to collect Linux performance counters and analyze the data using Windows Performance Monitor (**perfmon**).

xPL is written for x86- and x86\_64-based Linux platforms. It works on kernels 2.4 and 2.6 independent of the Linux distribution. Refer to 17.1, “Reliability and Performance Monitor console” on page 534 if you are not familiar with **perfmon**.

You can download the xPL binary and sources from:

<http://www.ibm.com/support/docview.wss?uid=psg1MIGR-64369>

xPL performs the following tasks, providing flexible and powerful monitoring:

- ▶ Trace counter data for specified time intervals or log file sizes
- ▶ Trace intervals down to milliseconds
- ▶ Allows breaking down the log into multiple files
- ▶ Allows overwriting the same log file (semi-circular)
- ▶ Allows creation of new parameter files
- ▶ Allows saving system information in a separate file

The following counters can be monitored. Each has multiple variables:

- ▶ CPU
- ▶ Interrupts
- ▶ Disk
- ▶ Memory
- ▶ Network

### 18.16.1 Counters descriptions

Because Windows does not interpret the entered values in the same way that Linux does, it is recommended that you do not use the Windows counters description to analyze the data. This section lists the counters that are collected by xPL and their definition, seen by a Linux system. Some of them are specific to the 2.6 kernel.

#### CpuStat

General CPU counters, which include the following:

- ▶ *%User Time*  
Per processor and total of all. Percent of total CPU time spent executing user processes.
- ▶ *% System Time*  
Per processor and total of all. Percent of total CPU time spent executing system (kernel) processors.

**Note:** In the 2.6 kernel, the %System Time that is reported by xPL includes IRQ and SoftIRQ times.

- ▶ *% Idle Time*  
Per processor and total of all. Percent of total CPU time being idle.

- ▶ *% Iowait Time* (2.6 kernel only)  
Per processor and total of all. Percent of total CPU time waiting for I/O to complete.
- ▶ *% IRQ Time* (2.6 kernel only)  
Per processor and total of all. Percent of total CPU time servicing interrupts.
- ▶ *% SoftIRQ Time* (2.6 kernel only)  
Per processor and total of all. Percent of total CPU time servicing softirqs.

**Note:** Under Linux, a *softirq* is an interrupt handler that runs outside of the normal interrupt context, runs with interrupts enabled, and runs concurrently when necessary. The kernel runs up to one copy of a softirq on each processor in a system. Softirqs are designed to provide more scalable interrupt handling in SMP settings.

- ▶ *% Processor Time*  
Per processor and total of all. Sum of user and system time.
- ▶ *Context Switches/sec*  
Total of all CPUs. Total number of context switches across all CPUs.

**Note:** A context switch (also sometimes referred to as a process switch or a task switch) is the switching of the CPU from one process or thread to another. A context is the contents of a CPU's registers and program counter at any point in time.

Context switching can be described as the kernel performing the following activities with regard to processes including threads) on the CPU:

- ▶ Suspending the progression of one process and storing the CPU's state (that is, the context) for that process somewhere in memory.
- ▶ Retrieving the context of the next process from memory and restoring it in the CPU's registers.
- ▶ Returning to the location indicated by the program counter (that is, returning to the line of code at which the process was interrupted) to resume the process.

## **IntStat**

IntStat provides interrupts counters per second per active interrupt per processor and totals of all per interrupt and per processor. Devices mapped to each interrupt are displayed after the interrupt number in the trace.

- ▶ *DskStat*  
General Physical disk statistics. Each stat is provided per Physical disk and total of all.
- ▶ *Reads/sec*  
Total number of read operations completed successfully per second.
- ▶ *Writes/sec*  
Total number of write operations completed successfully per second.
- ▶ *Transactions/sec*  
Sum of read and write operations per second.
- ▶ *Read bytes/sec*  
Total number of bytes read successfully per second.
- ▶ *Write bytes/sec*  
Total number of bytes written successfully per second.
- ▶ *Bytes/sec*  
Sum of read and write bytes per second.
- ▶ *Average bytes/read*  
Total number of bytes read successfully per total # of reads completed successfully.
- ▶ *Average Bytes/Write*  
Total number of bytes written successfully per total # of writes completed successfully.
- ▶ *Average Bytes/Transaction*  
Sum of reads and writes.
- ▶ *Average sec/read*  
This is the total number of seconds spent by all reads per total # of reads completed successfully.
- ▶ *Average sec/write*  
This is the total number of seconds spent by all writes per total # of writes completed successfully.

- ▶ *Average sec/transaction*  
Sum of reads and writes.
- ▶ *I/O operations in progress*  
Snapshot of current disk queue. Incremented as requests are given to appropriate request queue and decremented as they finish. It is the number of requests outstanding on the disk at the time the performance data is collected. It also includes requests in service at the time of the collection. This is a instantaneous snapshot, not an average over the time interval.

## MemStat

Provides memory counters.

- ▶ *Total MB*  
Total physical memory.
- ▶ *Free MB*  
Total unallocated physical memory. This is not a good indicator of available physical memory. To check to see whether you are out of available physical memory, watch for paging or swapping activity.
- ▶ *Page In KB/sec*  
Total number of kilobytes the system paged in from disk per second.
- ▶ *Page Out KB/sec*  
Total number of kilobytes the system paged out to disk per second.
- ▶ *Page KB/sec*  
Sum of Page In and Page Out KB/sec.
- ▶ *Swap In/sec*  
Total number of swap pages the system brought in per second.
- ▶ *Swap Out/sec*  
Total number of swap pages the system brought out per second.
- ▶ *Swaps/sec*  
Sum of Swap In and Swap Out /sec.

The fundamental unit of memory under Linux is the *page*, which is a non-overlapping region of contiguous memory. All available physical memory is organized into pages near the end of the kernel's boot process, and pages are issued to and revoked from processes by the kernel's memory management algorithms at runtime. Linux uses 4 KB pages for most processors.

Paging and swapping both refer to virtual memory activity in Linux. With paging, when the kernel requires more main memory for an active process, only the least recently used pages of processes are moved to the swap space. The page counters are similar to Memory Page Reads and Writes in Windows. Swapping means to move an entire process out of main memory and to the swap area on hard disk, whereby all pages of that process are moved at the same time.

## NetStat

NetStat provides network counters. Each stat is provided per network device and total of all:

- ▶ *Packets Sent/sec*
- ▶ *Packets Received/sec*
- ▶ *Packets/sec (sum of sent and received)*
- ▶ *Bytes Sent/sec*
- ▶ *Bytes Received/sec*
- ▶ *Bytes/sec (sum of sent and received)*

### 18.16.2 Instructions

Before using xPL, you need to set the parameters in the parameter file. The default file is called `input.prm`, but you can rename it. (If you do rename it, be sure that you refer to the correct file when executing xPL.) See 18.16.2, “Instructions” on page 637 for information about the syntax of the file.

To run xPL, you can simply run the `xpl` command, followed by the parameter file:

```
xpl parameter_file
```

After you run xPL, you can use the created output files with Windows System Monitor. The generated files are in CSV format. Refer to Windows System Monitor (`perfmon`) help for instructions about how to import a CSV log file.

**Tip:** Start `perfmon` by clicking **Start** → **Settings** → **Control Panel** → **Administrative Tools** → **Performance**, and then click the cylinder icon  on the toolbar.

You can also generate parameter files by issuing the following command:

```
xpl -g parameter_file
```

You can create a system info file (a file that includes the system configuration information) by issuing the following command:

```
xpl -s system_info_file
```

**Tip:** You can stop xPL by pressing Ctrl+C to perform a graceful shutdown.

### 18.16.3 Parameter file

The parameter file is the file that is read when xPL is launched. It includes the information about which counters are monitored and how xPL should monitor those counters. The file is divided in several sections:

- ▶ config section

Tracing configuration. Options are:

- 0 Time limited trace
- 1 Log file size limited

**Note:** If xPL is unable to write any further to disk (for example, if the disk is full), it will stop.

Trace interval consists of interval in seconds and interval in milliseconds. The total of the two is used for trace interval, and is separated for convenience. These values are required by all trace configurations.

- ▶ int\_s

Trace interval in seconds.

- ▶ int\_m

Trace interval in milliseconds, must be 0 or between 20 and 999.

Trace duration is only required by config 0 (Time limited trace).

- ▶ dur\_s

Trace duration in seconds.

- ▶ dur\_m

Trace duration in milliseconds must be 0 or between 20 and 999.

- ▶ log\_size

Log file size limit in MB, which applies to all configs. Set to 0 for no limit (stops when the disk is full or the time limit has been reached). Note that on average, xPL writes 3 KB per sample.

- ▶ new\_log

0: no; 1: yes

Start a new log file when log file size limit is reached, for all configs. If set to no (0) along with config 0, xPL will overwrite the file if the log\_size limit has been reached and the time limit is not reached. If set to yes (1), xPL will

create a new log file, incrementing the log file number (see `log_start`). If used along with config 1, xPL will not stop until it is manually stopped, or when the disk is full.

- ▶ `log_start`  
Starting number of log file, applies only if `new_log` is set to yes(1).
- ▶ `output_file`  
Output file name. xPL will append .csv extension to the end. No spaces here, xPL will only use the first portion before the first space.
- ▶ `output_buffer`  
Output buffer size in KB.  
If set to a number other than zero, xPL will wait until the buffer is full before writing to disk.
- ▶ `counter`  
List of counters to trace: 1 to trace, and 0 to not trace. (The counters are those previously listed.)

Example 18-15 shows a sample parameter file you can use with xPL. Using this file, you will monitor the CPUs only at 1-second intervals for a total of 5 seconds. The log file is named `output.csv`, and it is overwritten each time you launch xPL.

*Example 18-15 Sample xPL Parameter file*

---

```
# Parameter Input File for IBM xSeries Performance Logger for Linux --  
xPL (v 1.0.1)  
#  
# Trace Parameters  
# 0: Time limited  
# 1: Log file size limited  
config 0  
# Note you can use the interrupt signal to stop tracing at any time  
  
# Time Interval, applies to all configs  
# interval, seconds  
int_s 1  
# interval, milliseconds, 0 or between 20 and 999  
int_m 0  
  
# Trace duration, applies to config 0  
# trace duration, seconds  
dur_s 5  
# trace duration, milliseconds, 0 or between 20 and 999
```

```
dur_m    0

# Log file size limit in MB, applies to all configs.
# Set to 0 for no limit (stops when disk full or time limit has
reached.)
log_size    0

# Start a new log file when log file size limit has reached, for all
configs.
# If set to no along with config 0, xPL will overwrite the file
# if log_size limit has reached so long as the time limit is not
reached.
# If set to yes along with config 1, xPL will continue tracing until
disk is
# full or manually stopped.
# 0: no
# 1: yes
new_log    0

# Starting number of log file
# Applies only if new_log is set to 1.
log_start    0

# Log file name, no spaces (xPL will append .csv to the end)
output_file    output
# Log file buffer size in KB (0 means write on every sample)
output_buffer    8

# Set of counters to trace
# 0: don't trace
# 1: trace
counter CpuStat    1
counter IntStat    0
counter DskStat    0
counter MemStat    0
counter NetStat    0
```

---

Figure 18-10 shows an example of an activity log file that is generated with xPL on a dual-processor server. The log file (CSV file) has been opened with **perfmon**.

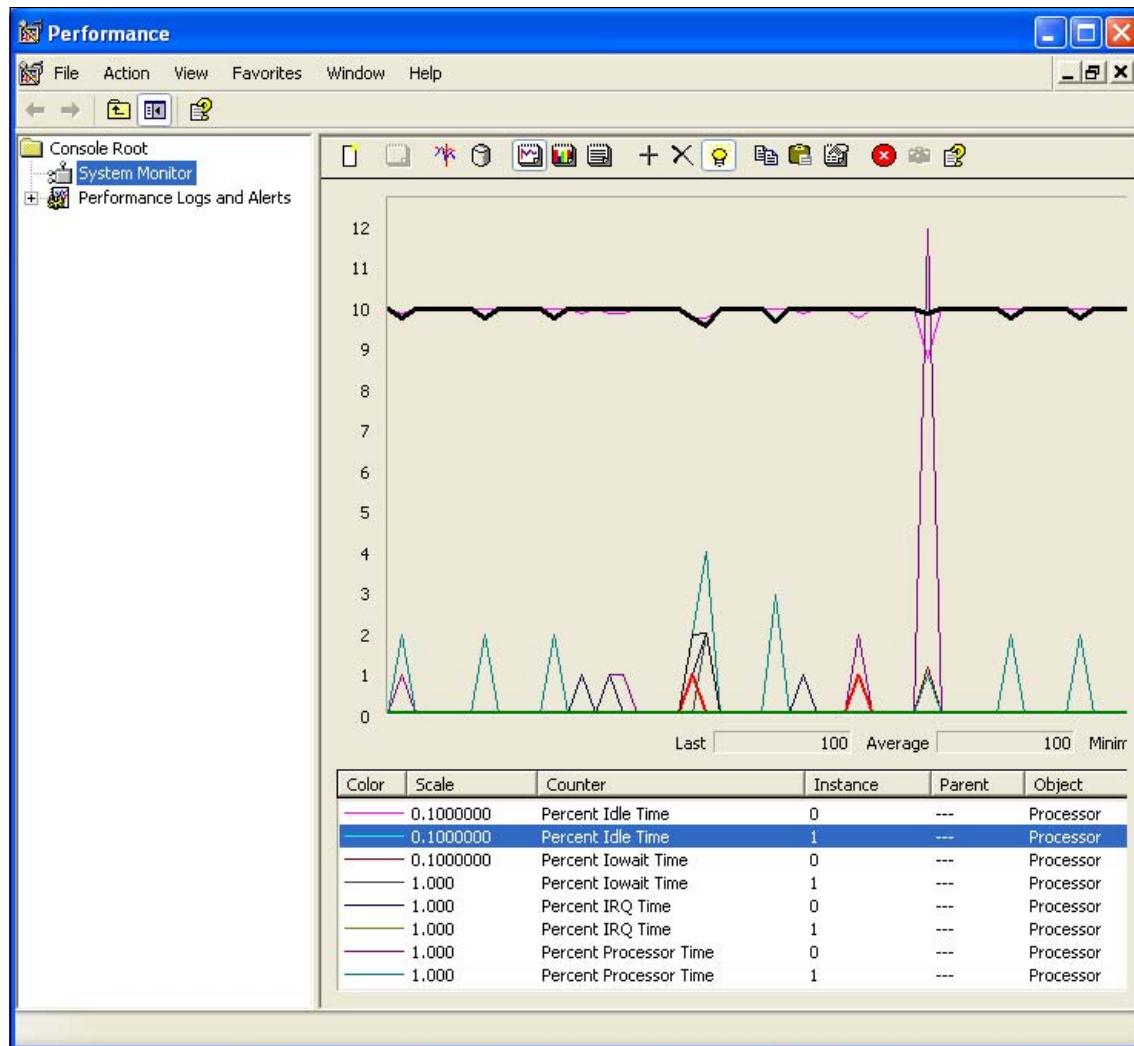


Figure 18-10 Trace log from xPL into Windows perfmon

**Note:** Perfmon can handle partially-written samples at the end of the log. You can use the `relog` command on Windows to manipulate log files (for example, to convert csv logs to binary). Refer to the relog help for more info.

Data is generated as it is read from /proc, so there is no limit on how often you can read it, although it might affect your system performance if the data is read too often.

## 18.17 The nmon tool

The `nmon` tool is an easy-to-use monitoring tool that was developed for AIX® platforms and has been ported onto Linux. This tool provides a significant amount of information within a single screen. Even though it is not supported officially by IBM, it is used during benchmarks to analyze bottlenecks or production systems to give a quick view of system utilization.

With nmon, you are able to monitor:

- ▶ CPU utilization
- ▶ Memory use
- ▶ Kernel statistics and run queue
- ▶ Disks I/O rates, transfers, and read/write ratios
- ▶ File system size and free space
- ▶ Disk adapters
- ▶ Network I/O rates, transfers, and read/write ratios
- ▶ Paging space and paging rates
- ▶ Machine details, CPU, and operating system specification
- ▶ Top processors
- ▶ User-defined disk groups

You can log the activity of a server using nmon. The generated log file is a comma-separated values (CSV) file that you can import and analyze through a spreadsheet. There is a Windows Excel® spreadsheet to automate this process. For more information, see “Data collection mode” on page 646.

The nmon tool reads the information from the server in the /proc file system. The /proc file system is used by the kernel to communicate with the various processes. It is a real-time file system where, basically, all activity counters are. The /proc file system resides in memory, not on the disk, which means that reading this file system is efficient.

Supported operating systems are:

- ▶ SUSE Linux Enterprise Server 8 and 9
- ▶ Debian
- ▶ Fedora
- ▶ Red Hat 9, EL 2.1, 3 and 4,
- ▶ Knoppix
- ▶ Linux on POWER
- ▶ Linux on System z

You can download **nmon** from:

<http://www-941.haw.ibm.com/collaboration/wiki/display/WikiPtype/nmon>

### 18.17.1 Using nmon

Installing **nmon** is very easy, because it is a binary that is compiled for every supported distribution. The syntax for using **nmon** is simple, and you need to specify only a few parameters. There are two different ways of using nmon:

- ▶ Interactive mode
- ▶ Data collection mode

#### Interactive mode

Interactive mode lets you use **nmon** as a monitoring tool without any logging of data. Interactive mode is useful for real-time diagnostics and to check quickly the impact of changing parameters. To run the monitoring, simply run **nmon** without any parameter.

Figure 18-11 shows the welcome screen, which provides the keys to use to display the counters. The values that are displayed are refreshed every two seconds by default, but you can change the refresh time if you prefer.

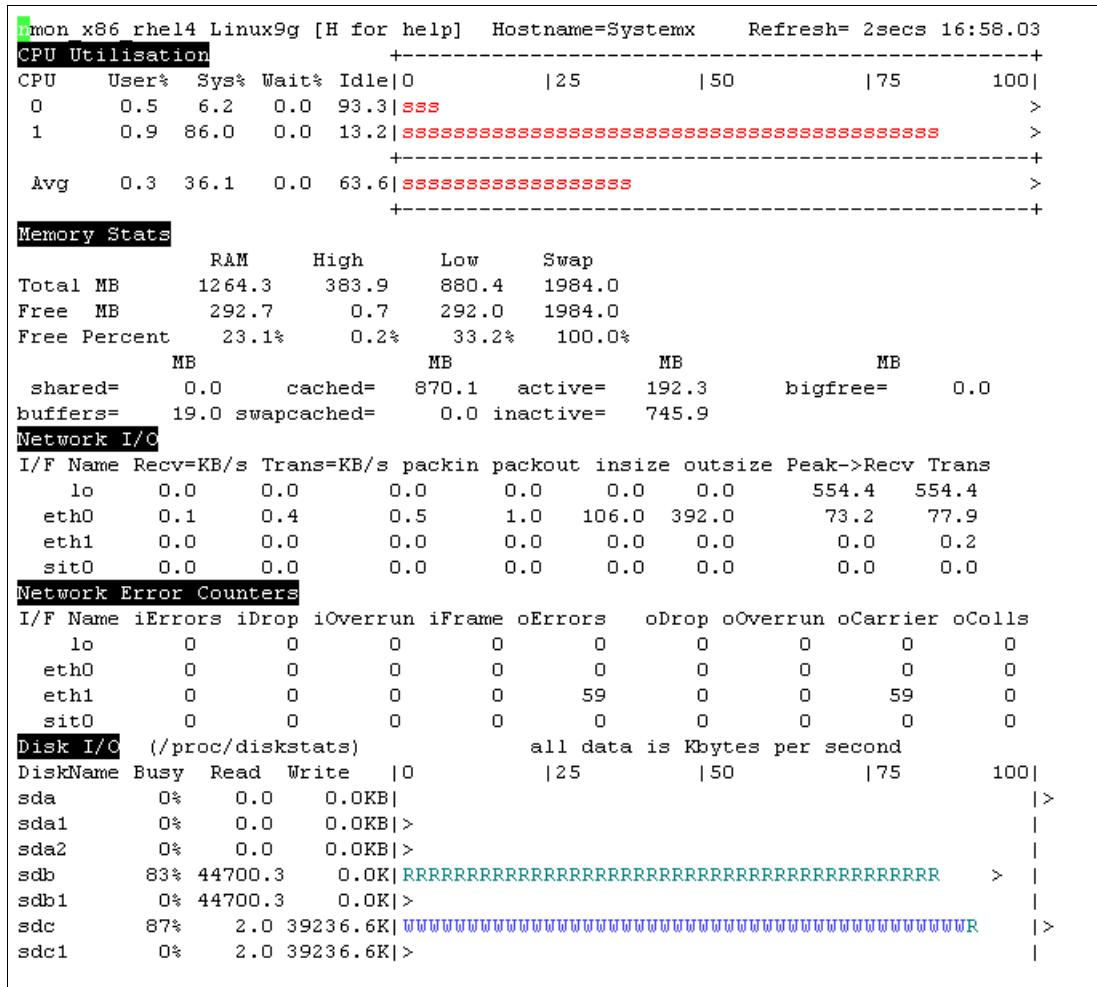
The screenshot shows a terminal window titled "root@Systemx:~". The title bar also displays "nmon x86\_rhel4 Linux9g [H for help] Hostname=Systemx Refresh= 2secs 16:49.37". The window content includes:

- A section of "#"-style comments at the top.
- Usage instructions:
  - "For help type H or ..."
  - "nmon -? - hint"
  - "nmon -h - full"
- A note: "To start the same way every time set the NMON ksh variable"
- A section titled "Use these keys to toggle statistics on/off:" with the following mappings:

c = CPU	l = CPU Long-term	- = Faster screen updates
m = Memory	j = Filesystems	+ = Slower screen updates
d = Disks	n = Network	v = Verbose hints
r = Resource	N = NFS	. = only busy disks/procs
k = kernel	t = Top-processes	
h = more options		

Figure 18-11 The nmon welcome window

As you press the keys corresponding to the counters that you want to monitor, new information appears. For example, if you press **c**, information for the CPU appears. Pressing **d** provides information for disks. Pressing **n** provides information for networks, and so on. Figure 18-12 is an example of nmon monitoring CPU, disks, memory, and network components.



*Figure 18-12 Monitoring server's activity with nmon*

## Data collection mode

The second way that you can use nmon is with trace logs in data collection mode. This mode allows you to run nmon for a specified period of time and trace the activity within given intervals (in seconds). This mode generates a log file that you can use for later analysis and performance reports.

For example, to use nmon to monitor all components every 10 seconds for one hour, issue the following command:

```
nmon -f -s10 -c360
```

This command appends all counters in a file called *<hostname>\_YYYYMMDD\_HHMM.nmon*, every 10 seconds, 360 times. The **-f** argument stands for file. If you specify **-f**, then it means you are using nmon in data collection mode and that you should specify, at least, the interval in seconds **-s** and the number of occurrences or count **-c**.

**Note:** If you omit the **-s** and **-c** arguments when using nmon in data collection mode, nmon will use the default values, which are 300 seconds and 288 occurrences. This corresponds to a 24-hour run.

You can use the default file name when tracing activity; in that case, a new file is created each time you launch nmon. Alternatively, you can specify your own file name (which can be overwritten if it already exists). To do so, use the **-F** flag (uppercase) instead of **-f**, followed by your own user-defined file name.

**Tip:** You can use both data collection mode and interactive mode at the same time. Simply launch nmon in data collection mode (with **-f** or **-F**) and then in interactive mode to log the activity while monitoring it.

After the data collection is finished, you can use a very simple program called **nmon2csv** to translate your nmon log file into a CSV file. The syntax is:

```
nmon2csv <filename.nmon>
```

The nmon2csv binary is available with the nmon tools downloads at:

<http://www-941.haw.ibm.com/collaboration/wiki/display/WikiPtype/nmon>

As a result, you will have a new file with the CSV extension. You can keep this file as activity logs, and you can use it with additional tools.

## 18.17.2 The nmon Analyser Excel macro

Also available is the nmon Analyser Excel macro, which you can use to generate graphs in Excel. You can download the nmon Analyser from:

<http://www-941.ibm.com/collaboration/wiki/display/WikiPtype/nmonanalyser>

**Tip:** If the monitored system and the Windows system have different regional settings, you need to specify, among other parameters, what character is used to separate values (period, comma, or semicolon) and what character is used to separate decimals, in the nmon Analyser Excel spreadsheet.

When you run the macro, you get a spreadsheet for each component of your system, including one or more graphs that represents the counters. For example, Figure 18-13 shows the CPU activity summary (on all processors or cores) and the disks transfers for a 20-minute period.

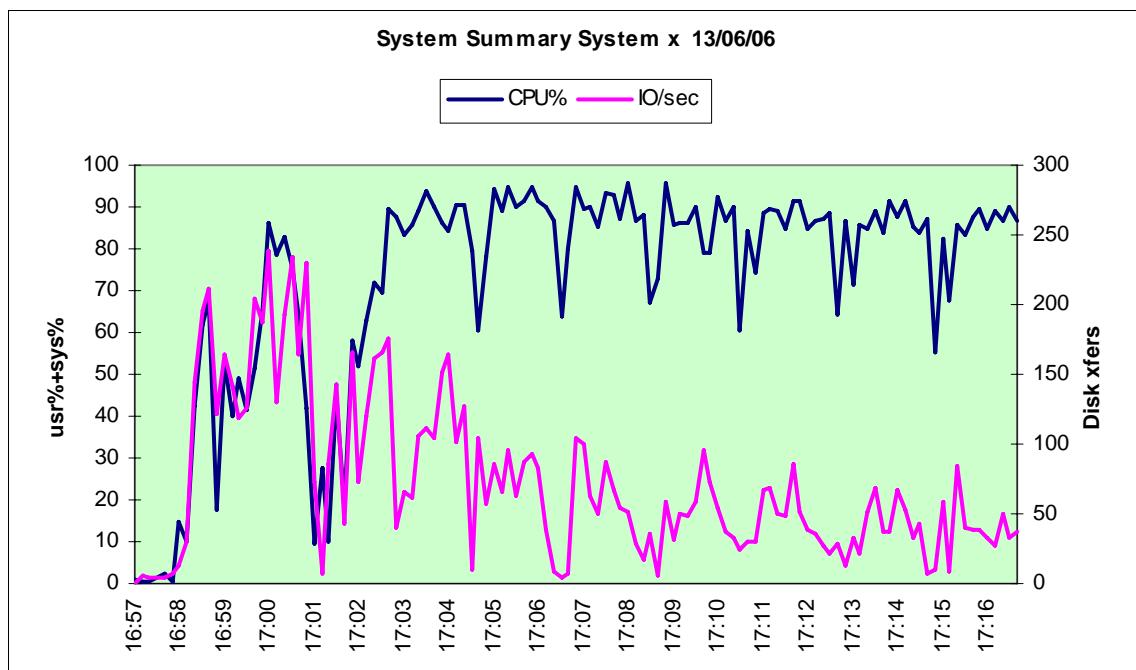


Figure 18-13 System summary graph generated with nmon Analyser

Although the system summary provides a useful overview of your system activity, you have a set of graphs for each component of your server (CPU, disks, memory, and network). More accurate data is then available. Figure 18-14 shows the network I/O on the system.

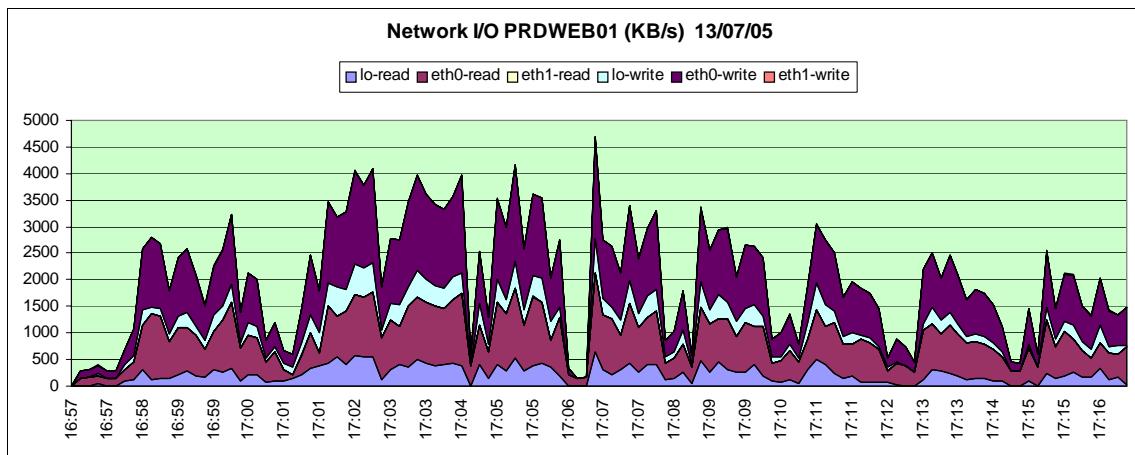


Figure 18-14 Network activity sample graph

The nmon help file includes all commands that you can use in interactive-mode or data-collect-mode. It also includes some very useful hints.

The nmon download site includes useful information, the nmon manual and FAQ, hints and tips, as well as an active forum for questions. To find this site, take the Performance Tools link at:

<http://www.ibm.com/systems/p/community/>



# VMware ESX tools

Virtualization provides many benefits. However, analyzing performance issues on a virtual machine is more complex than on a conventional system because there is one more factor, multiple virtual machines sharing common hardware. Without sound measurements, attempts to tune the VMware ESX system or any virtual machine remain pure speculation. VMware has extended their management tools to monitor performance.

Most monitoring tools have difficulties in identifying system performance and bottlenecks when installed in the Console OS. Thus, it is important to understand that the Console OS is simply a very privileged virtual machine with special interfaces to the VMware kernel. Issuing typical Linux performance monitoring commands (such as `top`) will only reveal the virtualized performance of the Console OS. In addition, other monitoring tools will make it difficult to understand the concept of page sharing that is implemented in VMware and can thus produce erroneous values.

It is generally sensible, however, to use application benchmarking tools for virtual machines with the limitation that VMware ESX is not designed to deliver peak performance but rather scalable performance over multiple virtual machines.

This chapter discusses the following topics:

- ▶ 19.1, “Benchmarks” on page 650
- ▶ 19.2, “The `esxtop` utility” on page 650
- ▶ 19.3, “VirtualCenter Console” on page 657

## 19.1 Benchmarks

There are currently two benchmark tools available to measure a system's performance and ability to run VMware ESX. These are the VMware benchmark VMmark, and the IBM/Intel co-developed tool vConsolidate.

Information about VMmark is available from:

<http://www.vmware.com/products/vmmark/>

VMware benchmark software is available to download from:

<http://www.vmware.com/download/vmmark/>

Along with IBM and Intel, VMware is a member of the SPEC Virtualization subcommittee, which is involved in developing a new, next generation benchmark standard that is likely to replace existing benchmarks. Information is available from:

<http://spec.org/specvirtualization/>

vConsolidate is a virtualization benchmark that runs multiple instances of consolidated Database, Mail, Web and Java workloads in multiple virtual machines to simulate real-world server performance in a typical environment.

A discussion about vConsolidate can be found at:

<http://www.intel.com/technology/itj/2006/v10i3/7-benchmarking/6-vconsolidate.htm>

## 19.2 The esxtop utility

The esxtop utility comes as a standard tool with VMware ESX. This utility enhances the classic top with the awareness of a virtualized system. When used in the Console OS, the esxtop utility reveals the current CPU and memory utilization. It also displays the various processes or virtual machines that run in the system, as well as their relative impact on overall system performance. Although esxtop delivers only current information, it is very easy to use and can be useful on a system that has a performance bottleneck.

Before you start troubleshooting a performance issue, ensure that you have a secure shell client (ssh) available. VMware recommends that Windows users use the PuTTY client, which you can download at no charge from:

<http://www.chiark.greenend.org.uk/%7Esgtatham/putty/download.html>

## 19.2.1 Starting esxtop

To start the esxtop utility, follow these steps:

1. Open your SSH client (such as PuTTY).
2. Enter the IP Address of the ESX Server system that you want to monitor and select the SSH protocol; see Figure 19-1. The standard port for SSH is 22. If your client does not add this port automatically, you will need to type it in.

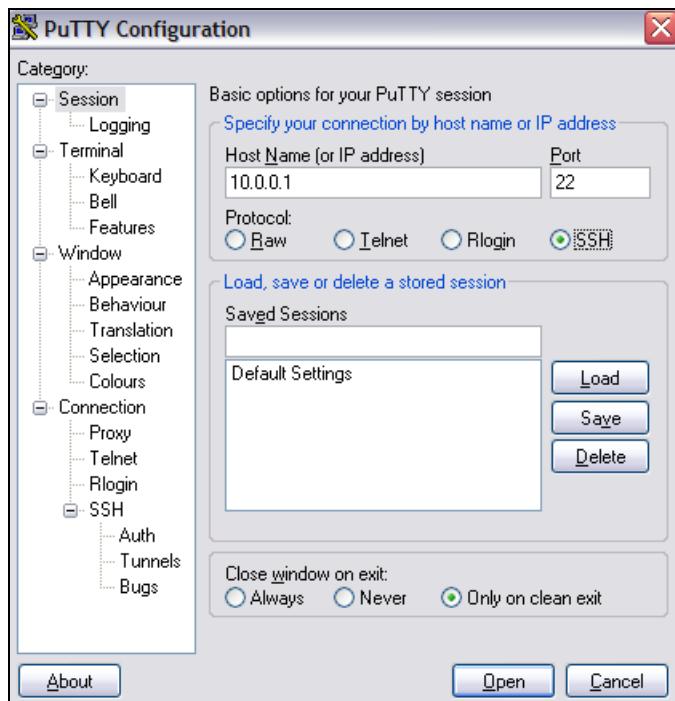


Figure 19-1 PuTTY configuration

3. Open the connection.

If you are using the standard security settings, you are not allowed to log on through SSH as root user. So, you need to log on as a different user than root.

If you want to log on as root, you have to edit the /etc/ssh/sshd\_config file before you log on to allow root access. However, we recommend that you use a regular account.

4. Switch to the root user. If you do not, you will not be able to run esxtop as shown in the error window in Figure 19-2.

```
login as: Daniel
Daniel@10.0.0.1's password:
[Daniel@esx Daniel]$ esxtop
esxtop: Need to run as user root
[Daniel@esx Daniel]$ su
Password:
[root@esx Daniel]#
```

*Figure 19-2 The SSH login*

5. Start the utility by entering **esxtop**.

The esxtop window opens, as shown in Figure 19-3 on page 653, which displays all processes that are running currently on the ESX Server.

Daniel@esx:/home/Daniel

```

6:05:05am up 1 day, 16:21, 45 worlds; CPU load average: 0.26, 0.26, 0.26
PCPU(%): 98.29, 1.90, 0.47, 0.33 ; used total: 25.25
CCPU(%): 91 us, 7 sy, 0 id, 2 wa ; cs/sec: 1142

ID GID NAME NMEM %USED %SYS %OVR LP %RUN %WAIT %BWAIT %TW
1 1 idle 4 298.36 0.00 0.00 1.32 0.00 0.00 0.00 0
2 2 system 5 0.00 0.00 0.00 0.00 498.72 0.00 0.00 498
6 6 console 1 97.22 0.00 0.07 97.81 0.00 1.93 1
7 7 helper 13 0.01 0.00 0.00 0.01 1296.77 0.00 0.00 1296
8 8 drivers 7 0.01 0.00 0.00 0.01 698.20 0.00 0.00 698
12 12 vmware-vmkauthd 1 0.00 0.00 0.00 0.00 99.74 0.00 0.00 99
14 14 W2k3 7 1.44 0.00 0.07 1.44 543.80 152.50 696
15 15 W2k32nd 7 1.14 0.00 0.06 1.14 553.90 142.75 696

6:07:28am up 1 day, 16:23, 45 worlds; CPU load average: 0.26, 0.26, 0.26
PCPU(%): 100.00, 0.70, 1.20, 0.97 ; used total: 25.72
CCPU(%): 94 us, 6 sy, 0 id, 0 wa ; cs/sec: 1170

ID GID NAME NMEM %USED %SYS %OVR LP %RUN %WAIT %BWAIT %
1 1 idle 4 297.32 0.00 0.00 2.52 0.00 0.00
2 2 system 5 0.02 0.00 0.00 0.02 500.00 0.00 0.00 5
6 6 console 1 99.43 0.01 0.06 100.01 0.00 0.00 0.00
6:07:44am up 1 day, 16:24, 45 worlds; CPU load average: 0.26, 0.26, 0.26
PCPU(%): 100.00, 1.75, 1.59, 0.90 ; used total: 26.06
CCPU(%): 91 us, 7 sy, 0 id, 2 wa ; cs/sec: 1110

ID GID NAME NMEM %USED %SYS %OVR LP %RUN %WAIT %BWAIT %
1 1 idle 4 302.16 0.00 0.00 1.60 0.00 0.00
2 2 system 5 0.00 0.00 0.00 0.00 500.00 0.00 0.00 5
6 6 console 1 98.47 0.01 0.07 98.09 0.00 1.97
7 7 helper 13 0.01 0.00 0.00 0.01 1300.00 0.00 0.00 13
8 8 drivers 7 0.01 0.00 0.00 0.01 700.00 0.00 0.00 7
12 12 vmware-vmkauthd 1 0.00 0.00 0.00 0.00 100.00 0.00 0.00 1
14 14 W2k3 7 1.17 0.00 0.07 1.17 530.99 168.14 6
15 15 W2k32nd 7 1.46 0.00 0.06 1.46 570.70 127.83 6

```

Figure 19-3 The esxtop start panel

To obtain a list of all the available commands, enter **h** on the command line; you will reach the window shown in Figure 19-4.

```
Esxtop version 3.0.0
Secure mode Off

Esxtop: top for ESX

These single-character commands are available:

^L      - redraw the screen
space   - update display
h or ?  - help; show this text
q       - quit

Interactive commands are:

fF      Add or remove fields
oO      Change the order of displayed fields
s       Set the delay in seconds between updates
#       Set the number of instances to display
W       Write configuration file ~/.esxtop3rc
e       Expand/Rollup Cpu Statistics

Sort by:
          U:%USED           R:%RDY           N:Default
Switch display
          m:ESX memory       d:ESX disk       n:ESX nic
```

Figure 19-4 *esxtop help*

The default is to display CPU usage. If you prefer to capture the memory, disk, or NIC usage, the last line of the display (Figure 19-4) shows the keys to press to switch:

- ▶ Press **m** to display memory usage
- ▶ Press **d** to display disk usage
- ▶ Press **n** to display network usage

In this section, we explain how to use esxtop with the CPU usage display. However, if you want to use another option, you can switch, and the operation is the same.

By entering the **f** command on the start window, you reach the panel to reorder the fields; see Figure 19-5. To reorder the fields, enter the corresponding letter. To exit and go back to the normal esxtop window, enter any key other than **A** through **H**.

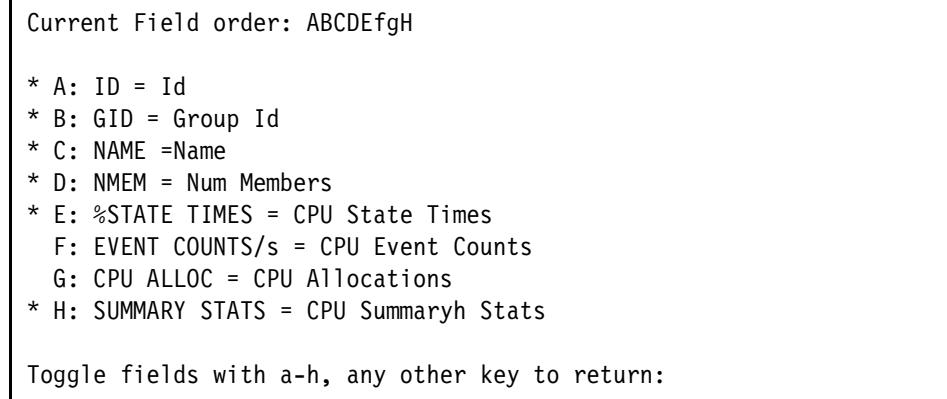


Figure 19-5 Field order panel

The esxtop utility also offers the ability to run in batch mode; for example, you can use this command:

```
esxtop -b -n iterations > logfile
```

For information about how to use this command, enter the following command on the SSH command line:

```
man esxtop
```

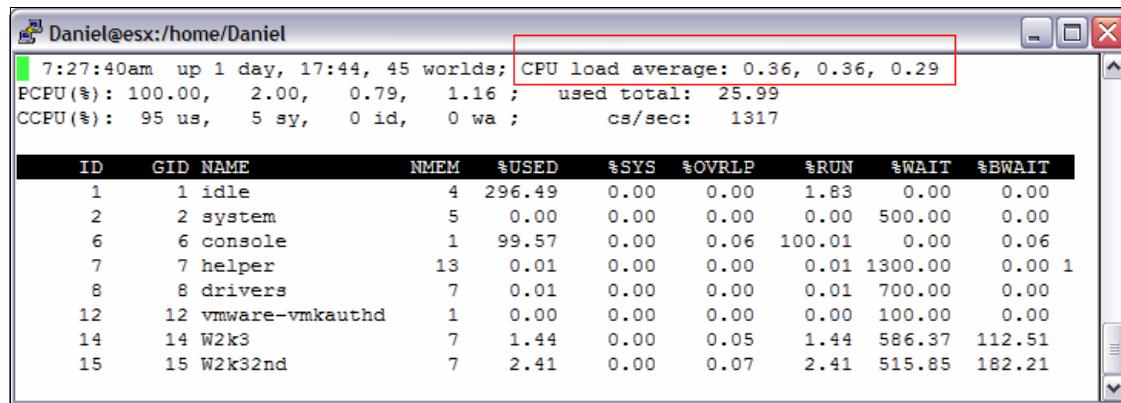
### 19.2.2 Using esxtop

After you have customized the esxtop main window, you can begin analyzing the bottlenecks in your system.

**Note:** Keep in mind that although the examples shown here are based on CPU usage, you can also monitor memory, disk, and network usage with this tool.

In the CPU usage main window, shown in Figure 19-6, the first line (highlighted in a red box) displays the load average for all physical CPUs on the ESX Server machine. A load average of 1.0 means that the physical CPU is fully utilized.

If this value is under 1.0, the CPU is *underutilized*. If this value is over 1.0, the CPU is *overutilized*, and you need to increase the number of physical CPUs or to decrease the number of virtual machines that are running on the server.



