PRESENTED BY RED HAT

LEARN. NETWORK.
EXPERIENCE OPEN SOURCE.

www.theredhatsummit.com

Red Hat Enterprise Linux Performance and Scalability

May 2011

D. John Shakshober (Shak)
Director Red Hat Performance Engineering
Larry Woodman
Consulting Engineer

dshaks@redhat.com
lwoodman@redhat.com

Agenda

Section 1 - RHEL6 Changes/Improvements

Section 2 - System Overview

Section 3 - Analyzing System Performance

Section 4 - Tuning Red Hat Enterprise Linux

Section 5 - Performance Analysis and Tuning Examples

Q & A

Section 1: RHEL6 Changes/Improvements

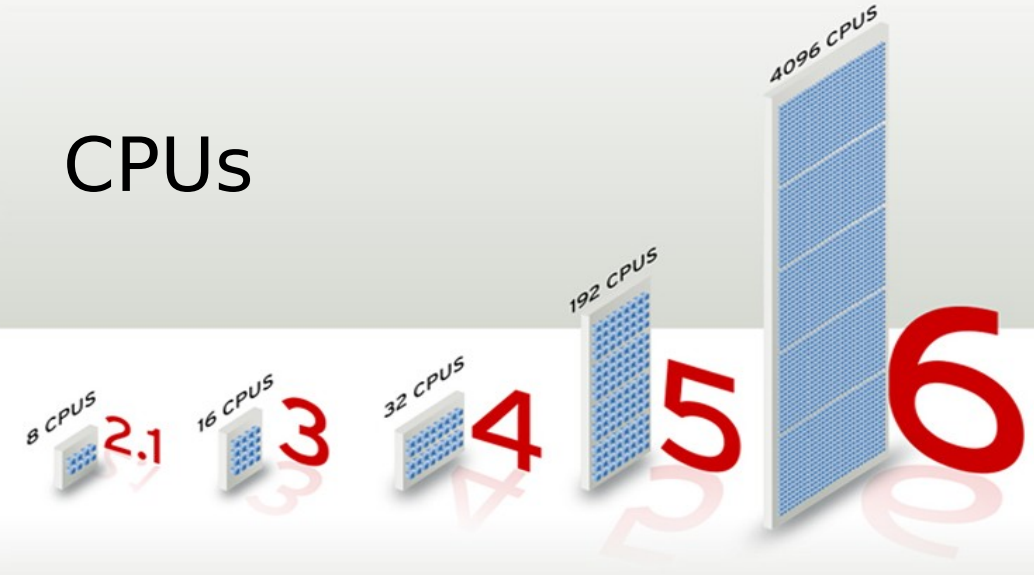
- Scalability improvements in RHEL6
- Kernel algorithms in RHEL6 for better performance
- New features in RHEL6
  - Cgroups, THP, transparent hugepages
- RHEL6 tuning, VM, tuned-adm and filesystems
- RHEL6 Networking
- RHEL6 KVM enhancments
- Performance tools in RHEL6

RHEL6 Scaling Improvements

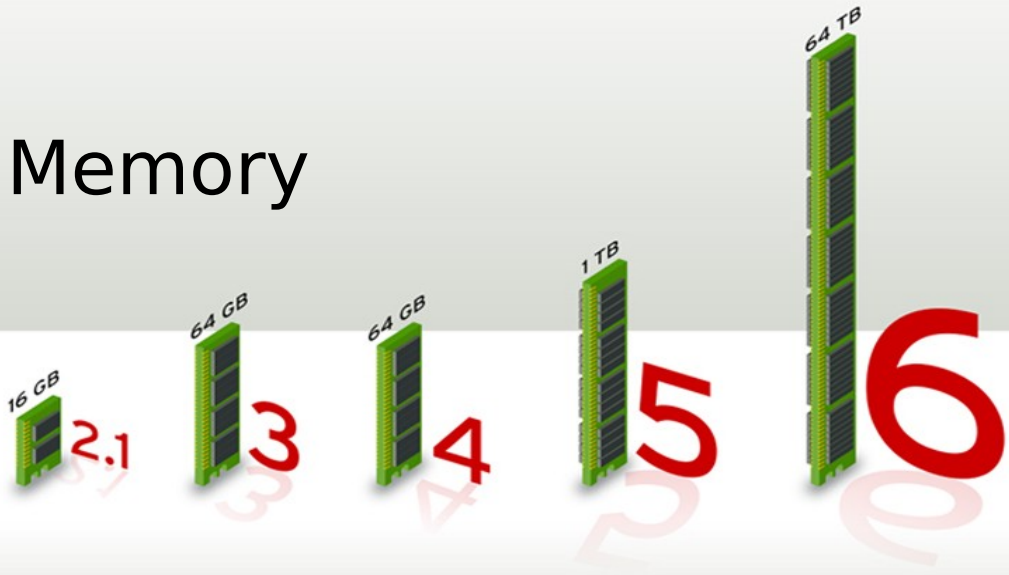
- Tickless kernel (2.6.17)
  - Reduced power consumption
- Split LRU (2.6.28)
  - Efficient reclaim (large systems)
- C – groups (2.6.18/2.6.29)
  - Better hardware utilization
- Ticket spinlocks (2.6.25 / 2.6.28)
  - Scalable / predictable locking
- Per-bdi flush (2.6.31)
  - Scalable flushing of dirty blocks
- Transparent Huge pages (2.6.31)
  - Automatically use huge pages

RHEL x86_64 version	CPUs	Memory
2.1	4	64GB
3	16	128GB
4	32	256GB
5	255	1TB
6	4096	64TB

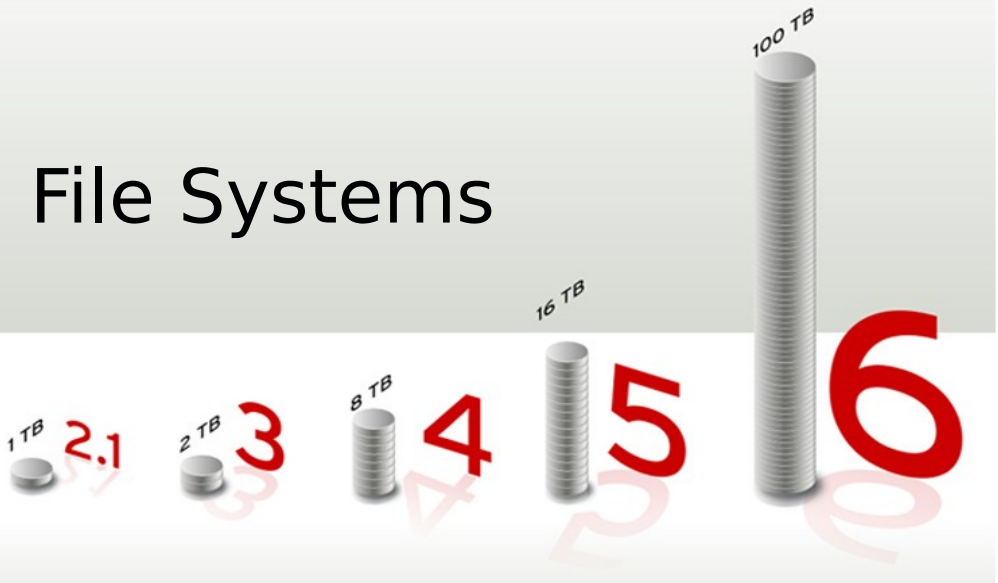
CPUs

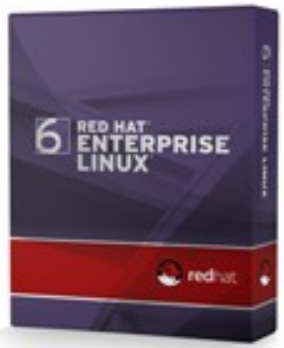


Memory



File Systems





Scalability

NUMA & Multi Core Support

Cpusets (2.6.12)