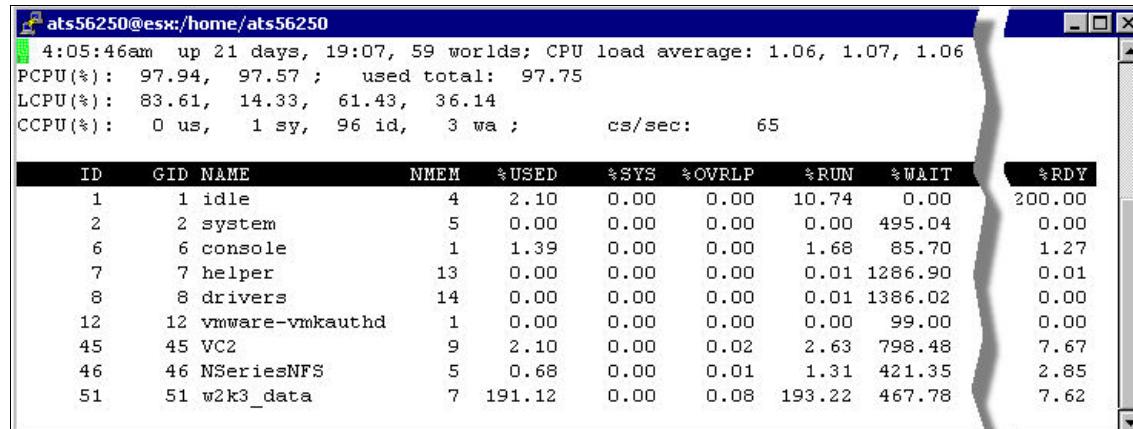
Daniel@esx:/home/Daniel

```
7:27:40am up 1 day, 17:44, 45 worlds; CPU load average: 0.36, 0.36, 0.29
PCPU(%): 100.00, 2.00, 0.79, 1.16 ; used total: 25.99
CCPU(%): 95 us, 5 sy, 0 id, 0 wa ; cs/sec: 1317
```

ID	GID	NAME	NMEM	%USED	%SYS	%OVR LP	%RUN	%WAIT	%BWAIT
1	1	idle	4	296.49	0.00	0.00	1.83	0.00	0.00
2	2	system	5	0.00	0.00	0.00	0.00	500.00	0.00
6	6	console	1	99.57	0.00	0.06	100.01	0.00	0.06
7	7	helper	13	0.01	0.00	0.00	0.01	1300.00	0.00 1
8	8	drivers	7	0.01	0.00	0.00	0.01	700.00	0.00
12	12	vmware-vmkauthd	1	0.00	0.00	0.00	0.00	100.00	0.00
14	14	W2k3	7	1.44	0.00	0.05	1.44	586.37	112.51
15	15	W2k32nd	7	2.41	0.00	0.07	2.41	515.85	182.21

Figure 19-6 CPU load average

The second line of the screen shows the PCPU usage, which is the usage for each individually installed CPU. The last value is the average percentage for all the CPUs. If you are using a multi-core CPU, each core is displayed as a separate CPU. If this value is under 80% utilization, your system should perform well. If this value is about 90% or higher, this is a critical warning that the CPU is overloaded. Figure 19-7 shows a system that has a CPU bottleneck.



ats56250@esx:/home/ats56250

```
4:05:46am up 21 days, 19:07, 59 worlds; CPU load average: 1.06, 1.07, 1.06
PCPU(%): 97.94, 97.57 ; used total: 97.75
LCPU(%): 83.61, 14.33, 61.43, 36.14
CCPU(%): 0 us, 1 sy, 96 id, 3 wa ; cs/sec: 65
```

ID	GID	NAME	NMEM	%USED	%SYS	%OVR LP	%RUN	%WAIT	%RDY
1	1	idle	4	2.10	0.00	0.00	10.74	0.00	200.00
2	2	system	5	0.00	0.00	0.00	0.00	495.04	0.00
6	6	console	1	1.39	0.00	0.00	1.68	85.70	1.27
7	7	helper	13	0.00	0.00	0.00	0.01	1286.90	0.01
8	8	drivers	14	0.00	0.00	0.00	0.01	1386.02	0.00
12	12	vmware-vmkauthd	1	0.00	0.00	0.00	0.00	99.00	0.00
45	45	VC2	9	2.10	0.00	0.02	2.63	798.48	7.67
46	46	NSeriesNFS	5	0.68	0.00	0.01	1.31	421.35	2.85
51	51	w2k3_data	7	191.12	0.00	0.08	193.22	467.78	7.62

Figure 19-7 PCPU usage

The NAME column provides the given name of your virtual servers, and each line provides information about the CPU usage. For our discussion, we concentrate only on the %USED and %RDY fields. Depending on your view, you might have to expand your SSH client window to see the %RDY field.

- ▶ %USED

This field shows the percentage of physical CPU resources that are used by a virtual CPU. If the virtual CPU is running at the full capacity of the physical CPU, you can identify the virtual machine that might be causing the bottleneck.

- ▶ %RDY

This field gives you information about the time that a virtual machine was ready but could not get scheduled to run on a physical CPU. As a rule of thumb, this value should remain under 5%.

If you are running into a CPU bottleneck on a virtual machine, the most common solutions are:

- ▶ Increase the number of CPUs or cores.
- ▶ Decrease the number of virtual machines.
- ▶ Move the virtual machine to another ESX Server system.

### **Memory usage**

As described in 19.2.1, “Starting esxtop” on page 651, you can change the esxtop view to show the memory usage by pressing **m**. We also recommend that you monitor the following values:

- ▶ The maximum available memory that is used by the virtual machine
- ▶ The amount of swapped memory that is used by a virtual machine

Swapping data out of memory can sometimes indicate that there is a shortage of available memory, but it can also simply be a process of removing inactive pages of data in anticipation of needing that space later. However, if the swap percentage increases, it could be a sign of a performance problem.

#### **19.2.3 Exiting esxtop**

To exit the utility, enter **q**.

### **19.3 VirtualCenter Console**

You can use VirtualCenter Console to monitor the utilization and performance of the host machine and all virtual guests. VirtualCenter Console replaces

vmkusage, which was previously a tool within ESX. The VMware Infrastructure client provides a graphical view of VirtualCenter Console, showing the host servers and virtual machines list on the left and various tabs for specific tasks on the right; see Figure 19-8. The Summary tab provides the actual view of resources available on the host and the current utilization.

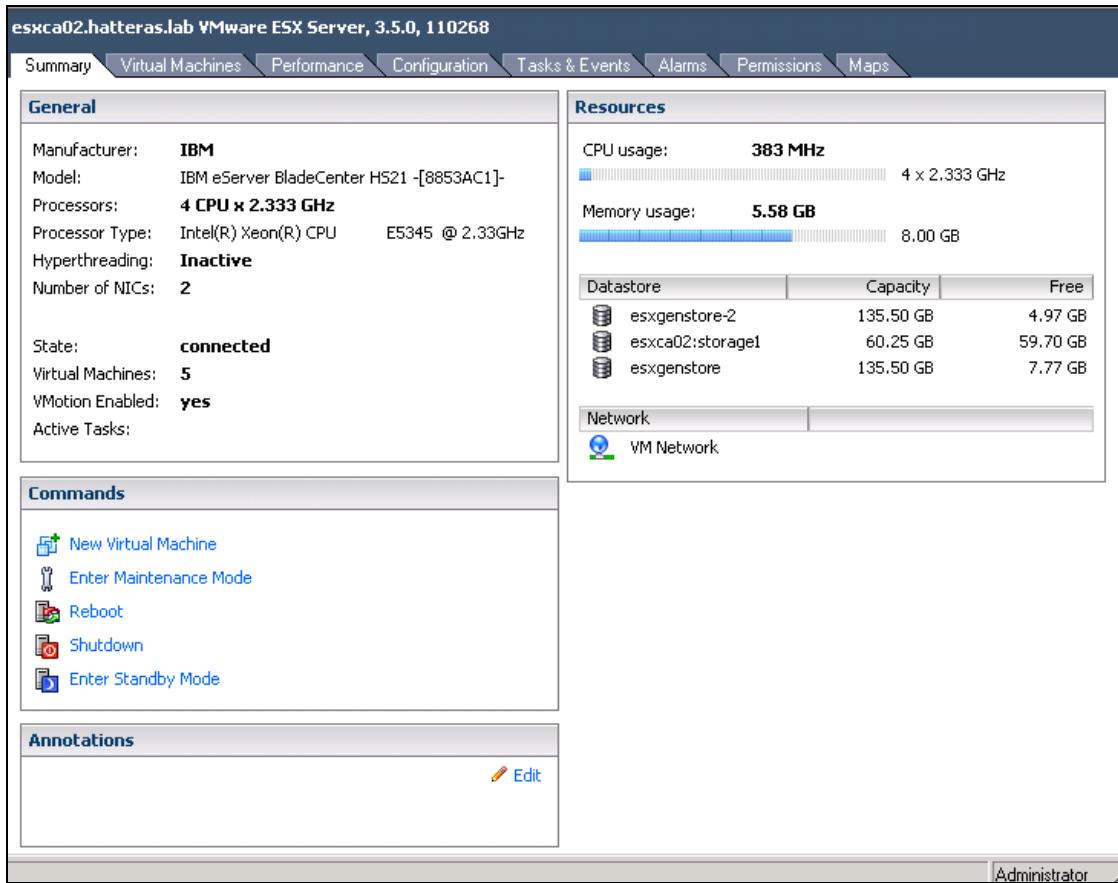


Figure 19-8 VMware VirtualCenter console summary tab

The Virtual Machines tab, shown in Figure 19-9 on page 659, displays VMs either on each ESX host or for all VMs in the cluster. This consolidated view shows where resources are allocated.

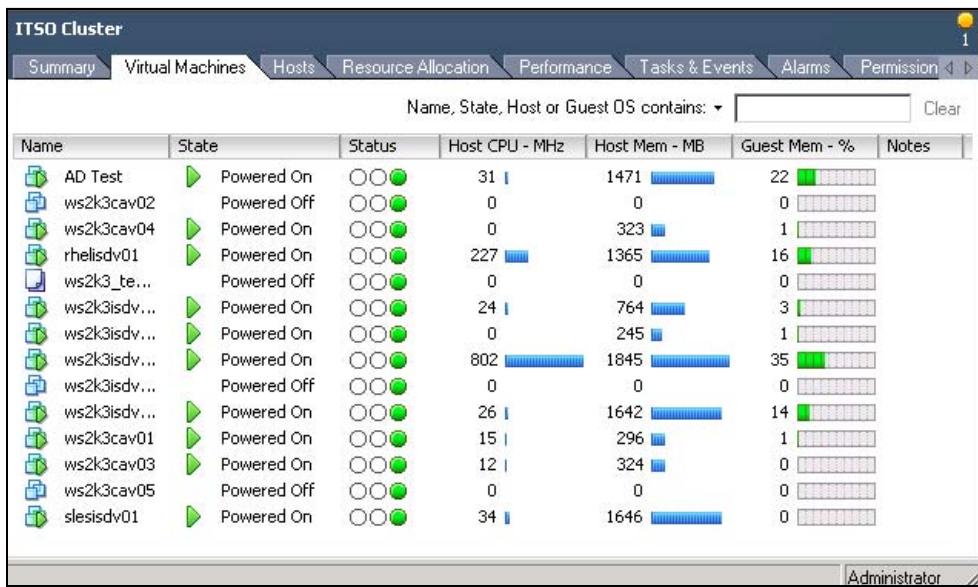


Figure 19-9 Virtual Machines view

The Performance tab, shown in Figure 19-10 on page 660, displays a real-time graph of the usage of various resources on the cluster, host ESX, or individual VM level. Performance counters are available for CPU, disk, memory, network, and overall systems performance.

Monitoring a specific VM for a period of time gives you an understanding of the actual resource usage by VM as shown in Figure 19-10 on page 660. If the VM does not utilize all memory or other resources, you have the option of reducing the resources allocated to that VM and adding more VMs to that host.

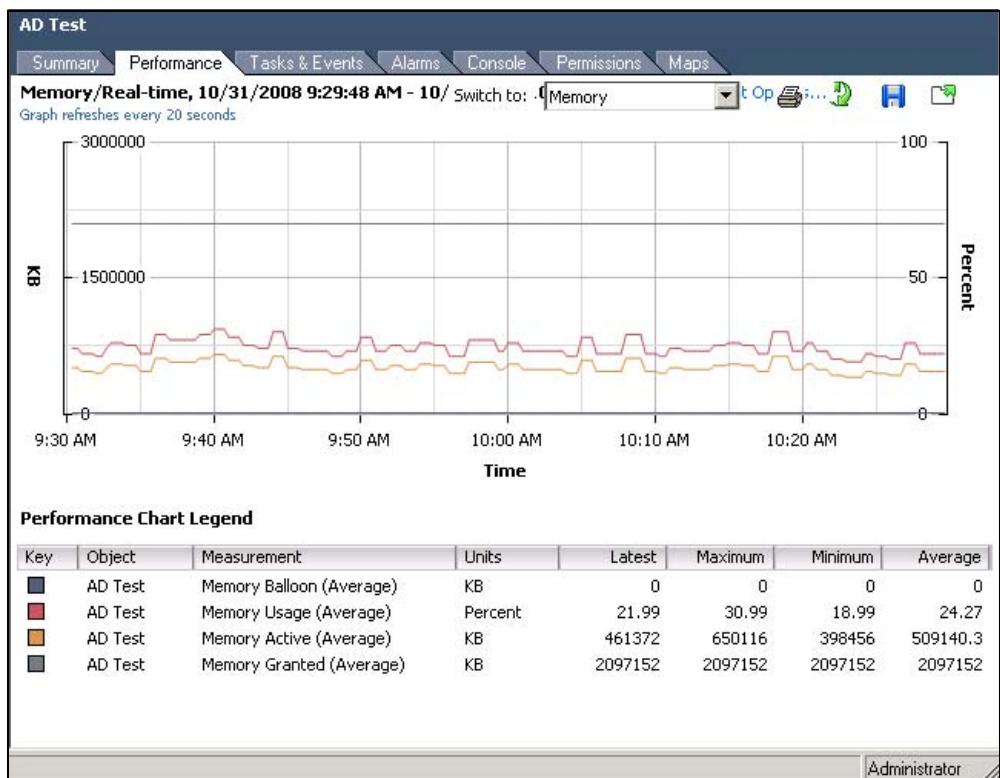


Figure 19-10 Performance tab: Memory usage by selected VM

# Working with bottlenecks

In this part, we demonstrate how to analyze your system to find performance bottlenecks and what to do to eliminate them. We describe an approach you can take to resolve a performance bottleneck, providing details about what to look for and how to solve problems.

We include a sample analysis of real-life servers, showing how tools can be used to detect bottlenecks and explaining the recommendations for particular systems.

This part includes the following chapters:

- ▶ Chapter 20, “Spotting a bottleneck” on page 663
- ▶ Chapter 21, “Analyzing bottlenecks for servers running Windows” on page 691
- ▶ Chapter 22, “Analyzing bottlenecks for servers running Linux” on page 719
- ▶ Chapter 23, “Case studies” on page 743





## Spotting a bottleneck

A *bottleneck* occurs when any server subsystem prevents the other subsystems from running at peak capacity.

This chapter can help you detect a bottleneck problem and show you what to look for so that you will have all the data that you need to identify possible solutions. Having the information that you need can be useful if you are facing a situation where a performance problem is already affecting a server.

This type of situation is a *reactive* situation, in which you need to follow a series of steps that lead to a concrete solution to restore the server to an acceptable performance level. In addition, over time, experience that you gain from solving server bottlenecks is very useful when performing new server configurations or server consolidation exercises.

There are a number of reasons why you need to fix performance problems, and there is usually a cost that is associated with each of them. To resolve a performance bottleneck problem, you need to be able to answer the following questions:

- ▶ Where is the bottleneck?
- ▶ How can it be fixed?
- ▶ How much will it cost?

You can use this chapter as a methodology to spot server performance bottlenecks, and we provide a number of worksheets that you can complete based on the performance measurements that you gather from the server. We also describe and recommend the following steps as a bottleneck detection strategy:

1. Know your system.
2. Determine if the bottleneck is real or simply a misunderstood expectation.
3. Back up the system.
4. Monitor and analyze each subsystem during the time the bottleneck is expected to occur.
5. Identify the primary bottleneck and any latent bottlenecks.
6. Fix the cause of the bottleneck by making only one change at a time.
7. Repeat from step 4 until you are satisfied with the performance of the system.

**Tip:** Document each step, as well as any changes that you make and their effect on performance.

The topics that we discuss in this chapter are:

- ▶ 20.1, “Achieving successful performance tuning” on page 665
- ▶ 20.2, “Step 1: Gathering information” on page 667
- ▶ 20.3, “Step 2: Monitoring the server’s performance” on page 669
- ▶ 20.4, “Step 3: Fixing the bottleneck” on page 687
- ▶ 20.5, “Conclusion” on page 689

**Note:** Throughout this chapter we use the term *processor* to generally represent the logical processors as they appear to the host operating system.

Due to the complexities of multiple cores per physical processor socket, as well as multi-threading technologies like Hyper-Threading (HT) and Simultaneous Multi-Threading (SMT), the number of logical processors appearing to the operating system can be much greater than the number of physical CPU sockets.

For the sake of bottleneck detection, we are primarily interested in the “per logical core” statistics only, and we use the term *processor* here to imply this unless otherwise stated.

## 20.1 Achieving successful performance tuning

To increase your success rate and reduce the time that you spend on each case, you must use a repeatable methodology. In this section, we list and explain how to achieve successful performance tuning:

- ▶ Perform general maintenance of the server.

The first task is to ensure that server maintenance is current. It is good practice to perform general maintenance on the server, including rebooting, defragmenting the drives, and applying the appropriate drivers, patches, and service packs. In some cases, performance bottlenecks are caused by improper service pack, BIOS, or driver configurations or incompatibilities. Ensure that the server has the latest system software before you waste time chasing bottlenecks.

- ▶ Develop accurate and realistic goals.

Performance improvement quests must have realistic boundaries, because unrealistic expectations might lead you to invest unreasonable amounts of time and money for little gain. It can also be useful to reevaluate your goals during the process.

Be sure to research what the expectations are for this server. For example, does the customer expect the new 8-way server to be twice as fast as the older 4-way server? If so, this may or may not be a valid expectation. When the expectation involves comparing one server to another, check published industry-standard benchmarks that are relevant to the production application environment to get an approximate idea of how the two servers should compare.

Keep in mind that benchmarks are performed by very skilled engineers who know how to extract the most performance from each system. In production environments, however, it might not be possible to obtain a similar result, so err on the conservative side. Also remember that response times can vary widely, depending upon the system and the supporting environment configuration. Transactional throughput is the only accurate way to compare the multiuser capacity of two different servers.

This period of investigation is also the perfect time to identify expectations. For example, if a customer says, “Our server is slow,” be aware that this can mean many things, and you need to ask how much faster they need it to be. Document the answer and try to obtain a reasonable expectation before you launch into an extensive bottleneck detection effort. If the customer wants the system to be five times faster, this might not be practical without a server replacement, application modification, or both.

- ▶ Gather relevant background information.

This information is key to successful bottleneck removal. To gather relevant background information, use an iterative process for questioning. A useful starting point is to ask the list of questions that we provide in 20.2, “Step 1: Gathering information” on page 667.
- ▶ Have a good understanding of the system.

This step is half of the solution. Do not try to diagnose a complex multiserver bottleneck if you are just learning the basics. Do your homework and ask for expert assistance if needed.
- ▶ Use a methodological approach.

Time and stress are parameters that are often encountered during a troubleshooting situation. Simply “trying this or that” will not get you anywhere. Prepare your battle plan before getting to the site, and keep it simple, methodical, and consistent.
- ▶ List all necessary performance metrics and counters.

Having a good understanding of the system and the performance problems should lead you to a set of parameters that can help you resolve the situation. As you understand the problem in more detail, use more specific counters to focus on the problem. Start with the simple counters first, then dig deeper into the bottlenecked component.
- ▶ Gather a baseline of the system’s current performance.

If you have stated your goals, you need to be able to measure the results of your actions. Without a baseline, you cannot tell if you have met your goals.
- ▶ Validate your interpretation of counter values.

Try, if possible, to record all counters for the recommended objects that we list in 20.3, “Step 2: Monitoring the server’s performance” on page 669. These counters give you all the data that you need to drill deeper into problems as you learn more about the bottleneck, without having to take multiple traces.
- ▶ Make a permanent record of your progress.

Document your steps, changes, and results in a dedicated performance notebook (loose pieces of paper tend to disappear into the recycling bin). You will thank yourself in six months when similar problems occur. Over time, such documentation can help you build your bottleneck detection expertise.
- ▶ If in doubt, contact an expert.

Replacing hardware is costly, not only in terms of parts but also of machine downtime. If you do not feel comfortable about a solution, consult experts. This consultation is where your good understanding of the system and your notebook can prove handy.

## 20.2 Step 1: Gathering information

Most likely, the only first-hand information you have are statements such as: “There is a problem with the server.” It is crucial to probe questions to clarify and to document what the problem is. Here is a list of questions that you need to ask to help clarify the problem:

- ▶ Can you give me a complete description of the server in question?
  - Model
  - Age
  - Configuration
  - Peripheral equipment
  - Operating system
- ▶ Can you tell me what the problem is *exactly*?
  - What are the symptoms?
  - Is the number of users on the server now the same as when the server was installed?
  - Description of any error messages.

Some people might have difficulty answering these questions, so any extra information that you uncover might provide hints about how to solve the problem. For example, someone might say: “It is really slow when I copy large files to the server.” This information might indicate a network problem or a disk subsystem problem.

Keep in mind that people often describe problems by discussing poor response time. However, server response time is only a part of the picture. The most important metric for predicting server performance is throughput. For example, how many transactions per second or bytes per second is the server sustaining, and how much is needed?

Knowing the answers to these questions can help you to determine whether the server will ever be able to support the required load. No amount of server optimization is going to improve performance to the desired level if the customer needs higher bandwidth than the network can possibly sustain.

- ▶ Who is experiencing the problem?

Is one person, one particular group of people, or the entire organization experiencing the problem? This information helps you to determine whether the problem exists in one particular part of the network or if it is application-dependent and so on. If only one user is experiencing the problem, then the problem might be the user’s personal computer.

The perception that clients have of the server is usually a key factor. From this point of view, a performance problem might not be directly related to the server. The network path between the server and the clients could easily be

the cause of the problem. This path includes network devices as well as services provided by other servers such as domain controllers.

► Can the problem be reproduced?

All reproducible problems can be solved. If you have sufficient knowledge of the system, you should be able to narrow the problem to its root and decide which actions should be taken.

If the problem can be reproduced, you can see the problem and understand it better. Document the sequence of actions necessary to reproduce the problem at any time:

- What are the steps necessary to reproduce it?

Knowing the steps can let you reproduce the same problem on a different machine under the same conditions. If this works, you have the opportunity to use a machine in a test environment and eliminate the risk of crashing the production server.

- Is it an intermittent problem?

If the problem is intermittent, your first task is to gather information and find a path to move the problem to the reproducible category. The goal here is to have a scenario to make the problem happen on command.

**Important:** This step is critical. If you cannot reproduce the problem, there is little chance of finding the bottleneck by taking a trace, unless you are extremely lucky.

- Does the problem occur at certain times of the day or certain days of the week?

This information might help you determine what is causing the problem. It might occur when everyone arrives at work or returns from lunch. Look for ways to change the timing (that is, make it happen less often or more often). If there are ways to do so, the problem becomes a reproducible one.

- Is it unusual?

If the problem falls into the non-reproducible category, you might conclude that it is the result of extraordinary conditions and classify it as fixed. In real life, however, there is a high percentage that it will happen again.

► When did the problem start? Was it gradual or did it occur very quickly?

If the performance issue occurred gradually, then it is likely to be a sizing issue. If it appeared overnight, then the problem could be caused by a change made to the server or peripherals.

- ▶ Have any changes been made to the server (minor or major), or are there any changes in the way clients are using the server?  
Did the customer alter something on the server or peripherals recently that might have caused the problem? Is there a log of all network changes available?  
Demands could change based on business changes, which could affect demands on servers and network systems.
- ▶ Are there any other servers or hardware components involved?
- ▶ Are there any logs available?
- ▶ What is the priority of the problem? When does it need to be fixed?
  - Does it need to be fixed in the next few minutes or days?
  - How massive is the problem?
  - What is the related cost of the problem?

## 20.3 Step 2: Monitoring the server's performance

**Important:** Before taking any troubleshooting actions, back up all data and the configuration information to prevent a partial or complete loss.

At this point, you should begin monitoring the server. The simplest way to monitor the server is to run monitoring tools from the server that is being analyzed. See the appropriate chapter for your operating system:

- ▶ Chapter 17, “Windows tools” on page 533
- ▶ Chapter 18, “Linux tools” on page 607
- ▶ Chapter 19, “VMware ESX tools” on page 649

**Note:** The remainder of this chapter applies to each of these operating systems. However, the specific counters are from Performance Monitor in Windows.

You need to create a performance log of the server during its peak time of operation (for example, 9:00 a.m. to 5:00 p.m.). When creating the log, if available, include a minimum of the following objects:

- ▶ Processor
- ▶ System
- ▶ Memory
- ▶ Physical disk
- ▶ Network interface

Based on server type, analyze the important subsystems that are likely to be the source of the performance bottleneck. See Chapter 2, “Understanding server types” on page 13 for details about which subsystems are important.

Then, complete the information in Table 20-1.

*Table 20-1 Server type and key subsystems*

Information	Your details
Server type	
Important subsystems (from Chapter 2, “Understanding server types” on page 13)	1. 2. 3. 4.

Before you begin, remember that a methodical approach to performance tuning is important. The process we recommend you use for System x server performance tuning is as follows:

1. Understand the factors affecting server performance. This book helps you to do so.
2. Measure the current performance to create a performance baseline to compare with your future measurements and to identify system bottlenecks.
3. Use the monitoring tools to identify a performance bottleneck. By following the instructions in the following sections, you should be able to narrow the bottleneck to the subsystem level.
4. Improve the component that is causing the bottleneck by performing the appropriate actions to improve server performance in response to demands.

**Important:** You can obtain the greatest gains from upgrading a bottlenecked component when the other components in the server have ample *capacity* left to sustain an elevated level of performance.

Figure 20-1 shows the position of five subsystems (CPU, memory, network, disk, and operating system) represented by a letter from *a* to *e* on a performance scale. A well-optimized system has all of its subsystems grouped together. This figure has one component (*a*) which represents the system bottleneck. Component *a* needs to be moved closer to the other subsystems.

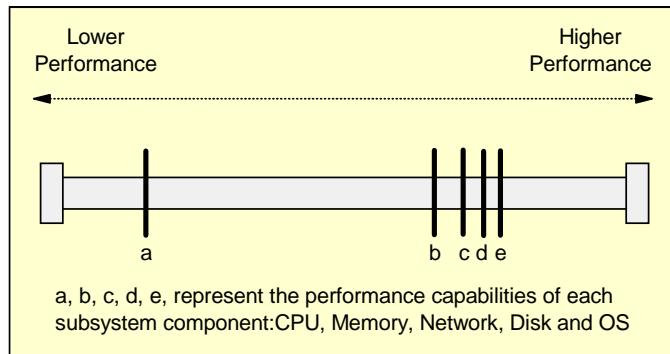


Figure 20-1 System with one primary bottleneck (*a*) but well-balanced other subsystems

In addition, ensure that other components in the server are not *latent* bottlenecks working just below the utilization of the bottlenecked component. Latent bottlenecks limit improvements that are realized by any upgrade. In general, components that have average utilization between 60% to 70% are likely to be latent bottlenecks.

If there are latent bottlenecks in a system, then you must reconfigure or upgrade both the primary component that is causing the bottleneck and the component that has a latent bottleneck to obtain optimal performance. Figure 20-2 shows one primary (component *a*) and one latent (component *b*).

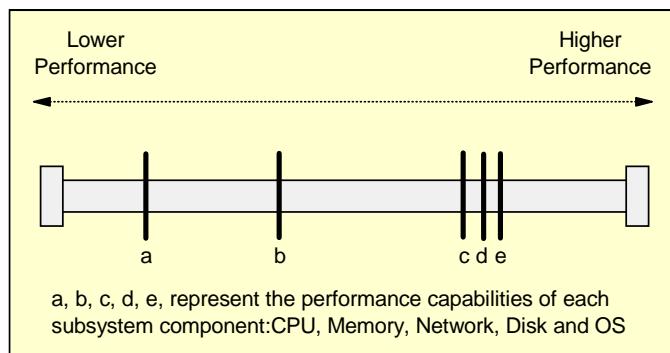


Figure 20-2 System with a primary (*a*) bottleneck and a latent (*b*) bottleneck

In this case, upgrading component *a* moves the primary bottleneck to component *b*, as shown in Figure 20-3. If several components are causing latent bottlenecks, perhaps the most cost-effective solution is to replace the entire server.

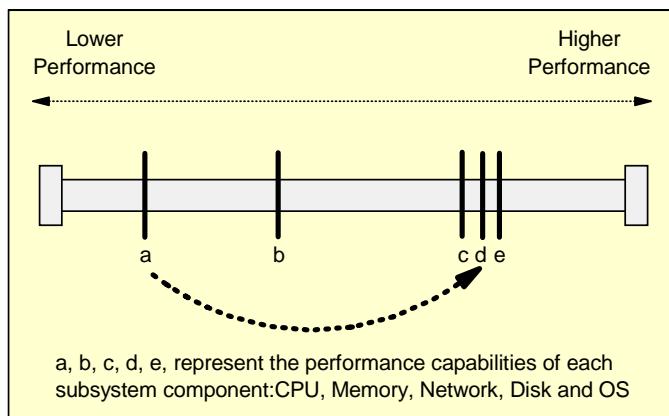


Figure 20-3 Moving the bottleneck from (a) to the latent bottleneck (b)

5. Measure the new performance so that you can compare the performance before and after the tuning steps.

Performance tuning is a continuous process, so you must maintain ample record-keeping to help analyze future demands. In this way, you can predict and avoid problems before they occur.

Analysis tools, such as System Monitor in Windows, give access to all the listed objects and counters in the remaining tables in this chapter. The number of objects and counters that you measure and how often you measure them depend on two questions:

- ▶ What are you trying to detect?
- ▶ What kind of disk resources do you have available?

The sampling period, or the amount of time between data collection points, always comes with a trade-off. The smaller the sampling period, the bigger the file will be. On the other hand, an extremely long sampling period will not be suited to detecting temporary peaks in the system utilization. For example, if you record all the counters listed in this section, you could expect a log file size for a one-second sampling rate to be close to 30 MB per hour of recording.

**Tip:** For an 8-hour trace, a sample time of 60 seconds or longer is typically sufficient to diagnose consistently slow server performance.

Keep in mind that the techniques recommended in this section do not require high-resolution sampling unless the problem you are diagnosing only occurs for a brief period of time. If the problem is a consistently slow server, then use sample times that result in manageable-sized trace files.

If space is a problem and you want to record for a long period of time, use the circular logging option in the Performance console. Using the alerts that we explain in 17.1.3, “Using Data Collector Sets” on page 546 can also help you.

Sampling can also be a two-step action: first, record using only counters from the different subsystems until you can narrow down the search to one or two subsystems. After that, you can increase the number of counters in one subsystem and decrease the sampling rate. The obvious downside to this method is the extra work and inconvenience of taking multiple traces.

### 20.3.1 Where to start

Now that you have a good idea about what you want to accomplish, how do you proceed? Which counters should you use, and how do you know when a bottleneck occurs? These are the crucial steps which cause many people to stumble, because there are an enormous number of complex performance objects and associated counters. However, detecting server subsystem bottlenecks is quite straightforward after you understand the primary subsystems and the primary subsystem counters that you can use to diagnose the health of each subsystem.

Our strategy for bottleneck detection uses a *top-down* approach. In a top-down approach, you take a high-level look at each of the server's primary subsystems by examining each of the primary counters that indicate bottlenecks are present.

First validate the health of each primary server subsystem. The best way to determine if a subsystem is performing poorly is to identify the primary counters and the corresponding thresholds that can be used to identify performance bottlenecks for that subsystem. Then examine each of the primary counters and compare them to the thresholds to determine whether the subsystem is healthy or unhealthy.

These primary performance counters are critical, because you can use them as a pass/fail test to determine the health of a subsystem. Only after a subsystem has failed the primary counter test do you need to perform more extensive analysis. This pass/fail testing makes bottleneck detection much easier, because you can avoid all the complex counters for any subsystem that passes the primary counter test because that system is healthy.

Table 20-2 lists the primary performance objects and associated counters, along with the corresponding threshold for each of the server subsystems that we used for our pass/fail test (in order of most likely to cause a bottleneck).

*Table 20-2 Primary performance objects*

Subsystem	Counter	Guidance	Your value
Disk	Physical Disk: Avg. Disk sec/Transfer	Must be lower than about 25 ms.	
Memory	Memory: Available Bytes	Should be no lower than 20% to 25% of installed memory. However, some applications like IIS, Exchange, and SQL Server will increase their working sets to consume available memory. In these cases, carefully monitor Paging activity.	
Memory	Memory: Page Reads/sec	Ideally should be zero (0), but sometimes this is not possible because some applications, such as Lotus Domino and SAP, use the page file for memory mapped file communication. In any event, the combined value of this counter and Page/Writes/sec should not be higher than about 150 I/Os per second per disk used in the page file device.	
Memory	Memory: Page Writes/sec	Ideally, should be zero (0).	
Processor	Processor: % Processor Time_Total	Total processor utilization should be lower than about 70% to 80%.	
Processor	Processor: % Processor Time_(N)	Each processor instance should be lower than about 70% to 80% utilization.	
Network	Network Interface: Bytes Total/sec	Should be lower than about 50% to 60% of maximum sustainable bandwidth. For Gigabit Ethernet in a Xeon-based system, this is about 70 to 80 MBps.	
Network	Network Interface: Packets/sec	Should be no higher than about 100,000 packets per second total for Gigabit Ethernet in a modern Xeon-based server.	

To reiterate, our top-down approach is to perform a pass/fail test for each of the primary counters listed in Table 20-2. This pass/fail testing provides a simple, objective way to determine systemically if each subsystem is healthy.

Only after you find one or more unhealthy subsystems do you then drill down deeper to learn more about the bottleneck. If you find bottlenecks, you can refer to Chapter 21, “Analyzing bottlenecks for servers running Windows” on page 691 to determine whether you relieve the bottleneck by upgrading or by tuning the specific subsystem.

After tracing hundreds of server configurations, we have learned that the most likely server hardware components to cause a bottleneck are, in order:

- ▶ Disk subsystem
- ▶ Memory subsystem

Disk and memory technologies have lagged significantly behind the performance curve of processors and network technology, which is one reason why disk and memory are the two most frequently found server bottlenecks.

However, equally significant is that many administrators configure server disk subsystems based solely on capacity requirements. Often this limitation means purchasing fewer, higher-capacity disks; this is especially an issue when the highest capacity drives are SATA-based, and thus the slowest technology. In these cases, there are fewer disk spindles to service the required I/O data rates.

Because the disk subsystem is so often the bottleneck, we start our analysis there.

### 20.3.2 Disk subsystem

The disk subsystem is comprised of the disk controller and its device driver, the SCSI, SAS, SATA, or Fibre Channel bus that connects the system to the disks, and finally the individual disk drives. One key point to understand about disk subsystems is that for most commercial server workloads, physical disk I/O is almost always random. Servers provide data storage for the entire population of network-attached users connected to that server. Each user is requesting different data from different locations on the disks of the server. The operating system works to cache data into buffers in memory, and the disk controller caches data into the controller cache. The on-disk file system will become fragmented over time as new and modified data is written to new areas of the physical disk surface.

So by the time a disk I/O actually reaches the disk drive, it is almost always to a very different address than the previous disk I/O request. This means the disk controller has to process the I/O command, send the command to the disk, move the head to the new data track (seek), wait for the correct data location on disk to rotate underneath the head (rotational latency), read or write the data, and send a completion status back to the controller to notify the operating system that the I/O is complete.

Even if server users are executing applications that are performing sequential I/O, because of all the caching and disk fragmentation, much of the physical I/O at the physical disk drive can still be random. Most commercial applications do not perform sequential I/O. Databases, e-mail, file serving, and most multiuser commercial applications generate random disk I/O, which introduces seek and rotational latency delays for nearly every disk access, thereby greatly reducing sustained throughput. Thus, we should not expect disks to generate very high data rates for most commercial servers.

Of course, there are always exceptions. High Performance Computing (HPC) servers might run a single process that reads a large array of data from disk, and then writes a large solution set back to disk. In this case, a single process will be accessing the disk, not a large number of concurrent users generating multiple unique disk I/O requests. Furthermore, HPC workloads tend to read disk data using very large disk I/O sizes, thereby increasing the sustained disk I/O bandwidth.

Another example can be decision support systems, where very large databases are scanned to process queries. While not all decision support environments will have sustained sequential I/O, these environments do tend to have higher bandwidth requirements than most other transactional workloads.

For these high bandwidth workloads, it is possible to saturate PCI subsystem or disk subsystem bandwidth limitations. However, in most commercial workloads, seek and latency operations dominate the I/O time and significantly lower sustained I/O rates.

A closer look at the time to perform disk I/O operations provides a critical understanding of how to avoid and diagnose disk subsystem bottlenecks for commercial workloads (random I/O).

Take a look at a typical high-speed 15,000 RPM disk drive. For this disk, the total access time can be calculated as:

Average seek	3.8 ms
Average latency	2.0 ms
Command and data transfer	< 1 ms
<b>Average random access time:</b>	<b>6.8 ms per operation</b> (147 operations/sec)

It takes about 0.0068 seconds for a 15,000 RPM disk to perform an average disk operation. Therefore, in one second the disk can only do about 147 I/Os per second. This is calculated as  $1 / 0.0068 \text{ sec} = 147 \text{ I/Os per second per disk}$ .

Because most commercial applications are accessing data on disk in 4 KB, 8 KB, or 16 KB sizes, the average bandwidth sustained by a disk drive can easily be calculated. For example, at 8 KB I/O size:

$$8 \text{ KB per I/O} * 147 \text{ I/Os per second} = 1.15 \text{ MBps per disk}$$

At about 1.15 MBps per disk, it takes a significant number of disks to stress the PCIe bus (x8 PCIe is rated at 2GBps per direction), the SAS bus (a single x4 SAS uplink is rated at 1.2 GBps), or Fibre Channel (4 Gbit = 400 MBps full duplex). Far too often people are concerned with PCIe and SAS bus configurations, when in fact they are usually no cause for concern.

In general, the number of disk drives, disk fragmentation, RAID strategy, and the ability of the application to queue a large number of disk I/O commands to the physical array are the leading causes of commercial server disk subsystem bottlenecks.

For 10,000 RPM disks, the same calculations can be:

Average seek	4.9 ms
Average latency	3.0 ms
Command and data transfer	< 1 ms
<b>Average random access time:</b>	<b>8.9 ms per operation (112 operations/sec)</b>

8.9 ms corresponds to  $1 / 0.0089 = 112$  I/O operations per second, and at 8 KB per I/O, a 10,000 RPM disk can sustain only about:

$$8 \text{ KB per I/O} * 112 \text{ I/Os per second} = 896 \text{ KBps per disk} (\approx 900 \text{ KBps}).$$

However, using these calculated I/O rates as direct indicators for disk performance bottlenecks could be a serious mistake, because I/O rates can vary wildly depending on disk usage. For example:

- ▶ In some special cases, disks perform sequential I/O. When this occurs, seek and rotational latency will be zero or near-zero, and disk I/O rates will increase dramatically. Even though this is rare, we do not want to use a performance indicator that works only some of the time.
- ▶ Our calculations assume average seek and latency times. Drive vendors produce average seek times from 1/3 track-seek range measurements. Full track seek times are much longer, and disks that are accessing more than 1/3 of capacity will have longer seek times and significantly lower sustained I/O rates.
- ▶ RAID strategy affects the number of physical I/Os a disk will actually perform. A random write to a RAID-5 disk array will generally produce two read and

two write disk operations to the RAID-5 array, but operating system disk counters count this as one disk I/O.

- ▶ Stripe size and I/O request size will affect the number of physical disk operations performed. For example, a very large disk read or write operation of 256 KB in size sent to an array that is using a 64 KB stripe size will generate 4 physical disk I/O operations. But the operating system disk counters will count this as one disk I/O because the operating system does not know anything about the stripe size the disk array controller is using.

Sustained disk I/O rates can vary greatly. Therefore, we do *not* recommend using average disk I/O rates as an indicator of a disk bottleneck unless you thoroughly understand the disk workload and storage configuration.

A better way to identify disk bottlenecks is to apply an understanding of disk operation combined with average response time. We know a 15,000 RPM disk requires about 7 ms of average disk access time and a 10,000 RPM disk has about 9 ms of disk access time. We can use this information to greatly simplify disk bottleneck detection. However, before we launch into that discussion, we need to discuss one more characteristic of disk drive operation: optimization.

Modern disk drives can actually increase the sustained throughput when given more work to do. If multiple read and write operations are sent to a disk drive, it can use *elevator seek optimization* and *rotational positioning optimization* to reorder the physical I/Os to increase the sustained I/O rate as compared to when processing a single disk read or write operation at a time.

By sending two or more disk commands to the disk, it can reorder the operations to reduce the amount of seek time and even rotational latency. However, even more significantly, when a seek is occurring for an existing I/O request, and another disk I/O command arrives, the disk can determine if it can access that data while performing the current seek command. That is, while a long seek operation is occurring, the processor on the disk determines if the head will pass over any of the data addresses for read or write commands that just arrived in its queue after the long seek was started. If so, the read or write command in the queue will be executed while the head is moving out to the track for the original I/O.

The key message from the discussion of disk I/O operation is this: disk drives perform best and have optimal throughput when given two to three (no more than three) disk operations at the same time.

As a rule of thumb, the optimal response time of a disk drive is about 2.5 times the normal access time.

- ▶ For 15,000 RPM disk drives, this is about 17 ms.
- ▶ For 10,000 RPM disk drives, the optimal response time is about 22 ms.

For bottleneck detection purposes, use a range of values instead of a precise number. A good rule of thumb to use is that a disk subsystem is healthy whenever the disk subsystem is performing read and write operations with less than about 20-25 ms per I/O. When the average disk latency is much greater than 25 ms, then the disk subsystem can be considered unhealthy and is a bottleneck.

**Rule of thumb:** When Avg. Disk Seconds/Transfer (the disk latency counter) is significantly greater than 25 ms, the disk subsystem is unhealthy and is a bottleneck. Remember, this counter does not tell us *how* to fix the problem, it only indicates there *is* a problem.

You can look at one simple Performance Monitor counter and know if your disk subsystem is healthy or not. Simply look at Avg. Disk Sec/Transfer for each physical disk drive or array in your server. Use the chart mode to identify the peaks, and if this counter spends a significant amount of time over 25 ms during the period where the server is considered to have a bottleneck, you can consider your disk subsystem unhealthy.

Do not order a new SAN if your average disk latency is 26 ms, however. Clearly, there is a range of latencies where performance will be acceptable. Each server administrator must decide when to consider the average latency too great. Some server administrators will use 30, 40 or 50 ms; other server administrators will want ultimate performance and take action at 25 ms.

However, if the disk subsystem is running at 60 or 80 ms on a regular basis, then the disk subsystem is clearly slowing down server performance. On many occasions, we have seen overloaded disk subsystems performing with 1 or 2 seconds of average latency (1000 or 2000 ms). This is a very slow disk subsystem.

Although most recent SAS or FC controllers will perform best for most workloads when their write back caches are enabled, there may be environments where this is not optimal. In any case, where the server is performing a large amount of write operations for RAID-5 with write-through disk controller settings, we want to use about 40-50 ms as our threshold for identifying disk performance bottlenecks.

After you have identified the disk subsystem as unhealthy, Chapter 21, “Analyzing bottlenecks for servers running Windows” on page 691 can help you understand how to improve the performance of an unhealthy disk subsystem.

Complete Table 20-3 with your results. When you are done, examine the primary disk counter in the table to determine if the disk subsystem is a bottleneck. If it is, then refer to 21.4.1, “Analyzing disk bottlenecks” on page 704 for more details regarding how to analyze and resolve the disk bottleneck.

*Table 20-3 Performance console counters for detecting disk bottlenecks*

Counter	Is a bottleneck if...	Your result
Physical Disk: Avg. Disk sec/Transfer	<p><b>This is the primary counter for detecting disk bottlenecks.</b></p> <p>This is the time to complete a disk I/O. For optimal performance, this should be less than 25 ms. Consistently running over .025 s (25 ms) indicates disk congestion.</p> <p><b>Note:</b> Examine the counters for physical disk, <i>not</i> logical disk.</p>	
Physical Disk: Avg. Disk Queue Length	<p>It is necessary to know how many disks are used in the RAID Array represented by this Physical Disk counter. In general, optimal performance is obtained when this counter averages 2 to 3 times the number of disks in each physical array.</p> <p>A high number indicates queuing at the physical volume. This is undesirable because it increases response time and degrades performance. A high number here indicates the application I/O workload will typically scale simply by adding disks to the array.</p>	
Physical Disk: Avg. Disk Bytes/Transfer	<p>This is the average number of bytes transferred to or from the disk during write or read operations.</p> <p>Also compare this value against the stripe size of the RAID array (if you are using hardware-based RAID). We recommend you configure the stripe size to be at least equal to the long-term average value of this counter. For example, if the Avg. Disk Bytes/Transfer is 16 KB, then use an 16 KB or larger stripe size on the RAID array volume. Note that many SAS adapters now default to large stripe sizes of 64 KB to 128 KB, which may not be optimal if this counter's average is significantly smaller.</p>	

Counter	Is a bottleneck if...	Your result
Physical Disk: Disk Bytes/sec	Sum this counter's value for each disk drive attached to the same SAS/Fibre Channel controller and compare it to 70% to 80% of the theoretical throughput. If these two numbers are close, the bus is becoming the disk subsystem's bottleneck. Review the disk subsystem data path.	
Physical Disk: Split IO/sec	A split I/O is a result of two different situations: the data requested is too large to fit in one I/O, or the disk is fragmented. If split IO/sec is more than ~10% of the total Physical Disk: Disk Transfers/sec, this could indicate high disk fragmentation, though any non-zero value may warrant further investigation.	

### 20.3.3 Memory subsystem

Most memory counters are related to virtual memory management. Virtual memory counters will not help identify if the server has insufficient physical memory capacity and is running poorly as a result of excessive disk paging. However, there are a few counters that can help you to determine if the memory configuration is healthy.

Complete Table 20-4 on page 682 by monitoring the memory counters on your server. Then examine the primary memory counters to determine if the memory capacity is causing a bottleneck. If it is, then refer to 21.3, “Analyzing memory bottlenecks” on page 697 for more detail regarding these counters.

Table 20-4 Performance console counters for detecting memory bottlenecks

Counter	Is a bottleneck if...	Your result
Memory: Page Reads/sec	<p><b>This is a primary memory bottleneck counter.</b></p> <p>Ideally, this value should be close to zero (0). However, sometimes it is not possible to eliminate paging because some applications (such as Lotus Domino) use the page file for communication between processes.</p> <p>However, if paging is so high as to saturate the page disk device then performance will suffer. If this counter is consistently higher than 150 I/Os per disk per second for the paging device, the server likely has a memory or paging device bottleneck.</p> <p>Page Reads/sec is the rate at which the disk was read to resolve hard page faults. It shows the number of read operations, without regard to the number of pages retrieved in each operation. Hard page faults occur when a process references a page in virtual memory that is not in a working set or elsewhere in physical memory, and must be retrieved from disk.</p> <p>This counter is a primary indicator of the kinds of faults that cause system-wide delays. It includes read operations to satisfy faults in the file system cache (usually requested by applications) and in non-cached mapped memory files. Compare the value of Memory:Pages Reads/sec to the value of Memory:Pages Input/sec to determine the average number of pages read during each operation.</p>	
Memory: Page Writes/sec	<p><b>This is a primary memory bottleneck counter.</b></p> <p>Consistent non-zero values for this counter are likely to indicate that memory is paging to disk.</p> <p>Page Writes/sec is the rate at which pages are written to disk to free up space in physical memory. Pages are written to disk only if they are changed while in physical memory, so they are likely to hold data, not code. This counter shows write operations, without regard to the number of pages written in each operation. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.</p>	
Memory: Available MB	<p><b>This is a primary memory bottleneck counter.</b></p> <p>This should be no lower than 20% to 25% of installed memory; however, some applications like IIS, Exchange, and SQL Server will increase their working sets to consume available memory. In these cases, carefully monitor Paging activity.</p>	

Counter	Is a bottleneck if...	Your result
Memory: Pool Nonpaged Bytes	This indicates the amount of RAM in the non-paged pool system memory area where space is acquired by operating system components as they accomplish their tasks. If this value has a steady increase without a corresponding increase in activity on the server, it might indicate that a process that is running has a memory leak, and should be monitored closely.	
Paging File: % Usage Peak	This is a bottleneck if the value consistently reaches 90%.	
Server: Pool Nonpaged Peak	This is the maximum number of bytes of nonpaged pool the server has had in use at any one point. It indicates how much physical memory the computer should have. Add 20% to this value to determine the amount of installed memory that the server should require.	
Server: Pool Nonpaged Failures	This is the number of times that allocations from the nonpaged pool have failed. If this value is not zero on a regular basis, the system likely needs additional memory.	

#### 20.3.4 Processor subsystem

Determining processor bottlenecks is much more straightforward than the other server subsystems. If the %Processor Time Total of any single processor is sustaining over 70% to 80% utilization, it should be considered a bottleneck.

Complete Table 20-5, then examine the primary processor counters to determine if the CPU is a bottleneck. If it is, refer to 21.2.1, “Finding CPU bottlenecks” on page 693 for more detail regarding these counters and how to eliminate processor bottlenecks.

Table 20-5 Performance console counters for detecting CPU bottlenecks

Counter	Is a bottleneck if...	Your result
Processor:% Processor Time	<p><b>This is the primary counter for detecting processor bottlenecks.</b></p> <p>The CPU is a bottleneck if the value is consistently running over 70% to 80% (excluding any processes that are running in the background at low priority, absorbing all spare CPU cycles).</p> <p><b>Note:</b> Examine the counters for each CPU installed as well as the _Total value to ensure there are no problems with “unbalanced” or processor affinitized applications.</p>	

Counter	Is a bottleneck if...	Your result
Processor: % Privileged Time	<p>This counter can be used (excluding any processes that are running in the background at low priority, absorbing all spare CPU cycles) to identify abnormally high kernel time, and it might indicate I/O driver problems.</p> <p><b>Note:</b> Examine the counters for each CPU installed as well as the _Total value to ensure there are no problems with “unbalanced” applications.</p>	
Processor: % User Time	<p>%User time represents the time spent by the user application on the server. This is an important counter because it shows a breakdown of how the server application is utilizing all the processors.</p> <p><b>Note:</b> Examine the counters for each CPU installed, as well as the _Total value, to ensure there are no problems with “unbalanced” application usage.</p> <p>Often, when applications do not scale, they will not start user threads for all the processors in the server. For example, if processors 0 and 1 are running at high % User Time while the remaining processors are running at much lower % User Time, this may indicate insufficient application threading.</p>	
System: Processor Queue Length	<p>A sustained queue much greater than four times the number of processors installed indicates processor congestion.</p> <p>However, an application that drives the processor queue to a long length can take advantage of a large number of processors.</p> <p>When the server is running at above 75% CPU utilization, check this counter to see if the average queue length is significantly greater than two times the number of installed processors. If so, that application will scale as additional processors are added, up to the point where the average Queue Length is equal to <math>2N</math>, where <math>N</math> is the number of processors.</p>	

Counter	Is a bottleneck if...	Your result
Processor: Interrupts/sec	<p>Processor Interrupts/sec should no longer be used as a simple indicator of performance.</p> <p>Modern device drivers have dynamic mechanisms and batch interrupts when the system is busy, doing more work per interrupt. This causes the number of interrupts per second to be dynamic.</p> <p>When a system is moderately busy, it might require an interrupt to process each LAN or DISK I/O operation. In this mode, the server will have a fairly high interrupt rate. However, as the server becomes busier, multiple disk and LAN operations will be sent under one interrupt request, thereby lowering the interrupt rate and improving interrupt efficiency.</p> <p>In summary, do not use this counter unless you have detailed information and expectations for the specific device drivers used in your server.</p>	<p>No need to measure; this is not normally used to analyze bottlenecks.</p>

### 20.3.5 Network subsystem

The network itself can be difficult to analyze. This is because the performance counters captured on the server represent only the load in the network that is destined to or from the particular server where the counters were captured. The counters do *not* reflect the entire load in the network, which could be causing a serious bottleneck for users connected to the same network as they try to communicate with the server.

Unfortunately, the only way to identify network bottlenecks that are not related to the server where the performance counters were monitored is to use a Network Analyzer. Further discussion of that topic, however, is beyond the scope of this publication.

The network subsystem counters can, however, be used to successfully diagnose network bottlenecks caused by excessive traffic to or from a particular server. This excessive traffic will manifest itself as a choke point at the network adapter. Network adapters can have two types of bottlenecks:

- ▶ When the sustainable throughput of the network adapter is reached
- ▶ When the maximum sustainable rate of the network adapter is reached

In general, these values vary with network adapter type and system configuration, so do not rely on them too heavily. However, in general, the counter Bytes Total/sec should be lower than about 50% to 60% of maximum sustainable bandwidth. This means that for a single Gigabit Ethernet port on a

Xeon-based server, you can reasonably sustain ~80 MBps in a typical bi-directional workload.

Similarly, the sustainable packet rates on a Xeon-based server with Gigabit Ethernet should be no higher than ~100,000 packets/sec total per port.

It is important to note that multiple ports may not scale these numbers as linear bandwidth or packet rate multipliers, because the actual sustainable throughput is largely a product of the transaction size being generated from the application. The same holds true for 10 Gigabit Ethernet adapters, where even tuned real-world environments will rarely see more than 4 to 5 Gbps of total sustained throughput.

Complete all fields in Table 20-6. Compare the primary network counters with the sustained thresholds listed to determine if the network subsystem is the bottleneck. If it is, refer to 21.5.1, “Finding network bottlenecks” on page 708 for more detail about how to resolve network bottlenecks.

**Note:** For Windows 2000 Server, you will need to install the Network Monitor Driver and SNMP services prior to performing this analysis.

To monitor network-specific objects in Windows 2000 Server, you need to install the Network Monitor Driver (this is not necessary for Windows Server 2003):

1. Open Network and Dial-up Connections in the Control Panel.
2. Select any connection.
3. Click **File → Properties**.
4. In the General tab, click **Install**.
5. Select **Protocol**.
6. Click **Add**.
7. Select **Network Monitor Driver**.
8. Click **OK** then **Close**.

Table 20-6 Performance console counters for detecting network bottlenecks

Counter	Is a bottleneck if...	Your result
Network Interface: Bytes Total/sec	<b>This is a network subsystem primary counter.</b>  Sustained values over 50% to 60% of the network adapter's available bandwidth are cause for concern. Expected maximum sustained application throughput for a Gigabit Ethernet in a modern server using a typical 70%/30% Read/Write Ration is about 160 MBps.  To be conservative, detailed network analysis is warranted if the Bytes Total/sec value is over about 90 MBps.	

Counter	Is a bottleneck if...	Your result
Network Interface: Packets/sec <i>and</i> Network Interface: Packets Sent/sec <i>and</i> Network Interface: Packets Received/sec	<b>These are network subsystem primary counters.</b> Packets/sec rates should be no higher than about 100,000 Packets/sec total, and no more than ~70,000 for Packets Sent/sec and Packets Received/sec. If these values are exceeded, the server's network, or the supporting network attached to the server, may need further investigation	
Network Interface: Bytes Received/sec	This counter is a network subsystem primary counter. Sustained values over 50% to 60% of the maximum throughput in the receive direction should be investigated by a network administrator to determine if the network is a bottleneck. Most Gigabit Ethernet adapters can sustain about 110 MBps in the receive direction. To be conservative, detailed network analysis is warranted if the Bytes Received/sec value is over about 60MBps.	
Network Interface: Bytes Sent/sec	Sustained values over 50% to 60% of maximum throughput in the send direction should be investigated by a network administrator to determine if the network is a bottleneck. Most Gigabit Ethernet adapters can sustain about 110 MBps for data sends. To be conservative, detailed network analysis is warranted if the Bytes Sent/sec value is over about 60MBps.	

## 20.4 Step 3: Fixing the bottleneck

After you determine which subsystem is the bottleneck, examine the options for solving the problem. We discuss these in the next three chapters. Depending on your specific situation, these options could include:

- ▶ CPU bottleneck:
  - Add more processors.
  - Switch to processors with larger caches.
  - Replace existing processors with faster ones.
- ▶ Memory bottleneck:
  - Add memory.

- ▶ Disk bottleneck:
  - Spread the I/O activity across drives or RAID arrays (logs, page file, and so on).
  - Add additional disks to the RAID array.
  - Use RAID-10 instead of RAID-5 or instead of single disks.
  - Tune the stripe size to match the I/O transfer size.
  - Use faster disks.
  - Add another RAID controller/channel or Fibre Channel host adapter.
  - For Fibre Channel, add a second RAID controller module.
  - If running in Write Back (WB) mode and if Disk sec/Transfer exceeds a sustained value of ~20 ms, selecting Write Through (WT) can yield a 20% to 30% increase in throughput for heavily loaded disk configurations compared to WB mode.

**Note:** WT mode can increase disk latencies for periods of lower disk activity, so this workaround may be best used as a temporary solution only.

- ▶ Network bottleneck:
  - Ensure network card configuration matches router and switch configurations (for example, frame size and flow control mechanisms).
  - Modify how your subnets are organized.
  - Use faster network cards.
  - Add network cards.

When attempting to fix a performance problem, remember the following points:

- ▶ Take baseline measurements. Before you upgrade or modify anything, take measurements so that you can tell if the change had any effect.
- ▶ Examine the options that involve reconfiguring existing hardware, and not simply those that involve adding new hardware.
- ▶ After you upgrade a specific subsystem, other latent bottlenecks might appear in other subsystems which could require attention.

Follow the steps in the flowchart shown in Figure 20-4 on page 689 as a first step to resolving performance problems.

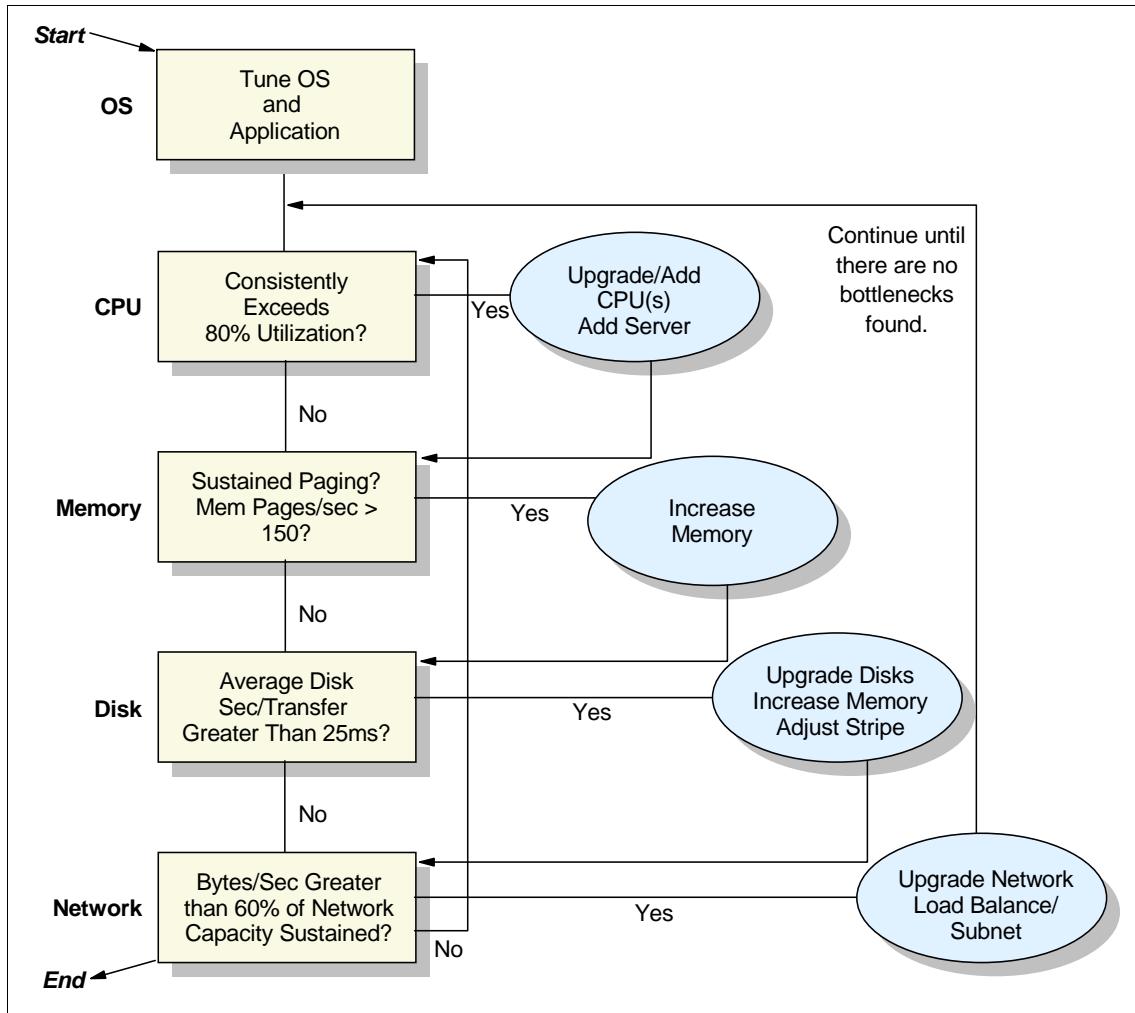


Figure 20-4 Bottleneck flowchart

## 20.5 Conclusion

This chapter provides a general approach that you should take when determining server bottlenecks. When trying to find bottlenecks, also take into consideration the type of server you are monitoring and what subsystems are potential bottlenecks.

Before making any recommendation for a server:

- ▶ Make sure you understand what is causing the bottleneck.
- ▶ Research your recommendations, and be sure what you are proposing will improve server performance.
- ▶ Know how much the upgrade or reconfiguration will cost.



# Analyzing bottlenecks for servers running Windows

This chapter discusses how to use the Performance Monitor console, which is the tool built into Microsoft Windows<sup>1</sup> for monitoring server performance. It also explains how to use the tool's output to analyze server subsystem bottlenecks. How you resolve server bottlenecks depends primarily on your bottleneck detection analysis and findings. Windows Server 2008 now refers to the Performance Monitor console as the Reliability and Performance Monitor. Windows Server 2003 refers to this same functionality as the System Monitor. In this chapter, we will use the terms Performance Monitor and System Monitor interchangeably to generally refer to this tool.

This chapter discusses the following topics:

- ▶ 21.1, “Introduction” on page 692
- ▶ 21.2, “CPU bottlenecks” on page 692
- ▶ 21.3, “Analyzing memory bottlenecks” on page 697
- ▶ 21.4, “Disk bottlenecks” on page 703
- ▶ 21.5, “Network bottlenecks” on page 707

Refer to Chapter 17, “Windows tools” on page 533 for details on using Performance Monitor, and Chapter 20, “Spotting a bottleneck” on page 663 for general tips on identifying a server bottleneck.

<sup>1</sup> Product screen captures and content are reprinted with permission from Microsoft Corporation.

## 21.1 Introduction

Consider the following statements when analyzing Windows Server performance:

- ▶ In general, the most frequently found hardware bottlenecks in servers are caused by the disk subsystem and available memory capacity. Often disk bottlenecks can result from too little available memory. Also, when disk bottlenecks are identified, remember that it is far easier to add memory than to reconfigure the disk subsystem. So, explore the option of adding memory when you discover disk bottlenecks.
- ▶ When you rule out disk subsystem and memory shortages, the processor and network subsystems are likely the next sources for contention.
- ▶ In general, you can achieve the greatest performance gain in Windows Server by tuning and sizing the memory subsystem, disk subsystem, processor configuration, and network subsystem properly, in that order.

## 21.2 CPU bottlenecks

The CPU is one of the first components suspected when there is a performance problem. All server operations, from servicing requests from network-attached clients, to performing the weekly server backups, are processed by the server's CPU. For servers having the primary role of application or database server, the CPU is clearly important. However, there is a common misconception that the CPU is the most important part of the server and can be the single measure when comparing system performance. Unfortunately, in practice, this is often *not* the case.

Modern CPUs have increased in speed at about double the rate that memory has increased in density, and disks have increased in throughput. Many CPUs are so fast that their power often far exceeds the other server components, especially when the server is configured improperly. Servers are often overconfigured with CPU and underconfigured with disks, memory, and network components.

Remember, CPU performance has increased significantly in recent years, and the CPU is usually only a bottleneck when memory, network, and disk subsystems are performing without any bottlenecks. Bottlenecks in any of these other subsystems means the CPUs must wait, which results in lower CPU utilization. Therefore, it usually pays to check the performance of the other subsystems before upgrading or adding additional CPUs to a poor-performing server.

Also consider that the latest processors have evolved to support multiple concurrent threads. Hyper-Threading and Simultaneous Multi-Threading are techniques employed by some Intel processors to better use their fast processor cores by multiplexing two software threads to run on the same processor core. This Hyper-Threading ensures that when one thread is waiting on slower resources such as memory or I/O, the other thread can execute and keep the processor core busier.

However, this dual-thread execution methodology shifts the burden onto the software to create twice as many threads as it did before Hyper-Threading was introduced. In many cases, applications have not evolved to take advantage of this greater parallelism and the additional threads are not always generated. In this case, Hyper-Threading might not yield a significant improvement in performance.

Also consider this same threading issue when migrating older applications to a server with a greater number of processors, especially when doing so to gain increased performance. When adding more processors to the system, an application must detect the additional processors and must introduce more threads to take advantage of the increased parallel execution capabilities. This process does not happen automatically, and older applications do not always detect additional processors. The result is often that the existing applications run no better on the new server with its greater number of processors than they did on the older server.

The application must be closely analyzed when adding additional processors to solve a processor bottleneck. Ask the software vendor if the application is designed to take advantage of Hyper-Threading or greater SMP capability. Also, look at the System: Processor Queue Length counter to see if the application is spawning a large number of threads. If so, this makes it a candidate for solutions with Hyper-Threading or a larger number of processors.

### 21.2.1 Finding CPU bottlenecks

You can use System Monitor processor object counters to help you determine if the processor is the bottleneck by creating a chart as discussed in 17.1.2, “Using Performance Monitor” on page 541. After gathering performance data, analyze it according to the recommendations in Table 21-1 on page 694.

Also consider that Windows Server 2003 and 2008 have much more efficient scheduling routines than the scheduler that is used by Windows 2000. Windows 2000 attempted to balance thread execution across all processors. In contrast, the newer versions work to keep threads that are associated with a particular process together by executing on a *home* or preferred processor.

This *affinity assignment* is done to enhance performance on NUMA-based systems (see 9.2.3, “NUMA” on page 161). It greatly reduces processor cache-to-cache migration of thread code and data that previously occurred in Windows 2000. Affinity assignment significantly increases system efficiency at the expense of a potentially less-balanced execution load. So, under some workloads, expect to see more unbalanced CPU utilization with new versions of Windows, which is more efficient and results in higher overall performance.

Examine the indications of processor bottlenecks based on the object counter readings that you have obtained, then perform the recommended actions in Table 21-1 to rectify the situation.

**Tip:** In Windows Server, you can set processor counters to monitor a specific processor individually or total processor usage of the server. Always examine both sets of counters to determine whether one or more processors is causing a bottleneck that you do not see by looking at the %Total Processor Utilization counter.

Table 21-1 Performance console counters for detecting CPU bottlenecks

Counter	Description
Processor: %Processor Time: <All>	This counter is the percentage of time the processor is busy. When this counter is consistently over 75% to 80%, the processor has become a system bottleneck. Examine processor utilization for each individual CPU (instance), as well as the average for all CPUs in your server. From this counter, you can tell if one or a few CPUs are being used significantly more than the others. This is often indicative of an affinitized application which has outgrown its allocated processing resources.
Processor: %Privileged Time: <All>	This counter measures the time the processor spends performing operating system functions and services. In general, most server applications spend about 20% to 50% of the time in privileged time. If you spot excessive privileged time, you will need to determine if the application is making excessive kernel calls or a device driver is operating incorrectly.
Processor: %User Time: <All>	This counter is the percentage of processor time spent in user mode executing the server application. The percentage of time spent in %User Time compared to %Privileged Time will help you to identify whether the application or the operating system and device drivers are likely causing the CPU bottleneck. If the %User Time makes up the overwhelming component of %Processor Time, the application is consuming most of the processing cycles. In general, this is ideal because you want the application to execute as much of the time as possible. But this also means the application has to be reconfigured or modified in some way for the system to be made more efficient. For verification, you might have to examine CPU usage by process to identify which application process is using the majority of the processing time.

Counter	Description
System: Processor Queue Length: <All>	<p>This counter is the instantaneous length of the processor queue in units of threads. All processors use a single queue in which threads wait for processor cycles. A sustained processor queue length that is greater than 4 times the number of logical processor cores (the number of “processors” that appear in Performance Manager) might indicate a processor bottleneck.</p> <p>This bottleneck means that the processor cannot handle the concurrent thread execution requirements.</p> <p>However, a high System Processor Queue Length indicates the application will scale to a higher performance level on a system configured with additional processors. When the System Processor Queue Length is low, the application might not scale on a system with a greater number of processors unless some configuration parameter is limiting the number of concurrent threads generated by the application.</p> <p>Some applications increase the thread depth when they detect additional processors. Because of this, it is difficult to know with certainty that an application will not take advantage of a greater number of processors. Confirm this with your software vendor.</p>

Figure 21-1 on page 696 shows a sample Performance console chart setting for detecting processor bottlenecks.

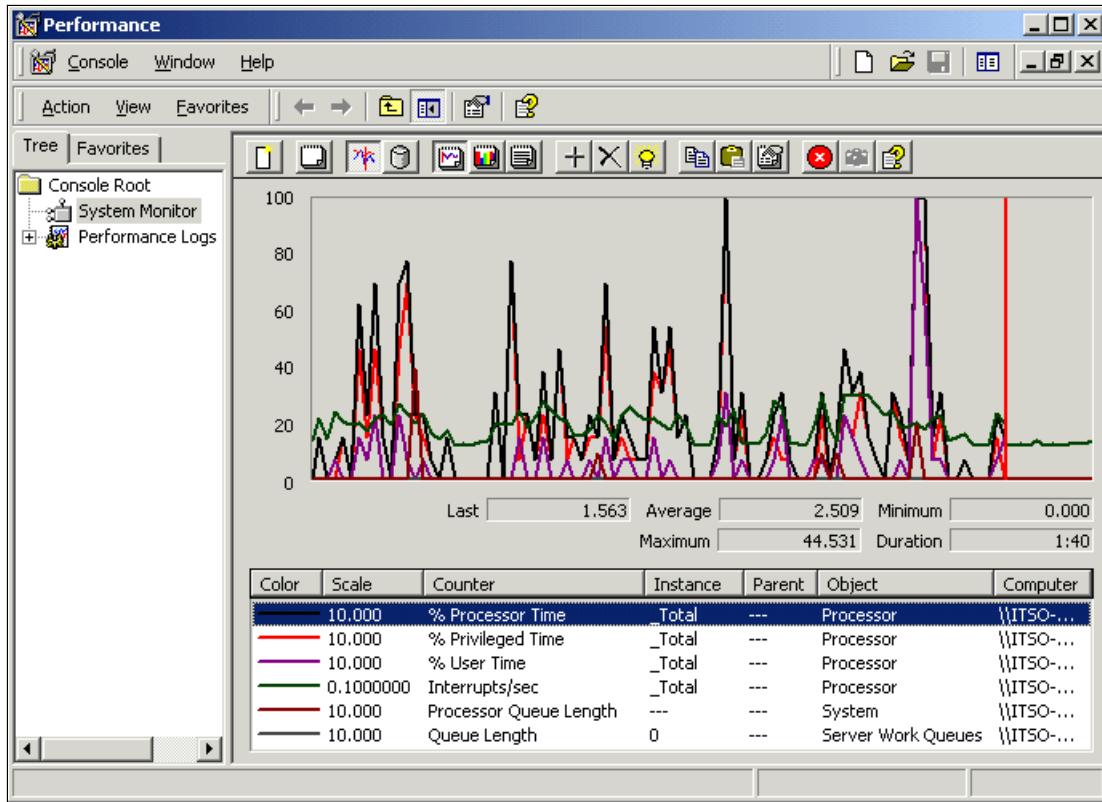


Figure 21-1 Chart setting for finding processor bottlenecks

### 21.2.2 Processor subsystem performance tuning options

If you have determined that your processor subsystem is bottlenecked and causing performance problems you have several basic choices:

- ▶ Upgrade to faster processors or processors with larger caches

Upgrading to faster processors is usually the safest way to solve a processor bottleneck because you can be assured the application will execute faster.

Adding processors places increased threading requirements on the software. However, adding *faster* processors executes the existing threads faster without any additional software support.

- ▶ Add processors

Only add multiple processors to improve performance when you are certain the applications have proper threading and can take advantage of the additional processors. Do *not* take this point lightly, because not all applications scale and can take advantage of additional processors.

In general, the majority of older server applications work well with one to four processor cores. More modern applications are typically written to take advantage of 4 to 8 total processor cores. In general, typically only enterprise middleware and database applications can take advantage of 8 or more cores natively.

A key point to remember is that if the current system is not constantly running at very high CPU utilization (that is, greater than 80% to 90%) then adding processors will likely *not* improve system level performance by any significant amount. If the CPU subsystem is not saturated, then adding more processors or faster processors will simply reduce total sustained CPU utilization with only a slight increase in performance at best. In addition, adding more processors always introduces some overhead (for example, greater scheduling and bus contention).

- ▶ Optimize the software environment

If %Kernel Time is high, it means that the operating system or device driver is using most of the processing time. This can occur if the application makes many calls to the operating system, if the system has an inefficient device driver, or if there is a operating system misconfiguration.

If the majority of time is spent in %User mode, then the applications are using most of the processing power. In this case, examine whether you can configure the application to be more efficient.

Also, examine whether you can schedule some CPU-intensive jobs to run during off-peak hours. For example, in Windows 2003, the AT command can be used to schedule tasks to off-peak hours, especially useful when doing system backups.

## 21.3 Analyzing memory bottlenecks

As mentioned in Chapter 13, “Microsoft Windows Server 2003” on page 351, Windows Server has the ability to self-optimize, but only for certain aspects of the operating system. This optimization process focuses on memory caching and virtual memory management. The Windows memory manager adjusts the amount of caching memory that is available to best suit operating conditions. However, memory is one of the most common sources for server bottlenecks, so do not overlook this section.

When configured for file serving mode, Windows Server favors a large disk cache where most of the available memory is assigned to the disk cache, and only some memory is available for loading programs. In application server mode, the operating system reduces the amount of memory available to the disk cache and

maximizes the memory available to run applications. See 13.6, “File system cache” on page 365 for details.

The amount of memory that the operating system and applications have available will significantly influence your server's performance. You can find more information about the memory subsystem in Chapter 10, “Memory subsystem” on page 183.

### 21.3.1 Paged and non-paged RAM

Physical memory in Windows Server is divided into *paged* and *non-paged* areas, as shown in Figure 21-2. Non-paged memory is critical memory used for drivers, the kernel, or application contents that must remain in RAM. This memory is never paged to the page file device because the non-paged memory is needed for general operation of the system. If it were paged out to disk, the operating system might not have all the data it needs to access the page file device, which could create a deadlock and the system might fail. To avoid this potential for a crash, the operating system marks critical memory as non-paged, thereby pinning these memory locations into physical memory.

Paged memory is non-critical memory buffers that can be written out to the page file on disk because the contents are not required for general execution of the operating system.

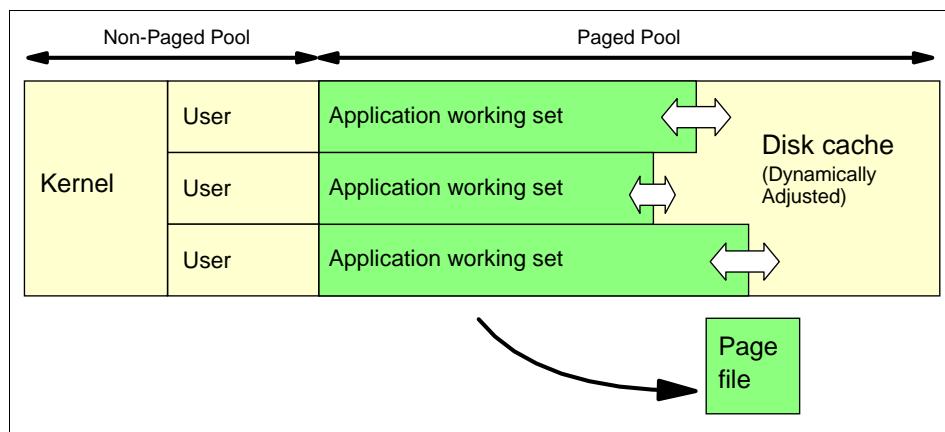


Figure 21-2 Windows 2000 memory definition

Programs can have a portion of their memory space set as non-pageable, but the majority of program functions are pageable. At program load, Windows Server loads all code that is needed for execution. However, usually much of the code and data is not in constant use as the program executes. As a result, the operating system marks the unused memory storing *data* as available to send

into the page file and marks the rarely used memory storing *code* as able to be deleted. Code is never paged, because it can simply be read back from disk; only data is written to the page file.

However, these unused objects in memory are not automatically paged out to the page file on disk. The actual paging to disk occurs only when other programs (or the caches) require additional memory and no free memory is available. When this occurs, the operating system frees memory used to store code that has been marked as unused, and writes the data that has been marked as unused to the pagefile. This frees up memory which in turn is allocated to the application making the request for more memory.

When this memory paging occurs, we call it *memory pressure*. Memory pressure occurs when applications are constantly requesting memory buffers that are not freely available in the system, and the operating system is forced to perform housekeeping to delete the least recently used code memory, and swap the least recently used data to the swap device. Clearly, when memory pressure occurs, system performance suffers. So if we can detect memory pressure (physical swapping), we can detect the slowdown due to insufficient memory.

### 21.3.2 Virtual memory system

Because almost all of the System Monitor performance counters for memory relate to virtual memory, they cannot be used to directly diagnose physical memory pressure. However, there are several counters that you can use to indicate that the lack of sufficient memory is causing a performance problem. These counters are:

- ▶ Memory: Available MBytes

This counter is the amount of physical memory that is available to processes that are running on the server (in MB).

- ▶ Memory: Page Reads/Sec
- ▶ Memory: Page Writes/Sec

These two counters typically indicate hard paging to disk is occurring. Nearly every server has pages per second counts that occur during normal operation, because page misses occur and the resulting miss can be serviced from a memory page that has not yet been paged out to disk.

Remember, unused pages are marked for swapping out to the page file, but will remain in system memory as long as available physical memory is not constrained. Thus, it would be incorrect to use Pages/Sec as an indicator to determine if physical memory capacity is insufficient. Only when paged data is actually written or read from the page file on disk is the server experiencing a performance bottleneck from lack of memory.

You can use the Performance console memory object counters listed in Table 21-2 to help you determine memory bottlenecks.

Table 21-2 Performance console counters for detecting memory bottlenecks

Counter	Description
Memory: Page Reads/sec	This counter is the number of disk read operations performed for physical paging. Generally, if the combined value of Page Writes/sec and Page Reads/sec exceeds ~150, it indicates a great deal of paging activity, and memory capacity might be the bottleneck in your system.
Memory: Page Writes/sec	This counter is the number of disk write operations performed for physical paging. Generally, if the combined value of Page Writes/sec and Page Reads/sec exceeds ~150, it indicates a great deal of paging activity, and memory capacity might be the bottleneck in your system.
Memory: Available MBytes	This counter indicates the amount of remaining <i>physical</i> memory in MB available to applications. If the server is configured to be optimized for file serving applications, this counter can normally be low, because Disk Cache Manager uses extra memory for caching and then returns it when requests for memory occur. If this value stays at less than 20% to 25% of installed RAM, it is an indication that you may not have enough memory in the system. Note also that some applications like IIS, Exchange, and SQL will increase their working sets to consume the majority of available memory. In these cases, carefully monitor Paging activity to determine a memory bottleneck as well. If this counter is consistently dropping over time, this may indicate that the application has a memory leak.
Memory: Pool Nonpaged Bytes	This counter indicates the amount of RAM in the non-paged pool system memory area where space is acquired by operating system components as they accomplish their tasks. If this value has a steady increase without a corresponding increase in activity on the server, it might indicate that a process that is running has a memory leak, and it should be monitored closely.

### 21.3.3 Performance tuning options

Typically, you have two choices for solving memory bottlenecks. These options are, in order of importance:

- ▶ Increase the total memory capacity of the server.
- ▶ If paging is unavoidable, move the page file to a faster disk or array.

#### Adding memory

Many performance problems are typically solved by simply increasing memory capacity of the server. However, you need to consider carefully the memory object counters, the system hardware, and application configuration, to ensure maximum performance gains.

There is always a point at which adding memory capacity will not help increase throughput. In general, for most servers running a single, older application, this limit is at about 3 GB to 4 GB of memory, or whenever the application has no need for additional memory or the bottleneck moves to other parts of the system.

Figure 21-3 shows the effect of adding memory to a file server.

**Note:** When adding DIMMs, ensure the server has DIMMs populated in all the slots needed to utilize the maximum interleave and concurrency the memory subsystem offers. See 10.2.10, “Memory interleaving” on page 199 for additional details.

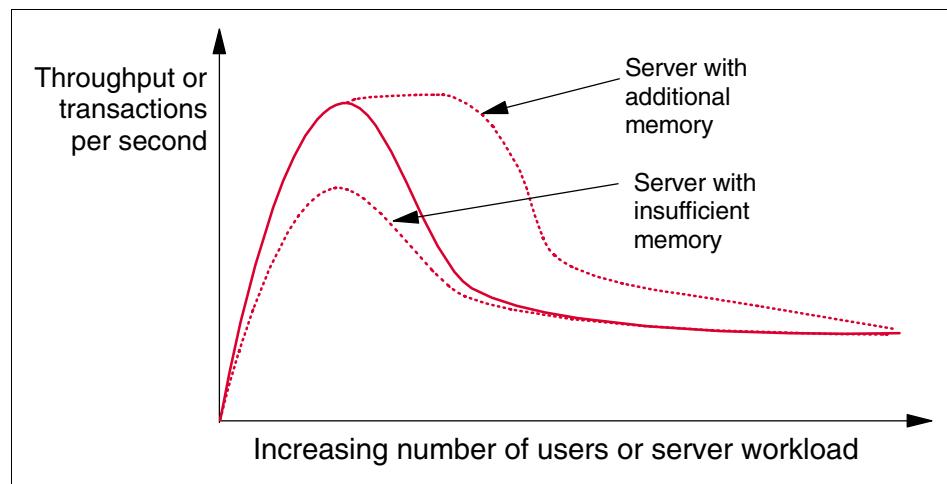


Figure 21-3 Effect of memory changes in a file server

Adding memory affects the performance of the server in the following ways:

- ▶ Adding memory to a server improves the file cache hit rate or reduces system paging, which increases the sustained server throughput rate (assuming that the caching algorithms that are used are effective for the particular application).

This reduces the disk I/O rate and increases network utilization because the server is now able to respond to requests at a faster rate. As a result, a slow network adapter can end up becoming the next bottleneck after memory is added to address the memory bottleneck.

- ▶ Higher disk cache hit rates or lower paging also translate into higher CPU utilization, again because the processors are no longer waiting as much for disk I/O to complete. So, poor CPU headroom can also reduce the potential performance gains from adding memory.

These points emphasize the importance of configuring a server properly for balanced performance.

## Unavoidable paging

The optimal situation for Windows servers is to have low, or no, sustained paging. Because Windows needs a page file, there will usually be some paging at initial startup of the server. Paging activity often occurs as an application warms up and in most cases, should be minimal afterwards.

In some cases, however, even with sufficient memory capacity, excessive paging cannot be eliminated. For example, the Windows operating system provides a facility to support data sharing between applications that uses the paging feature of Windows. This facility is called *memory mapped files*.

Memory mapped files enable applications to share large data items that cannot fit in physical memory by using the page file system as a virtual large shared memory buffer. In this case, one process can store data quickly into the page file and another process can access that data quickly without the overhead of using the file system to share data on disk.