- Enable CPU & Memory assignment to sets of tasks
- Allow dynamic job placement on large systems

Numa-aware slab allocator (2.6.14)

Optimized locality & management of slab creation

Swap migration. (2.6.16)

Swap migration relocates physical pages between nodes in a NUMA system while the process is running – improves performance

Huge page support for NUMA (2.6.16)

Netfilter ip_tables: NUMA-aware allocation (2.6.16)

Multi-core

- Scheduler improvements for shared-cache multi-core systems (2.6.17)
- Scheduler power saving policy

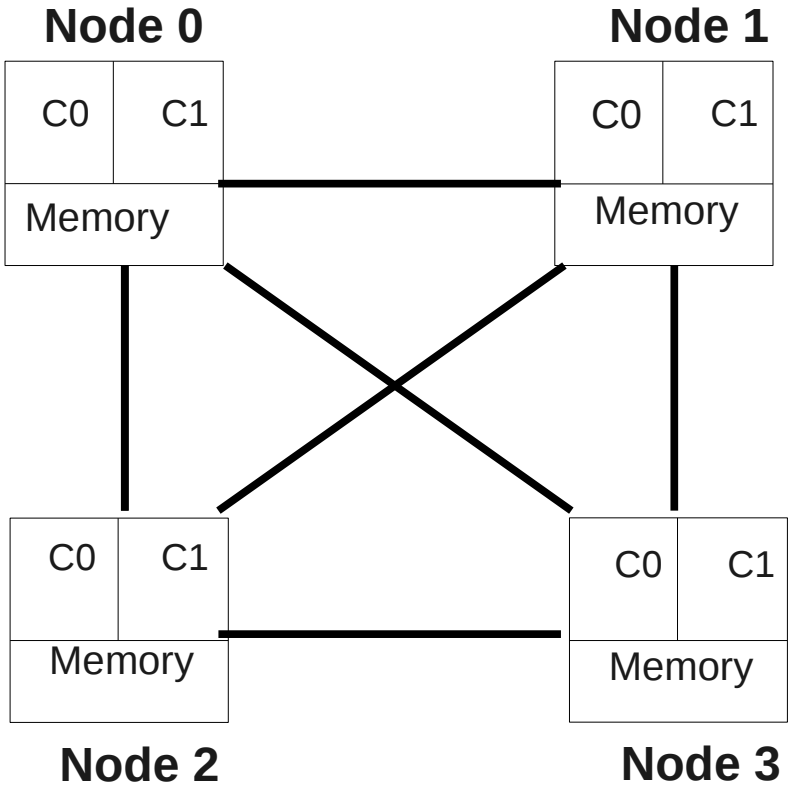
Power consumption improvements through optimized task spreading

Additional NUMA awareness in scheduler(2.6.32)

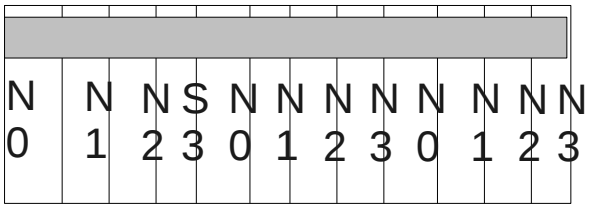
More NUMA aware hugepage allocation(2.6.32)

Significant scale-up(2.6.32)

Typical NUMA System Layout

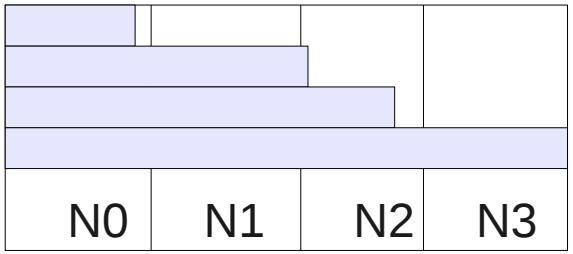


Process memory on N1C0



ınterleaved (Non-NUMA)

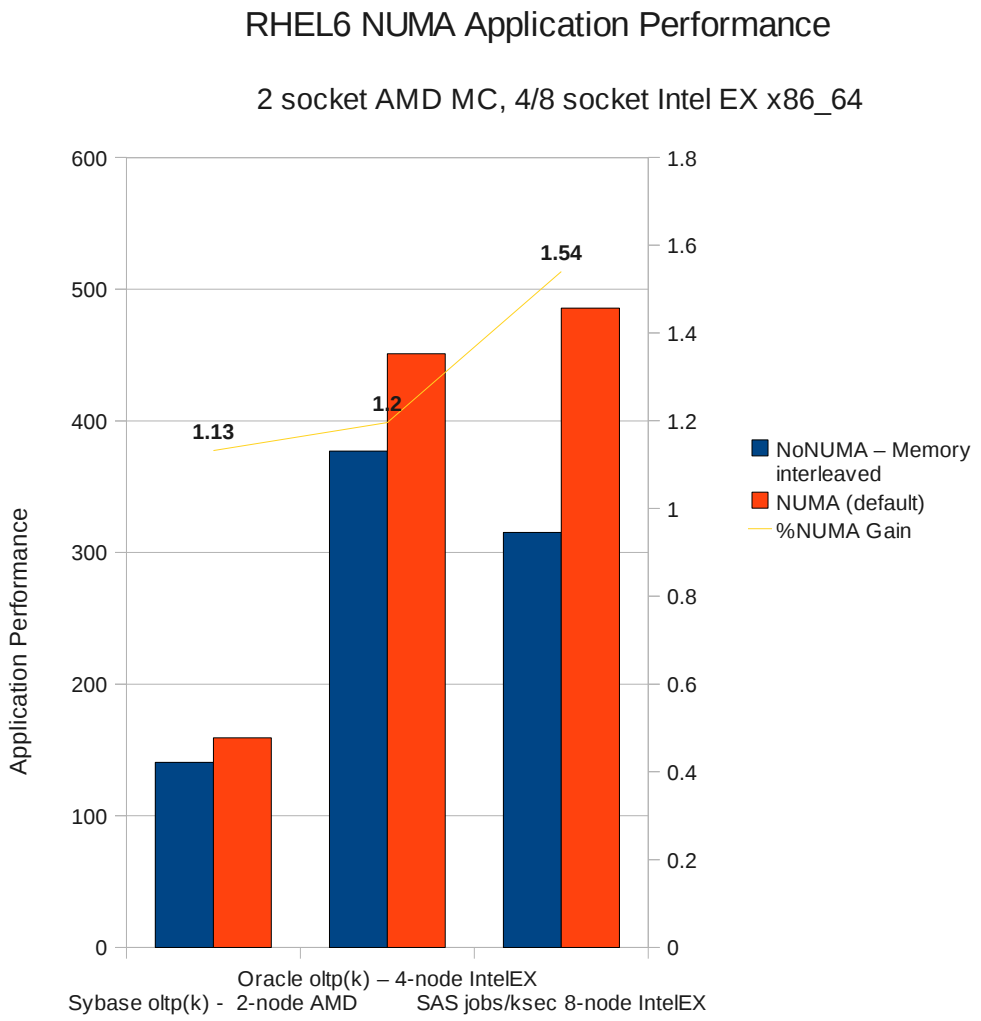
Process memory on N1C0



Non-Interleaved (NUMA)