For more information about memory mapped files, see the MSDN® Web site:

[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dngen1ib/html/msdn\\_manamemo.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dngen1ib/html/msdn_manamemo.asp)

However, when applications are using memory mapped files, significant paging occurs and no amount of additional memory reduces the memory mapped component of paging. Although there are only a few applications that use memory mapped files, if your application appears to have available memory or is in an environment where the application is expected to consume the available memory, and paging is still occurring, the application is likely using memory mapped files.

In such instances, our recommendation is to put the page file on the fastest disk array possible, such as a local RAID array.

## Recommendations

The recommendations for solving memory capacity issues are:

- ▶ Check Page Reads/sec and Page Writes/sec. If the sum of these two counters is greater than 150 pages/sec, add more RAM to the server up to the limit of what the operating system and application can support.
- ▶ If the server still pages significantly after the memory capacity is increased to the limits that the application supports, then assume the application is using

memory mapped files and proceed to move the page file to a fast RAID device.

- ▶ Check the Memory Available MB counter to see if the available memory is low. This is an indicator that the server is running low on physical memory. Again, add memory up to the limit for the application, then recheck paging.

**Note:** Refer to Chapter 10, “Memory subsystem” on page 183 for additional detail about the memory subsystem and its effect on performance.

## 21.4 Disk bottlenecks

Because servers ultimately retrieve all programs and data from the disk subsystem, this can be the most important aspect of server performance. Disk problems can stem from other factors, such as lack of memory, so always check memory capacity before modifying the disk configuration.

General disk subsystem operation is discussed in Chapter 11, “Disk subsystem” on page 237. Review that chapter before analyzing disk bottlenecks.

Performance console disk counters are available with either the LogicalDisk or PhysicalDisk objects:

- ▶ LogicalDisk monitors the operating system partitions stored on physical drives or physical arrays. After identifying a physical disk bottleneck, if multiple logical partitions are on the physical array, this object is useful for determining which partition is causing the disk activity, thereby possibly indicating the application or service that is generating the requests.
- ▶ PhysicalDisk counters reflect the actual activity to the physical hard disk drives, or arrays of hard disk drives. It is useful for monitoring all disk activity to the drives. Our initial analysis is always performed using physical disk counters because we first want to identify if the system has a hardware-level performance problem. If after determining the physical disk configuration is not optimal, review the logical disk counters to determine if disk I/O to one or more logical drives on the busy physical drive can be moved to another physical disk or array with less I/O traffic.

**Tip:** For initial analysis of disk performance bottlenecks, always use physical disk counters.

## 21.4.1 Analyzing disk bottlenecks

You can use the physical disk object counters listed in Table 21-3 to help you determine if you have disk bottlenecks. Then examine the indications of disk bottlenecks based on the object counter readings. Afterwards, you should perform appropriate actions to respond to the situation.

Table 21-3 Performance counters for detecting disk bottlenecks

Counter	Description
Physical Disk: Avg. Disk sec/Transfer	This is a key counter that indicates the health of the disk subsystem. This counter is the average time to complete each disk I/O operation. For optimal performance, this should be less than 25 ms for most environments. In general, this counter can grow to be very high when insufficient numbers of disks, slow disks, poor physical disk layout, or severe disk fragmentation occurs.
Physical Disk: Avg. Disk Queue Length	This counter is the average number of both read and write requests queued to the selected disk during the sample interval. If this value is consistently more than 2 to 3 times the number of disks in the array (for example, 8 to 12 for a 4-disk array), it indicates that the application is waiting too long for disk I/O operations to complete. To confirm this assumption, always check the Avg. Disk Sec/Transfer counter. Also, the Avg. Disk Queue Length counter is a key counter for determining whether a disk bottleneck can be alleviated by adding disks to the array. Remember, adding disks to an array only results in increased throughput when the application can issue enough requests to the array to keep all disks in the array busy. For optimal disk performance, you want the Avg. Disk Queue Length to be no more than 2 or 3 times the number of physical disks in the array. Also, in most cases the application has no knowledge of how many disks are in an array because this information is hidden from the application by the disk array controller. So unless an application configuration parameter is available to adjust the number of outstanding I/O commands, an application will simply issue as many disk I/Os as it needs to accomplish its work, up to the limit supported by the application or disk device driver. Before adding disks to an array to improve performance, always check the Avg. Disk Queue Length counter and only add enough disks to satisfy the 2 to 3 disk I/Os per physical disk rule. For example, if the array shows an Avg. Disk Queue Length of 30, then an array of 10 to 15 disks should be used.
Physical Disk: Avg. Disk Bytes/Transfer	This is the average number of bytes transferred to or from the disk during write or read operations. This counter can be used as an indicator of the stripe size that should be used for optimal performance. For example, always create disk arrays with a stripe size that is at least as large as the average disk bytes per transfer counter value as measured over an extended period of time.

**Note:** Never use the %Disk Time physical disk counter to diagnose server bottlenecks. This counter is the percentage of elapsed time that the selected disk drive is busy servicing read or write requests.

However, this counter is only useful with IDE drives, which, unlike SCSI disks, can only perform one I/O operation at a time. The %Disk Time counter is derived by assuming the disk is 100% busy when it is processing an I/O, and 0% busy when it is not. The counter is a running average of the 100% versus 0% count (binary).

SCSI array controllers can perform many hundreds or even thousands of I/Os per second before they encounter bottlenecks. Most array controllers can perform two to three disk I/Os per drive before a bottleneck occurs. For example, if an array controller with 60 drives has one disk I/O to perform at all times, it will be 100% utilized according to the % Disk Time counter. However, that array could actually be issued 120-180 I/Os before a true bottleneck occurs.

#### 21.4.2 Performance tuning options

After verifying that the disk subsystem is a bottleneck, a number of solutions are possible. These solutions include:

- ▶ Verify stripe size is at least as great as the sustained Avg. Bytes/transfer counter value for each array. If not, the array could be doing multiple physical disk I/Os to satisfy each request.
- ▶ Offload files that are experiencing heavy I/O processing to another server or to another array on the same server.
- ▶ Add more RAM.
- ▶ Use faster speed disks.
- ▶ Add more disk drives to an array in a RAID environment. This spreads the data across multiple physical disks and yields increased I/O rates.

Figure 21-4 shows the effect of putting a faster disk subsystem on a file server.

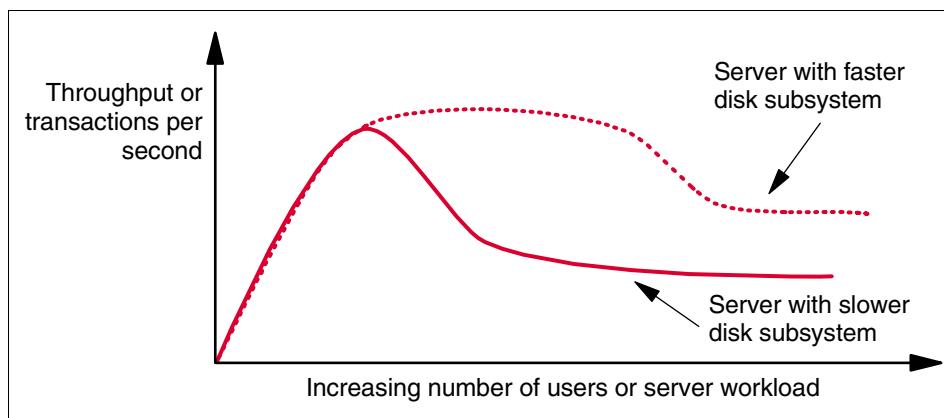


Figure 21-4 Effect of adding a faster disk subsystem to the file server

A faster disk subsystem usually improves the high load sustained transaction rate for the server, shown as the right portion of the curves in Figure 21-4. The peak of the curve is the peak sustainable throughput of the network adapter, and the lower part of the curve to the right represents the sustainable throughput of the disk subsystem.

The disk subsystem might only slightly affect performance under light loads because most requests are serviced directly from the disk cache. In this case, network transfer time is a relatively large component, and disk transfer times are hidden by a high frequency of disk cache accesses.

As the server disk performance improves, increased network adapter and CPU performance is required to support greater disk I/O transaction rates.

- ▶ Adding memory will increase file system cache, which in effect reduces disk I/O traffic, thereby improving server throughput and response times. Adding memory should be the first course of action before reconfiguring the disk subsystem. Again, before you act on a disk bottleneck, review 21.3, “Analyzing memory bottlenecks” on page 697 to make certain that memory capacity is optimal.
- ▶ When disk bottlenecks are detected, one option is to replace the slow disk with a faster one. However, consider that faster disks usually result in system level improvements on the order of 20% to 40%, not 2 or 3 times. So if the performance problem is a mild one, using faster disks can be considered. But in general, where performance improvements must be significant, adding disks to the array is usually the best choice.

- ▶ Adding disks is the safe way to improve performance and usually the most cost effective because you do not have to replace current server hardware. However, always determine the concurrent I/O demand of your server application before adding disks.

Checking the Avg. Disk Queue length will help you understand how many disks to add to the array. This can be calculated simply by dividing the Avg. Disk Queue length counter by 2 for very best performance, or by 3 for best price-performance. For example, if the sustained Avg. Disk Queue length is 12, then configure the array with 6 disks (for optimal performance) and 4 disks for best price-performance.

## 21.5 Network bottlenecks

Network performance troubleshooting can be a very complex task. This complication is in part because the performance counters obtained by System Monitor only represent traffic flow to and from the server for which the counters were monitored. The performance counters do not reflect the total traffic in the network. Thus, although the counters for a particular server might seem reasonable, the network itself could be experiencing heavy load and be causing slow server response times as seen by the users.

In general, when poor network performance is suspected, you must rely on help from an experienced networking professional. The complete debug of network performance problems is complicated and beyond the scope of this book. However, there are a few performance counters that can be used to perform a basic diagnosis of server network adapter bottlenecks and help you know when to call in the experts.

## 21.5.1 Finding network bottlenecks

The network performance object counters that should be investigated are listed in Table 21-4. Examine the indications of network bottlenecks based on the object counter readings. Also included here are suggestions that could help alleviate the situation.

Table 21-4 Performance console counters for detecting network bottlenecks

Counter	Description
Network Interface: Bytes Total/sec	Sustained values over 50% to 60% of the network adapter's available bandwidth are cause for concern. Expected maximum sustained application throughput for a Gigabit Ethernet in a modern server using a typical 70% - 30% Read/Write Ratio is about 160 MBps. To be conservative, detailed network analysis is warranted if the Bytes Total/sec value is over about 90 MBps.
Network Interface: Bytes Received/sec	This counter is a network subsystem primary counter. Sustained values over 50% to 60% of the maximum throughput in the receive direction should be investigated by a network administrator to determine if the network is a bottleneck. Most Gigabit Ethernet adapters can sustain about 110 MBps in the receive direction. To be conservative, detailed network analysis is warranted if the Bytes Received/sec value is over about 60 MBps.
Network Interface: Bytes Sent/sec	Sustained values over 50% to 60% of maximum throughput in the send direction should be investigated by a network administrator to determine if the network is a bottleneck. Most Gigabit Ethernet adapters can sustain about 110 MBps for data sends. To be conservative, detailed network analysis is warranted if the Bytes Sent/sec value is over about 60MBps.
Network Interface: Packets/sec <i>and</i> Network Interface: Packets Sent/sec <i>and</i> Network Interface: Packets Received/sec	Packets/sec rates should be no higher than about 100,000 Packets/sec total, and no more than ~70,000 for Packets Sent/sec and Packets Received/sec. If these values are exceeded, the server's network, or the supporting network attached to the server, may need further investigation.

## 21.5.2 Analyzing network counters

A network adapter can have two primary types of performance bottlenecks:

- ▶ Data rate bottleneck (saturating the network interface)
- ▶ Packets per second bottleneck (saturating the adapter's processor)

### Data rate bottlenecks

Data rate bottlenecks occur when the network adapter is running at the maximum sustainable data rate of the network technology. Data rate bottlenecks are the first bottleneck most people think of when diagnosing server network performance bottlenecks. However, they are actually the least likely to occur in most production networks.

In general a packet per second rate bottleneck, or the number of packets per second that the network adapter can process, is the more likely network bottleneck to occur in servers. This bottleneck is caused when server applications communicate by rapidly sending small messages, about 64 to 512 bytes in size. These smaller data packets do not saturate the data bandwidth of the network, but they can often saturate the ability of the LAN adapter or operating system network stack to process these packets. Therefore, to diagnose server LAN adapter bottlenecks, you need a method to identify both bandwidth bottlenecks and packet rate bottlenecks.

Diagnosing bandwidth bottlenecks from a server is not necessarily a straightforward task. In general, you want to determine if the LAN adapter is moving data at a rate which approaches the maximum sustainable rate of the network adapter being used. Keep in mind, however, that the server being analyzed is most likely not the only server in the network. Therefore, when examining the performance counters, keep in mind that these counters can only reflect the data rate to and from the monitored server, and not the total traffic of the network.

To use System Monitor performance counters as an indicator for network bottlenecks, a conservative approach is recommended. When the server is sustaining more than about 50% of the available network bandwidth, examine the entire network as a potential bottleneck.

**Tip:** If your sustained network bandwidth is 50% or greater than the potential bandwidth of your server, engage a network expert to help you diagnose the bottleneck.

However, if you know that the server is the only server in the network, and there is little peer-to-peer traffic, then you can use a less conservative value of up to 75% to 80% of sustainable throughput to indicate a network bottleneck.

The network throughput seen by Performance Monitor represents actual data throughput. The line-level network traces and network switch performance monitoring show total network throughputs, including all protocol overheads.

Because Ethernet is full duplex, a 1 Gbps Ethernet controller can have a theoretical throughput of 2000 Mbps (1000 Mbps in each direction). However, due to protocol and network stack processing overheads, actual sustainable data throughput is lower than the theoretical maximums. In practice, you can expect the best case sustained data rate in each direction to be about 800 Mbps, or about 100 MBps.

**Note:** Although network technologies are typically referred to in some number of *bits* per second, Performance Monitor measures throughput in *Bytes* per second. Therefore, 1000 Mbps = 125 MBps.

It is also important to note that because Ethernet is full duplex, it is possible to have a transmit bottleneck, a receive bottleneck, or both. The methodology you use must check all of these to determine the presence of any network adapter bottlenecks.

Note that all these limits are based on the assumption that the system is capable of driving the TCP/IP stack to the maximum bandwidth of the adapter. Although nearly all modern servers have no issue in driving 1 Gigabit load levels, we can still see limitations with multiple adapters or high bandwidth 10 Gigabit cards. Refer to 12.2, “Factors affecting network controller performance” on page 300 for an analysis of the effects of these server components on network throughput.

In general, it is recommended that you investigate the network in more detail whenever the sustained bandwidth is greater than about half the total sustainable bandwidth in any direction of the network. For 1 Gbps Ethernet, then, you might start further investigation after spotting about 50 MBps of sustained bandwidth, in either transmit or receive directions.

### Packet rate bottlenecks

Although network bandwidth analysis has hard upper limit (line speed) boundaries, packet rate thresholds are much more difficult to define. Because many applications communicate using small packets of data, it is entirely possible for the server to have a packet per second rate bottleneck at throughputs levels that are only a very small fraction of the maximum sustainable network bandwidth.

Networks transmit data in data chunks called *frames*. The frame size determines the largest piece of data that can traverse the network. It is important to note that the operating system’s network stack will segment network traffic that exceeds

the network's frame size, so very large data chunks sent by an application will be broken up into lots of smaller pieces. For typical networks, the frame size is ~1500 Bytes, and because of this, we will not typically see packet sizes reported in Performance Monitor above this size. However, systems and network adapters that implement network offload technologies may occasionally show packet sizes larger than the network's frame size.

It is nearly impossible to offer specific advice for packet rate bottleneck detection without detailed knowledge of the server configuration, workload, and particular network adapter. Further, the network stack processing overhead varies greatly based on the size of the network I/O operations that the application performs. Some applications transmit numerous small messages, and others transmit data in huge chunks. Unfortunately, although there can be a vast performance difference between these, Performance Monitor does not provide efficient mechanisms for determining this information. Nevertheless, there are still indicators that can help you detect whether a network issue is occurring.

The first step is to determine the average Read Packet Size, Write Packet Size, and Total Packet Size seen by the network stack. As mentioned in Table 21-4 on page 708, this is accomplished by dividing the appropriate Bytes/sec counter with its correlating Packets/sec counter. This result will provide a Bytes per Packet result for Reads, Writes, and Total.

Using this information, you can then apply the following very general rules of thumb:

- ▶ If the Read or Write Bytes/Packet < ~1000 Bytes, the application has a high percentage of small data elements.

In this case, monitor the Packets Sent/sec and Packets Received/sec counters to determine if either of these exceed ~70,000 packets/sec. Also verify that the total Packets/sec counter is not exceeding ~100,000 packets/sec. If any of these thresholds are exceeded, the network adapter or server could be saturating on small block network processing.

- ▶ If the Read or Write Bytes/Packet > ~1000 Bytes, this tends to indicate that the server has a higher percentage of large block data movement.

In this case you need to apply the preceding rules. You must also pay special attention to the Bytes Sent, Bytes Received, and Bytes Total counters as described in Table 21-4 on page 708 to rule out the possibility of data throughput limitations.

## Network scaling

For most modern systems, the throughput rules can generally be scaled to four 1 Gigabit Ethernet adapters, and packet rate rules can generally be scaled to two 1 Gigabit adapters. The reason for the difference in number is due to the

overheads. Small block network processing, which is the primary limiting factor when discussing Packet Rate limitations, has a much higher overhead than the large block processing that is commonly found in throughput-limited networks.

### 21.5.3 Solving network bottlenecks

After verifying that the network subsystem is the bottleneck, a number of solutions are possible. Here are some actions that you can take to respond to network bottlenecks.

However, always consult a network expert before making modifications to your system, because any modifications must be done only after considering the total network traffic. In many cases, this total network traffic will include many aspects not seen by the System Monitor counters.

#### Hardware approach

Consider the following items:

- ▶ Use current generation PCIe-based LAN adapters if available. These adapters are faster and more efficient than first-generation 1 Gigabit adapters.
- ▶ Consider consolidating to higher bandwidth networks, like 10 Gigabit. Note, however, that there is still great variance in functionality and efficiency between adapter vendors. Figure 21-5 shows the effect of adding a faster network adapter to the file server.

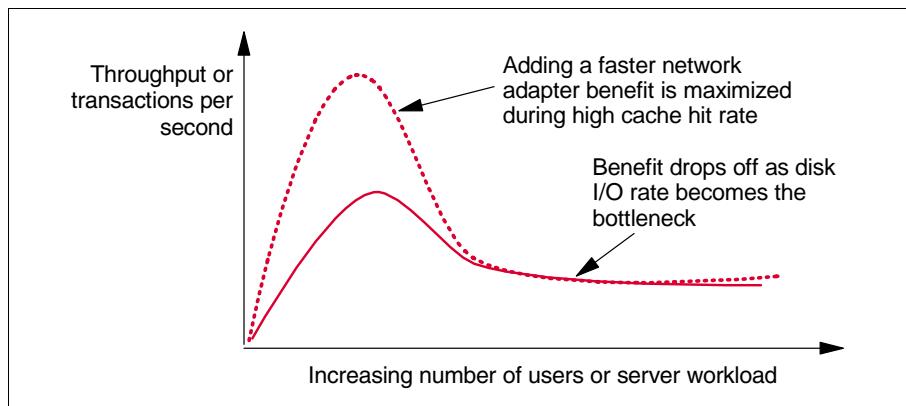


Figure 21-5 Effect of adding faster network adapter to the file server

For this example, the network adapter's speed impacts the overall performance of the file server in the following ways:

- It improves the maximum peak transaction rate of the server.

- It might only slightly affect performance under heavy loads because the server is disk bound as it waits for seeks. In this case, the total network transfer time is a relatively small component of the overall transaction.
- As network adapter performance improves, increased CPU performance is required to service the increased request rates from the users on the LAN.
- ▶ Balance the network load across multiple network adapters. The most efficient mechanism for doing this is through simple network subnetting, because it has no additional processing overheads.

A alternate approach is to use one of the many supported port trunking techniques supported by your LAN adapter and switches. Although these mechanisms can provide added benefits such as failover in case an adapter, cable, or switch fails, these mechanisms add some overhead to the network stack, and can significantly reduce performance.

**Note:** Use port trunking techniques carefully, because these failover mechanisms all add some amount of overhead to the network stack, increasing processor utilization and potentially reducing throughput.

Although not feasible for all environments, simple network subnetting will allow distribution of load across network ports without additional stack processing overheads.

- ▶ Create multiple networks or subnetworks. This helps in handling unnecessary broadcasts over the network.
- ▶ Upgrade to better performing routers and bridges. These smart networking devices themselves can add significant latencies to the network, and can have the affect of limiting the networking capabilities of your server.
- ▶ Add more servers to the network. In this way, you distribute the processing load to other servers.

## Software tuning options

For best network performance, consider implementing the following items:

- ▶ Use recent LAN adapter device drivers. LAN adapter manufacturers usually develop their latest drivers to address bugs and inefficiencies in previous releases. This is especially critical for fast-evolving technologies like 10 Gigabit, but it still applies to 1 Gigabit technologies, as well.
- ▶ The tool INTFILTR, or INTPOLICY in Windows 2008, are interrupt affinity tools that allow you to bind device interrupts to specific processors on SMP servers. This is a useful technique for maximizing performance, scaling, and

partitioning of large servers. It can provide up to a 20% network performance increase. For more information, go to one of the following:

Windows 2000 Server:

<http://support.microsoft.com/?kbid=252867>

Windows Server 2003:

<http://www.microsoft.com/downloads/details.aspx?familyid=9d467a69-57ff-4ae7-96ee-b18c4790cff&displaylang=en>

Windows Server 2008:

<http://www.microsoft.com/whdc/system/sysperf/IntPolicy.mspx>

- ▶ For best performance, remove any unneeded network services and protocols to reduce network stack overheads. In extreme cases, extra protocols can have a significant effect on networking performance. To do this, follow these general steps:
  - a. Click **Control Panel** → **Network Connections**.
  - b. Select any connection, and then select **File** → **Properties** from the menu.
  - c. Choose a service to be removed, and then click **Uninstall**, or simply uncheck the selection box for that service, which will not uninstall the service from the machine completely, but will prevent its usage with this network interface.
  - d. Confirm **Yes** if prompted, and then **Close**. Figure 21-6 on page 715 shows you how to remove network services from the Windows 2000 server. The methods for Windows Server 2003 and 2008 are similar and can be extrapolated from this example.

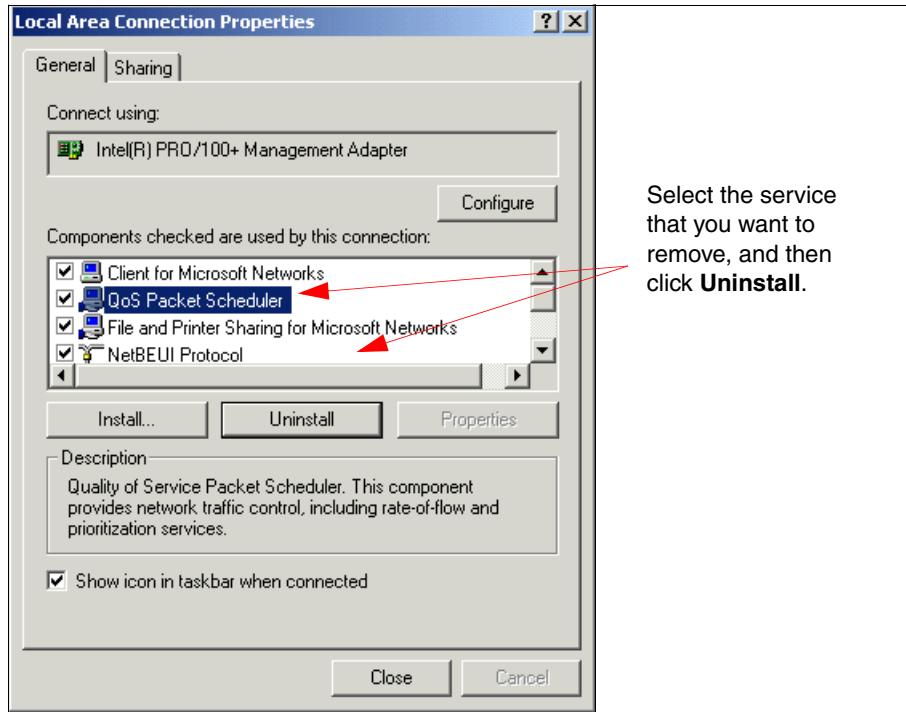


Figure 21-6 Removing network services from Windows 2000 Server

If additional network services are to be removed outside of those listed in the Connection Properties screen, you will need to use the Add/Remove Programs option of Windows.

- a. Click **Control Panel** → **Add/Remove Programs** → **Add/Remove Windows Components**.
- b. Select **Networking Services** or **Other Network File and Print Services**.
- c. Click **Details...** and then deselect services that you do not need.
- d. Click **OK**, then **Next**, and **Finish**. Figure 21-7 on page 716 shows you how to remove additional network services from the Windows 2000 server.

**Important:** In Windows 2008, both the IPV4 and IPV6 network stacks are loaded by default, although most environments will only use one or the other. It is highly recommended that the unneeded protocol stack (most commonly IPV6, due to the slow adoption of this new network standard) be deselected to reduce the potential for network issues that could arise from having both protocols running simultaneously.

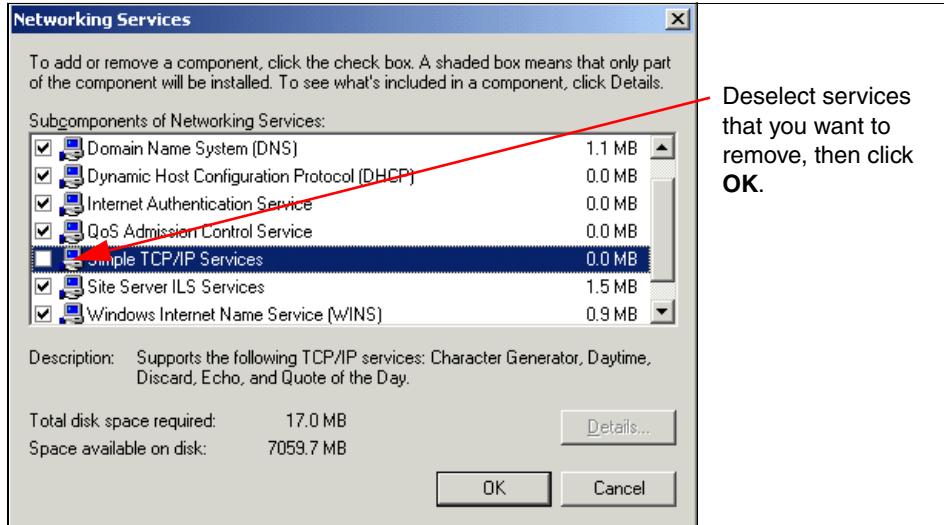


Figure 21-7 Removing unnecessary Windows 2000 networking services

#### 21.5.4 Monitoring network protocols

In addition to the performance objects and counters, it is also important to monitor how network protocols affect the network. Network protocol analysis enables you to monitor broadcasts and retransmissions of the server. By monitoring the right counters for the protocols you selected, you can have a better understanding of the use of the network bandwidth.

With the widespread adoption of Internet-related services, the TCP/IP protocol has become the standard communication mechanism for both public and private networks. TCP/IP supports open connectivity across hardware platforms and operating systems, and also supports routing for intranet and Internet applications. The TCP/IP protocol in Windows Server 2003 and 2008 includes a suite of tools that are common to most UNIX systems as well as TCP/IP supporting systems.

TCP/IP counters are added to a system when the TCP/IP protocol and the SNMP Service have been installed. The SNMP Service includes the objects and counters shown in Table 21-5 on page 717 for TCP/IP-related protocols.

Table 21-5 TCP/IP counters

Object	Description
TCP: Segments/sec	The number of TCP segments (frames) that are sent and received over the network. This value is usually high, indicating high throughput.
TCP: Segments Retransmitted/sec	The number of frames (segments) that are retransmitted in the network. This value should be low. If sustained high values are observed, upgrade your physical hardware or segment your network. Also, look at flow control policies of your NICs and Switches, because high Retransmissions are often due to mismatched flow control settings.
UDP: Datagrams/sec	The number of UDP datagrams (such as broadcasts) that are sent and received. This value should be low. If sustained high values are observed, reduce your network broadcast.
Network Interface: Output Queue Length	The length of the output packet queue (in packets). Generally, a queue longer than two indicates congestion, and analysis of the network structure to determine the cause is necessary. This value should be low. If sustained high values are observed, upgrade the LAN adapter, add an additional LAN adapter, or verify the physical network components for failures.

**Tip:** A high rate of Segments Retransmitted/sec often indicates that the flow control policies of a LAN adapter is mismatched to the switch port it is attached to, potentially affecting the server's networking performance.





# Analyzing bottlenecks for servers running Linux

This chapter is useful if you are facing a reactive situation where a performance problem is already affecting a server. It presents a series of steps which lead to a concrete solution that you can implement to restore the server to an acceptable performance level.

This chapter discusses the following topics:

- ▶ 22.1, “Identifying bottlenecks” on page 720
- ▶ 22.2, “CPU bottlenecks” on page 724
- ▶ 22.3, “Memory subsystem bottlenecks” on page 728
- ▶ 22.4, “Disk bottlenecks” on page 733
- ▶ 22.5, “Network bottlenecks” on page 739

## 22.1 Identifying bottlenecks

Linux manages system resources differently from other popular operating systems. Because of this, it is important to know how to understand the different performance metrics. Because the Linux distributors may have already tuned the systems for generic performance parameters, identifying bottlenecks is important, especially when working on enterprise servers and on scale-out systems. We used the following steps as our quick tuning strategy:

1. Know your system.
2. Back up the system.
3. Monitor and analyze the system performance.
4. Narrow down the bottleneck and find its cause.
5. Fix the cause of the bottleneck by trying a single change at a time.
6. Repeat from step 3 until you are satisfied with the performance of the system.

**Tip:** Document each step, especially the changes that you make and their effect on performance. Never change more than one factor at a time during your testing.

In many of the cases described in this chapter, there is a set of tasks that can be performed to tune or resolve a performance issue. Some of these require advanced Linux skills and an understanding of how the Linux kernel works, so check with your system administrator (or double-check, if you are the system administrator) before executing changes to avoid creating additional bottlenecks when there were none previously.

### 22.1.1 Gathering information

Most likely, the only first-hand information you have access to includes statements such as “There is a problem with the server.” In this situation, it is crucial that you ask probing questions to clarify and document the problem. Here is a list of questions to ask to obtain a better picture of the system:

- ▶ Could you give a complete description of the server in question?
  - Model
  - Age
  - Configuration
  - Peripheral equipment
  - Operating system version and update level

- ▶ Can you describe the problem *exactly*?
  - What are the symptoms?
  - Describe any error messages.

Some people can have problems answering this question. Any extra information you can learn might allow you to diagnose the problem. For example, you might hear “It is really slow when I copy large files to the server.” Slow performance might indicate a network problem or a disk subsystem problem.

- ▶ Who is experiencing the problem?

Is one person, one particular group of people, or the entire organization experiencing the problem? This type of question helps you to determine whether the problem exists in one particular part of the network, or whether it is application-dependent, and so on. If only one user is experiencing the problem, then the issue might be with the user’s personal computer (or it may be one of perception).

The perception that clients have of the server is usually a key factor. From this point of view, performance problems might not be related directly to the server. The network path between the server and the clients can easily be the cause of the problem. This path includes network devices as well as services that are provided by other servers, such as domain controllers.

- ▶ Can the problem be reproduced?

All reproducible problems can be solved. If you have sufficient knowledge of the system, you should be able to narrow the problem to its root and decide which actions to take.

The fact that the problem can be reproduced allows you to see and to understand it better. Document the sequence of actions that are necessary to reproduce the problem at any time:

- What are the steps to reproduce the problem?

Knowing the steps might let you reproduce the same problem on a different machine under the same conditions. If this works, it gives you the opportunity to use a machine in a test environment and removes the chance of crashing the production server.

- Is it an intermittent problem?

If the problem is intermittent, the first thing to do is to gather information and find a path to move the problem in the reproducible category. The goal here is to have a scenario to make the problem occur “on command.”

- Does it occur at certain times of the day or certain days of the week?

This information might help you to determine what is causing the problem. The problem might occur when everyone arrives for work or returns from

lunch. Look for ways to change the timing (that is, make it happen less often or more often). If there are ways to do so, the problem becomes a reproducible one.

- Is it unusual?

If the problem falls into the non-reproducible category, you can conclude that it is the result of extraordinary conditions and classify it as fixed. In real life, there is a high probability that it will happen again.

A useful way to troubleshoot a hard-to-reproduce problem is to perform general maintenance on the server: reboot, or apply current drivers and patches to the machine.

- ▶ When did the problem start? Was it gradual or did it occur very quickly?  
If the performance issue appeared gradually, then it is likely to be a sizing issue. If it appeared overnight, then the problem could be caused by a change made to the server or peripherals.
- ▶ Have any changes been made to the server (minor or major), or are there any changes in the way clients are using the server?  
Did the customer alter something on the server or peripherals to cause the problem? Is there a log of all network changes available?

Demands could change based on business changes, which could affect demands on a servers and network systems, for example:

- ▶ Are there any other servers or hardware components involved?
- ▶ Are there any logs available?
- ▶ What is the priority of the problem? When does it need to be fixed?
  - Does it need to be fixed in the next few minutes, or in days? You might have some time to fix it, or it might already be time to operate in panic mode.
  - How massive is the problem?
  - What is the related cost of that problem?

### 22.1.2 Analyzing the server's performance

**Important:** Before initiating any troubleshooting actions, back up all data and the configuration information to prevent a partial or complete loss.

At this point, begin monitoring the server. The simplest way to monitor the server is to run monitoring tools from the server that you are analyzing.

Create a performance log of the server during its peak time of operation (for example, 9:00 a.m. to 5:00 p.m.). The peak performance time for your server will depend upon what services are provided and who is using these services. When creating the log, if available, include the following objects:

- ▶ Processor
- ▶ System
- ▶ Server work queues
- ▶ Memory
- ▶ Page file
- ▶ Physical disk
- ▶ Redirector
- ▶ Network interface

Before you begin, remember that a methodical approach to performance tuning is important. Our recommended process, which you can use for your System x server performance tuning process, is as follows:

1. Understand the factors that affect server performance as explained in the first chapters of this book. Especially try to understand the logical connection of the various subsystems and the distinction between a hardware and a software bottleneck.
2. Measure the current performance to create a performance baseline to compare with your future measurements and to identify system bottlenecks.
3. Use available monitoring tools to identify a performance bottleneck. By following the instructions in this chapter, you should be able to narrow down the bottleneck to the subsystem level.
4. Improve the component that is causing the bottleneck by performing actions to improve server performance in response to demands.

**Note:** It is important to understand that the greatest gains are obtained by upgrading a component that has a bottleneck when the other components in the server have ample power left to sustain an elevated level of performance.

5. Measure the new performance so that you can compare the performance before and after the tuning steps.

When attempting to fix a performance problem, remember the following:

- ▶ Take measurements before you upgrade or modify anything so that you can tell whether the change had any effect (that is, take baseline measurements).
- ▶ Examine the options that involve reconfiguring existing hardware, not just those that involve adding new hardware.

## 22.2 CPU bottlenecks

For servers with the primary role of application or database server, the CPU is a critical resource and can often be a source of performance bottlenecks. It is important to note that high CPU utilization does not always mean that a CPU is busy doing work. It might, in fact, be waiting on another subsystem.

When you perform proper analysis, it is very important that you look at the system as a whole and examine all subsystems, because there can be a cascade effect within the subsystems.

**Note:** There is a common misconception that the CPU is the most important part of the server. Unfortunately, this is often not the case and, as such, servers are often overconfigured with CPU and underconfigured with disks, memory, and network subsystems. Only specific applications that are truly CPU-intensive can take advantage of today's high-end processors.

To identify CPU bottlenecks, it is important to understand the processor metrics. In this section we review the various parameters that may help in this task, independently of the method or tool used to obtain this information.

The following list describes the main processor metrics and explains the impact they may have, to help identify the bottleneck:

- ▶ CPU utilization

This refers to the overall utilization per processor. If the CPU utilization is over 80% for a sustained period of time, a CPU bottleneck is most likely occurring.
- ▶ Runnable processes

This refers to the processes that are ready to run. This value should not exceed ten times the amount of physical processors or cores for a sustained period of time.
- ▶ Blocked processes

This refers to the processes that cannot execute because they are waiting for an I/O operation to finish. Blocked processes may point to an I/O bottleneck.
- ▶ User time

This measures the percentage of time that the CPU spends on user processes, including the nice time. High values of this parameter are OK, because they mean that the CPU is actually processing user workload.

- ▶ System time
 

This measures the CPU percentage spent on kernel operations, including the IRQ handling and softirq handling. High and sustained time values may point to a bottleneck in the network or in the driver stack.
- ▶ Idle time
 

This refers to the percentage of time the CPU was idle waiting for tasks.
- ▶ Nice
 

This refers to the CPU time consumed by processes that have had their nice value changed.
- ▶ Context switch
 

This refers to the number of switches between threads that occur in the system. A high number of context switches in connection with a large amount of interrupts may point to driver or application issues.
- ▶ Waiting
 

▶ This value refers to the total amount of CPU time spent waiting for an I/O operation to occur.
- ▶ Interrupts
 

This value includes the CPU clock and both hard and soft interrupts.
- ▶ Steal
 

This value refers to the time spent in involuntary wait because no virtual CPU resources are available.

These values can be found in the /proc file system and also gathered with any of the CPU tools, such as **top**. Example 22-1 displays the output of the command line tool **mpstat**, which provides information about the status of CPU utilization.

*Example 22-1 Output of mpstat command*

---

```
root@blade1]# mpstat -P ALL 2 10
Linux 2.6.9-5.ELsmp (bc1srv7)   04/10/2008

01:55:14 PM  CPU  %user  %nice %system %iowait    %irq   %soft
%idle    intr/s
01:55:16 PM  all   0.00   0.00  20.80   11.28    5.01   17.79
45.11  27901.50
01:55:16 PM    0   0.00   0.00   14.00    6.00    9.50   29.00
41.50  26665.00
01:55:16 PM    1   0.00   0.00   27.50   16.50    1.00    6.50
49.00  1227.50
```

---

## 22.2.1 Finding bottlenecks with the CPU

Determining bottlenecks with the CPU can be accomplished in several ways. As discussed in Chapter 18, “Linux tools” on page 607, Linux has a variety of tools to help determine CPU bottlenecks. The question is, which tools should you use?

One such tool is **uptime**. By analyzing the output from the uptime tool, you can form an approximate idea of what has happened in the system for the last 15 minutes (Example 22-2). For a more detailed explanation of this tool, refer to 18.2, “The uptime command” on page 609.

*Example 22-2 The uptime tool output from a CPU-strapped system*

---

```
18:03:16 up 1 day, 2:46, 6 users, load average: 182.53, 92.02, 37.95
```

---

Using KDE System Guard and the CPU sensors allows you to view the current CPU workload.

**Tip:** Be careful not to add to CPU problems by running too many tools at one time. You might find that using several different monitoring tools at one time can contribute to the high CPU load.

Also keep in mind that X-based monitoring tools bring some overhead with them due to the GUI. Never attempt to measure a imminent memory bottleneck with the aid of GUI-based tools, because they increase memory demand even further.

Using **top**, you can see CPU utilization and also what processes are the biggest contributors to the problem, as shown in Example 18-3 on page 611. If you have set up **sar**, you are collecting a great deal of information, some of which is CPU utilization over a period of time. Analyzing this information can be difficult, so use **isag**, which can take sar output and plot a graph.

Otherwise, you might want to parse the information through a script and use a spreadsheet to plot it to see any trends in CPU utilization. You can also use **sar** from the command line by issuing **sar -u** or **sar -U processor-number**.

To gain a broader perspective of the system and current utilization of more than just the CPU subsystem, a useful tool is **vmstat**, which is described in greater detail in 18.6, “The vmstat command” on page 615.

## 22.2.2 Multi-processing machines

Issues with multi-processing machines can be difficult to detect. In an SMP environment, the concept of *CPU affinity* implies that you bind a process to a

CPU. CPU affinity is useful with CPU cache optimization, which is achieved by keeping the same process on one CPU rather than moving the process between processors. When a process moves between CPUs, the cache of the new CPU must be flushed. Thus, a process that moves between processors causes many cache flushes to occur. Therefore, an individual process takes longer to finish.

This scenario is very difficult to detect because it appears that the CPU load is very balanced and that it is not necessarily peaking on any CPU. Affinity is also useful in NUMA-based systems (such as servers based on the AMD Opteron and the System x 3850 and System x 3950), where it is important to keep memory, cache, and CPU access local to one another.

### 22.2.3 Performance tuning options for the CPU

When attempting to tune the CPU, first ensure that the system performance problem is caused by the CPU and not one of the other subsystems. If it is the processor that is the server bottleneck, then you can take a number of steps to improve performance, including:

- ▶ Ensure that no unnecessary programs are running in the background by using `ps -ef`. If you find unnecessary programs that are running, stop these programs and use `cron` to schedule them to run at off-peak hours.
- ▶ Identify non-critical, CPU-intensive processes by using `top`, and modify their priority using `renice`.
- ▶ In an SMP-based machine, try using `taskset` to bind processes to CPUs to make sure that processes are not hopping between processors and causing cache flushes.

**Important:** Using `taskset` requires in-depth knowledge of both the system and the application. If you do not feel qualified to use `taskset`, then avoid using it because you can impact performance by trying. For example, binding a multithreaded application to a single core will have a severe adverse effect.

- ▶ Based on the application that is running, decide whether it is better to scale up (bigger CPUs) than scale out (more CPUs). This decision is a function of whether your application is designed to take advantage of more processors effectively. For example, a single-threaded application scales better with a faster CPU and not with more CPUs.
- ▶ Ensure that you are using the latest drivers and firmware, as this can affect the load on the CPU.

## 22.3 Memory subsystem bottlenecks

On a Linux system, many programs run at the same time. These programs support multiple users, and some processes are more used than others. Some of these programs use a portion of memory while the rest are “sleeping.” When an application accesses cache, the performance increases because an in-memory access retrieves data, thereby eliminating the need to access slower disks.

The operating system uses an algorithm to control which programs use physical memory and which programs are paged out. This paging of memory is transparent to user programs. *Page space* is a file created by the operating system on a disk partition to store user programs that are not used currently. Typically, page sizes are 4 KB or 8 KB. In Linux, the page size is defined in the kernel header file `include/asm-<architecture>/param.h`, using the variable `EXEC_PAGESIZE`. The process that is used to page out a process to disk is called *pageout*.