RHEL6 Differences w/NUMA
AMD MC 2node/ Intel EX 4/8 node

- Split LRU (2.6.28) / NUMA
  - CFS NUMA scheduling
  - Efficient reclaim
  - Better hardware utilization
- Ticket spinlocks (2.6.25 / 2.6.28)
  - Scalable / predictable locking
- Per-bdi flush (2.6.31)
  - Scalable flushing of dirty blocks



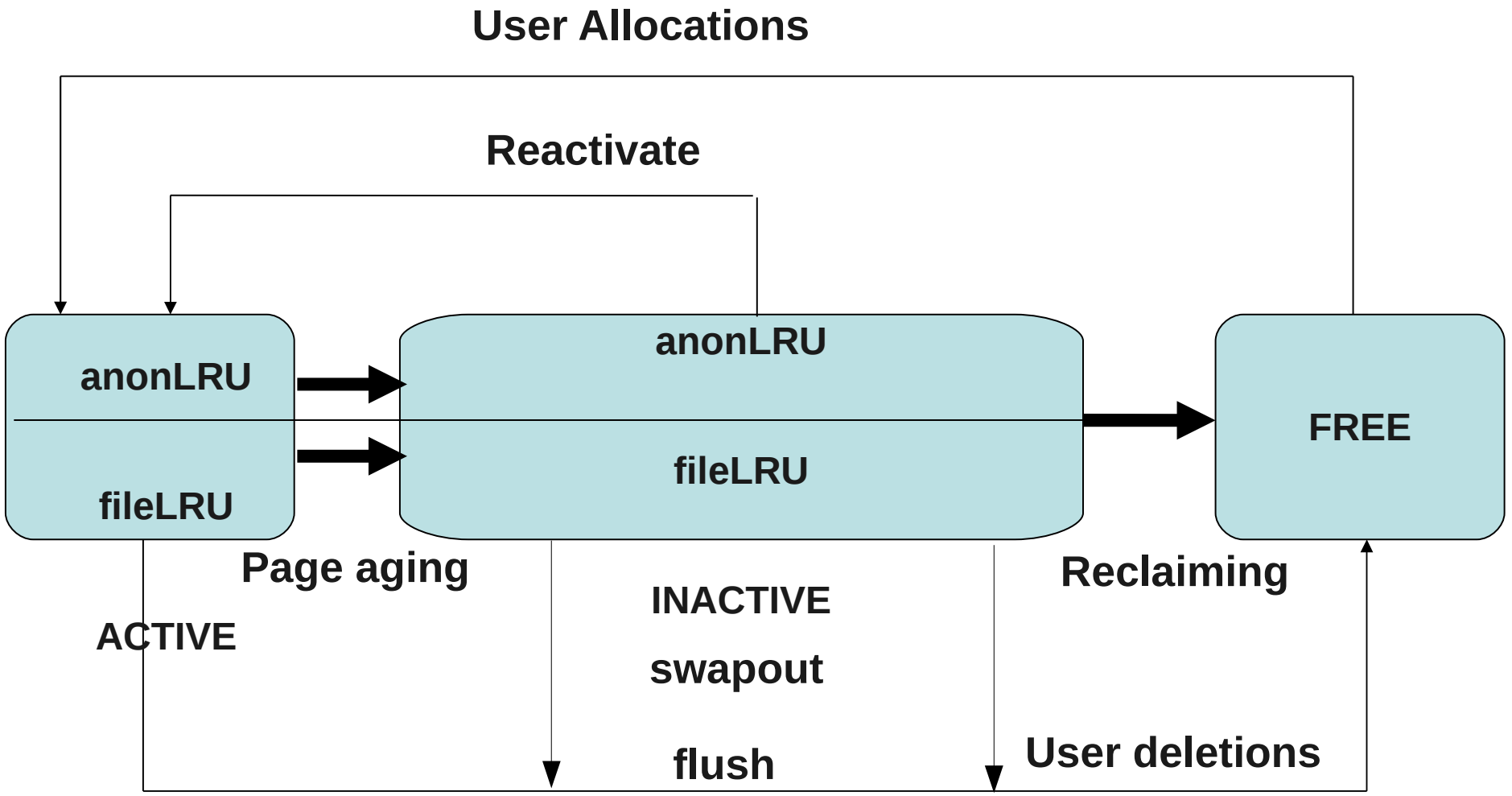
New kernel features in RHEL6

- Separate page-lists for anonymous & pagecache pages
- Ticketed spin-locks
- NUMA aware Hugepages
- Transparent hugepages
- 1GB hugepage support
- One flush daemon per bdi/filesystem
- cgroups
- Finer grained tuning for very large systems

Split LRU pagelists

- Separate page-lists for anononymous and pagecache
- Prevents mixing of anonymous and filebacked pages on active and inactive LRU lists
- Eliminates long pauses when all CPUs enter direct reclaim during memory exhaustion
- Prevents swapping when copying very large files
- Prevents swapping of database cache during backup.

Per Node/Zone split LRU Paging Dynamics



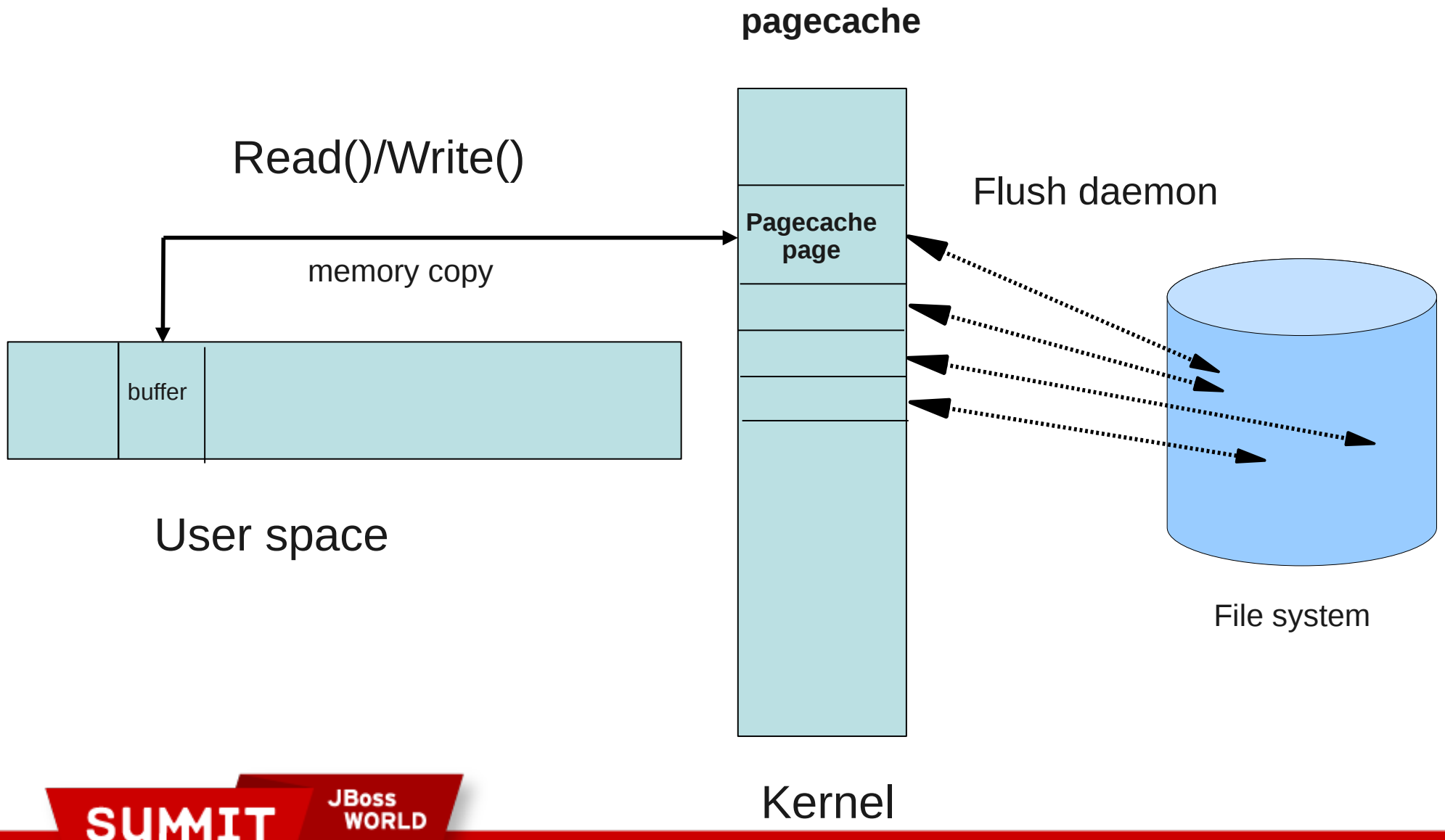
Ticketed spinlocks

- Logically converts the simple spinlock into into a FIFO queue.
- Eliminates remote node spinlock starvation on NUMA systems.
- Eliminates unfair spinlock access on all systems
- This is an x86_64 only feature.

Per device/file/LUN page flush daemon

- Each file system or block device has its own flush daemon
- Allows different flushing thresholds and resources for each daemon/device/file system.
- Prevents some devices from not getting flushed because a shared daemon blocks used all resources
- Replaces pdflushd where a pool of threads flushed all devices.

Per file system flush daemon



RHEL6 “tuned-adm” profiles

tuned-adm list

Available profiles:

- **default** - CFQ elevator (cgroup), IO barriers on, ondemand power savings, upstream VM, 4 msec quantum

- **latency-performance** – elevator=deadline, power=performance

- **throughput-performance** – latency + 10 msec quantum, readahead 4x, VM dirty_ratio=40

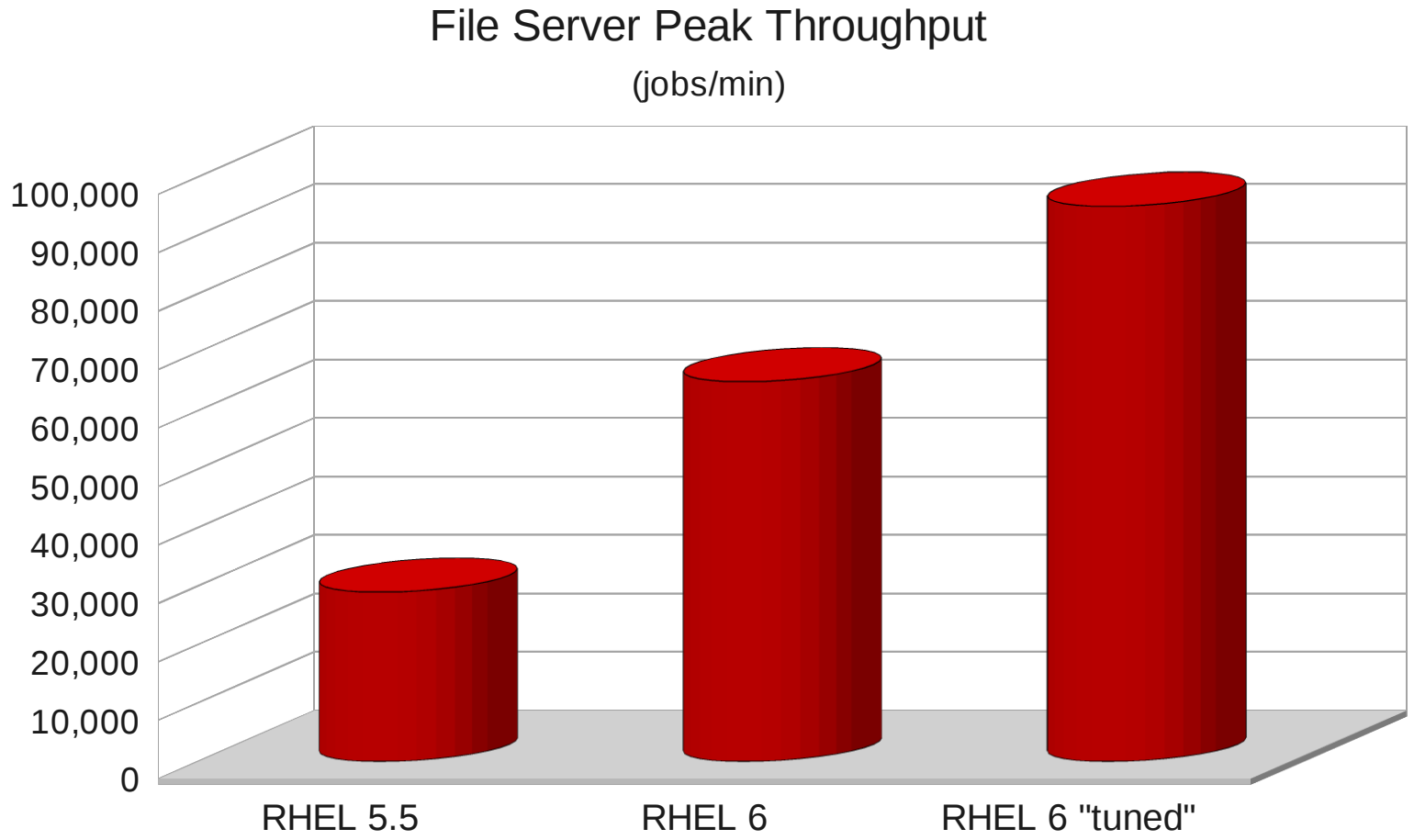
- **enterprise-storage** – throughput + barrier=0

Example

tuned-adm profile enterprise-storage

Recommend “enterprise-storage” w/ KVM

High End HP DL 980 AIM7 results w/ “ktune” (r5) “tuned-adm” (r6)



HP DL980 64-core/256GB/30 FC/480 lun AIM7 results w/ “tuned”

Control Groups

- Create hierarchical subsets of a total system
  - Memory
  - CPU
  - Disk
  - Network
- Aggregate sets of tasks & future children into subsets(cgroups)
- Constrain the tasks to the limits a cgroup

cgroups

1GB/2CPU subset of a 16GB/8CPU system

```
#mount -t cgroup xxx /cgroups
#mkdir -p /cgroups/test
#cd /cgroups/test
#echo 1 > cpuset.mems
#echo 2-3 > cpuset.cpus
#echo 1G > memory.limit_in_bytes
#echo $$ > tasks
```