It is also very important to understand the memory metrics that the system provides, to identify the potential memory bottlenecks. The following list describes the main metrics:

Free memory	This is the amount of available memory in the system.
Swap usage	If you have values above 200 to 300 pages per second for a sustained period of time, most likely you face a memory bottleneck.
Buffer & cache	This refers to the cache memory allocated as file system and block device cache.
Slabs	This value is related to the kernel usage of memory. Note that this memory cannot be allocated out to disk.
Active vs. Inactive memory	This provides information about the active memory usage of the system memory. Inactive memory is likely to be swapped to the disk

Figure 22-1 on page 729 illustrates how Linux memory is organized, and how to interpret the `free` command to understand the information it provides.

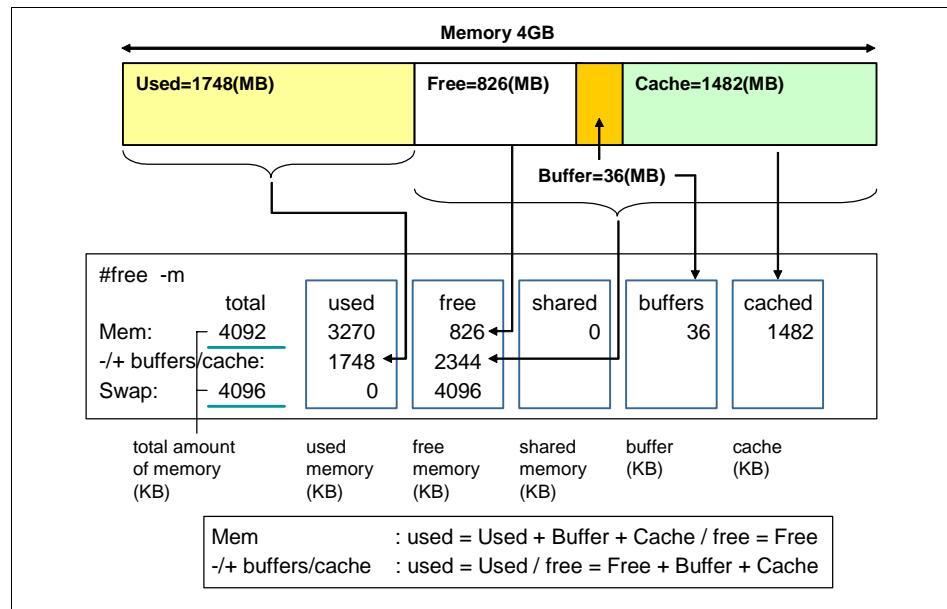


Figure 22-1 Linux memory

### 22.3.1 Finding bottlenecks in the memory subsystem

To find bottlenecks in the memory subsystem, start your analysis by listing the applications that are running on the server. Determine how much physical memory and swap each of the applications needs to run. Figure 22-2 on page 730 shows KDE System Guard monitoring memory usage.

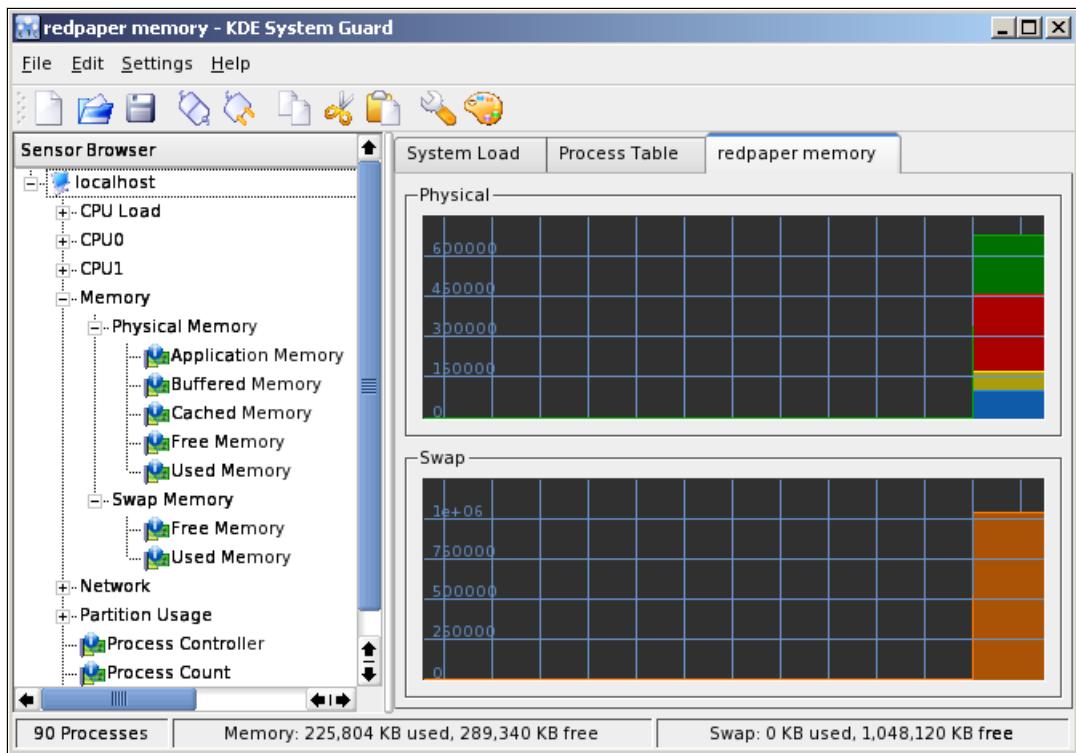


Figure 22-2 KDE System Guard memory monitoring

The indicators in Table 22-1 can also help you to define a problem with memory.

Table 22-1 Indicator for memory analysis

Memory indicator	Analysis
Memory available	This indicates how much physical memory is available for use. If, after you start your application, this value decreases significantly, you might have a memory leak. Check the application that is causing it and make the necessary adjustments. Use <code>free -l -t -o</code> to obtain additional information.
Page faults	There are two types of page faults: <ul style="list-style-type: none"> <li>- Soft page faults, when the page is found in memory</li> <li>- Hard page faults, when the page is not found in memory and must be fetched from disk</li> </ul> Accessing the disk slows down your application considerably. The <code>sar -B</code> command can provide useful information for analyzing page faults, specifically columns <code>ppgin/s</code> and <code>ppgout/s</code> .
File system cache	This is the common memory space used by the file system cache. Use the <code>free -l -t -o</code> command, for example.

Memory indicator	Analysis
Private memory for process	This represents the memory that is used by each process running on the server. You can see how much memory is allocated to specific processes by using the <code>pmap</code> command.

## Paging and swapping indicators

In Linux, as with all UNIX-based operating systems, there are differences between paging and swapping. *Paging* moves individual pages to swap space on the disk. *Swapping* is a bigger operation that moves the entire address space of a process to swap space in one operation.

Swapping can have one of two causes:

- ▶ A process enters sleep mode. Sleep mode normally occurs because the process depends on interactive action; editors, shells, and data entry applications spend most of their time waiting for user input. During this time, they are inactive.
- ▶ A process behaves poorly. Paging can be a serious performance problem when the amount of free memory pages falls below the minimum amount specified, because the paging mechanism is not able to handle the requests for physical memory pages and the swap mechanism is called to free more pages. This type of paging increases I/O to disk significantly and degrades a server's performance quickly.

If your server is always paging to disk (a high page-out rate), consider adding more memory. However, for systems with a low page-out rate, adding memory might not have any effect.

## Using NMON

NMON displays available physical memory, low space memory, high memory, and swap space in total, free, and free percentage. With NMON, you can view very quickly whether your system is swapping by looking at the free percentage value of the swap memory column. You can also see whether your server is running out of memory, or whether the applications or operating system is consuming too much memory.

To display the memory statistics in NMOM, press the `m` key. Figure 22-3 on page 732 illustrates the memory monitoring with NMON.

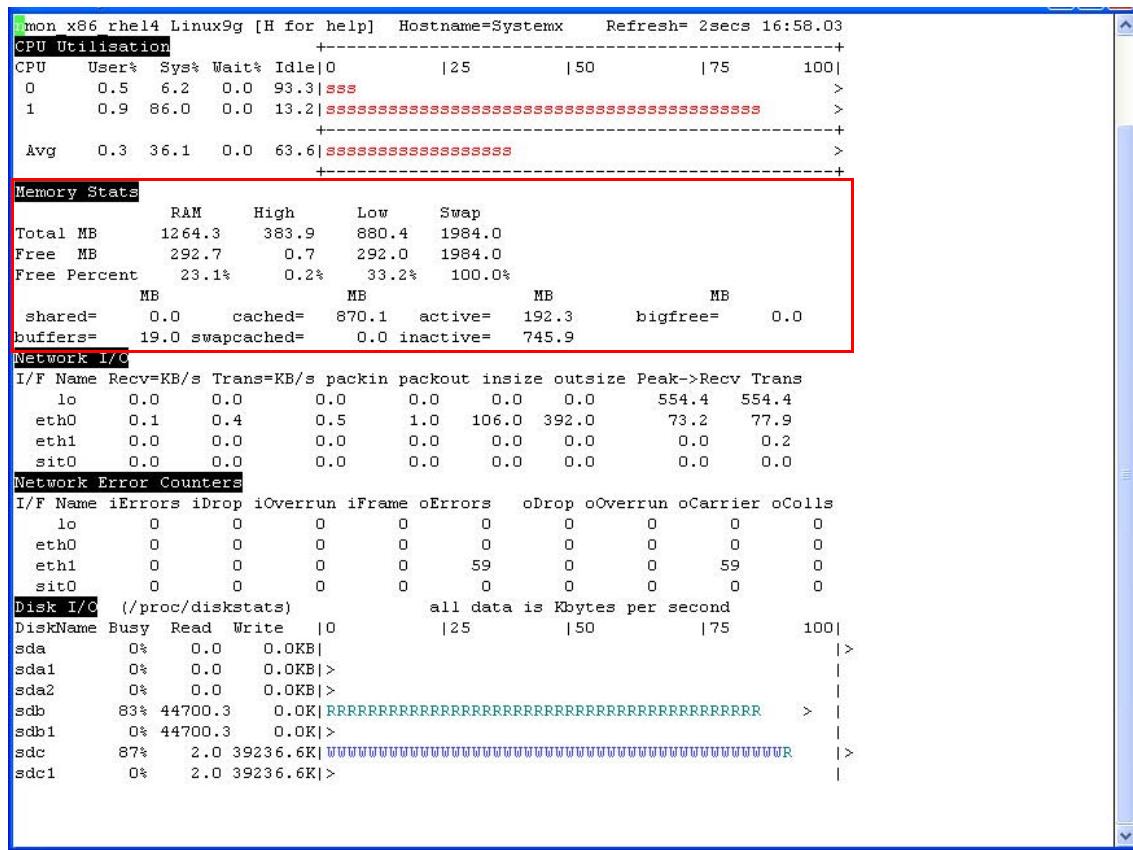


Figure 22-3 NMON monitoring memory

### 22.3.2 Performance tuning options for the memory subsystem

If you think that there is a memory bottleneck, consider these actions:

- ▶ Tune the swap space using bigpages, hugetlb, or shared memory.
- ▶ Increase or decrease the size of pages.
- ▶ Improve the handling of active and inactive memory.
- ▶ Adjust the page-out rate.
- ▶ Limit the resources that are used for each user on the server.
- ▶ Stop the services that you do not need as discussed in 15.2, “Working with daemons” on page 455.
- ▶ Add memory.

## 22.4 Disk bottlenecks

The disk subsystem is often the most important aspect of server performance, and it is usually the most common bottleneck. However, problems can be hidden by other factors, such as lack of memory. Applications are considered to be I/O-bound when CPU cycles are wasted simply waiting for I/O tasks to finish.

The most common disk bottleneck is having too few disks. Most disk configurations are based on capacity requirements, not performance. The least expensive solution is to purchase the smallest number of the largest-capacity disks possible. However, this places more user data on each disk, causing greater I/O rates to the physical disk and allowing disk bottlenecks to occur.

The second most common problem is having too many logical disks on the same array, which increases seek time and greatly lowers performance.

We discuss the disk subsystem in greater detail in 15.9, “Tuning the file system” on page 480.

As with the other components of the Linux system we discussed, disk metrics are important when identifying performance bottlenecks. Some of the values that may point to a disk bottleneck are:

- ▶ **lwait**

This is the time the CPU spends waiting for an I/O to occur.

- ▶ **Average queue length**

This is the number of outstanding I/O requests. In general, when the value is higher than 2 to 3, it means there may be a disk I/O bottleneck. This applies to systems with a single disk.

In disk arrays, however, the queue length may be different and not necessarily indicate a Linux bottleneck; it may be under the control of the I/O controller using cache or other methods.

- ▶ **Average wait**

This is a measurement of the average time in ms that it takes for an I/O request to be serviced. The wait time consists of the actual I/O operation and the time it waits in the I/O queue.

- ▶ **Transfers per second**

This refers to the number of I/O operations per second (reads and writes).

- ▶ **Blocks read/write per second**

This refers to the reads/writes per second in blocks of 512 bytes in the kernel 2.6 style.

- ▶ kBytes per second read/write

This refers to the reads/writes from/to the block device in kBytes.

These parameters can be found in the proc file system or they can be gathered with tools like **iostat**, as shown in Example 22-3.

*Example 22-3 Sample of iostat output*

---

```
root@blade1]# iostat -kx /dev/sda
Linux 2.6.9-5.ELsmp (bc1srv7)   04/06/2005

Device:  rrqm/s wrqm/s   r/s    w/s    rsec/s   wsec/s   rkB/s   wkB/s
avgrq-sz avgqu-sz   await   svctm %util
sda      0.00  6223.38  0.00  588.56  0.00  54013.93  0.00  27006.97  91.77
          81.74    175.31   0.67   39.70

avg-cpu: %user   %nice   %sys %iowait   %idle
          0.25     0.00    36.00   21.50   42.25
```

---

**Note:** All Linux blocks are 512 bytes in size, although the kernel refers to blocks as being a larger size: Kernel 2.4 refers to a block as 4096 bytes and kernel versions earlier than 2.4 refer to a 2048 byte block size.

#### 22.4.1 Finding bottlenecks in the disk subsystem

A server that exhibits the following symptoms might be suffering from a disk bottleneck (or a hidden memory problem):

- ▶ Slow disks result in memory buffers that fill with write data or that wait for read data, which delays all requests because free memory buffers are unavailable for write requests.

Alternatively, the response waits for read data in the disk queue, or there is insufficient memory because there is not enough memory buffers for network requests, which can cause synchronous disk I/O.

- ▶ Disk or controller use is typically very high.
- ▶ Most LAN transfers occur only after disk I/O has completed, which causes very long response times and low network utilization.
- ▶ Because disk I/O can take a relatively long time and disk queues can become full, the CPUs are idle or have low utilization because they wait a long time before processing the next request.

The disk subsystem is perhaps the most challenging subsystem to configure properly. In addition to looking at raw disk interface speed and disk capacity, it is important to understand the workload. Is disk access random or sequential? Is

there large I/O or small I/O? Answering these questions can provide you with the necessary information to make sure that the disk subsystem is tuned adequately.

Disk manufacturers tend to showcase the upper limits of their drive technology's throughput. However, taking the time to understand the throughput of your workload can help you to set true expectations for your underlying disk subsystem.

*Table 22-2 Exercise showing true throughput for 8K I/Os for different drive speeds*

Disk speed	Latency	Seek time	Total Random Access Time <sup>a</sup>	I/Os per second per disk <sup>b</sup>	Throughput given 8 KB I/O
15 000 RPM	2.0 ms	3.8 ms	6.8 ms	147	1.15 MBps
10 000 RPM	3.0 ms	4.9 ms	8.9 ms	112	900 KBps
7 200 RPM	4.2 ms	9 ms	13.2 ms	75	600 KBps

a. Assuming that the handling of the command + data transfer < 1 ms, total random access time = latency + seek time + 1 ms.  
b. Calculated as 1/total random access time.

Random read/write workloads usually require several disks to scale. The bus bandwidths of SCSI or Fibre Channel are of lesser concern. Larger databases with random access workload benefit from having more disks. Larger SMP servers scale better with more disks. Given the I/O profile of 70% reads and 30% writes of the average commercial workload, a RAID-10 implementation performs 50% to 60% better than a RAID-5.

Sequential workloads tend to stress the bus bandwidth of disk subsystems. Pay special attention to the number of SCSI buses and Fibre Channel controllers providing a greater connection bandwidth where maximum throughput is desired. Given the same number of drives in an array, RAID-10, RAID-0, and RAID-5, all have similar streaming read and write throughput.

There are two ways to approach disk bottleneck analysis:

- ▶ *Real-time monitoring* must be done while the problem is occurring. Real-time monitoring might not be practical in cases where system workload is dynamic and the problem is not repeatable. However, if the problem is repeatable, this method is very flexible because of the ability to add objects and counters as the problem becomes well understood.
- ▶ *Tracing* is the collecting of performance data over time to diagnose a problem. This method is a useful way to perform remote performance analysis. Some drawbacks of this method include the potential for having to analyze large files when performance problems are not repeatable, and the

potential for not having all the key objects or parameters in the trace and having to wait for the next time the problem occurs for the additional data.

## The vmstat command

You can use the **vmstat** tool to track disk usage on a Linux system. The columns of interest in vmstat with respect to I/O are the bi and bo fields. These fields monitor the movement of blocks in and out of the disk subsystem. Having a baseline is key to being able to identify any changes over time.

Example 22-4 shows an example of vmstat output.

Example 22-4 vmstat output

---

r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id	wa
2	1	0	9004	47196	1141672	0	0	0	950	149	74	87	13	0	0
0	2	0	9672	47224	1140924	0	0	12	42392	189	65	88	10	0	1
0	2	0	9276	47224	1141308	0	0	448	0	144	28	0	0	0	100
0	2	0	9160	47224	1141424	0	0	448	1764	149	66	0	1	0	99
0	2	0	9272	47224	1141280	0	0	448	60	155	46	0	1	0	99
0	2	0	9180	47228	1141360	0	0	6208	10730	425	413	0	3	0	97
1	0	0	9200	47228	1141340	0	0	11200	6	631	737	0	6	0	94
1	0	0	9756	47228	1140784	0	0	12224	3632	684	763	0	11	0	89
0	2	0	9448	47228	1141092	0	0	5824	25328	403	373	0	3	0	97
0	2	0	9740	47228	1140832	0	0	640	0	159	31	0	0	0	100

---

## The iostat command

You can also encounter performance problems when too many files are opened and read and written to, and then closed repeatedly. This problem can become apparent as seek times (the time it takes to move to the exact track where the data is stored) begin to increase.

Using the **iostat** tool, you can monitor the I/O device loading in real time. Different options for this tool allow you to drill down even further to gather the necessary data.

Example 22-5 shows a potential I/O bottleneck on the device /dev/sdb1. This output shows average wait times (await) of around 2.7 seconds and service times (svctm) of 270 ms.

Example 22-5 Sample of an I/O bottleneck as shown with iostat 2 -x /dev/sdb1

---

avg-cpu:	%user	%nice	%sys	%idle
	11.50	0.00	2.00	86.50

---

```

Device: rrqm/s wrqm/s r/s w/s rsec/s wsec/s rkB/s wkB/s avgrq-sz avgqu-sz
await svctm %util
/dev/sdb1 441.00 3030.00 7.00 30.50 3584.00 24480.00 1792.00 12240.00 748.37 101.70
2717.33 266.67 100.00

avg-cpu: %user %nice %sys %idle
          10.50   0.00   1.00  88.50

Device: rrqm/s wrqm/s r/s w/s rsec/s wsec/s rkB/s wkB/s avgrq-sz avgqu-sz
await svctm %util
/dev/sdb1 441.00 3030.00 7.00 30.00 3584.00 24480.00 1792.00 12240.00 758.49 101.65
2739.19 270.27 100.00

avg-cpu: %user %nice %sys %idle
          10.95   0.00   1.00  88.06

Device: rrqm/s wrqm/s r/s w/s rsec/s wsec/s rkB/s wkB/s avgrq-sz avgqu-sz
await svctm %util
/dev/sdb1 438.81 3165.67 6.97 30.35 3566.17 25576.12 1783.08 12788.06 781.01 101.69
2728.00 268.00 100.00

```

---

The **iostat -x** command (for extended statistics) provides low-level detail of the disk subsystem. The output for this command gives you the following information:

- ▶ **%util**: percentage of time one or more I/Os are outstanding to the device
- ▶ **svctm**: average time required to complete a request, in milliseconds
- ▶ **await**: average amount of time an I/O waited to be served, in milliseconds
- ▶ **avgqu-sz**: average queue length
- ▶ **avgrq-sz**: average size of request
- ▶ **rrqm/s**: the number of read requests merged per second that were issued to the device
- ▶ **wrqm/s**: the number of write requests merged per second that were issued to the device

For a more detailed explanation of the fields, see the man page for **iostat**.

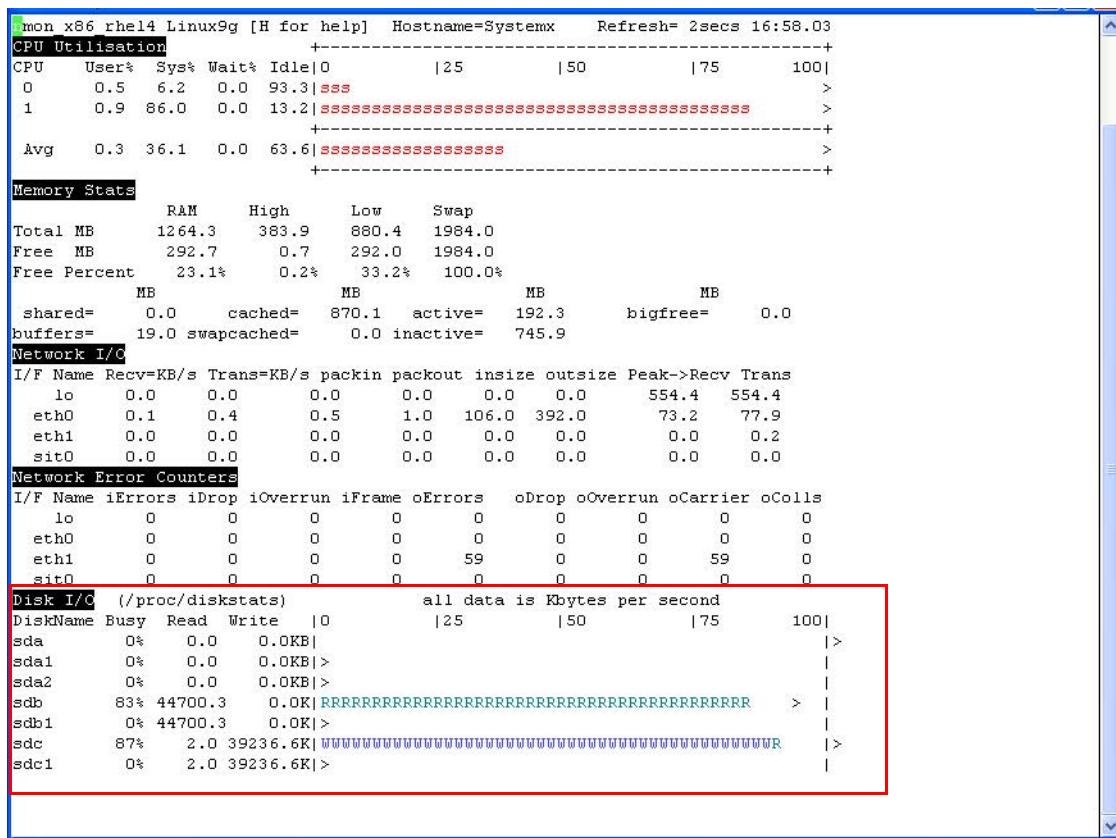
Changes made to the elevator algorithm, as described in “Tuning the elevator algorithm (kernel 2.4 only)” on page 484, will be seen in the **avgrq-sz** (average size of request) and **avgqu-sz** (average queue length).

Because the latencies are lowered by manipulating the elevator settings, the **avgrq-sz** will go down. You can also monitor the **rrqm/s** and **wrqm/s** to see the effect on the number of merged reads and writes that the disk can manage.

## Using NMON

When using NMON, you can view disk activity. NMON gives you a quick overview of disk and partition throughput in KBps.

To display disk activity with NMOM, press the d key. NMON also displays what kind of I/O is performed (read or write), as well as disk utilization. The refreshing period is user-defined, and you can monitor disk activity in timed intervals (every two seconds, for example). Figure 22-4 illustrates the NMON display with, among other information, disk activity.



*Figure 22-4 NMON monitoring disks*

#### **22.4.2 Performance tuning options for the disk subsystem**

After verifying that the disk subsystem is a bottleneck, a number of solutions are possible, including:

- If the workload is of a sequential nature and it is stressing the controller bandwidth, the solution is to add a faster disk controller. However, if the

workload is more random in nature, then the bottleneck is likely to involve the disk drives, and adding more drives will improve performance.

- ▶ Add more disk drives in a RAID environment to spread the data across multiple physical disks and improve performance for both reads and writes. This addition increases the number of I/Os per second. Also, use hardware RAID instead of the software implementation that is provided by Linux. If hardware RAID is used, the RAID level is hidden from the operating system.
- ▶ Offload processing to another system in the network (either users, applications, or services).
- ▶ Add more RAM. Adding memory increases system memory disk cache, which in effect improves disk response times.

## 22.5 Network bottlenecks

A performance problem in the network subsystem can be the cause of many problems, such as a kernel panic. To analyze these anomalies to detect network bottlenecks, each Linux distribution includes traffic analyzers.

### 22.5.1 Finding network bottlenecks

We recommend using KDE System Guard because of its graphical interface and ease of use. The tool is also available on the distribution CDs. For more information about this tool, see 18.9, “KDE System Guard” on page 617.

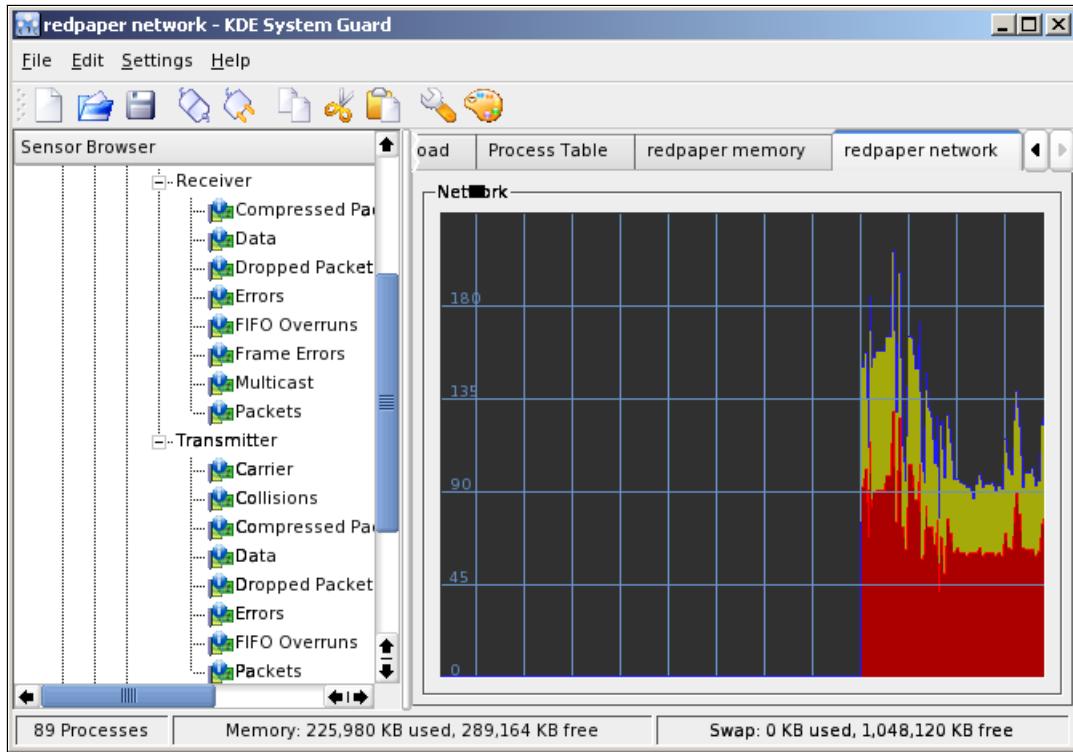


Figure 22-5 KDE System Guard network monitoring

For SUSE Linux, you can also use the traffic-vis package, which is an excellent network monitoring tool. This tool allows you to capture the network data, and then analyze the results using a Web browser. For details, see 18.11, “Traffic-vis” on page 625.

It is important to remember that there are many possible reasons for these performance problems and that sometimes, problems occur simultaneously, making it even more difficult to pinpoint the origin. The indicators listed in Table 22-3 can help you to determine the problem with your network.

Table 22-3 Indicators for network analysis

Network indicator	Analysis
Packets received Packets sent	Shows the number of packets that are coming in and going out of the specified network interface. Check both internal and external interfaces.

Network indicator	Analysis
Collision packets	Collisions occur when there are many systems on the same domain. The use of a hub might be the cause of many collisions.
Dropped packets	Packets can be dropped for a variety of reasons, but the result might impact performance. For example, if the server network interface is configured to run at 100 Mbps full duplex, but the network switch is configured to run at 10 Mbps, the router might have an ACL filter that drops these packets, for example: <pre>iptables -t filter -A FORWARD -p all -i eth2 -o eth1 -s 172.18.0.0/24 -j DROP</pre>
Errors	Errors occur if the communications lines (for example, the phone line) are of poor quality. In these situations, corrupted packets must be present, thereby decreasing network throughput.
Faulty adapters	Network slowdowns often result from faulty network adapters. When this kind of hardware fails, it can begin to broadcast junk packets in the network.

## 22.5.2 Performance tuning options for the network subsystem

To solve problems related to network bottlenecks:

- ▶ Ensure that the network card configuration matches router and switch configurations (for example, frame size).
- ▶ Modify how your subnets are organized.
- ▶ Use faster network cards.
- ▶ Tune the appropriate IPV4 TCP kernel parameters; refer to Chapter 15, “Linux” on page 453, for more information about this topic. Some security-related parameters can also improve performance, as described in that chapter.
- ▶ If possible, change network cards and check performance again.
- ▶ Add network cards and bind them together to form an adapter team, if possible.





## Case studies

In this chapter, we present four case studies for servers with performance problems. In these studies, we show how the subsystems behave for different load conditions and how to detect bottlenecks using monitoring tools. We also analyze the data that is generated by these tools, arrive at a conclusion based on this analysis, and recommend a course of action to improve performance.

We begin our discussion with a general overview of the two modes for system monitoring. We then discuss the following case studies:

- ▶ 23.2, “Case 1: SQL Server database server” on page 745
- ▶ 23.3, “Case 2: File servers hang for several seconds” on page 758
- ▶ 23.4, “Case 3: Database server” on page 766

## 23.1 Analyzing systems

To analyze a system, it is best to start with an overview and then move down in details to the bottleneck components. Essentially, you can divide bottleneck components into *hardware*, *software*, and *external* (users and network). There are two modes for monitoring a system: trace and real-time.

### Trace mode

Trace mode allows you to collect data for a given period of time, so that you can monitor the system during a specific activity period. You should collect all possible counters so that there is no need to rerun the workload if a counter is forgotten. You should then collect the following objects and counters:

- ▶ Processor: <All Counters>
- ▶ Memory: <All Counters>
- ▶ Physical disk: <All Counters>
- ▶ Network: <All Counters>
- ▶ System: Processor queue length

For advanced software bottlenecks, you may need to look at the following objects, however, these counter objects have a higher overhead to collect, and consume significantly more space to store, so it is best to only collect these as needed.

- ▶ Process: <All Counters>
- ▶ Thread: <All Counters>

### Real-time monitoring

Real-time monitoring is useful while a problem is actually occurring. This mode requires a steady state problem that is occurring on a server with no dynamic workload. Otherwise, it can be difficult to isolate the bottlenecks. This mode allows you to add and to remove any counter at any time. So, you can examine each counter as you begin to understand the problem. However, it is important to keep in mind that Perfmon displays averages for most counters.

We created all of the case studies in this chapter using Perfmon. Perfmon is the default and most commonly used performance monitoring tool for Windows systems. However, you can also use Perfmon to display Linux performance logs when used with xPL.

For more information about Perfmon, see 17.1, “Reliability and Performance Monitor console” on page 534. For more information about xPL, see 18.16, “System x Performance Logger for Linux” on page 632.

## 23.2 Case 1: SQL Server database server

This case study does not focus on a particular behavior from the system. It is instead more an analysis of how the server is actually working and what can be done to improve overall performance. The server in this case is a 4-core Xeon using a 100 Mbps Ethernet interface and three drive arrays, defined further within the case study.

Figure 23-1 illustrates the maximum values overview in Perfmon. These values provide a useful view of the data and allow us to detect multiple bottlenecks. They also show an example of trace mode system monitoring.

\GAALPSVR01AO					
Memory					
Available MBytes	1294				
Page Reads/sec	141.725	①			
Page Writes/sec	33.671				
Network Interface					
Broadcom Advanced Server Program Driver for Windows 2000					
Bytes Received/sec	7600768.815				
Bytes Sent/sec	3613564.141				
Packets/sec	8571.162				
PhysicalDisk					
Avg. Disk sec/Read	0 C:	1 D:	2 V:		
③ Avg. Disk sec/Write	0.058	4.012	0.647		
Split IO/Sec	2.890	4.265	4.083		
	26.258	238.618	2.014		
Processor					
% Privileged Time	_Total	0	1	2	3
④ % Processor Time	66.664	99.996	70.310	99.996	62.498
% User Time	100.000	100.000	100.000	100.000	100.000
	124.653	121.913	126.023	126.023	124.653

Figure 23-1 Maximum values overview

In Figure 23-1, the Page Reads and Page Writes per second counters (①) for memory indicate a moderate value that might imply a memory bottleneck. The network interface is indicating 7.6 MBps of received data (②). Given that this is a 100 Mbps Ethernet interface, this value may be high.

There are other potential bottlenecks on the disks, as well. More than four seconds as an average disk seconds per write (③) indicates a bottleneck. Finally, the percentage of processor time on each CPU is 100% (④) indicating every CPU is saturated.

Let us analyze each of these components more thoroughly.

### 23.2.1 Memory analysis

Memory is analyzed using the paging counters and the cache hit ratio. This analysis indicates if there is enough installed memory and if some data is written to, or read from, disks instead of memory. A trace mode monitoring of the memory subsystem provides some information about how memory is accessed and used by SQL Server.

Figure 23-2 shows that only 0.3 pages are read per second as an average, while the peak is reaching 142 page reads (white lines). This information tells us the paging activity is not a primary bottleneck but is effecting performance negatively. Additional checks of SQL Server Buffer Cache hit rate might be necessary to determine whether additional memory will reduce paging.

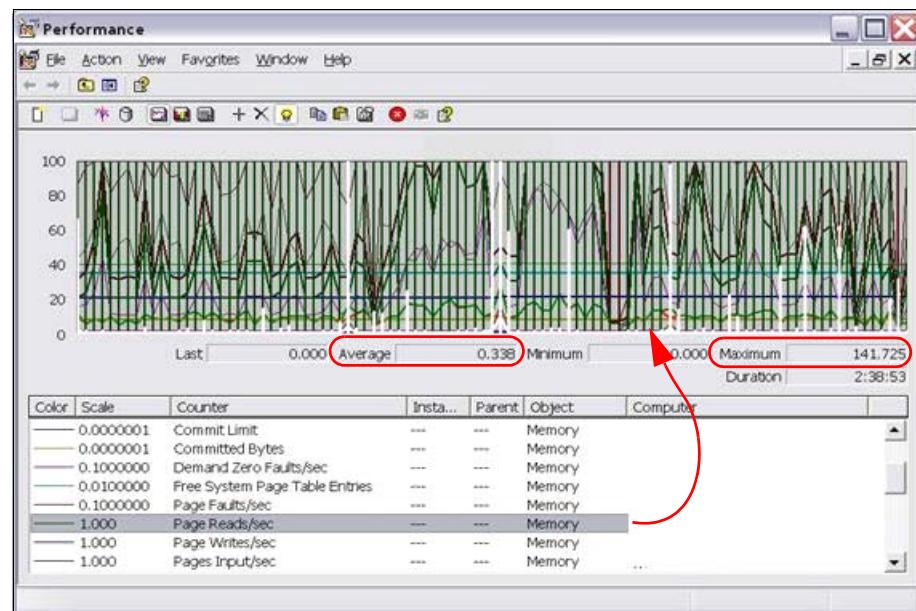


Figure 23-2 Memory analysis

Figure 23-3 shows cache faults per second. Cache faults indicate that server cache pages were requested but not available in cache (probably because of low cache memory capacity). Combined with page accesses (read and write) per second, this analysis indicates that these pages are being accessed on the paging disk. The server is then short on operating system and application memory.

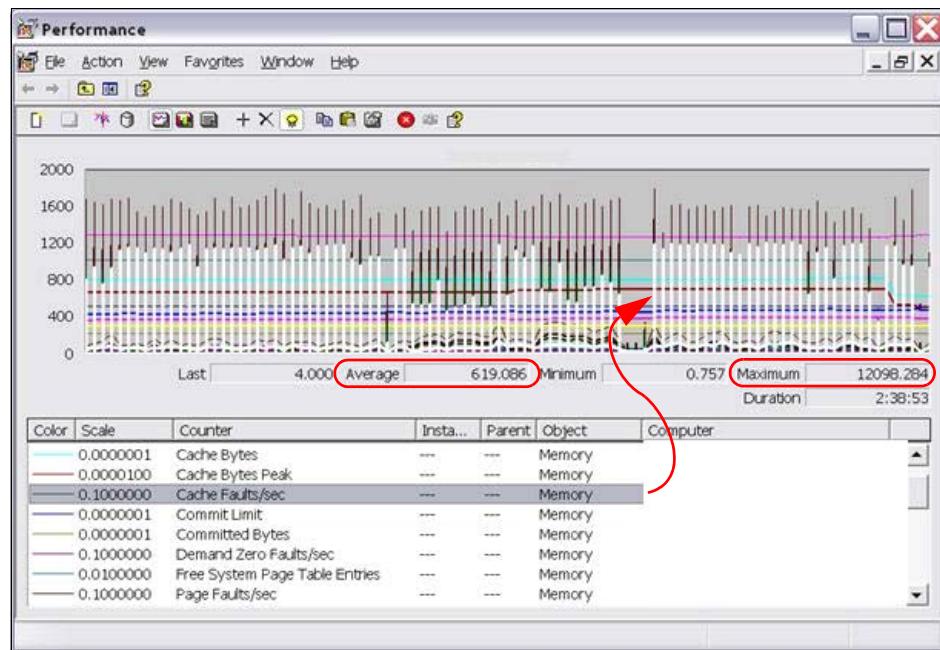


Figure 23-3 Cache faults per second

From this data, it appears that the server is acting as a file server or running some application that is using memory outside of SQL Server. This activity is producing cache page misses and pressuring the reserved memory for the cache. For optimal performance, you should add at least another 1 to 2 GB of memory for operating system use.

### 23.2.2 Processor analysis

The average processor utilization is more than 40%, with frequent peaks at 100%. This usage is divided in privileged time with an average of 12%, maximum at 67% and a user time, which is the application usage. This application time represents the majority of the CPU's utilization.

Figure 23-4 shows that regular peaks at 100% of the processor utilization are occurring. However, interrupt and privileged time percentages indicate optimal operating system and drivers efficiency. Therefore, for optimal performance, it might be necessary to offload some workload from the server, to upgrade the server with faster processors, or even to replace the server with a new server with faster processors.



Figure 23-4 Processor time

### 23.2.3 Network analysis

Figure 23-5 shows an average counters analysis and Figure 23-6 shows the corresponding maximums from the trace logs for the network subsystem. From this analysis, the average throughput for the only network adapter is around 2 Mbps with a maximum reaching 84 Mbps. For a 100 Mbps network connection, the maximum value is very close to the limit, though the average throughput (read and write) is very low. More detailed investigation of the network subsystem is warranted.

Further analysis shows that the network is experiencing heavy receive traffic and almost no send traffic. The receive traffic itself is not linear and peaks occur during a third of the run period, which reduces the average value dramatically.

Network Interface	Broadcom Advanced Server Program Driver for Windows 2000
Bytes Received/sec	245294.173
Bytes Sent/sec	18876.944
Bytes Total/sec	262412.873
Current Bandwidth	100000000
<b>Output Queue Length</b>	0
Packets Outbound Discarded	0
Packets Outbound Errors	0
Packets Received Discarded	0
Packets Received Errors	0
Packets Received Non-Unicast/sec	33.950
Packets Received Unicast/sec	181.958
Packets Received Unknown	113555936
Packets Received/sec	270.279
Packets Sent Non-Unicast/sec	0.002
Packets Sent Unicast/sec	112.145
Packets Sent/sec	112.147
Packets/sec	382.426

Figure 23-5 Average network utilization

Network Interface	Broadcom Advanced Server Program Driver for Windows 2000
Bytes Received/sec	7600768.815
Bytes Sent/sec	3613564.141
Bytes Total/sec	8398657.345
Current Bandwidth	100000000
<b>Output Queue Length</b>	4
Packets Outbound Discarded	0
Packets Outbound Errors	0
Packets Received Discarded	0
Packets Received Errors	0
Packets Received Non-Unicast/sec	472.321
Packets Received Unicast/sec	5373.440
Packets Received Unknown	113812991
Packets Received/sec	5460.588
Packets Sent Non-Unicast/sec	1.008
Packets Sent Unicast/sec	3110.574
Packets Sent/sec	3110.574
Packets/sec	8571.162

Figure 23-6 Maximum network utilization

The peak receive traffic during a period of time might show a network limitation. Indeed, the maximum values are almost reached, and a network subsystem improvement would increase the upper limit. Moreover, if the other components are improved (memory and processor, for example), the throughput demand might increase, and the network might then act as a bottleneck.

Thus, you could implement the following solutions to improve the network performance:

- ▶ Add a new adapter and set up adapter teaming.
- ▶ Upgrade to a faster network connection (1 or 10 Gigabit). You may need to upgrade the remainder of the network components as well, such as switches.
- ▶ Upgrade the current version of the operating system from Windows 2000 to Windows Server 2003 or 2008, because these newer operating systems have improved network stacks capable of sustaining higher utilizations with better efficiency.