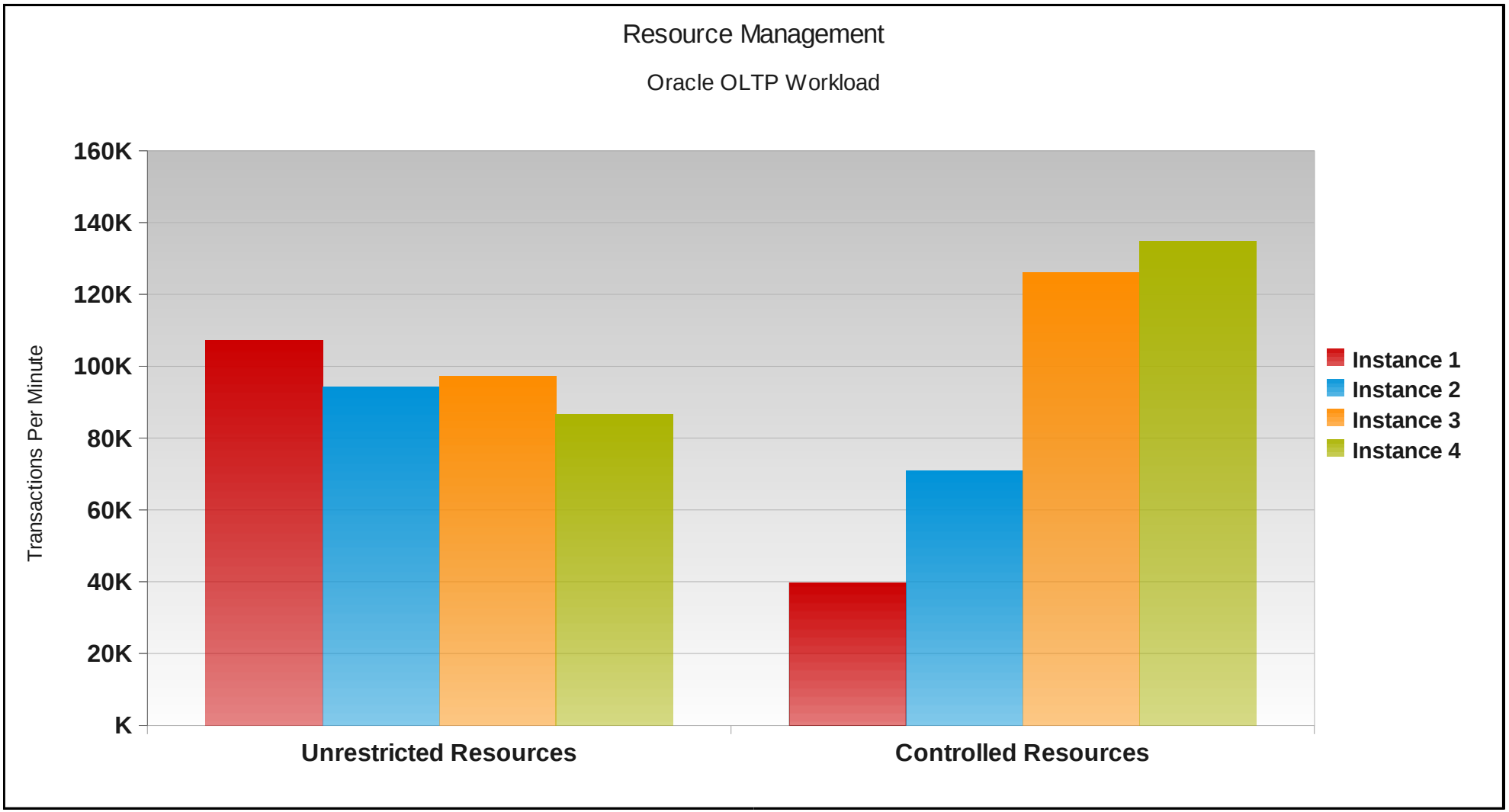

cgroups

```
[root@dhcp-100-19-50 ~]# memory 2GB &
[root@dhcp-100-19-50 ~]# vmstat 1
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu------
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0      0 15465636  33636 459612    5   67    16    68   46   27  1  0 99  0  0
 0  0      0 15465504  33636 459612    0    0     0     0  246  160  0  0 100  0  0
 1  0      0 14598736  33636 459612    0    0     0     0 1648  299  1  5 94  0  0
 1  0 114092 14484980  33636 459528    0 114176     0 114176 2974 1031  0  6 82 12  0
 0  1 264672 14479896  33636 459508    0 150496     0 150496 2630  568  0  2 90  7  0
 0  1 375612 14479524  33636 459612    0 110940     0 110940 2301  322  0  4 76 19  0
 0  1 500064 14477788  33636 459692    0 124452     0 124452 1869  273  0  2 91  7  0
 1  0 609908 14477540  33636 459628    0 109888     0 109888 1960  198  0  8 76 15  0
```

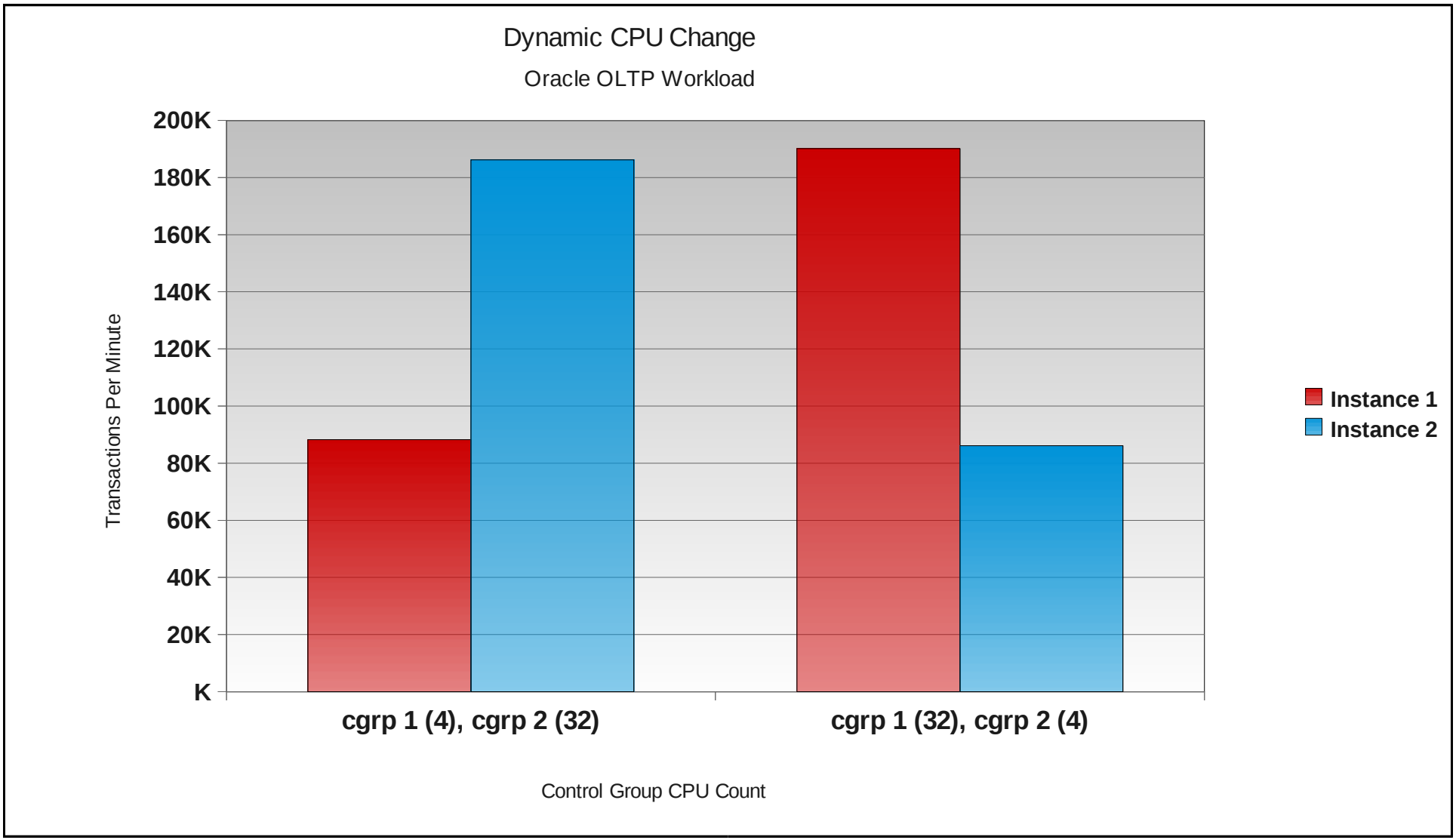

cgroups

```
[root@dhcp-100-19-50 ~]# forkoff 20MB 100procs &
[root@dhcp-100-19-50 ~]# top -d 5
top - 12:24:13 up  1:36,  4 users,  load average: 22.70, 5.32, 1.79
Tasks: 315 total,  93 running, 222 sleeping,   0 stopped,   0 zombie
Cpu0  :  0.0%us,  0.2%sy,  0.0%ni, 99.8%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Cpu1  :  0.0%us,  0.2%sy,  0.0%ni, 99.8%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Cpu2  :100.0%us,  0.0%sy,  0.0%ni,  0.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Cpu3  : 89.6%us, 10.0%sy,  0.0%ni,  0.0%id,  0.0%wa,  0.2%hi,  0.2%si,  0.0%st
Cpu4  :  0.4%us,  0.6%sy,  0.0%ni, 98.8%id,  0.0%wa,  0.0%hi,  0.2%si,  0.0%st
Cpu5  :  0.4%us,  0.0%sy,  0.0%ni, 99.2%id,  0.0%wa,  0.0%hi,  0.4%si,  0.0%st
Cpu6  :  0.0%us,  0.0%sy,  0.0%ni,100.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Cpu7  :  0.0%us,  0.0%sy,  0.0%ni, 99.8%id,  0.0%wa,  0.0%hi,  0.2%si,  0.0%st
Mem:  16469476k total,  1993064k used, 14476412k free,    33740k buffers
Swap:  2031608k total,   185404k used,  1846204k free,   459644k cached
```

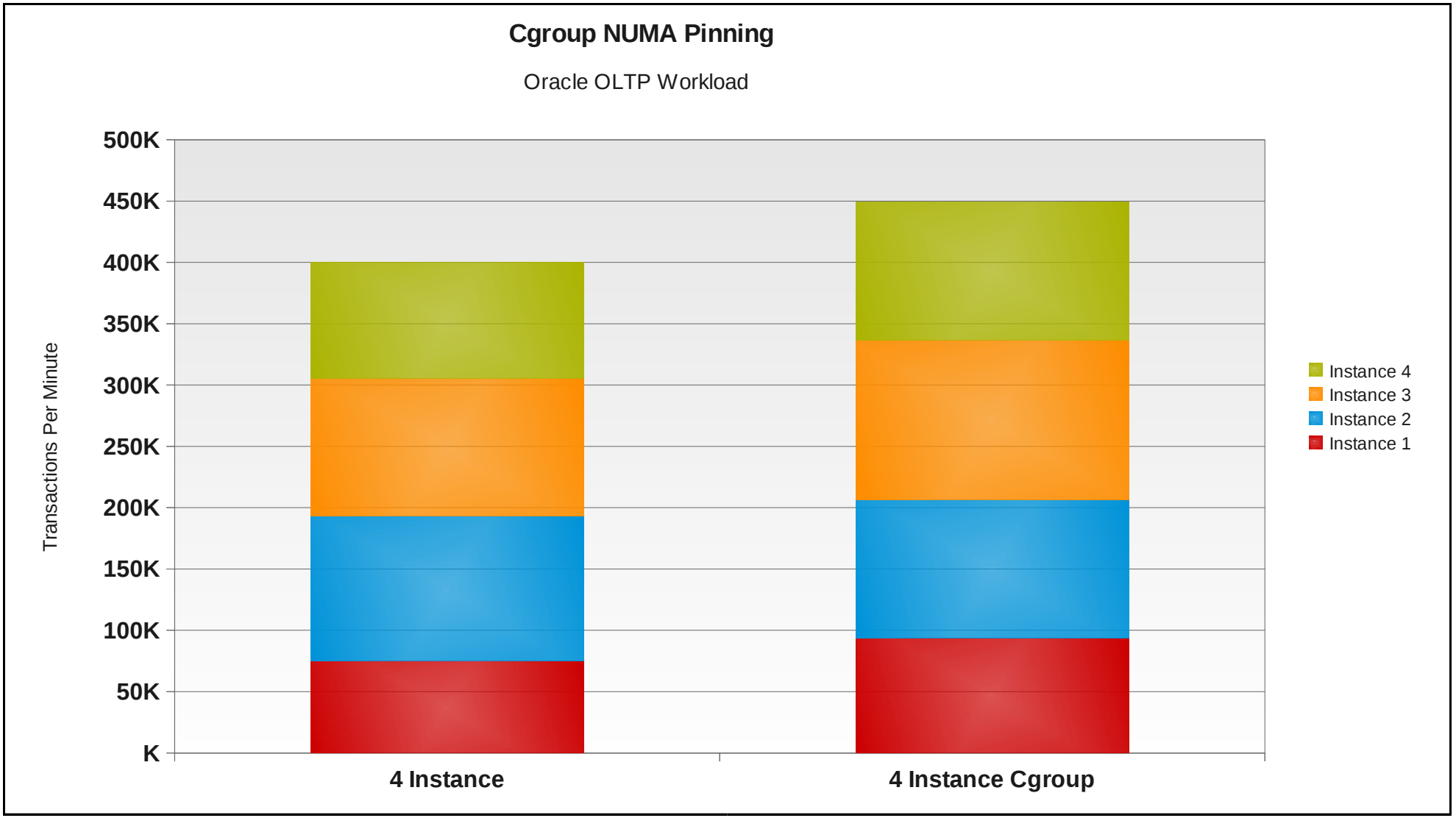

Cgroup – Resource management



C-group Dynamic resource control



Cgroup – NUMA pinning



Hugepages

- Standard HugePages
  - 2MB
  - Reserve/free via /proc/sys/vm/nr_hugepages
  - Used via hugetlbfs
- GB Hugepages
  - 1GB
  - Reserved at boot time/no freeing
  - Used via hugetlbfs
- Transparent HugePages
  - 2MB
  - On by default via boot args or /sys
  - Used for anonymous memory

Huge Pages

The Translation Lookaside Buffer (TLB) is a small CPU cache of recently used virtual to physical address mappings