## 23.2.4 Disk analysis on the C: drive

The C: drive is used to host the operating system, the application programs, and the system page file. Additionally, all Perfmon logs are being saved on the C: drive.

The optimal performance for a given disk subsystem is obtained when latency is not more than ~25 ms. Figure 23-7 shows the C: drive read latency. The average read latency is around 7 ms, well within the acceptable limits. Some peak values sometimes reach 60 ms, which is moderate and may deserve investigation.

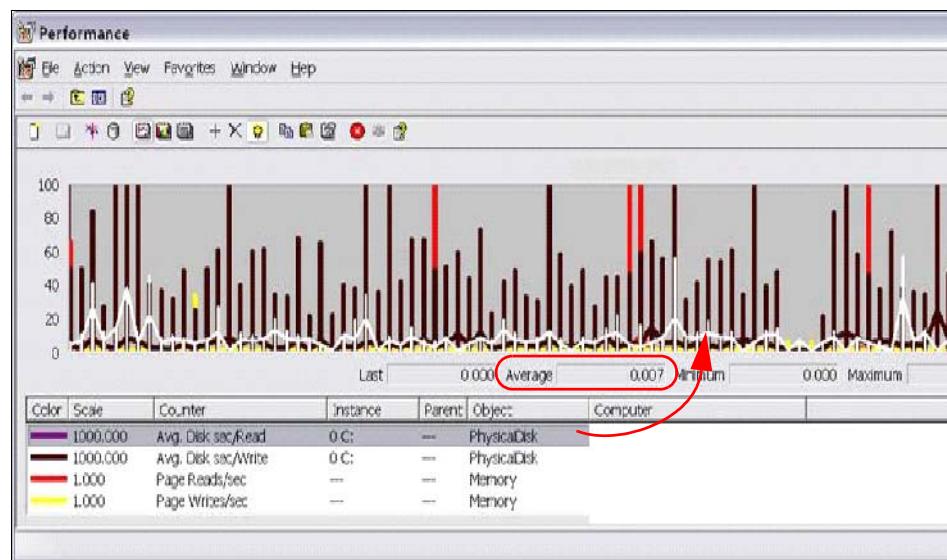


Figure 23-7 C: drive read latency

Figure 23-8 shows write latency on the same disk. The data shows an average latency around 6 ms (very good), but more importantly shows peaks at more than 2.5 seconds (very high). Some frequent write latency peaks indicate a significant bottleneck.

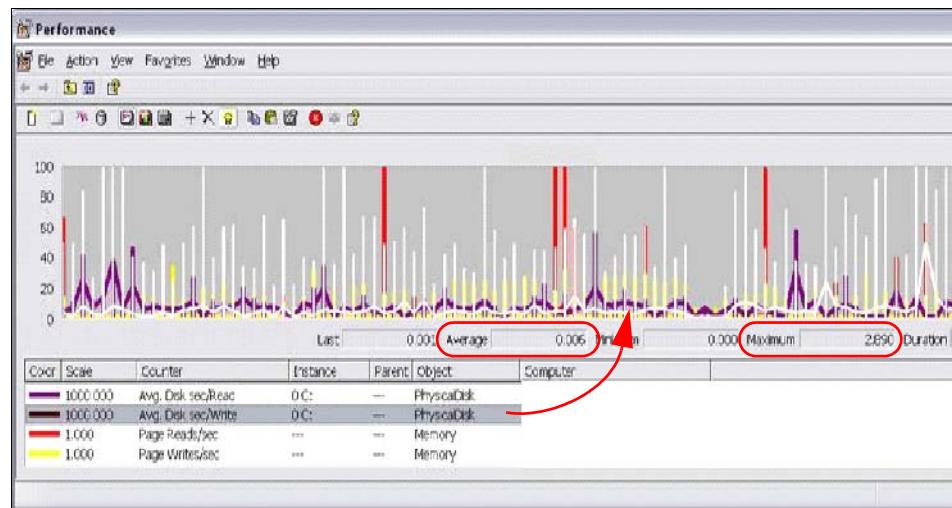


Figure 23-8 C: drive write latency

The write latency peaks correlate to the page write activity, indicating that these peaks can be reduced by adding more memory to the system. Perfmon log writes also occur to this disk and can increase the write latency as well. However, this behavior is insignificant compared to the overall performance.

Further analysis shows that the Average “Disk Bytes per Read” (which is the read size) is around 4 KB with peaks at 64 KB. Also, the drive write size gives frequent maximum values at 64 KB. Investigation into the RAID adapter’s stripe size indicates that it is set to 8 KB. At this stripe size, each 64 KB read or write access actually reads or writes eight 8KB block I/Os. This analysis shows that changing the stripe size for the C: drive array to 64 KB will improve performance by reducing latency.

Finally, disk throughput analysis shows an average read throughput is around 4 KBps, with a maximum of only 1.5 MBps and an average write throughput of 90 KBps with a maximum of 5 MBps. These values are very low, even for a RAID-1 array of two disks. Consequently, we can say there is absolutely no disk throughput bottleneck on the C: drive.

### 23.2.5 Disk analysis on the D: drive

The D: drive is used for logs and as temporary data store (tempDB). This drive array is composed of four 10 k RPM and uses an 8 KB Stripe.

Latency monitoring on that disk shows extremely high read and write values (up to 4.3 seconds). Additional analysis on the disk's stripe size show an average read size of 66 KB with a maximum of 262 KB, and an average write size of 51 KB with a maximum of 521 KB (as shown in Figure 23-9).

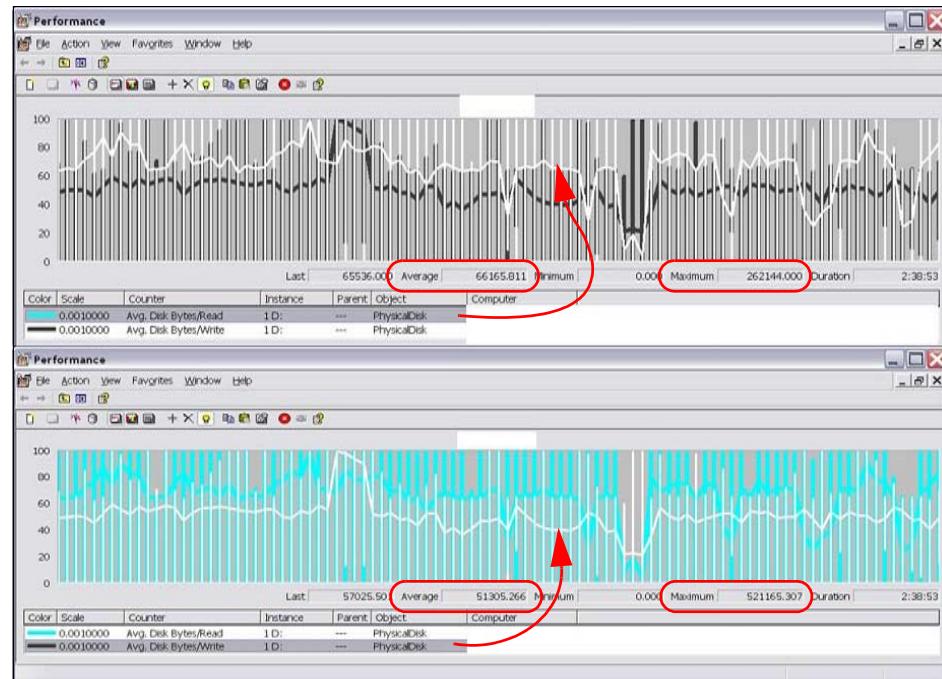


Figure 23-9 Read/write strip sizes on disk D:

In addition, I/O analysis shows very heavy load on write accesses with an average of 125 operations per second and a maximum of 1148 operations per second. With a default stripe size set at 8 KB, there are too many physical operations occurring on the disk array. Given that the array is based on four disks, the maximum number of operations per second is very high. You can reduce the number of operations per second by doing the following:

- ▶ Increasing the number of disks (to dispatch the load)
- ▶ Increasing the stripe size to 64 KB or more
- ▶ Replacing the disks with faster disks (15 K RPM instead of 10 K RPM)

The analysis shows that the disk accesses take a very long time to complete, and that the stripe size is not optimized, thus increasing the number of I/Os significantly.

### 23.2.6 Disk analysis of the V: drive

The V: disk is used for the database, and is configured with 6 disks using an 8 KB stripe.

When the latency on disk V: is monitored, we notice an average read latency around 20 ms, with a maximum of 647 ms. While the average is acceptable, the peaks are very high, indicating we should analyze the disk further. In addition, the average write latency is 29 ms with extremely high peaks at 4 seconds.

Further analysis shows an average read size of 12.4 KB with some peaks at 94 KB and an average write size of 51 KB with frequent peaks at 538 KB. However, most of the write peaks are under 64 KB. Thus, the long write latency that is associated with large and frequent random writes produces a bottleneck on the drive. Again, the default stripe size is here 8 KB, which generates eight physical disk I/O for the frequent 64 KB writes, which partially explains the long latency.

A deeper analysis of I/O performance (as shown in Figure 23-10 on page 755) shows a disk read operation per second that is around 35 as an average and 935 operations per second as maximum peaks for read. This load is moderate for a six-disk array. Additionally, the write operations average 52 operations per second and reach peaks at 1584 operations per second.

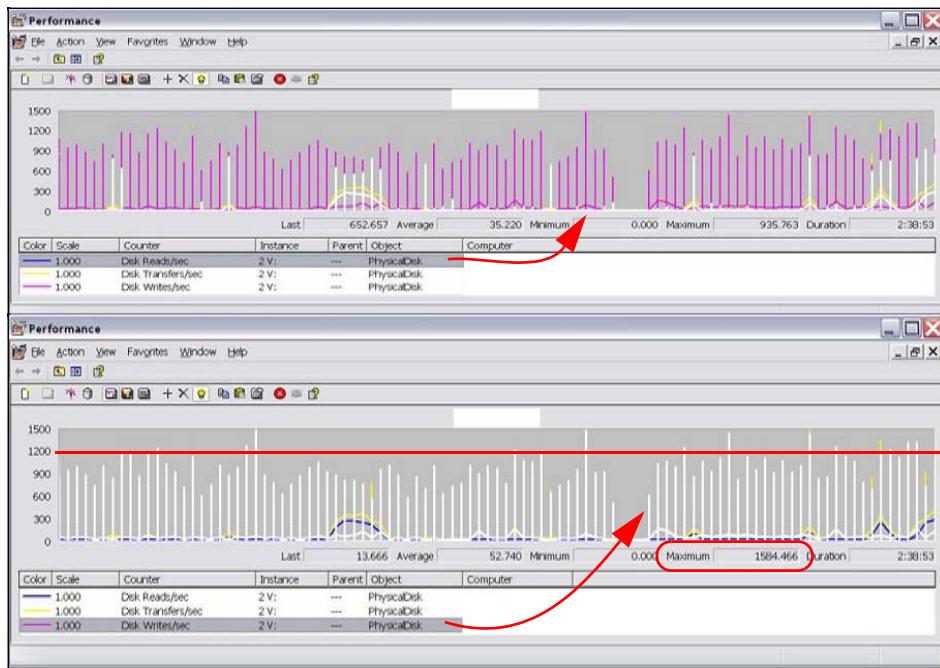


Figure 23-10 Disk V: I/Os

When looking at the combined read/write rate, we see an average of 87 I/Os per second (35 read operations + 52 write operations) and a peak of 1584 operations per second for disk write peaks. Given that the array of six disks can deliver at most 200 to 250 I/O per second for each disk, the maximum operations per second that this array could perform is around 1200 to 1500 I/O per second. This limit is exceeded with the write peaks (see the bottom graph in Figure 23-10).

The heavy I/O load on the V: disk array, coupled with very large writes, causes multiple physical I/Os and long latency. This implies that the system is bottlenecked on that drive. To improve performance, you should increase the stripe size up to 64 KB and increase the number of drives to accommodate the peak load requirements.

The throughput analysis shows light to moderate throughput on the V: drive with an average read rate of about 602 KBps (peaks at 49 MBps). The write transfer rates have an average of 2.7 MBps and peaks at 38 MBps. These limits are well within the capabilities of the RAID controller, indicating no throughput bottleneck.

### 23.2.7 SQL Server analysis

Figure 23-11 illustrates that free pages are sometimes dropping to zero (0). The average free pages is 5246 pages (or 41.9 MB), which is very low. In addition, the Procedure cache is consuming all memory. As a consequence, starving the procedure cache can cause excess I/O to tempDB and might explain the high read rate to D: drive.



Figure 23-11 SQL Server buffer trace logs

Adding 2 GB of memory dedicated to SQL server would improve performance in this situation.

### 23.2.8 Summary of Case 1

All subsystems of this server are considered bottlenecks mainly because a single component is having issues in performing the I/Os and, thus, the entire system is impacted. In summary, the trace logs of Perfmon indicate:

- ▶ The memory subsystem is the greatest bottleneck in the system, experiencing excessive paging and memory starvation while heavy paging is associated with slow disk performance for the C: drive. It appears that the server is acting as a file server or running some application that is using memory outside of SQL Server. This activity is producing cache page misses and pressuring the reserved memory for the cache.

For optimal performance, it would be recommended to add another 1 GB to 2 GB of memory just for the operating system. In addition, because SQL Server is memory-starved, it is suggested that you add at least 2 GB just for SQL

Server use. In total, 4 GB of additional memory should alleviate these memory bottlenecks.

- ▶ The processor subsystem is a bottleneck as well. Processors are frequently a bottleneck with regular peaks at 100%. Interrupt percentages and Privileged time percentages indicate optimal operating system and driver efficiency.

For optimal performance, it is recommended to offload some workload from the server, upgrade the server with faster processors, or replace the server. When the disk and memory bottlenecks are addressed, processor utilization will significantly increase over the current 43% average.

- ▶ The network subsystem is considered a bottleneck. While not severe, the network is often running at maximum receive rates. Resolving other bottlenecks within the memory, disk, and processor configuration results in greater network throughput demand and potentially significant network bottlenecks.

A recommendation is to explore updating the network infrastructure to 1 Gigabit or 10 Gigabit, in this case.

- ▶ The disk subsystem acting as a bottleneck because of usage of the 8 KB stripe sizes and frequent 256 KB to 512 KB I/O to log and database files. For the C: drive, long write latency is associated with large and frequent paging writes. A 8 KB stripe size translates to eight physical disk I/Os for the frequent 64 KB writes and would partly explain high latencies.

Analysis shows that about half of the write traffic is paging traffic. Changing the stripe (which is a destructive operation, in most cases) may help alleviate the disk bottlenecks, as will reductions in paging due to a memory increase.

- ▶ Another analyzed drive is the D: drive. Analysis shows that peak latencies of 4 seconds occur when peak I/O requests are queued to LUN D:. Heavy load on the D: drive, especially because the stripe size is set to 8 KB (default), are noticed. Also, long latency is associated with very large 512 KB I/O to the D: disk.

The write cache policy of the RAID card should be verified, with caching enabled if possible. Alternatively, consider moving tempDB to a larger LUN because that is the source of most read traffic and is about half the total disk load.

- ▶ The final disk that was analyzed was the V: drive. Analysis showed moderate to heavy I/O load on the V: drive array coupled with very large writes causing long latency. The system is disk bottlenecked on V: due to very large I/O sizes and high I/O rates.

It is recommended to upgrade the stripe size to at least 64 KB and increase the number of drives to 12 or more for this array.

### **23.3 Case 2: File servers hang for several seconds**

The configuration for this case study uses multiple servers, which consist of both file servers and application servers. The issue of file servers that hang for several seconds implies a long wait time for users who are based in Web browsers. To help determine the problem and to solve it, the case study monitors the file server's subsystems.

Again, to gain a complete picture of what is happening on the file server's subsystems, we need to monitor each key subsystem: memory, processors, network, and disks. Each component is related to the others, and a bottleneck issue on one component can impact the others. However, sometimes the actual bottleneck is not that obvious.

### 23.3.1 Memory analysis

We begin the scenario by analyzing the memory subsystem. Figure 23-11 on page 756 shows that free memory is about 2.6 GB to 3.0 GB. There is no significant paging, and the cache is stable. Therefore, we can say there are no configuration issues, and this memory configuration is suitable for the current workload.

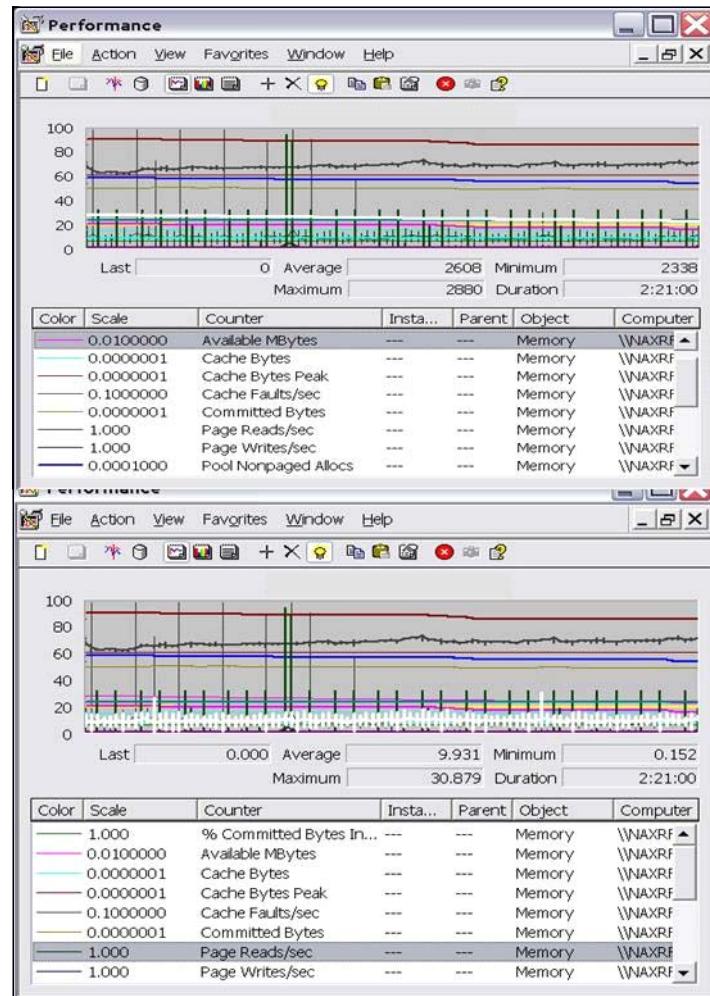


Figure 23-12 Memory analysis

### 23.3.2 Processor analysis

Figure 23-13 shows the analysis of the processor subsystem. This analysis is interesting because it shows when the server hangs. The average processor utilization is only 15%. However, the analysis shows huge peaks at 99.9% when the server hangs.

The deeper analysis shows that this processor utilization is essentially kernel time (meaning the operating system or drivers). From this analysis, you can determine that upgrading the processor to a faster one will not solve the problem, because the processor is busy an average of 15% and hang events would not likely be solved.

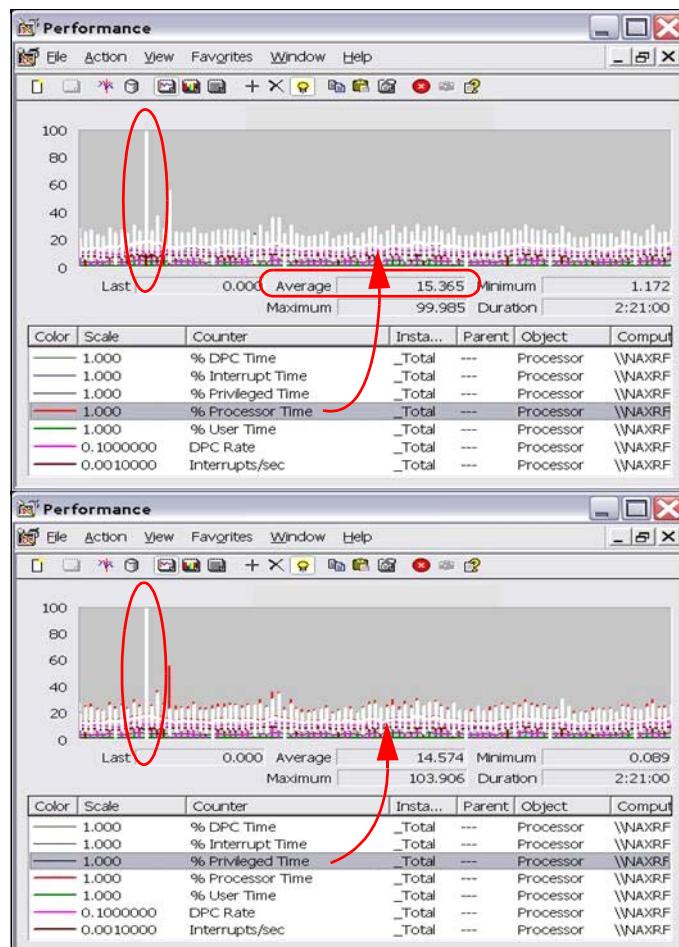


Figure 23-13 Processor analysis shows hang events on peak utilization

### 23.3.3 Network analysis

The network analysis also shows hang events. The network subsystem appears to be underutilized and, therefore, is not causing a bottleneck. However, as shown in Figure 23-14, there is a peak throughput correlated to the spike in CPU utilization. The network average throughput is about 64 Mbps, and there is an average of 15 500 packets sent or received every second.

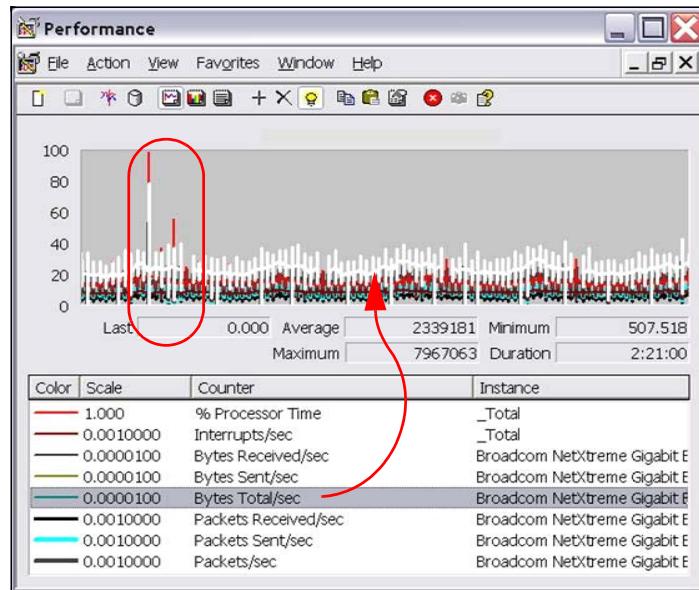


Figure 23-14 Network analysis shows hang events

The analysis of this subsystem shows that it is healthy. The LAN traffic is correlated with the CPU spike, which proves it is in response to a greater request in file servers load. The servers are then responding to the increased load from the application server.

### 23.3.4 Disks analysis of the V: drive

The V: drive is used to store the data from the file server. Figure 23-15 shows the average read latency is around 3 ms, which is acceptable. Some minor spikes occur but never above 20 ms, so they can be considered negligible. Moreover, the second graph shows an average size for read accesses around 4 KB.



Figure 23-15 Disk read accesses do not indicate significant bottleneck

Alternatively, when monitoring the write accesses, Figure 23-16 shows an average write latency around 3 ms (which is good) and a peak at 630 ms. This peak is probably not due to the disk configuration and must be related to the CPU peak occurring when server hangs. With an average write operation of 5 KB, we can say that the stripe size for this drive should be 8 KB.

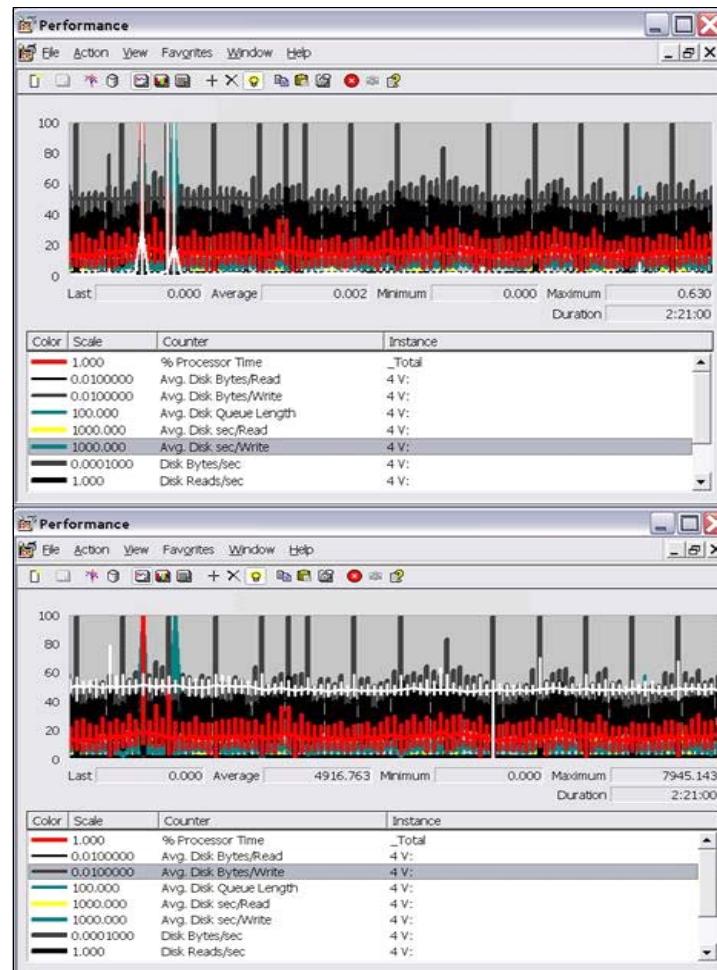


Figure 23-16 V: drive write accesses analysis does not show bottlenecks

Device analysis did not show why the server hangs for several seconds. Only CPU spikes are indicating there is a problem occurring, but the network, the memory, and the disk subsystems do not appear to be bottlenecks. Thus, at this point, we need to perform further analysis on a system level.

### 23.3.5 System-level analysis

To find the cause of the hang event, we keep the counter demonstrating the event (in our case, the processor time percentage) and add progressively the counters for each object to be examined. These objects include System, Server, and Process. You can monitor many counters for each object.

For this scenario, the first counter that we monitor is the File Control Operations Per Second on the system object (as shown in Figure 23-17, the white line). On average, the server is performing about 1300 file control operations per second. But during the CPU spike, the application server produces a storm of 10,000 file control operations per second. This spike consumes the CPU to nearly 100% utilization and appears to be a root cause of the performance issue.

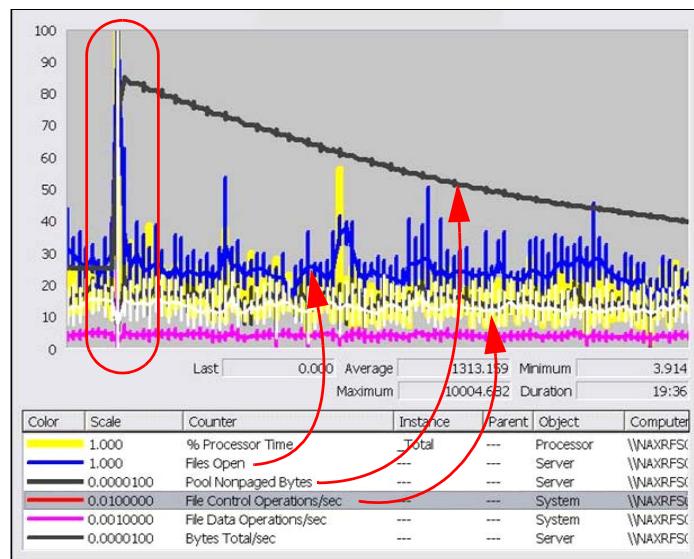


Figure 23-17 Analyzing file operation counters

Another counter monitored is the File Open counter on server object (Figure 23-17, the blue line) Here again, the number of files that are open correlate with the CPU utilization spike. This correlation confirms that the server is opening a large number of files and that drives the CPU utilization to nearly 100%.

During normal operation, the server has only about 25 files open. However, when the peak occurs, the application servers request that the file server open as many as 200 files. To buffer the file control structures, the server allocates Pool Nonpaged Bytes (Figure 23-17, the grey line).

Pool Nonpaged Bytes is memory that is non-pageable and allocated to store file control structures (directory and file pointers). Figure 23-17 on page 764 shows that, at the peak moment, 8 MB are allocated to nonpage pool, which is not excessive. Because fewer files are open, less memory is used and that is why there is the progressive decrease of bytes allocated to nonpage pool.

At the time of the peak, the amount of data that is sent or received by the file server drops to zero (0) because the CPU is 100% utilized processing file control operations. Because file control operations spike to over 10000 control operations per second, we can confirm that this is the root cause for high CPU utilization and, therefore, for the application hang.

### 23.3.6 Summary of Case 2

A solution to this case would be to increase the number of file servers so that the opened files are spread over multiple nodes, thus reducing the load to any one server.

## 23.4 Case 3: Database server

This case study analyzes a heavily loaded database server and suggests methods to improve its performance. Let us take a look at the performance statistics of the different server subsystems.

### 23.4.1 CPU subsystem

Because this is a dual processor machine, we first check whether the workload is evenly balanced across both CPUs. Uneven CPU balancing is often a problem with existing applications that do not account for multi-core machine architectures.

Figure 23-18 shows that both CPUs are utilized equally well. Although minimum and maximum utilization varies greatly, the average load in the CPUs is 42% to 44%. This usage indicates that the CPU subsystem is not the bottleneck for this system. A value below 70% is acceptable and leaves enough headroom for occasional spikes.

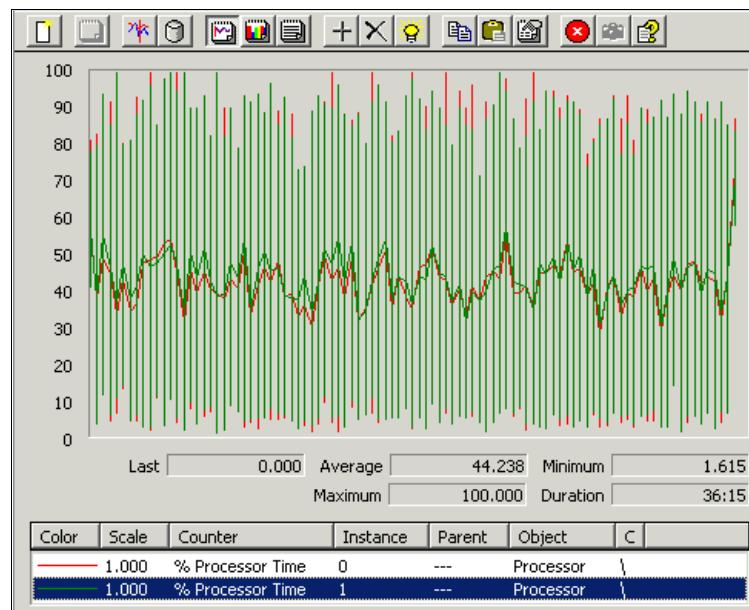


Figure 23-18 CPU utilization

## 23.4.2 Memory subsystem

To determine if the memory subsystem is a bottleneck, we determine if the system is paging a significant amount of data to the disk, which might be a symptom of a system running low on memory. To give an overall impression of memory utilization, we also take a look at the Memory:Available Bytes counter. Figure 23-19 shows that there is plenty of system memory available, yet the system is performing a substantial amount of page reads. (Note that the Page Reads/sec counter is shown on a 0.10 scale, so the average in this case is ~250 Page Reads / sec.) We need to determine why this bottleneck is occurring.

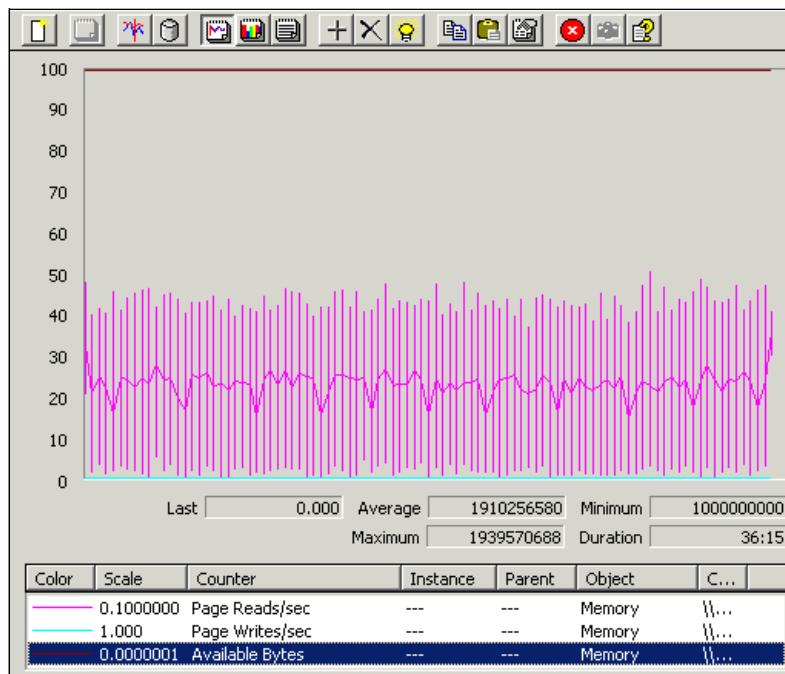


Figure 23-19 Memory utilization

Recall what the Page Reads/sec and Page Write/sec counters actually represent:

- ▶ *Page Reads/sec* is the number of times the disk was read to resolve hard page faults. Hard page faults occur when a process requires data that is not in its working set or elsewhere in physical memory and must be retrieved from disk.

This counter is designed as a primary indicator of the kinds of faults that cause system-wide delays. It includes reads to satisfy faults in the file system cache (usually requested by applications) and in non-cached mapped memory files. This counter counts numbers of disk read operations, without regard to the number of pages retrieved by each operation.

- ▶ *Page Writes/sec* is the number of times pages were written to disk to free up space in physical memory. Pages are written to disk only if they are changed while in physical memory, so they are likely to hold data, not code. This counter counts write operations, without regard to the number of pages written in each operation.

The high number of page reads suggests that the system is running low on memory and is paging heavily. However, when we look at the page writes for this scenario, they are constantly zero (0). This pattern is not an indicator of a low memory problem.

Instead, this behavior suggests that the paging activity is done by the application itself. The use of memory mapped files in an application usually accounts for this. See the following Knowledge Base entry for information about memory mapped files:

<http://support.microsoft.com/default.aspx?scid=kb;EN-US;q139609>

Because this issue appears to be with the application itself, the first recourse is to get detailed tuning information from the application vendor. If this is not possible, ensure that the paging device is a fast array (ideally, many 15 K RPM disks in a RAID-10 array) with a 64 KB stripe size to optimize for the page request size. The average Page Reads per Second value of 235 keeps two to three 10 K RPM disk drives busy.

### 23.4.3 Disk subsystem

Next, we look at the disk array to determine whether the disk subsystem is a bottleneck (Figure 23-20).

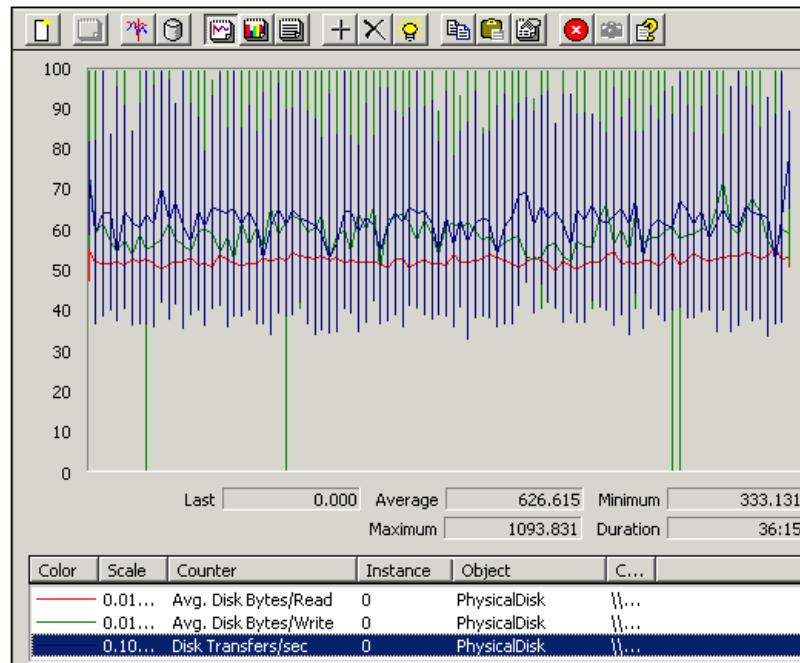


Figure 23-20 Disk counters for Disk 0

From our analysis, we can determine the following key disk metrics:

- ▶ Stripe size

Average disk bytes/read are 5267 and the maximum is 6498 bytes. So, as far as reads are concerned, the stripe size at 8 KB is acceptable because this is larger than both maximum read and write sizes.

Looking at write traffic, the Average Disk bytes/write are 5943 but the maximum is 38 297 bytes. The data shows a regular write size above 16 KB, so a 32 KB stripe size would be best for this array.

► I/O rates

Disk transfers/sec are 626 average and 1093 at peak. These are well under the limits of any modern RAID controller, indicating that the controller is not under stress.

So far, the only problem we have encountered is that the array uses a non-optimal stripe size. We move on and examine the Average disk sec/write counter, which describes the amount of time a write operation takes on a given disk drive.

► Disk latency

Figure 23-21 shows the writes are experiencing the most delay, but reads (which are not shown here for clarity) are not much better. Average disk sec/write are 23 ms while the maximum is 212 ms. The maximum is an extremely long period because we want to keep disk operations below 20 ms to 25 ms.

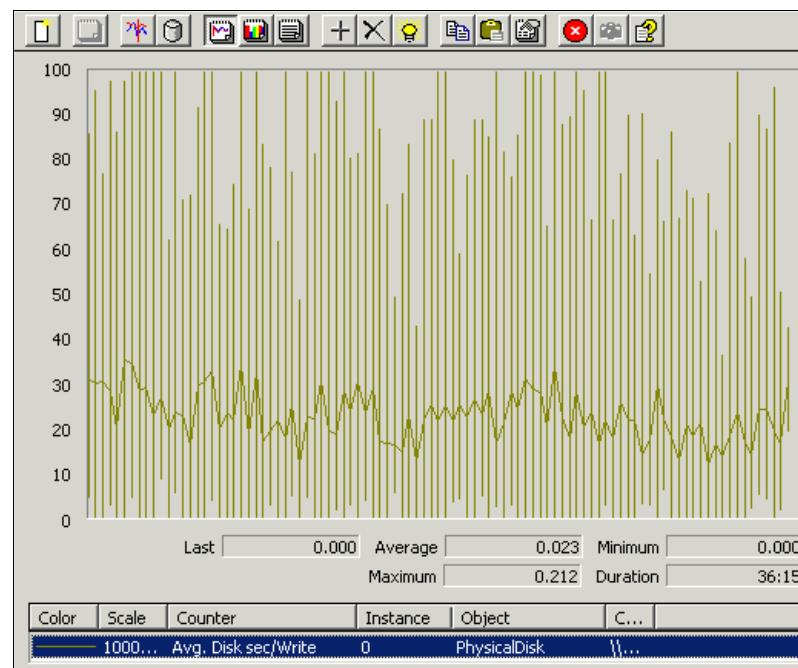


Figure 23-21 Average disk sec/write counter

The disk controller has been verified to use Write-Back caching as would be suggested, but is configured with an 8 KB stripe size, forcing two physical disk I/Os each time a 16 KB request is issued. For the few cases where the request is greater than 24 KB, three disk I/Os occur.

By looking at the distribution of the Average disk bytes/write counter in Figure 23-20 on page 769, we can estimate that about 2.3 to 2.5 physical disk I/Os occur each time the software requests a logical I/O. Adjusting to a larger stripe size reduces this to something closer to one disk I/O per request.

Looking at Figure 23-20 on page 769, it is clear the disk subsystem is performing regularly over 800 I/Os per second. However, remember this is the logical I/O rate (the I/O rate requested by the application). Because the stripe size is incorrect, the disks are doing about 2.3 real I/Os for each application request. With this 8 KB stripe size and using the array size of 13 disks, the disks are really doing about  $2.3 * 626$  or 1439 I/Os per array, and  $1439/13 = 110$  I/Os per disk. Given that this array uses older 10k RPM drives, this rate is pretty high and explains the long latency.

So, with the stripe size changed to 32 KB, the disk bottleneck will improve and throughput will increase. By switching to 32 KB, you effectively double disk throughput because each request generates, on average, one disk I/O (instead of 2.3 with the 8 KB stripe). System performance should improve about 50% to 60% in this case.

**Rule of thumb:** Each time you double disk I/O throughput on a disk bottlenecked application, you can see about a 50% improvement in system throughput assuming no other bottlenecks are encountered.

The disks are performing about 1400 I/Os per second (with higher server throughput) and the 13 disks will bottleneck before the server hits maximum performance (100% CPU).

- ▶ Number of disks, RPM, and RAID levels

Given that the server CPUs are running at about 42% to 44%, you could get about double the current performance by adding more drives, or using newer, faster drives that can handle a greater I/O rate. Further, changing from RAID-5 to RAID-10 can help in cases where random writes are a significant component of the workload.

For example, at 1400 I/Os and 70% CPU utilization (accounting for the 50-60% increase due to the stripe size modification), you need to increase system performance by 30% to reach 100% CPU. Thus, you need  $1400 * 1.3 = 1680$  I/Os per second. For the server to run at 100% CPU utilization with acceptable response times, you must add drives so that each drive does no more than about 100 I/Os per second. This translates into  $1680 \text{ I/Os} / 100 \text{ I/sec/disk} = 17$  RAID-5 disks.

Thus, the optimal RAID-5 solution is to configure 17 disks. Moving to 15K RPM disks could reduce this by ~25%, to ~14 disks. Alternatively, because

this is a write-intensive workload, RAID-10 could be used to reduce the disks by up to half, assuming the disk size is still sufficient.

#### 23.4.4 Summary of Case 3

In this case study, we find that the server's CPUs are underutilized. The Page device is being consumed doing memory-mapped file activity, and may need a more capable array to increase performance. Most notably, the data disk is showing high latencies and indicates that more drives and drive tuning may be needed to optimize the system. With these changes, it can be possible to more than double the performance of this server with a relatively small investment.

# Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

## IBM Redbooks publications

For information about ordering these publications, see “How to get IBM Redbooks publications” on page 783. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *Application Switching with Nortel Networks Layer 2-7 Gigabit Ethernet Switch Module for IBM BladeCenter*, REDP-3589
- ▶ *DB2 UDB V7.1 Performance Tuning Guide*, SG24-6012
- ▶ *DB2 UDB V8.2 on the Windows Environment*, SG24-7102
- ▶ *Domino 7 Performance Tuning Best Practices to Get the Most Out of Your Domino Infrastructure*, REDP-4182
- ▶ *IBM eServer i5 and iSeries System Handbook: IBM i5/OS Version 5 Release 3 October 2004*, GA19-5486
- ▶ *IBM eServer xSeries Clustering Planning Guide*, SG24-5845
- ▶ *IBM System Storage Solutions Handbook*, SG24-5250
- ▶ *IBM System x3950 M2 and x3850 M2 Solution Assurance Product Review Guide*, REDP-4363
- ▶ *IBM XIV Storage System: Concepts, Architecture, and Usage*, SG24-7659
- ▶ *Implementing Cisco InfiniBand on IBM BladeCenter*, REDP-3949
- ▶ *Implementing IBM Director 5.20*, SG24-6188
- ▶ *Implementing VMware ESX Server 2.1 with IBM TotalStorage FASTt*, SG24-6434
- ▶ *Implementing Windows Terminal Server and Citrix MetaFrame on IBM eServer xSeries Servers*, REDP-3629
- ▶ *Introducing IBM TotalStorage FASTt EXP100 with SATA Disks*, REDP-3794
- ▶ *Introducing Windows Server x64 on IBM eServer xSeries Servers*, REDP-3982
- ▶ *Netfinity and Domino R5.0 Integration Guide*, SG24-5313

- ▶ *Planning and Installing the IBM eServer X3 Architecture Servers*, SG24-6797
- ▶ *Planning, Installing, and Managing the IBM System x3950 M2*, SG24-7630
- ▶ *Running the Linux 2.4 Kernel on IBM eServer xSeries Servers*, REDP-0121
- ▶ *ServeRAID Adapter Quick Reference*, TIPS0054
- ▶ *SQL Server 2005 on the IBM eServer xSeries 460 Enterprise Server*, REDP-4093
- ▶ *The Green Data Center: Steps for the Journey*, REDP-4413
- ▶ *Tuning IBM System x Servers for Performance*, SG24-5287
- ▶ *Understanding IBM eServer xSeries Benchmarks*, REDP-3957
- ▶ *Using iSCSI Solutions' Planning and Implementation*, SG24-6291
- ▶ *Virtualization on the IBM System x3950 Server*, SG24-7190
- ▶ *VMware ESX Server: Scale Up or Scale Out?*, REDP-3953

## Other publications

- ▶ *Windows Management Instrumentation (WMI)*, New Riders, by Matthew Lavy and Ashley Meggitt, ISBN 1578702607

## Online resources

The following Web sites are referenced in the book.

### IBM products and services

- ▶ Active Energy Manager  
<http://www.ibm.com/systems/management/director/plugins/actengmgr/index.html>
- ▶ Green IT Energy efficiency solutions  
[http://ibm.com/systems/optimizeit/cost\\_efficiency/energy\\_efficiency/](http://ibm.com/systems/optimizeit/cost_efficiency/energy_efficiency/)
- ▶ Green technology services  
[http://ibm.com/systems/optimizeit/cost\\_efficiency/energy\\_efficiency/services.html](http://ibm.com/systems/optimizeit/cost_efficiency/energy_efficiency/services.html)
- ▶ Press release: IBM BladeCenter Systems Up to 30 Percent More Energy Efficient Than Comparable HP Blades  
<http://www.ibm.com/press/us/en/pressrelease/20633.wss>

- ▶ IBM Cool Blue Technology  
<http://www.backhomeproductions.net/ibm/ftp/coolblue/flash/>
- ▶ IBM Linux Technology Center  
<http://www.ibm.com/linux/ltc/>
- ▶ IBM Performance White Papers  
<http://www.ibm.com/servers/eserver/xseries/benchmarks/related.html>
- ▶ IBM System x and BladeCenter Power Configurator  
<http://www.ibm.com/systems/bladecenter/powerconfig/>
- ▶ IBM Performance Benchmarks for BladeCenter servers  
<http://www.ibm.com/systems/bladecenter/resources/benchmarks/>
- ▶ IBM Power Systems Community  
<http://www.ibm.com/systems/p/community/>
- ▶ IBM System x benchmarks  
<http://www.ibm.com/systems/x/resources/benchmarks/>
- ▶ IBM xRef Reference Sheets  
<http://www.redbooks.ibm.com/xref>
- ▶ IBM System Storage and TotalStorage products  
<http://ibm.com/systems/storage/product/interop.html>

## **IBM support Web sites**

- ▶ IBM support home  
<http://ibm.com/support>
- ▶ IBM ServerProven  
<http://www.ibm.com/servers/eserver/serverproven/compat/us/>
- ▶ Racks and power solutions  
<http://www.ibm.com/servers/eserver/xseries/storage/rack.html>
- ▶ nmon for AIX & Linux Performance Monitoring  
<http://www-941.haw.ibm.com/collaboration/wiki/display/WikiPtype/nmon>
- ▶ nmonanalyser documentations  
<http://www-941.ibm.com/collaboration/wiki/display/WikiPtype/nmonanalyser>
- ▶ IBM UpdateXpress  
<http://www.ibm.com/support/docview.wss?uid=psg1MIGR-53046>

- ▶ IBM xSeries Performance Logger for Linux  
<http://www.ibm.com/support/docview.wss?uid=psglMIGR-64369>
- ▶ IBM Chipkill Memory  
<http://www.ibm.com/systems/support/supportsite.wss/docdisplay?brandid=5000008&lnocid=MCGN-46AMQP>

## **Intel Web sites**

- ▶ Intel Platform Memory  
<http://developer.intel.com/technology/memory/>
- ▶ Intel Virtualization Technology for Directed I/O - Architecture Specification  
[http://download.intel.com/technology/computing/vptech/Intel\(r\)\\_VT\\_for\\_Direct\\_Io.pdf](http://download.intel.com/technology/computing/vptech/Intel(r)_VT_for_Direct_Io.pdf)
- ▶ White paper: Multi-core and Linux Kernel  
<http://software.intel.com/sites/oss/pdf/mclinix.pdf>
- ▶ Technology brief: Intel Virtualization Technology for Connectivity  
[http://softwarecommunity.intel.com/isn/downloads/virtualization/pdfs/20137\\_1ad\\_vtc\\_tech\\_brief\\_r04.pdf](http://softwarecommunity.intel.com/isn/downloads/virtualization/pdfs/20137_1ad_vtc_tech_brief_r04.pdf)
- ▶ Intel Launches vConsolidate to Promote Virtualization Technology and Multi-Core Application Usage  
<http://www.intel.com/pressroom/archive/releases/20070417gloc1.htm>
- ▶ Intel 5400 Chipset  
<http://www.intel.com/products/server/chipsets/5400/5400-overview.htm>
- ▶ Intel Technology Journal: Intel Virtualization Technology: Hardware support for efficient processor virtualization  
<http://www.intel.com/technology/itj/2006/v10i3/1-hardware/8-virtualization-future.htm>
- ▶ Intel Technology Journal: Intel Virtualization Technology: Redefining server performance characterization for virtualization benchmarking  
<http://www.intel.com/technology/itj/2006/v10i3/7-benchmarking/6-vconsolidate.htm>
- ▶ Intel QuickPath Technology  
<http://www.intel.com/technology/quickpath>
- ▶ VTune Performance Analyzer for Windows evaluation download  
<https://registrationcenter.intel.com/EvalCenter/EvalForm.aspx?ProductID=585>

## **AMD Web sites**

- ▶ AMD Virtualization  
[http://www.amd.com/us-en/0,,3715\\_15781,00.html](http://www.amd.com/us-en/0,,3715_15781,00.html)
- ▶ White paper: Virtualizing Server Workload  
[http://www.amd.com/us-en/assets/content\\_type/DownloadableAssets/AMD\\_WP\\_Virtualizing\\_Server\\_Workloads-PID.pdf](http://www.amd.com/us-en/assets/content_type/DownloadableAssets/AMD_WP_Virtualizing_Server_Workloads-PID.pdf)
- ▶ Paper: Estimating Total Power Consumption by Servers in the U.S. and the World  
<http://enterprise.amd.com/Downloads/svrpwrusecompletesfinal.pdf>

## **Microsoft Web sites**

- ▶ Collecting User-Mode Dumps  
[http://msdn.microsoft.com/en-us/library/bb787181\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/bb787181(VS.85).aspx)
- ▶ Windows Performance Analyzer (WPA)  
<http://msdn.microsoft.com/en-us/library/cc305187.aspx>
- ▶ Managing Memory-Mapped Files in Win32  
[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dng\\_enlib/html/msdn\\_manamemo.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dng_enlib/html/msdn_manamemo.asp)
- ▶ Performance Counter Classes  
[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wmi\\_sdk/wmi/performance\\_counter\\_classes.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wmi_sdk/wmi/performance_counter_classes.asp)
- ▶ Using AWE  
<http://msdn2.microsoft.com/en-us/library/ms175581.aspx>
- ▶ TCP/IP and NetBT configuration parameters for Windows 2000 or Windows NT  
<http://support.microsoft.com/?kbid=120642>
- ▶ Server service configuration and tuning  
<http://support.microsoft.com/?kbid=128167>
- ▶ How to stop the NT Executive from paging to disk  
<http://support.microsoft.com/?kbid=184419>
- ▶ Description of Windows 2000 and Windows Server 2003 TCP Features  
<http://support.microsoft.com/?kbid=224829>

- ▶ Terminal Server Client Connections and Logon Limited by MaxWorkItem and MaxMpxCt Values  
<http://support.microsoft.com/?kbid=232476>
- ▶ How to overcome the 4,095 MB paging file size limit in Windows  
<http://support.microsoft.com/?kbid=237740>
- ▶ How to Install and Use the Interrupt-Affinity Filter Tool  
<http://support.microsoft.com/?kbid=252867>
- ▶ Windows 2000 Does Not Use Configured TCPWindowSize Registry Parameter When Accepting a Connection  
<http://support.microsoft.com/?kbid=263088>
- ▶ MaxMpxCt and MaxCmnds limits in Windows 2000  
<http://support.microsoft.com/?kbid=271148>
- ▶ About Cache Manager in Windows Server 2003  
<http://support.microsoft.com/?kbid=837331>
- ▶ PerfMon: High Number of Pages/Sec Not Necessarily Low Memory  
<http://support.microsoft.com/default.aspx?scid=kb;EN-US;q139609>
- ▶ Intel Physical Addressing Extensions (PAE) in Windows 2000  
<http://support.microsoft.com/kb/268363>
- ▶ Large memory support is available in Windows Server 2003 and in Windows 2000  
<http://support.microsoft.com/kb/283037>
- ▶ A description of the 4 GB RAM Tuning feature and the Physical Address Extension parameter  
<http://support.microsoft.com/kb/291988>
- ▶ Use of the /3GB switch in Exchange Server 2003 on a Windows Server 2003-based system  
<http://support.microsoft.com/kb/823440>
- ▶ The Microsoft Windows Server 2003 Scalable Networking Pack release  
<http://support.microsoft.com/kb/912222>
- ▶ Information about the TCP Chimney Offload, Receive Side Scaling, and Network Direct Memory Access features in Windows Server 2008  
<http://support.microsoft.com/kb/951037>

- ▶ Performance and Reliability Monitoring Step-by-Step Guide for Windows Server 2008  
<http://technet.microsoft.com/en-us/library/cc771692.aspx>
- ▶ Windows Sysinternals  
<http://technet.microsoft.com/en-us/sysinternals>
- ▶ Microsoft Windows Server 2003 TCP/IP Implementation Details  
<http://www.microsoft.com/downloads/details.aspx?FamilyID=06c60bfe-4d37-4f50-8587-8b68d32fa6ee&displaylang=en>
- ▶ Windows Server 2003 Resource Kit Tools  
<http://www.microsoft.com/downloads/details.aspx?familyid=9d467a69-57ff-4ae7-96ee-b18c4790cff&displaylang=en>
- ▶ Windows Server 2003 Deployment Guide  
<http://www.microsoft.com/resources/documentation/WindowsServ/2003/all/deployguide/en-us/46656.asp>
- ▶ Microsoft Windows Server 2003 TCP/IP Implementation Details  
<http://www.microsoft.com/technet/prodtechnol/windowsserver2003/technologies/networking/tcpip03.mspx>
- ▶ Receive-Side Scaling Enhancements in Windows Server 2008  
[http://www.microsoft.com/whdc/device/network/NDIS\\_RSS.mspx](http://www.microsoft.com/whdc/device/network/NDIS_RSS.mspx)
- ▶ Processor Power Management™ in Windows Vista and Windows Server 2008  
<http://www.microsoft.com/whdc/system/pnppwr/powermgmt/ProcPowerMgmt.mspx>
- ▶ Interrupt-Affinity Policy Tool  
<http://www.microsoft.com/whdc/system/sysperf/IntPolicy.mspx>
- ▶ Performance Tuning Guidelines for Windows Server 2008  
[http://www.microsoft.com/whdc/system/sysperf/Perf\\_tun\\_srv.mspx](http://www.microsoft.com/whdc/system/sysperf/Perf_tun_srv.mspx)
- ▶ Windows Server 2003 x64 Editions Deployment Scenarios  
<http://www.microsoft.com/windowsserver2003/64bit/x64/deploy.mspx>
- ▶ What's New in Windows Server 2003 R2  
<http://www.microsoft.com/windowsserver2003/R2/whatsnewinr2.mspx>
- ▶ Comparison of Windows Server 2003 Editions  
<http://www.microsoft.com/windowsserver2003/evaluation/features/comparefeatures.mspx>

- ▶ Performance Tuning Guidelines for Windows Server 2003  
<http://www.microsoft.com/windowsserver2003/evaluation/performance/tuning.mspx>
- ▶ Benefits of Microsoft Windows 2003 x64 Editions  
<http://www.microsoft.com/windowsserver2003/techinfo/overview/x64benefits.mspx>
- ▶ Windows Server 2008: Compare Technical Features and Specifications  
<http://www.microsoft.com/windowsserver2008/en/us/compare-specs.aspx>
- ▶ Windows Server 2008 Overview of Editions  
<http://www.microsoft.com/windowsserver2008/en/us/editions-overview.aspx>

## **Linux Web sites**

- ▶ SYSSTAT Utilities home page  
<http://perso.wanadoo.fr/sebastien.godard/>

## **VMware Web sites**

- ▶ VMware VMmark  
<http://www.vmware.com/products/vmmark/>
- ▶ Download VMware VMmark  
<http://www.vmware.com/download/vmmark/>
- ▶ VMware Infrastructure 3 SDK - Performance Counters  
[http://www.vmware.com/files/pdf/technote\\_PerformanceCounters.pdf](http://www.vmware.com/files/pdf/technote_PerformanceCounters.pdf)
- ▶ Performance Best Practices and Benchmarking Guidelines - VMware Infrastructure 3 version 3.5 with ESX 3.5, ESXi 3.5, and VirtualCenter 2.5  
[http://www.vmware.com/pdf/VI3.5\\_Performance.pdf](http://www.vmware.com/pdf/VI3.5_Performance.pdf)
- ▶ Performance Tuning Best Practices for ESX Server 3  
[http://www.vmware.com/pdf/vi\\_performance\\_tuning.pdf](http://www.vmware.com/pdf/vi_performance_tuning.pdf)
- ▶ Timekeeping in VMware Virtual Machines  
[http://www.vmware.com/pdf/vmware\\_timekeeping.pdf](http://www.vmware.com/pdf/vmware_timekeeping.pdf)
- ▶ ESX Configuration Guide - ESX 4.0 and vCenter Server 4.0  
[http://www.vmware.com/pdf/vsphere4/r40/vsp\\_40\\_esx\\_server\\_config.pdf](http://www.vmware.com/pdf/vsphere4/r40/vsp_40_esx_server_config.pdf)

- ▶ vSphere Resource Management Guide - ESX 4.0, ESXi 4.0 and vCenter Server 4.0  
[http://www.vmware.com/pdf/vsphere4/r40/vsp\\_40\\_resource\\_mgmt.pdf](http://www.vmware.com/pdf/vsphere4/r40/vsp_40_resource_mgmt.pdf)
- ▶ VMware Compatibility Guide  
<http://www.vmware.com/resources/compatibility/>
- ▶ Installing and Configuring NTP on VMware ESX Server  
[http://www.vmware.com/support/kb/enduser/std\\_adp.php?p\\_faqid=1339](http://www.vmware.com/support/kb/enduser/std_adp.php?p_faqid=1339)
- ▶ VMware Documentation  
<http://www.vmware.com/support/pubs/>
- ▶ ESX Server 2.x Documentation  
[http://www.vmware.com/support/pubs/esx\\_pubs.html](http://www.vmware.com/support/pubs/esx_pubs.html)
- ▶ VMware Technical Resources  
[http://www.vmware.com/vmtn/resources/esx\\_resources.html](http://www.vmware.com/vmtn/resources/esx_resources.html)

## Other sites

- ▶ Hypertransport Consortium  
<http://www.hypertransport.org/>
- ▶ Understanding the detailed Architecture of the AMD 64-bit Core  
[http://chip-architect.com/news/2003\\_09\\_21\\_Detailed\\_Architecture\\_of\\_AMDs\\_64bit\\_Core.html#3.18](http://chip-architect.com/news/2003_09_21_Detailed_Architecture_of_AMDs_64bit_Core.html#3.18)
- ▶ SPEC Virtualization Committee  
<http://spec.org/specvirtualization/>
- ▶ Advanced Configuration & Power Interface specification  
<http://www.acpi.info/spec.htm>
- ▶ Troubleshooting Cisco Catalyst Switches to NIC Compatibility Issues  
[http://www.cisco.com/en/US/products/hw/switches/ps700/products\\_tech\\_note09186a00800a7af0.shtml](http://www.cisco.com/en/US/products/hw/switches/ps700/products_tech_note09186a00800a7af0.shtml)
- ▶ STREAM: Sustainable Memory Bandwidth in High Performance Computers  
<http://www.cs.virginia.edu/stream>
- ▶ EPA Report to Congress on Server and Data Center Energy Efficiency  
[http://www.energystar.gov/index.cfm?fuseaction=prod\\_development.server\\_efficiency#epa](http://www.energystar.gov/index.cfm?fuseaction=prod_development.server_efficiency#epa)

- ▶ FLUENT Benchmarks  
<http://www.fluent.com/software/fluent/f15bench>
- ▶ Eaton products  
[http://www.powerware.com/ibm/US/Products/UPS\\_systemx.asp](http://www.powerware.com/ibm/US/Products/UPS_systemx.asp)
- ▶ vConsolidate performance and power on IBM and HP quad-core Intel processor-based servers  
<http://www.principledtechnologies.com/Clients/Reports/IBM/IBMvCon1p0808.pdf>
- ▶ High Performance Buildings: Data Centers - Server Power Supplies  
[http://hightech.lbl.gov/documents/PS/Final\\_PS\\_Report.pdf](http://hightech.lbl.gov/documents/PS/Final_PS_Report.pdf)
- ▶ The Raised Floor  
<http://theraisedfloor.typepad.com/blog/>
- ▶ T/TCP -- TCP Extensions for Transactions Functional Specification  
<http://www.ietf.org/rfc/rfc1644.txt>
- ▶ Great Internet Mersenne Prime Search  
<http://www.mersenne.org>
- ▶ Filesystem Hierarchy Standard  
<http://www.pathname.com/fhs>
- ▶ Serial ATA International Organization  
<http://www.sata-io.org>
- ▶ SPEC CPU2006  
<http://www.spec.org/cpu2006/>
- ▶ SPECjAppServer2004  
<http://www.spec.org/jAppServer2004/>
- ▶ SPECjbb2005  
<http://www.spec.org/jbb2005/>
- ▶ SPECpower\_ssj2008  
[http://www.spec.org/power\\_ssj2008/](http://www.spec.org/power_ssj2008/)
- ▶ SPECpower results for IBM System x3200 M2  
[http://www.spec.org/power\\_ssj2008/results/res2008q2/power\\_ssj2008-20080506-00050.html](http://www.spec.org/power_ssj2008/results/res2008q2/power_ssj2008-20080506-00050.html)
- ▶ SPECweb2005  
<http://www.spec.org/web2005/>

- ▶ Tolly Report - Enhanced Network Performance with Microsoft Windows Vista and Windows Server 2008  
<http://www.tolly.com/docdetail.aspx?docnumber=208306>
- ▶ The Linpack Benchmark  
<http://www.top500.org/project/linpack>
- ▶ Top Crunch benchmark results  
[http://www.topcrunch.org/benchmark\\_results\\_search.sfe](http://www.topcrunch.org/benchmark_results_search.sfe)
- ▶ TPC-C benchmark  
<http://www.tpc.org/tpcc/>
- ▶ TPC-E benchmark  
<http://www.tpc.org/tpce/>
- ▶ TPC-H benchmark  
<http://www.tpc.org/tpch>
- ▶ The Uptime Institute White Papers  
<http://www.uptimeinstitute.org/whitepapers>

## How to get IBM Redbooks publications

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks, at this Web site:

[ibm.com/redbooks](http://ibm.com/redbooks)

## Help from IBM

IBM Support and downloads

[ibm.com/support](http://ibm.com/support)

IBM Global Services

[ibm.com/services](http://ibm.com/services)



# Abbreviations and acronyms

<b>AC</b>	alternating current	<b>BTU</b>	British Thermal Unit
<b>ACK</b>	acknowledgment	<b>CAS</b>	column address strobe
<b>ACPI</b>	advanced control and power interface	<b>CD</b>	compact disk
<b>AD</b>	Active Directory	<b>CD-ROM</b>	compact disc read only memory
<b>AEM</b>	Active Energy Manager	<b>CDM</b>	Cedar Mill
<b>ALU</b>	arithmetic logic unit	<b>CEE</b>	Converged Enhanced Ethernet
<b>AMB</b>	Advanced Memory Buffer	<b>CFQ</b>	Completely Fair Queuing
<b>AMD</b>	Advanced Micro Devices	<b>CIFS</b>	Common Internet File System
<b>ANSI</b>	American National Standards Institute	<b>CL</b>	CAS Latency
<b>API</b>	application programming interface	<b>CMT</b>	Center for Microsoft Technologies
<b>APIC</b>	Advanced Programmable Interrupt Controller	<b>COM</b>	Component Object Model
<b>ARC</b>	Advanced Risc Computing	<b>CPI</b>	Cycles Per Instruction
<b>AS</b>	Australian Standards	<b>CPU</b>	central processing unit
<b>ASCII</b>	American Standard Code for Information Interchange™	<b>CRAC</b>	computer room air conditioner
<b>ASM</b>	Advanced System Management	<b>CRC</b>	cyclic redundancy check
<b>ATA</b>	AT attachment	<b>CSG</b>	Chip Select Group
<b>ATAPI</b>	ATA packet interface	<b>CSI</b>	Common System Interface
<b>ATM</b>	asynchronous transfer mode	<b>CSV</b>	comma separated variable
<b>AVC</b>	Access Vector Cache	<b>CTCP</b>	Compound TCP
<b>AWE</b>	Address Windowing Extensions	<b>DAS</b>	Direct Attached Storage
<b>BI</b>	Business Intelligence	<b>DBS</b>	demand-based switching
<b>BIOS</b>	basic input output system	<b>DC</b>	domain controller
<b>BLG</b>	binary log file	<b>DCU</b>	data cache unit
<b>BMC</b>	Baseboard Management Controller	<b>DDPM</b>	Dual Dynamic Power Management
<b>BPA</b>	Best Practices Analyzer	<b>DDR</b>	Double Data Rate
<b>BTB</b>	branch target buffer	<b>DEP</b>	Data Execution Prevention
		<b>DFS</b>	Distributed File System
		<b>DHCP</b>	Dynamic Host Configuration Protocol

<b>DIMM</b>	dual inline memory module	<b>FRS</b>	File Replication service
<b>DLL</b>	dynamic link library	<b>FS</b>	fast skinny
<b>DMA</b>	direct memory access	<b>FSB</b>	front-side bus
<b>DMZ</b>	demilitarized zone	<b>FTP</b>	File Transfer Protocol
<b>DNS</b>	Domain Name System	<b>GB</b>	gigabyte
<b>DOS</b>	disk operating system	<b>GIF</b>	graphic interchange format
<b>DP</b>	dual processor	<b>GPR</b>	general purpose register
<b>DPC</b>	deferred procedure call	<b>GPT</b>	GUID partition table
<b>DRAM</b>	dynamic random access memory	<b>GRUB</b>	Grand Unified Bootloader
<b>DRS</b>	Distributed Resource Scheduler	<b>GUI</b>	graphical user interface
<b>EB</b>	Exabytes	<b>HA</b>	high availability
<b>ECC</b>	error checking and correcting	<b>HAM</b>	hot-add memory
<b>ECN</b>	Explicit Congestion Notification	<b>HBA</b>	host bus adapter
<b>EFI</b>	Extensible Firmware Interface	<b>HDD</b>	hard disk drive
<b>EIDE</b>	enhanced IDE	<b>HE</b>	high end
<b>EJB</b>	Enterprise Java Beans	<b>HPC</b>	high performance computing
<b>EL</b>	execution layer	<b>HPMA</b>	High Performance Memory Array
<b>EMEA</b>	Europe, Middle East, Africa	<b>HT</b>	Hyper-Threading
<b>EPT</b>	Extended Page Tables	<b>HTML</b>	Hypertext Markup Language
<b>ERP</b>	enterprise resource planning	<b>HTTP</b>	Hypertext Transfer Protocol
<b>ESCON</b>	enterprise systems connection	<b>HW</b>	hardware
<b>ESI</b>	Enterprise Southbridge Interface	<b>I/O</b>	input/output
<b>EXP</b>	expansion	<b>I/OAT</b>	I/O Acceleration Technology
<b>FAMM</b>	Full Array Memory Mirroring	<b>IBM</b>	International Business Machines
<b>FAQ</b>	frequently asked questions	<b>ICMP</b>	Internet control message protocol
<b>FAT</b>	file allocation table	<b>ID</b>	identifier
<b>FB-DIMM</b>	Fully Buffered DIMMs	<b>IDE</b>	integrated drive electronics
<b>FBDIMM</b>	Fully Buffered DIMM	<b>IDF</b>	Intel Developer Forum
<b>FC</b>	Fibre Channel	<b>IEEE</b>	Institute of Electrical and Electronics Engineers
<b>FDDI</b>	fiber distributed data interface	<b>IIS</b>	Internet Information Server
<b>FIFO</b>	first in first out	<b>IM</b>	instant messaging
<b>FP</b>	floating point		

<b>IOAT</b>	I/O Acceleration Technology	<b>MAN</b>	metropolitan area network
<b>IOPS</b>	I/O operations per second	<b>MB</b>	megabyte
<b>IP</b>	Internet Protocol	<b>MBR</b>	Master Boot Record
<b>IPI</b>	intelligent peripheral interface	<b>MCH</b>	memory controller hub
<b>IPSEC</b>	IP Security	<b>MESI</b>	modified exclusive shared invalid
<b>IPX</b>	Internetwork Packet eXchange™	<b>MIOC</b>	Memory and I/O Controller
<b>IRQ</b>	interrupt request	<b>MMC</b>	Microsoft Management Console
<b>ISV</b>	independent software vendor	<b>MMU</b>	memory management unit
<b>IT</b>	information technology	<b>MOESI</b>	modified owner exclusive shared invalid
<b>ITSO</b>	International Technical Support Organization	<b>MP</b>	multiprocessor
<b>JDBC</b>	Java Database Connectivity	<b>MSDN</b>	Microsoft Developer Network
<b>JIT</b>	just in time	<b>MSI-X</b>	Extended Message Signaled Interrupts
<b>JSC</b>	Joint Solutions Center Java Statistical Classes	<b>MSS</b>	maximum segment size
<b>JSP</b>	JavaServer™ Pages	<b>MTU</b>	maximum transmission unit
<b>JVM</b>	Java Virtual Machine	<b>NAS</b>	network attached storage
<b>KB</b>	kilobyte	<b>NCQ</b>	Native Command Queuing
<b>KCC</b>	Knowledge Consistency Checker	<b>NDIS</b>	network driver interface specification
<b>KDE</b>	K Desktop Environment	<b>NFS</b>	network file system
<b>KVM</b>	keyboard video mouse	<b>NGN</b>	next-generation network
<b>LAN</b>	local area network	<b>NIC</b>	network interface card
<b>LAN/WAN</b>		<b>NL</b>	near-line
<b>LBA</b>	Logical Block Address	<b>NLB</b>	Network Load Balancing
<b>LDAP</b>	Lightweight Directory Access Protocol	<b>NMP</b>	Native Multipathing Plugin
<b>LDS</b>	Lightweight Directory Services	<b>NPT</b>	Nested Page Tables
<b>LPT</b>	line printer	<b>NPTL</b>	Native POSIX Thread Library
<b>LSD</b>	Loop Stream Detector	<b>NTC</b>	Network Transformation Center
<b>LSM</b>	Linux Security Modules	<b>NTFS</b>	NT File System
<b>LTC</b>	Linux Technology Center	<b>NTP</b>	Network Time Protocol
<b>LUN</b>	logical unit number	<b>NUMA</b>	Non-Uniform Memory Access
<b>LVD</b>	Low Voltage Differential	<b>OEM</b>	other equipment manufacturer
<b>MAC</b>	media access control	<b>OLTP</b>	online transaction processing

<b>OOB</b>	out of band	<b>RAS</b>	remote access services; row address strobe
<b>OS</b>	operating system	<b>RBS</b>	redundant bit steering
<b>OSI</b>	Open Systems Interconnect	<b>RDBMS</b>	relational database management system
<b>PAE</b>	Physical Address Extension		
<b>PATA</b>	parallel ATA	<b>RDC</b>	Remote Desktop Connection
<b>PC</b>	personal computer	<b>RDM</b>	Remote Deployment Manager
<b>PCI</b>	Peripheral Component Interconnect	<b>RDP</b>	Remote Desktop Protocol
<b>PCI-E</b>	PCI Express	<b>RDS</b>	Reliable Datagram Sockets
<b>PCPU</b>	physical CPU	<b>RFC</b>	request for comments
<b>PCU</b>	Power Control Unit	<b>RHEL</b>	Red Hat Enterprise Linux
<b>PDC</b>	primary domain controller	<b>RODC</b>	Read-only domain controller
<b>PDU</b>	power distribution unit	<b>ROM</b>	read-only memory
<b>PID</b>	process ID	<b>RPM</b>	revolutions per minute
<b>PMTU</b>	Path Maximum Transmission Unit	<b>RPO</b>	rotational positioning optimization
<b>POST</b>	power-on self test	<b>RSA</b>	Remote Supervisor Adapter
<b>PPM</b>	processor power management	<b>RSS</b>	Receive-side scaling
<b>PSA</b>	POSIX semantic agent	<b>RT</b>	real time
<b>PSSC</b>	Products and Solutions Support Center	<b>RTM</b>	release to manufacturing
<b>PTE</b>	Page Table Entry	<b>RTT</b>	round trip time
<b>PUE</b>	Power Usage Effectiveness	<b>RVI</b>	Rapid Virtualization Indexing
<b>PVSCSI</b>	Paravirtualized SCSI	<b>RWDC</b>	Read-Write Domain Controllers
<b>PXE</b>	Preboot eXecution Environment	<b>SAN</b>	storage area network
<b>QDR</b>	quad data rate	<b>SAS</b>	Serial Attached SCSI
<b>QPI</b>	QuickPath Interconnect	<b>SATA</b>	Serial ATA
<b>QSTAT</b>		<b>SATP</b>	Storage Array Type Plugin
<b>QUED</b>		<b>SBCCS</b>	single byte command code set
<b>RAID</b>	redundant array of independent disks	<b>SCSI</b>	Small Computer System Interface
<b>RAID-M</b>	redundant array of inexpensive DRAMs for memory	<b>SCTP</b>	Stream Control Transmission Protocol
<b>RAM</b>	random access memory	<b>SDRAM</b>	static dynamic RAM
		<b>SEC</b>	single edge connector
		<b>SIMD</b>	single instruction multiple data

<b>SLAT</b>	Second Level Address Translation	<b>TCP/IP</b>	Transmission Control Protocol/Internet Protocol
<b>SLES</b>	SUSE Linux Enterprise Server	<b>TCQ</b>	tagged command queueing
<b>SMB</b>	server message block	<b>TLB</b>	thermal design power
<b>SME</b>	subject matter expert	<b>TOE</b>	translation lookaside buffer
<b>SMI</b>	synchronous memory interface	<b>TPC</b>	TCP offload engine
<b>SMP</b>	symmetric multiprocessing	<b>TSO</b>	Transaction Processing Performance Council
<b>SMS</b>	System Managed Space	<b>UDDI</b>	TCP Segmentation Offload
<b>SMT</b>	Simultaneous Multi-Threading		Universal Description, Discovery and Integration
<b>SNMP</b>	Simple Network Management Protocol	<b>UDP</b>	user datagram protocol
<b>SNP</b>	Scalable Networking Pack	<b>UEFI</b>	Unified Extensible Firmware Interface
<b>SPEC</b>	Standard Performance Evaluation Corporation	<b>UID</b>	unique ID
<b>SQL</b>	Structured Query Language	<b>UPS</b>	uninterruptible power supply
<b>SRAT</b>	Static Resource Allocation Table	<b>URL</b>	Uniform Resource Locator
<b>SRQ</b>	System Request Queue	<b>USB</b>	universal serial bus
<b>SSA</b>	serial storage architecture	<b>VI</b>	VMware Infrastructure
<b>SSD</b>	solid state drive	<b>VLAN</b>	virtual LAN
<b>SSE</b>	Streaming SIMD Extensions	<b>VM</b>	virtual machine
<b>SSH</b>	Secure Shell	<b>VMCB</b>	virtual machine control block
<b>SSL</b>	Secure Sockets Layer	<b>VMCS</b>	virtual-machine control structure
<b>SSP</b>	Serial SCSI Protocol	<b>VMFS</b>	virtual machine file system
<b>STA</b>	SCSI Trade Association	<b>VMM</b>	Virtual Machine Manager
<b>STG</b>	Server & Technology Group	<b>VMX</b>	Virtual Machine Extensions
<b>STP</b>	SATA Tunneled Protocol	<b>VPID</b>	Virtual Processor Identifiers
<b>SUMO</b>	Sufficiently Uniform Memory Organization	<b>VRM</b>	voltage regulator module
<b>SWAT</b>	Samba Web Administration Tool	<b>VS</b>	Virtual Server
<b>TB</b>	terabyte	<b>VT</b>	Virtualization Technology
<b>TCB</b>	Transport Control Block	<b>WAN</b>	wide area network
<b>TCO</b>	total cost of ownership	<b>WB</b>	write back (cache)
<b>TCP</b>	Transmission Control Protocol	<b>WDS</b>	Windows Deployment Services
		<b>WER</b>	Windows Error Reporting

<b>WINS</b>	Windows Internet Naming Service
<b>WMI</b>	Windows Management Instrumentation
<b>WSRM</b>	Windows System Resource Manager
<b>WSUS</b>	Windows Server Update Services
<b>WT</b>	write through (cache)
<b>WWN</b>	World Wide Name
<b>XML</b>	Extensible Markup Language
<b>XOR</b>	exclusive or

# Index

## Symbols

- /3GB parameter in BOOT.INI 390, 393
- /PAE parameter 392
- /proc 467

## Numerics

- 10 Gigabit Ethernet 313
- 4 GB, more than 207
  - 64-bit 120
  - Linux 474
  - Windows 391
- 40K7547, Passthru card for x3755 115
- 64-bit computing 116–122
  - 64-bit mode 118
  - addressable memory 121
  - AMD Opteron 109
  - applications 119
  - benefits 119
  - definition 117
  - Intel 64 Technology 116
- 8.3 filenames 415

## A

- abstract xxiii
- accelerated memory technology 234
- actions 687
  - disk
    - Linux 738
    - Windows 705
  - memory
    - Linux 732
    - Windows 700
  - network
    - Linux 741
    - Windows 712
  - processor
    - Linux 727
    - Windows 696
- Active Directory server 15
- Active Energy Manager 78
- Active Memory 47
- adapter keying 148

- adding drives 270
- Adjacent Sector Prefetch 130
- advanced ECC memory 212
- Advanced Memory Buffer 192–193

airflow 86

AMD

- See also* Opteron
- AMD64 109, 117
- AMD64 modes 118
- AMD-V 138, 141
- HyperTransport 177–178, 180
- Pacifica 110, 138
- Rapid Virtualization Indexing 142

AMD Opteron

*See* Opteron

analyzing performance 669

Linux 719–741

Windows 691–717

apmd daemon 455

application scalability 168

arptables 455

associativity, cache 123

ATM 294

auditing 418

authentication servers 15

autofs 456

AWE 210

## B

- BaanERP benchmark 39
- background information 667
- backup software 419
- bandwidth, memory 199
- Barcelona 111
- baseline measurements 4, 666
- bcopies 310
- bdflush 476
- benchmarks 6, 31–41
  - BaanERP 39
  - benchmark centers 8
  - Client Benchmark Centers 8
  - energy efficiency 88
  - Fluent 40

industry-standard 32  
Java benchmark 36  
Linpack benchmark 38  
LS-DYNA 40  
Oracle 39  
results 34  
SAP 39  
SPEC CPU2006 37  
SPECjAppServer2004 37  
SPECjbb2005 36  
SPECpower\_ssj2008 38, 88  
SPECweb2005 36  
STREAM 202  
System x 33  
System x Performance Lab 5  
TPC-C 35  
TPC-E 35  
TPC-H 36  
types 32  
understanding 41  
unregulated 32  
vConsolidate 39  
VMmark 38  
workload 32  
binary translation 135  
BIOS  
    memory configuration 232  
    UEFI 170  
Blackford 172  
BladeCenter 14, 82  
block diagrams  
    Dunnington processor 102  
    NUMA (x3950 M2) 162  
    Opteron 180  
    x3755 113  
BOOT.INI  
    /3GB parameter 390, 393  
    /PAE parameter 392  
bottlenecks  
    actions 687  
    disk  
        Linux 738  
        Windows 705  
    memory  
        Linux 732  
        Windows 700  
    network  
        Linux 741  
        Windows 712

processor  
    Linux 727  
    Windows 696  
CPU subsystem  
    Linux 726  
    Windows 692  
data rate 709  
determining 665  
disk subsystem 703  
ESX Server 649  
latent 671  
Linux 719–741  
memory subsystem 697  
network subsystem 295, 707  
Windows 691–717  
worksheets 670  
buffered SDRAMs 188  
busmaster devices 145

## C

cables, SAS 245  
cache  
    associativity 123  
    cache coherency filter 168  
    disk 675  
    effect on performance 127  
    RAID adapter 282  
    write-back versus write-through 281  
capping of power 59  
CAS 200  
CAS latency 201  
case studies 743  
Center for Microsoft Technologies 7  
checksum offload 315  
Chimney Offload 332  
Chipkill 212  
chipsets 157–182  
    AMD 177  
    cache coherency filter 168  
    design 159  
    eight-way configurations 167  
    eX4 175  
    HyperTransport 178  
    Intel 5000 family 171  
    latency 162  
    MESI 165  
    MOESI 166  
    NUMA 161

overview 158  
performance 159  
QuickPath Interconnect 181  
scalability 160  
snoop cycles 165  
SRAT table 163  
chkconfig command 457, 460  
Citrix Presentation Server  
    scale-out 14  
clock speed 94, 128  
Clovertown 98  
Common System Interface  
    *See* QuickPath Interconnect  
communication servers 24  
compatibility mode 118  
Compound TCP 441  
compression  
    NTFS 416  
Converged Enhanced Ethernet 346  
Core microarchitecture 103  
cores, processor 94  
CPU subsystem 93  
    64-bit computing, benefits of 119  
    64-bit mode 118  
    adding processors 696  
    Adjacent Sector Prefetch 130  
    Advanced Digital Media Boost 105  
    Advanced Smart Cache 104  
affinity  
    Linux 726  
    Windows 387  
AMD64 109, 117  
analysis example 766  
Barcelona 111  
bottlenecks 683  
    Linux 727  
    Windows 692  
cache 46, 127  
cache associativity 123  
cache coherency filter 168  
clock speed 94, 128  
Clovertown 98  
comparison 123  
compatibility mode 118  
Core microarchitecture 103  
cores 94  
C-states 58  
DCU Prefetcher 131  
Demand-Based Switching 98  
design 159  
dual core processors 95  
Dual Dynamic Power Management 112  
Dunnington 100–101  
eight-way configurations 167  
EM64T 117  
encryption 119  
energy efficiency 54  
Extended Page Tables 139  
Hardware Prefetcher 131  
hardware scalability 160  
Harpertown 100  
history 94  
HyperTransport 177, 180  
HyperTransport links 114  
IA-32e mode 118  
integrated memory controller 106  
Intel 64 Technology 117  
Intelligent Power Capability 103  
introduction 46  
IP Prefetcher 131  
latency 162  
legacy mode 118  
Linux  
    bottlenecks 724  
    tuning 473  
long mode 118  
Loop Stream Detector 108  
Macrofusion 108  
memory addressing 121  
memory, affect on 170  
MESI 165  
MOESI 166  
naming convention 108  
Nehalem architecture 105  
nested paging 137  
network, effect on 310  
NUMA 161  
Opteron 109  
paging, nested 137  
Passthru card for x3755 113  
performance 122, 159  
performance analysis 683  
Power Control Unit 59  
prefetch 130  
primary counters 674  
P-states 57  
quad-core processors 98  
queue length 695

QuickPath Interconnect 105, 181  
Rapid Virtualization Indexing 142  
replace processors 696  
scaling 129  
Shanghai 112  
shared L3 cache 95  
Simultaneous Multi-Threading 107  
Smart Memory Access 104  
SMP  
    defined 160  
    linear improvement 169  
    Linux 474  
    type of server 169  
SMP scaling 129  
snoop cycles 165  
software scalability 168  
Static Resource Allocation Table 163  
technology 94  
Tigerton 97, 99  
TOE 317  
transition latency 140  
T-states 57  
Tulsa 95  
tuning options 696  
UEFI 170  
upgrade processors 696  
Virtual Processor Identifier 139  
Wide Dynamic Execution 103  
Wolfdale 97  
Woodcrest 96  
    worksheet 683  
cpuspeed 456  
C-states 58  
cups 456  
Cycles Per Instruction 158