TLB misses are extremely expensive on today's very fast, pipelined CPUs

Large memory applications
can incur high TLB miss rates

HugeTLBs permit memory to be managed in very large segments

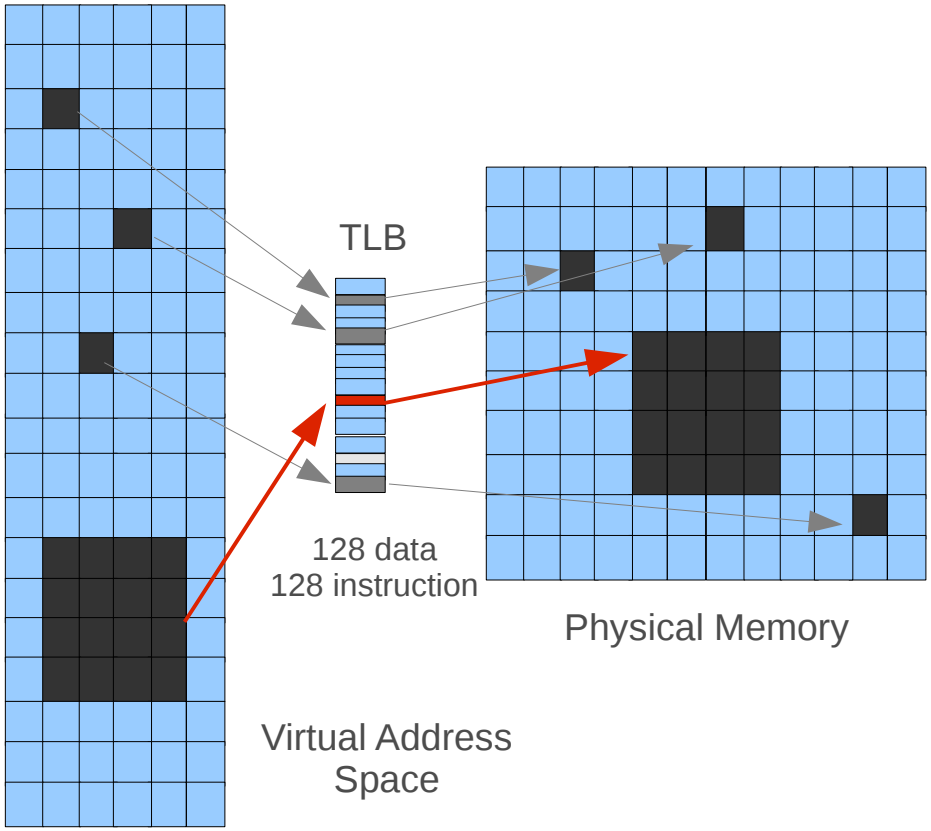
Example: x86_64

- Standard page: 4KB
- Huge page: 2MB
  - 512:1 difference
- 1GB Hugepage
  - 262144:1 difference

File system mapping interface

Example: 128 entry TLB can fully map 256MB

* RHEL6 – 1GB hugepage support



Transparent Hugepages

- Boot argument: transparent_hugepages=always (enabled by default)
- Dynamic:
  - # echo always > /sys/kernel/mm/redhat_transparent_hugepage/enabled

```
[root@dhcp-100-19-50 code]# time ./memory 15GB
   real     0m7.024s
   user     0m0.073s
   sys      0m6.847s
```

```
[root@dhcp-100-19-50 ~]# cat /proc/meminfo
  ...
  AnonHugePages:     15572992 kB
  ...
```


RHEL6 Transparent Hugepages

- echo never > /sys/kernel/mm/transparent_hugepages=never

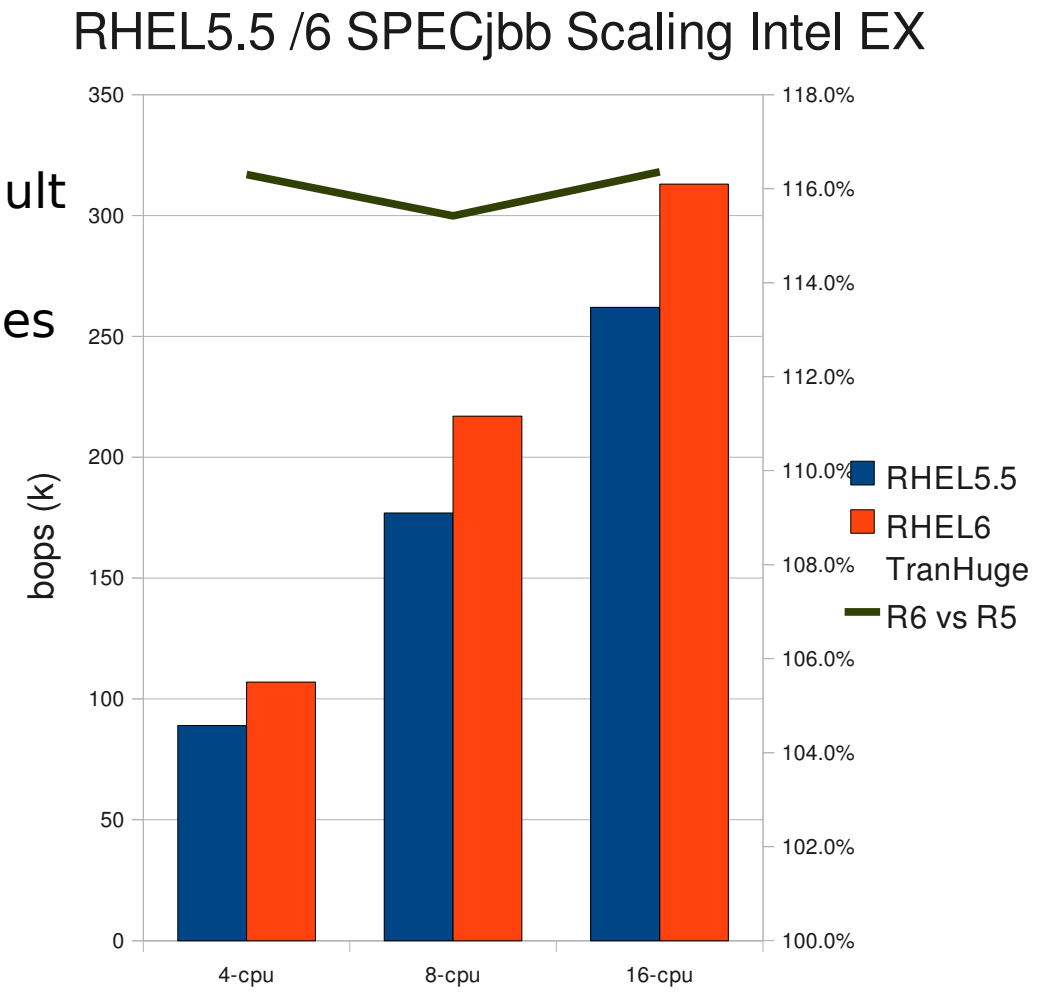
```
[root@dhcp-100-19-50 code]# time ./memory 15 0
real    0m12.434s
user    0m0.936s
sys     0m11.416s
```

```
[root@dhcp-100-19-50 ~]# cat /proc/meminfo
AnonHugePages:     0 kB
```