## D

daemons 455–459  
    tunable 455  
Data Execution Prevention 209, 357  
database servers 18  
datagrams 295  
DB2  
    processors, number of 130  
    server type 18  
DCU Prefetcher 131  
DDR memory 189  
DDR2 189

AMD Opteron support 109  
eX4 Architecture 232  
DDR3 memory 190  
defrag utility 592  
Demand-Based Switching 98  
device drivers 284  
DHCP servers 27  
DIMMs 185, 232  
direct-attach storage 241  
disk subsystem 237–291  
    *See also* Fibre Channel  
    *See also* RAID levels  
    *See also* SAS  
    *See also* SCSI  
    *See also* Serial ATA  
    *See also* ServeRAID  
    access time calculations 676  
    active data set size 271  
    adapter cache size 282  
    adding drives 269, 705, 739  
    analysis example 769  
    bottlenecks 675  
    cables 245  
    cache size 282  
    calculations 676  
    capacity 269  
    command overhead 273  
    common bottlenecks 677  
    data set size 271  
    data transfer rate 238  
    device drivers 284  
    direct-attach storage 241  
    drive performance 273  
    drives, number of 269  
    EIDE 238, 246  
    energy efficiency 64  
    Ext3 file system 482  
    Fibre Channel 285  
    firmware 284  
    fragmentation 417  
    I/O operation 675  
    interface data rate 273  
    interleave depth 275  
    introduction 47  
    iSCSI 238, 255  
    large arrays 266  
    latency 238, 770  
    Linux 480–481  
    Linux bottlenecks 733

logical drive configurations 274  
Low Voltage Differential 239  
media data rate 273  
NAS 253  
NL SAS 249  
operation 240  
optimization 678  
paging 702  
Parallel ATA 246  
performance analysis 675  
performance factors 267  
platters 238  
primary counters 674  
protocols 243  
RAID levels 257  
RAID rebuild time 284  
RAID strategy 268, 677  
rebuild time 284  
ReiserFS file system 483  
relative speed 237  
remote storage 250  
rotational latency 241, 273  
rotational positioning optimization 274, 678  
rules of thumb 291, 679  
SAN 251  
SAS 238, 242  
SAS nearline 249  
SATA 238, 244  
seek operation 240  
seek time 238, 273, 677  
sequential I/O 677  
Serial ATA 246  
serial technology 239  
servo track 240  
solid state drives 249  
spread of data 271  
SSA 239  
steps 240  
stripe size  
    affect on performance 678  
    case study 769  
    concept 275  
    Linux 487  
stroke 271  
tuning options 705  
Windows bottlenecks 703  
worksheet 675  
write-back versus write-through 281  
DISKPERF command 280, 568

DMA devices 145  
DMA transfers 295, 298  
dmesg command 610  
DNS servers 26  
domain controller 15  
domain controllers 15  
downloads 159  
DRAM chips 185  
DS4000  
    SCSI protocol 288  
    throughput 286  
    two controllers 290  
Dual Dynamic Power Management 112  
Dunnington 100–101  
dynamic Web pages 21

**E**  
ECC memory 185, 212  
EIDE 238, 246  
EM64T  
    *See Intel 64 Technology*  
e-mail server 20  
encryption 418  
energy efficiency 49–91  
    Active Energy Manager 78  
    airflow 86  
    balance 51  
    benchmarks 88  
    BladeCenter 82  
    CPUs 54  
    C-states 58  
    data center 53  
    data center-level solutions 84  
    definition 49  
    drives 64  
    efficiency of components 54  
    governors 72  
    green DIMMs 61  
    iDataPlex 82  
    importance 51  
    intelligent PDUs 81  
    Linux 69  
    memory 60  
    memory layout 62  
    Nehalem processors 59  
    operating systems 68  
    PDUs 81  
    power budget 53

power capping 59  
Power Configurator 76  
power demands 49  
power savings 59  
power supplies 65  
processors 54  
Project Big Green 50  
P-states 57  
rack-level solutions 76  
Rear Door Heat eXchanger 83  
resources 91  
savings 59  
server-level solutions 53  
software 68  
solutions 53  
SPECpower\_ssj2008 88  
trending 80  
T-states 57  
virtualization 72  
Windows Server 2008 68

ESX Server  
    energy efficiency 73  
    esxtop 650–657  
    measuring performance 649  
    performance measurement 649  
    VirtualCenter Console 657

esxtop 650  
    batch mode 655  
    columns to display 655  
    commands 654  
    CPU information 654, 656  
    disk information 654  
    exit 657  
    logging 655  
    memory information 654, 657  
    network information 654  
    PCPU usage 656  
    starting 651

Ethernet 293, 346–347  
    *See also* network subsystem  
    10 Gb Ethernet 345  
    10 Gigabit Ethernet 313  
    1480 byte packet size 303  
    FC over Ethernet 347  
    jumbo frames 312  
    linear scaling 308

eX4 architecture  
    chipset 175  
    memory implementation 231

examples 743  
EXEC\_PAGESIZE 728  
Execute Disable Bit 209  
Ext3 file system 482  
Extended Page Tables 139

## F

FB-DIMMs  
    Advanced Memory Buffer 193  
    energy efficiency 60  
    implementation 191  
    low-power DIMMs 194  
    performance 195

FC over Ethernet 347  
fiber optic cabling 255

Fibre Channel  
    *See also* disk subsystem  
    FC over Ethernet 347  
    I/O operation 285  
    I/O size 285, 288  
    iSCSI comparison 256  
    protocol layers 287  
    redundant paths 255  
    rules of thumb 290  
    scalability 255  
    SCSI protocol 288  
    SCSI, comparison with 254  
    segment size 276  
    throughput 285, 290

file server 17  
    stripe size 277

file system cache, Windows 365

Fluent benchmark 40

four phases 4

fragmentation, disk 417, 419

frames, Ethernet 295

free command 625, 730

fsutil command 415

Fully Buffered DIMMs  
    *See* FB-DIMMs

## G

Gigabit Ethernet  
    *See also* network subsystem  
    10 Gb Ethernet 345  
    10 Gigabit Ethernet 313  
    checksum offload 315

gpm 456

Greencreek 172  
groupware servers 22  
guidelines 5

**H**  
hardware assists 138  
Hardware Prefetcher 131  
hardware scalability 160  
Harpertown 100  
High Performance Computing 676  
high performance computing 28  
HPMA 232  
hpoj 456  
HugeTLBfs 479  
Hyper-Threading 99, 693  
    interrupt processing 389  
    kernel selection 474  
    Linux 473  
    software scalability 169  
HyperTransport 177–178, 180  
Hyper-V 437  
    energy efficiency 73  
hypervisor 135

**I**  
I/O Acceleration Technology 324  
    *See also* IOAT  
IA-32e mode 118  
IBM Center for Microsoft Technologies 7  
iDataPlex 82  
industry-standard benchmarks 32  
InfiniBand 29, 151  
integrated memory controller 106  
Intel  
    5000 chipset 171  
    Advanced Digital Media Boost 105  
    Advanced Smart Cache 104  
    chipsets 158  
    Core microarchitecture 103  
    Data Execution Prevention 209  
    EM64T 117  
    Execute Disable Bit 209  
    Extended Page Tables 139  
    integrated memory controller 106  
    Intel 64 Technology 117  
    Intel VT 139  
    Intelligent Power Capability 103  
    naming convention 108

Nehalem architecture 105  
QuickPath Interconnect 105, 181  
Simultaneous Multi-Threading 107  
Smart Memory Access 104  
transition latency 140  
UEFI 170  
vConsolidate 39  
Virtual Processor Identifier 139  
Virtualization Technology 138  
VTune 605  
Wide Dynamic Execution 103  
Intel 64 Technology  
    architecture 117  
    modes 118  
intelligent PDUs 81  
interleaving  
    interleave depth (disk) 275  
    memory 199  
Internet Explorer  
    Performance console, use with 569  
interrupt assignment 388  
INTFILTR utility 388, 713  
introduction 3  
IOAT 324  
    adapters supported 326  
    Clovertown processors 98  
    data flow 326  
    implementation 324  
    operating system support 330, 339  
    TOE comparison 329  
iostat command 613, 734, 736  
IP datagrams 295  
IP Prefetcher 131  
IPSEC, iSCSI 257  
irqbalance 456  
isag command 726  
iSCSI 238, 255, 339  
    encapsulation 342  
    encryption 257  
    Fibre Channel comparison 256  
    hardware initiator 343  
    host bus adapter 344  
    infrastructure 344  
    initiator 340  
    latency 257, 344  
    NAS comparison 340  
    network 344  
    network load 344  
    OSI model 341

performance 256  
remote boot 344  
SCSI comparison 256  
security 344  
session layer 341  
software initiator 342  
TCP/IP packets 342  
technology 340  
throughput 257  
TOE, combined with 343  
isdn 456

## J

Java Server Benchmark 36  
journaling  
Linux 480  
options for Ext3 486  
Jumbo Frames  
VMware ESX 503  
jumbo frames 312

## K

KDE System Guard 617–624  
memory monitoring 730  
network bottlenecks 739  
kernel  
/proc file system 467  
Hyper-Threading 474  
parameters 466, 469  
powertweak 466  
selection 474  
sysctl command 468  
kernel swap behavior 478  
Kirkland center 7  
Knowledge Consistency Checker 16  
kswapd 478  
kudzu 456

## L

lanes 149  
LargeSystemCache 369  
last access time  
Linux 483  
Windows 414  
latency 200  
disk 238  
NUMA 162

latent bottlenecks 671  
LDAP 15  
legacy mode 118  
Linpack benchmark 38  
Linux 453–500  
/proc 467  
2.6 kernel 454  
accept\_redirects 493  
access time updates 483  
affinity 726  
analyzing bottlenecks 719–741  
anticipatory elevator 485  
CFQ scheduler 485  
chkconfig command 460  
CPU affinity 726  
CPU bottlenecks 724  
CPU subsystem 473  
daemons, disabling 455  
deadline elevator 485  
disk bottlenecks 733  
disk subsystem 480–481  
dmesg command 610  
elevator algorithm 484  
elvtune command 484  
energy efficiency 69  
EXEC\_PAGESIZE 728  
Ext3 file system 482  
file system 480  
free command 625, 730  
governors 72  
GUI, do not run 461  
Hyper-Threading 473  
I/O scheduler 485  
icmp\_echo\_ignore\_broadcasts 493  
init command 462  
interrupt handling 475  
IOAT support 330, 339  
iostat command 613, 734, 736  
ipfrag\_low\_thresh 496  
isag command 726  
journaling 480  
KDE System Guard 617–624  
memory monitoring 730  
network bottlenecks 739  
kernel  
parameters 469  
which one to use 474  
kernel 2.6 454  
last access time 483

mdadm 482  
memory bottlenecks 728  
memory bottlenecks 728  
memory subsystem 476, 480  
mingetty 463  
mpstat command 631, 725  
network bottlenecks 739  
network subsystem 492  
nice command 612  
nmon 731, 738  
noatime 483  
NOOP scheduler 485  
notail 487  
NUMA 454, 475  
NUMA optimizations 164  
page size 728  
partitioning recommendations 488  
partitions 489  
performance bottlenecks 719–741  
PLPerf 632  
pmap command 628  
RAID 481  
ReiserFS file system 483  
rmem\_max 495  
rp\_filter 495  
RSS support 339  
sar command 615, 726  
secure\_redirects 492  
Security Enhanced Linux 464  
send\_redirects 493  
sleep mode 731  
SMP-based systems 726  
Static Resource Allocation Table 163  
strace command 629  
stripe size 277, 487  
swap partition 490  
sysctl command 468  
tagged command queueing 487  
taskset command 727  
tcp\_fin\_timeout 495  
tcp\_keepalive\_time 495  
tcp\_max\_syn\_backlog 496  
tcp\_rmem 495  
tcp\_tw\_recycle 493  
tcp\_tw\_reuse 493  
tcp\_wmem 495  
TIME-WAIT 493  
TOE support 323, 339  
tools 607–632

top command 473, 611, 726  
Traffic-vis utility 625, 740  
ulimit command 630  
uptime command 609, 726  
vmstat command 615, 726, 736  
wmem\_max 495  
xPL 632  
zombie processes 613  
Linux Technology Center 7  
loaded latency 202  
local memory 205  
logical drive configurations 274  
logical drive migration 271  
logman utility 593  
long mode 118  
Loop Stream Detector 108  
Lotus Domino  
    stripe size 277  
Low Voltage Differential 239  
LS-DYNA benchmark 40

## M

Macrofusion 108  
mail server 20  
maximize throughput for file sharing 366  
mdadm 482  
memory mapped I/O 235  
memory subsystem 183–236  
    4 GB, more than 207  
    add memory 700  
    addressability 121  
    addressable memory 210  
    advanced ECC memory 212  
    Advanced Memory Buffer 192–193  
    analysis example 767  
    AWE 210  
    bandwidth 199  
    BIOS settings 232  
    bottlenecks 681  
    buffered 188  
    capacity 185  
    CAS 200  
    CAS latency 201  
    Chipkill 212  
    clock cycles 200  
    DDR memory 189  
    DDR2 memory 189  
    DDR3 memory 190

DIMM layout 198  
DIMMs 185  
double-ranked DIMMs 187  
DRAM chips 185  
ECC 185, 212  
energy efficiency 60, 62  
eX4 architecture 231  
FB-DIMM performance 195  
FB-DIMMs 191  
green DIMMs 61, 194  
HPMA 232  
insufficient memory 184  
integrated memory controller 106  
interleaving 199  
introduction 46  
latency 200, 202  
layout of DIMMs 198  
levels of memory 184  
Linux 476, 480  
Linux bottlenecks 728  
loaded latency 202  
local memory 205  
low-power FB-DIMMs 61, 194  
maximum memory addressable 121  
memory mapped I/O 235  
Memory ProteXion 233  
MetaSDRAM 195  
mirroring 47, 213  
NUMA 204  
Opteron 205  
PAE 208  
paged and non-paged RAM 698  
paging 184  
paging to disk 235, 702  
PC1600-PC3200 specifications 196  
peak throughput 197  
performance analysis 681  
primary counters 674  
processor performance, affect on 170  
QuickPath Interconnect 105  
rank 187  
RAS 200  
registered 188  
remote memory 205  
rules of thumb 234, 236  
SDRAM 188  
    buffered/unbuffered 188  
single-rank DIMMs 187  
STREAM benchmark 202  
technology 185  
timing 198  
tuning options 700  
unbuffered 188  
utilization rules of thumb 236  
virtual memory 699  
Windows bottlenecks 697  
working set 235  
worksheet 681  
x3755 memory technology 234  
Xcelerated Memory Technology 234  
MESI 165  
MetaSDRAM 195  
Microsoft Management Console 536  
Microsoft Scalable Networking Pack 332  
Microsoft Word  
    Performance console, use with 570  
mirroring, memory 213  
MMC 536  
MOESI 166  
Molex cable 245  
monitoring tools  
    ESX Server 649  
    Linux 607–632  
    Windows 533–605  
mpstat command 631, 725  
MTU size  
    Linux 470  
    Windows 403  
multimedia servers 23  
Myrinet 29

**N**

NAS 253  
    iSCSI comparison 340  
nearline SAS 249  
Nehalem  
    architecture 105  
    Power Control Unit 59  
nested paging 137  
NET SERVER CONFIG command 420  
NetBEUI 716  
netfs 456  
NetWare  
    stripe size 277  
Network Analyzer, use of 685  
Network Load Balancing 14  
Network Monitor 580

capturing network traffic 584  
configuring filters 584  
filters 583  
installing 581  
packet analysis 586  
promiscuous mode 581  
raw data 587  
starting 581  
System Management Server 581  
tips 588  
using 581  
versions 581  
viewing data 585

Network Monitor Driver 686  
network subsystem 293–347  
*See also iSCSI*  
10 Gb Ethernet 345  
10 Gigabit Ethernet 313  
1480 byte packet size 303  
adapter command overhead 295  
adapters 294  
auto-negotiation 381  
bcopies 310  
bottlenecks  
    assumptions 295  
    finding 685  
    solving 712  
    two types 709  
busmaster adapter 295  
checksum offload 315  
Chimney Offload 332  
command overhead 295  
Converged Enhanced Ethernet 346  
CPU count 311  
CPU efficiency 322  
CPU performance 310  
CPU utilization 307  
data rate bottlenecks 709  
design 716  
DMA transfers 295  
    network subsystem 298  
duplex setting 381  
efficiency vs utilization 304  
Ethernet frames 295  
frame size 300  
frame, maximum 303  
interrupts handled by a specific CPU 388  
Intfilter utility 388  
INTFILTR 388, 713

introduction 47  
IOAT 324  
IOAT-TOE comparison 329  
iSCSI 339  
jumbo frames 312  
large packet sizes 303  
limiting factors 295  
linear scaling 308  
link speed setting 381  
Linux 492, 739  
memory copies 298, 310  
multiple ports 304  
NetBEUI 716  
number of processors 311  
packet  
    defined 295  
    packets per second limit 295  
    size 300, 314  
PCI bus 298  
PCI busmaster 295  
performance 294, 300, 685  
ports, multiple 304  
primary counters 674  
processor speed 310  
protocols 716  
receive-side scaling 308, 334  
Scalable Networking Pack 332  
small packet sizes 302  
SMP scaling 315  
summary 314  
TCP Chimney Offload 332  
TCP offload engine 316  
TCP/IP  
    performance 296  
    Windows 716  
TOE 307, 316  
TOE-IOAT comparison 329  
transfer size 300, 314  
tuning options  
    Linux 741  
    Windows 709, 712  
Windows bottlenecks 707  
Windows protocols to remove 374  
worksheet 685  
xcopy, use of 314  
network-attached storage 253  
networking  
    Chimney Offload 440  
nfslock 456

nice command 612  
nmon 642–648, 731, 738  
    Analyser Excel macro 647  
    batch mode 646  
    command-line 646  
    count 646  
    data collection mode 646  
    download 643  
    Excel macro 647  
    file name 646  
    graphs 647  
    interactive mode 643  
    interval 646  
    Linux 642  
    nmon2csv 646  
    using 643, 738  
NTFS  
    compression 416  
    Last Access Time 414  
    use of 415  
NUMA 204  
    defined 161  
    Linux 454, 475  
    Linux kernel 474  
    Opteron 179  
    Windows 694

**O**  
on-going performance analysis 4  
operating system levels 134  
operating systems  
    introduction 48  
    Linux 453–500  
    Windows Server 2003 351–424  
Opteron 109  
    AMD64 109  
    Barcelona 111  
    block diagram 180  
    DDR2 support 109  
    Dual Dynamic Power Management 112  
    HyperTransport 114, 178, 180  
    memory access 178  
    memory addressable 122, 211  
    memory subsystem 205  
    NUMA 179  
    NX feature 209  
    Pacifica 110  
    Passthru card for x3755 113

PCI Express support 110  
Revision F 109  
Shanghai 112  
Split-plane 112  
SUMO 163, 179  
Oracle  
    processors, number of 130  
    server type 18  
    stripe size 277  
Oracle benchmarks 39  
OSI model 295

**P**  
Pacifica 138  
packet  
    defined 295  
    segmentation 315  
PAE  
    defined 208  
    parameter in BOOT.INI 392  
    Windows 391  
PAGEFILE.SYS 361  
paging  
    disable kernel paging 407, 411  
    Linux 728  
    RAID-5 not recommended 363  
    unavoidable 702  
    Windows 361  
Parallel ATA 246  
paravirtualization 136  
Passthru card for x3755 113  
PC1600-PC300 memory specifications 196  
PCI Express 46, 149  
    2.0 151  
    AMD Opteron support 110  
    bandwidth 151  
    bridge not required 153  
    compared with PCI-X 151  
    lanes 149  
    link 149  
    overhead 152  
    PCI Express 2.0 151  
    performance 152  
    physical size 151  
    slot compatibility 150  
    uses 151  
    x3650 block diagram 154  
PCI subsystem 145–155

*See also* PCI Express  
*See also* PCI-X  
agent 146  
busmaster devices 145  
design 146  
initiators 146  
introduction 46  
modes 148  
multiplexed address and data bus 146  
notches in PCI adapters 148  
PCI transaction 146  
targets 146  
turnaround phase 146

**PCI-X**  
*See also* PCI subsystem  
adapter keying 148  
attribute phase 147  
bridging 154  
compared with PCI 147  
disconnect boundary 147  
frequencies 147  
modes and speeds 148  
split transactions 147  
throughput 147  
x3650 block diagram 154

pcmcia 456  
PDUs 81  
perfmon  
*See* Performance console  
performance  
analyzing 669  
bottlenecks, finding 663  
cache size 127  
case studies 743  
chipsets 159  
CPU clock speed 128  
CPU subsystem 122, 159  
data set size 271  
disk subsystem 267, 273  
drives, number of 269  
Ethernet adapter 294  
FB-DIMMs 195  
Fibre Channel 285, 289  
I/O transfer size 285  
interleaving 199  
IOAT 326, 328  
latent bottlenecks 671  
logical drive configurations 274  
memory bandwidth 199

network subsystem 293  
page file 363  
primary counters 674  
RAID adapter cache size 282  
RAID rebuild time 284  
RAID strategy 268  
spread of data 271  
stripe size 275  
TCP/IP 296  
TOE 316, 319  
tools  
ESX Server 649  
Linux 607–632  
Windows 533–605  
transfer size for Ethernet 300  
well balanced system 671  
write-back versus write-through 281  
X3 Architecture 129

**Performance console** 534–573  
adding counters 544  
alerts 559  
creating 559  
saving 563  
chart view 537, 541  
counter log 547  
creating 548  
deleting 556, 564  
importing 556  
saving 556  
starting 555  
time frame of view 559  
counters 539  
**Memory**  
Available Bytes 674, 767  
Available MBytes 682, 699–700  
Cache Faults/sec 747  
Page Reads/sec 674, 682, 699–700, 746, 767  
Page Writes/sec 674, 682, 699–700, 768  
Pool Nonpaged Bytes 683, 700

**Network Interface**  
Bytes Received/sec 687, 708  
Bytes Sent/sec 687, 708  
Bytes Total/sec 674, 686, 708  
Output Queue Length 717  
Packets Received/sec 687, 708  
Packets Sent/sec 687, 708  
Packets/sec 674, 687, 708

Page Reads/sec 745  
Page Writes/sec 745  
Paging File  
    % Usage Peak 683  
    %Usage Max 364  
PhysicalDisk  
    Avg. Disk Bytes/Read 769  
    Avg. Disk Bytes/Transfer 278, 289, 680, 704  
    Avg. Disk Bytes/Write 769  
    Avg. Disk Queue Length 680, 704  
    Avg. Disk sec/Read 751  
    Avg. Disk sec/Transfer 674, 680, 704  
    Avg. Disk sec/Write 745, 752, 770  
    Disk Bytes/sec 290, 681  
    Disk Transfers/sec 770  
    Split IO/sec 681  
Processor  
    % Privileged Time 684, 694  
    % Processor Time 674, 683, 694, 745, 748, 766  
    % User Time 684, 694  
    Interrupts/sec 685  
Server  
    Pool Nonpaged Failures 683  
    Pool Nonpaged Peek 683  
SQL Server  
    Free Pages 756  
System  
    Processor Queue Length 684, 695  
TCP  
    Segments Retransmitted/sec 717  
    Segments/sec 717  
UDP  
    Datagrams/sec 717  
CPU bottlenecks 693  
data collector set 546  
database servers, use with 569  
deleting objects 546  
disabled counters 572  
disk counters 568  
DISKPERF command 280, 568  
Fibre Channel 289  
highlighting a counter 546  
histogram view 537, 543  
icons 542, 548  
instances 539  
Internet Explorer, use with 569  
Linux 632  
logical drive counters 568  
LogicalDisk object 703  
logs 547  
Microsoft Word, use with 570  
network counters 569  
objects 539  
Performance Logs and Alerts 537, 546  
physical drive counters 568  
PhysicalDisk object 703  
real-time monitoring 744  
remote machines, accessing 547  
report 557  
report view 537, 543  
schedule 555  
spreadsheet applications, use with 569  
starting 536  
System Monitor 537, 541  
templates 549  
toolbar 542, 548  
trace log  
    buffer settings 568  
    creating 565  
    trace logs 547  
    TSV file format 546  
    views 537  
    word processors, use with 569  
Performance Lab 5  
Performance Monitor  
    Linux 632  
    Windows  
        See Performance console  
performance tuning 665  
phases 4  
ping command 403  
PLPerf 632  
    *See also* xPL  
pmap command 628  
portmap 456  
power  
    *See* energy  
power benchmark 38  
power capping 59  
Power Configurator 76  
power supplies, energy efficiency 65  
preemptive multitasking 359  
prefetch 130  
primary counters 674  
print servers 18  
privilege levels 134

processor subsystem  
    *See* CPU subsystem  
Project Big Green 50  
protocol layers, Fibre Channel 287  
P-states 57

## **Q**

quad-core processors 98  
questions to ask 667  
QuickPath Interconnect 105, 181

## **R**

Radware 14  
RAID array 4  
RAID levels 257  
    composite RAID levels 266  
    page file recommendation 363  
    RAID-0 258  
    RAID-00 267  
    RAID-1 259  
    RAID-10 266  
    RAID-1E 260  
    RAID-1E0 267  
    RAID-5 261  
        not for page files 363  
    RAID-5E 262  
    RAID-5EE 262  
    RAID-6 265  
    rebuild time 284  
        strategy 268  
    RAID-M 213  
rank 187  
Rapid Virtualization Indexing 142  
RAS 200  
rawdevices 456  
Rear Door Heat eXchanger 83  
rebuild time 284  
receive-side scaling 308, 334  
Red Hat Enterprise Linux  
    *See also* Linux  
    Ext3 file system 482  
    hugeblk\_pool 472  
    inactive\_clean\_percent 472  
    inet\_peer\_gc\_maxtime 472  
    pagecache 472  
Redbooks Web site 783  
    Contact us xxix  
redundant bit steering 233

redundant paths 255  
registered SDRAMs 188  
ReiserFS 483  
    notail 487  
Reliability and Performance Monitor  
    *See* Performance console  
remote memory 205  
remote storage 250  
rings 134  
rotational positioning optimization 274, 678  
rpc 456  
RSS  
    operating system support 339  
RTP Performance Lab 5  
rules of thumb  
    cache size 127  
    disk subsystem 291, 679  
    Fibre Channel 290  
    memory subsystem 234, 236  
    write-back versus write-through 282

## **S**

SAN 251  
    backend zone 252  
    design 251  
    frontend zone 252  
    iSCSI 256  
SAP benchmark 39  
sar command 615, 726  
SAS 242  
    *See also* disk subsystem  
    cables 245  
    components 242  
    defined 238  
    introduction 47  
    lanes 245  
    layers 243  
    protocols 243  
    speed negotiation 244  
SATA 244, 246–248  
    defined 238  
    introduction 47  
scalability 14  
Scalable Networking Pack 332  
scale-up 160  
    versus scale-out 14  
scenarios 743  
screen savers 419

SCSI 238  
array controller operation 240  
Fibre Channel, comparison 254  
iSCSI comparison 256  
Logical Block Address 240  
rotational latency 241  
SDRAM 188  
    *See also* memory subsystem  
SEC 212  
sector prefetch 130  
seek operation 240  
seek time 238  
segment size, Fibre Channel 276  
segments, TCP 295  
SELinux 464  
sendmail 456  
Serial ATA 246–248  
    features 247  
    ServeRAID 248  
    standard 246  
server selection 4  
server types 13  
    Active Directory server 15  
    communication servers 24  
    database servers 18  
    DHCP servers 27  
    DNS servers 26  
    domain controller 15  
    domain controllers 15  
    e-mail servers 20  
    file server 17  
    groupware servers 22  
    HPC 28  
    multimedia servers 23  
    print servers 18  
    terminal server 25  
    virtualization servers 28  
    Web 2.0 22  
    Web servers 21  
    WINS servers 27  
    worksheet 670  
ServeRAID  
    *See also* RAID  
    *See also* SAS  
    *See also* SCSI  
    *See also* Serial ATA  
    cache size 282  
    firmware 284  
    large arrays 266  
logical drive migration 271  
operation 240  
rebuild time 284  
Serial ATA adapter 248  
stripe size 275, 277  
write-back versus write-through 281  
ServerWorks  
    chipsets 158  
set associativity 123  
Shanghai 112  
shared L3 cache 95  
Simultaneous Multi-Threading 107  
single-rank DIMMs 187  
smartd 456  
SMP 160, 203  
    effect on performance 129  
snoop cycles 165  
software considerations 693  
software scalability 168  
solid state drives 249  
    energy efficiency 64  
SPEC benchmarks 36  
SPECpower\_ss2008 88  
Split-plane 112  
SQL Server  
    analysis 745  
    example 745  
    processors, number of 130  
    server type 18  
    stripe size 277  
SSA 239  
Static Resource Allocation Table 163  
static Web pages 21  
Storage Attached Network 251  
strace command 629  
STREAM benchmark 202  
stripe size 275  
    affect on performance 678  
    Linux 487  
    page file 280  
    video file server 277  
    Web server 277  
stroke 271  
subsystems, important  
    Active Directory 17  
    communication servers 24  
    database servers 18  
    DHCP servers 27  
    DNS servers 26

domain controller 15, 17  
file servers 17  
groupware servers 23  
HPC servers 28  
mail servers 20  
multimedia servers 23  
print server 18  
terminal server 25  
virtualization servers 28  
Web 2.0 servers 22  
Web servers 21  
WINS servers 27  
SUMO 163, 179  
SUSE Linux Enterprise Server  
*See also* Linux  
accept\_redirects 470  
autoconf parameter 470  
dad\_transmits 470  
heap-stack-gap 471  
ip\_conntrack\_max 470  
powertweak 466  
regen\_max\_retry 471  
ReiserFS file system 483  
router\_solicitation\_delay 470  
router\_solicitation\_interval 470  
router\_solicitations 470  
sched\_yield\_scale 469  
shm-bigpages-per-file 469  
shm-use-bigpages 469  
temp\_preferred\_lft 470  
temp\_valid\_lft 470  
Traffic-vis utility 625  
virtualization 497  
vm\_anon\_lru 471  
vm\_lru\_balance\_ratio 471  
vm\_mapped\_ratio 471  
vm\_passes 471  
vm\_shmem\_swap 471  
vm\_vfs\_scan\_ratio 471  
Xen 496  
YaST 461  
SYN requests 398  
sysctl commands 468  
Sysinternals utilities 589  
System Monitor  
*See* Performance console  
System x Performance Lab 5  
System x Performance Logger 632  
*See also* xPL

System x3755  
Passthru card 113

**T**

Task Manager 573–580  
columns 575  
performance 577  
processes 574  
starting 573  
taskset command 727  
TCP Chimney Offload 332  
TCP offload engine 316  
*See also* TOE  
TCP segments 295  
TCP table 400  
TCP/IP 296, 716  
counters 716  
Linux kernel parameters 469  
MaxUserPort 399  
MTU size  
    Linux 470  
    Windows 403  
operations 296  
Path MTU 405, 442  
TCP acknowledgement frequency 402  
TCP connection retransmissions 398  
TCP Control Block table 400  
TCP data retransmissions 399  
TCP window scaling 396  
TCP windows size 395, 441  
TIME-WAIT  
    Linux 493  
    Windows 399  
    Windows tuning 394  
thermal sensors, DIMMs 191  
Tigerton 97, 99  
TIME-WAIT  
    Linux 493  
    Windows 399  
TOE 316  
adapter support 324  
benefits 317  
data flow 317  
IOAT comparison 329  
iSCSI, combined with 343  
operating system support 323, 339  
purpose 316  
throughput 319

tools  
    ESX Server 649  
    Linux 607–632  
    Windows 533–605  
top command 473, 611, 726  
TPC benchmarks 35  
tracerpt utility 593  
Traffic-vis utility 625, 740  
transition latency 140  
Translation Lookaside Buffer 479  
trending 80  
T-states 57  
Tulsa 95  
tunable daemons 455  
typeperf utility 593

## U

UEFI 170  
ulimit command 630  
unbuffered/unregistered SDRAMs 188  
unregulated benchmarks 32  
uptime command 609, 726

## V

VBScript 593  
vConsolidate 39  
Video file server 277  
video subsystem 48  
virtual memory, Windows 361  
Virtual Processor Identifier 139  
VirtualCenter Console 657  
virtualization 134  
    energy efficiency 72  
    Xen 496  
virtualization hardware assists 133–144  
virtualization servers 28  
virus scanner applications 419  
VMmark 38  
VMotion 527  
vmstat command 615, 726, 736  
VMware  
    VMmark benchmark 38  
VMware ESX  
    affinity 511  
    BIOS settings 516  
    CPU 510  
    disk 505, 515  
    disk types 507

driver 522  
ESX 3.5 overview 523  
esxtop 509  
hardware subsystems 503  
I/O request sizes 506  
iSCSI 508  
Jumbo Frames 503  
kernel 518  
load balancing 530  
memory 513  
memory allocation 522  
network speed 519  
network subsystem 503  
over commit 508  
overview of 3.5 523  
page sharing 518  
partitions 515  
Queue Depth 508  
routing policies 505  
SCSI driver 522  
See ESX Server  
server farm 528  
single-threaded applications 511  
sizing 524  
storage 505, 515  
storage sizing 528  
swap 520  
switches 504  
time synchronization 523  
virtual disk modes 507  
virtual memory 513  
virtual switch 504  
VM tuning 520  
VMotion 527  
vSMP 524  
VMX 138  
VT 138  
VTune 600

## W

Web 2.0 22  
Web server  
    server type 21  
    stripe size 277  
well balanced system 671  
Win32PrioritySeparation 360  
Windows  
    EnablePMTUDiscovery 442

Path MTU 442  
TCP window size 441  
Windows Management Instrumentation 593  
Windows NT  
    4 GB, more than 207  
Windows Performance Toolkit 588  
Windows Server 2003 351  
    /3GB parameter 390, 393  
    /PAE parameter 392  
    32-bit editions 353  
    4 GB memory limit 391  
    64-bit editions 353–354  
    8.3 filenames 415  
    active data set size 416  
    addressable memory 355  
    affinity 387  
    analyzing bottlenecks 691–717  
    auditing 418  
    auto-negotiation 381  
AWE 210  
AWE support 391  
background services 360  
Base Priority 384  
binding order 376  
checksum offload 315  
clustering support 353  
Coalesce Buffers setting 381  
CPU affinity 387  
CPU-bound applications 386  
CPUs supported 353  
Data Execution Prevention 357  
disable services 371  
DisablePagingExecutive 407, 411  
drive arrays 417  
dump file 363  
duplex setting 381  
dynamic priority 382  
EM64T 357  
EnablePMTUDiscovery 405  
encryption 418  
energy efficiency 69  
features 353  
file system cache 365  
foreground boost 359  
fragmentation 417  
fsutil command 415  
high priority 385  
I/O locking operations 409  
idle priority 385  
Intel 64 Technology 357  
interrupts 388  
INTFILTR utility 388, 713  
IOAT support 339  
LanmanServer key 424  
large amount of RAM installed 370  
large TCP window scaling 396  
LargeSystemCache 369  
Last Access Time 414  
link speed setting 381  
log off the server 419  
MaxCmnds 413  
MaxFreeTcbs 402  
MaxFreeTWTcbs 402  
MaxHashTableSize 401  
maximize throughput for file sharing 368  
maximum segment size 395  
Maximum Transmission Unit 403  
MaxMpxCt 413  
MaxUserPort 399  
MaxWorkItems 413  
monitoring tools 533–605  
MTU 403  
multitasking 359  
NET SERVER CONFIG command 420  
network card settings 378  
network control blocks 413  
Network Load Balancing 14  
Network Monitor 580  
network provider order 377  
normal priority 385  
NumTcbTablePartitions 401  
offload features 382  
outstanding network requests 412  
packet segmentation 315  
PAE support 391  
page file stripe size 280  
paged pool 356  
PagedPoolSize 408  
paging 361, 363  
Patch-Guard 357  
Path MTU 405  
performance bottlenecks 691–717  
performance options window 360  
performance tools 533–605  
print provider order 377  
priority 382  
priority, when to change 386  
processor affinity 387

product family 352  
protocols  
    binding order 376  
    remove 374  
quantum 383  
R2 358  
RAM supported 353  
realtime priority 385  
Receive Buffers setting 381  
receive window 395  
RSS support 339  
Scalable Networking Pack 332  
scheduler 382  
screen savers 419  
server roles 419  
service startup recommendations 373  
services, disable 371, 714  
SMB 412  
SNMP service 716  
START command 385  
Static Resource Allocation Table 163  
stripe size 277, 280, 417  
system cache 365  
SystemPages 409  
Task Manager 383  
    *See* Task Manager  
TCP acknowledgement frequency 402  
TCP connection retransmissions 398  
TCP Control Block table 400  
TCP data retransmissions 399  
TCP TIME-WAIT delay 399  
TCP window scaling 396  
TCP window size 395  
TCP/IP operation 299  
TCP/IP tuning 394  
TCP1323Opts parameter 397  
TcpWindowSize 396  
Terminal Services  
    *See* Windows Terminal Services  
TOE support 323, 339  
tools 533, 605  
transfer size 418  
Transmit Descriptors setting 382  
user memory 355  
virtual memory 355, 361, 699  
VTune 600  
Win32PrioritySeparation 360  
Windows on Windows 64 emulator 356  
WMI 593  
work items 413  
x64 353  
Windows Server 2008 ??–452  
    Active Directory 16  
    analyzing bottlenecks 691–717  
    Chimney Offload 440  
    command-line version 432  
    Compound TCP 441  
    defrag utility 592  
    disk capacity 443  
    disk subsystem 443  
    disk write-caching 444  
    domain controller 436  
    editions 428  
    energy efficiency 68  
    Explicit Congestion Notification 442  
    family 427  
    features 431  
    GPT partitions 446  
    hardware RAID 444  
    Hyper-V 437  
    INTFILTR utility 713  
    IOAT support 339  
    last-access time stamp 446  
    logman utility 593  
    MBR partitions 446  
    monitoring tools 533–605  
    Network DMA 440  
    Network Load Balancing 14  
    Network Monitor 580  
    networking 439  
    NTFS 443  
    partition offset 446  
    Path Maximum Transmission Unit 442  
    performance bottlenecks 691–717  
    performance tools 533–605  
    performance tuning 426  
    power management 68  
    R2 450  
    RAID 444  
    Read-Only Domain Controller 436  
    receive window 441  
    registry 442  
    Reliability and Performance Monitor  
        *See* Performance console  
    Reliability Monitor 535  
    RODC 436  
    roles 431  
    RSS 440

RSS support 339  
self-healing NTFS 443  
Server Core 432  
Server Manager tool 431  
server roles 431, 435  
servermanagercmd command 431  
service packs 429  
services 448  
SMB 439  
software RAID 444  
Static Resource Allocation Table 163  
Sysinternals utilities 589  
System Configuration utility 448  
TCP Chimney Offload support 332  
TCP/IP improvements 440  
TOE support 323, 339  
tracerpt utility 593  
tuning tips 426  
typeperf utility 593  
visual effects 447  
VTune 600  
Windows Error Reporting 449  
Windows Performance Toolkit 588  
Windows System Resource Manager 438  
WMI 593  
Windows System Resource Manager 438  
Windows Terminal Services  
    L2 cache 25  
    memory 25  
    network 25  
    number of users 26  
    performance 26  
    processor 25  
    server type 25  
    subsystems 25  
WINS servers 27  
Wolfdale 97  
Woodcrest 96  
word processors  
    Performance console, use with 569  
workload benchmarks 32  
worksheets 670  
write-back versus write-through 281  
WSRM 438

**X**

X3 Architecture  
    cache associativity 127

scaling performance 129  
x3755  
    memory technology 234  
    Passthru card 113  
Xcelerated Memory Technology 234  
xcopy, use of 314  
Xen 136, 496  
    binary translation 499  
    drivers 500  
    hypervisor layer 500  
    I/O virtualization 500  
    paravirtualization 498  
    performance 499  
    virtualization 497  
    XenSource 496  
Xeon 94–108  
    5100 96  
    5200 97  
    5300 98  
    5400 100  
    5500 100  
    7100 95  
    7300 99  
    7400 101  
    addressable memory 211  
    Clovertown 98  
    Dunnington 101  
    Harpertown 100  
    Nehalem 105  
    Tigerton 99  
    Tulsa 95  
    Wolfdale 97  
    Woodcrest 96  
xfs 457  
xPL 632  
    command line 637  
    counter descriptions 633  
    CPU counters 633  
    CSV file 637  
    example 639  
    input.prm file 637  
    interrupts counters 635  
    log file 637  
    memory counters 636  
    network counters 637  
    parameter file 637–638  
    Performance Monitor, importing into 637  
    sample 639  
    settings 638

starting 637  
stopping 637  
xRef 159  
xSeries  
*See* System x

## Y

YaST 461

**IBM**



**Redbooks**

# Tuning IBM System x Servers for Performance

(1.5" spine)  
1.5" <-> 1.998"  
789 <-> 1051 pages







# Tuning IBM System x Servers for Performance



**Redbooks**®

**Identify and  
eliminate  
performance  
bottlenecks in key  
subsystems**

**Expert knowledge  
from inside the IBM  
performance labs**

**Covers Windows,  
Linux, and VMware  
ESX**

This IBM Redbooks publication describes what you can do to improve and maximize the performance of your business server applications running on IBM System x hardware and either Windows, Linux, or VMware operating systems. It describes how to improve the performance of the System x hardware, the operating system, and specific server applications.

The book is divided into five parts. Part 1 introduces performance tuning, server types and benchmarking. Part 2 explains the technology implemented in the major subsystems in System x servers, and shows what settings can be selected or adjusted to obtain the best performance. Part 3 describes the performance aspects of key operating systems: Microsoft Windows Server 2003 and 2008, Red Hat Enterprise Linux, SUSE Linux Enterprise Server, and VMware ESX.

Part 4 introduces the performance monitoring tools that are available to users of System x servers. Part 5 shows you how to analyze your system to find performance bottlenecks, and what to do to eliminate them.

This book is targeted at people who configure Intel and AMD processor-based servers running Windows, Linux, or VMware ESX, and seek to maximize performance. Some knowledge of servers is required. Skills in performance tuning are not assumed.

**INTERNATIONAL  
TECHNICAL  
SUPPORT  
ORGANIZATION**

**BUILDING TECHNICAL  
INFORMATION BASED ON  
PRACTICAL EXPERIENCE**

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

**For more information:  
[ibm.com/redbooks](http://ibm.com/redbooks)**