SPEEDUP 12.4/7.0 = 1.77x, 56%

Performance – R6 vs R5 “out-of-the box” Specjbb Java – Transparent Huge Pages

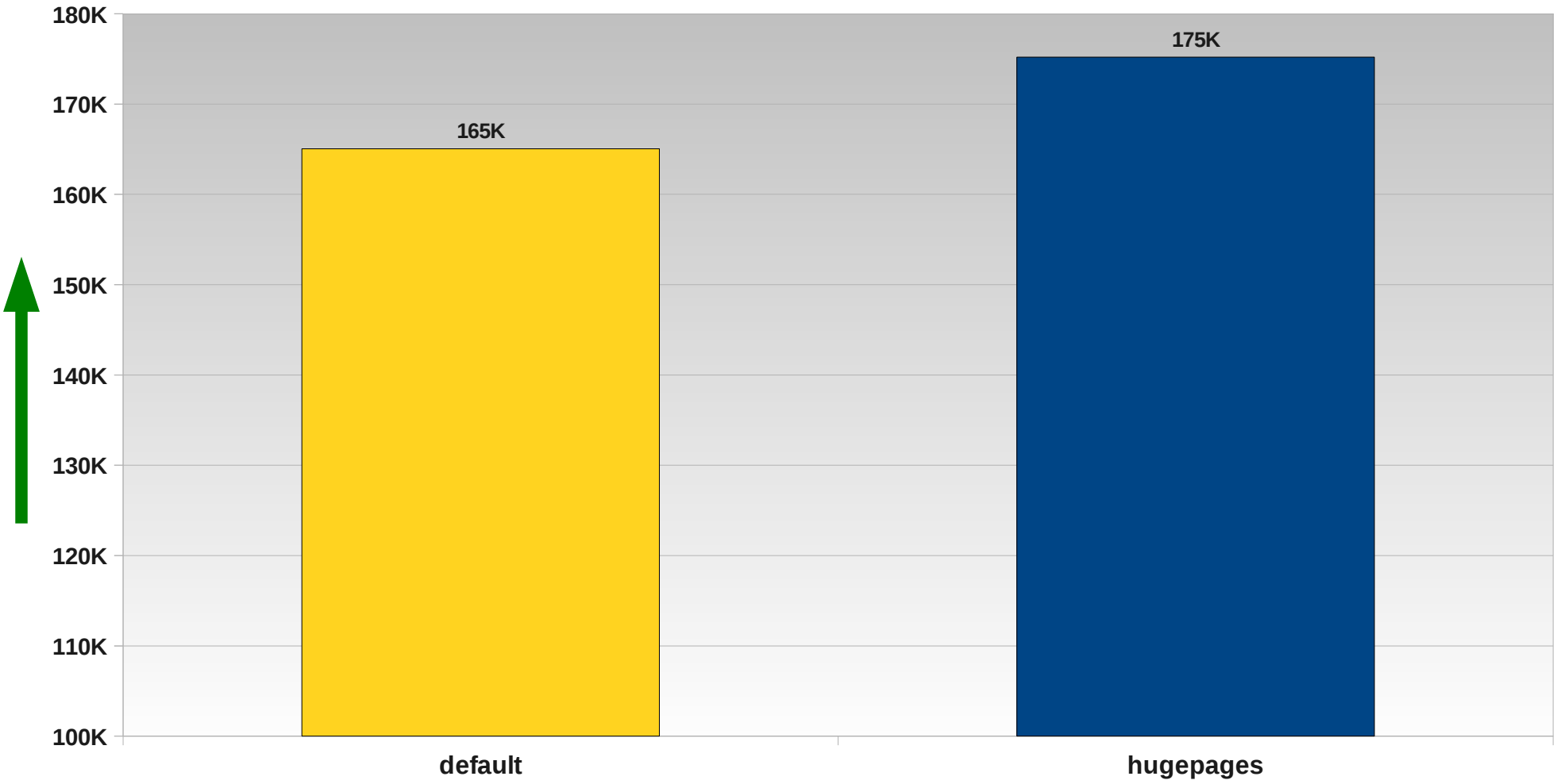
- Transparent Huge pages (2.6.31)
  - Use 2M x86_64 page vs 4k default page
  - < RHEL6, static use of hugepages
    - Static pages wired-down
    - Need application support DB/Java etc
  - Automatically use huge pages
    - For all anonymous memory
  - Daemon to gather free dynamically



Memory Tuning – Huge Pages – Sybase - OLTP

Sybase Huge Pages Testing - RHEL 5.5

OLTP transactional throughput on a Quad Core 4 Socket 2.5Ghz – 96G Physical



1GB Hugepages

Boot arguments -

- default_hugepagesz=1G
- hugepagesz=1G
- hugepages=8

cat /proc/meminfo | more

```
HugePages_Total:       8
HugePages_Free:        8
HugePages_Rsvd:        0
HugePages_Surp:        0
Hugepagesize:    1048576 kB
DirectMap4k:        7104 kB
DirectMap2M:     2088960 kB
DirectMap1G:    14680064 kB
```


1GB Hugepages

#mount -t hugetlbfs none /mnt

```
# ./mmapwrite /mnt/junk 33
 writing 2097152 pages of random junk to file /mnt/junk
 wrote 8589934592 bytes to file /mnt/junk
```

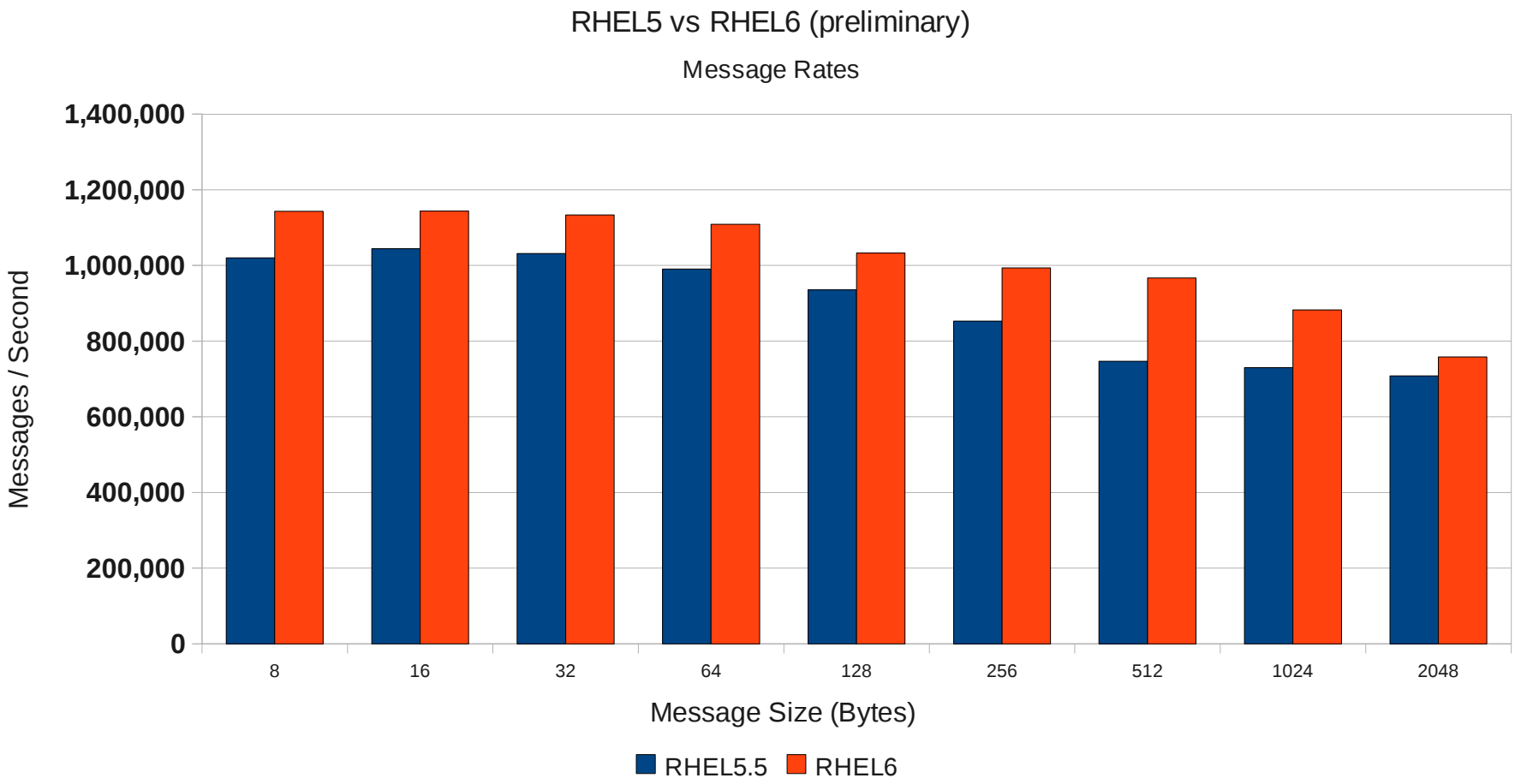
cat /proc/meminfo | more

```
HugePages_Total:       8
HugePages_Free:        0
HugePages_Rsvd:        0
HugePages_Surp:        0
Hugepagesize:    1048576 kB
DirectMap4k:        7104 kB
DirectMap2M:     2088960 kB
DirectMap1G:    14680064 kB
```


RHEL6 Technology Innovation

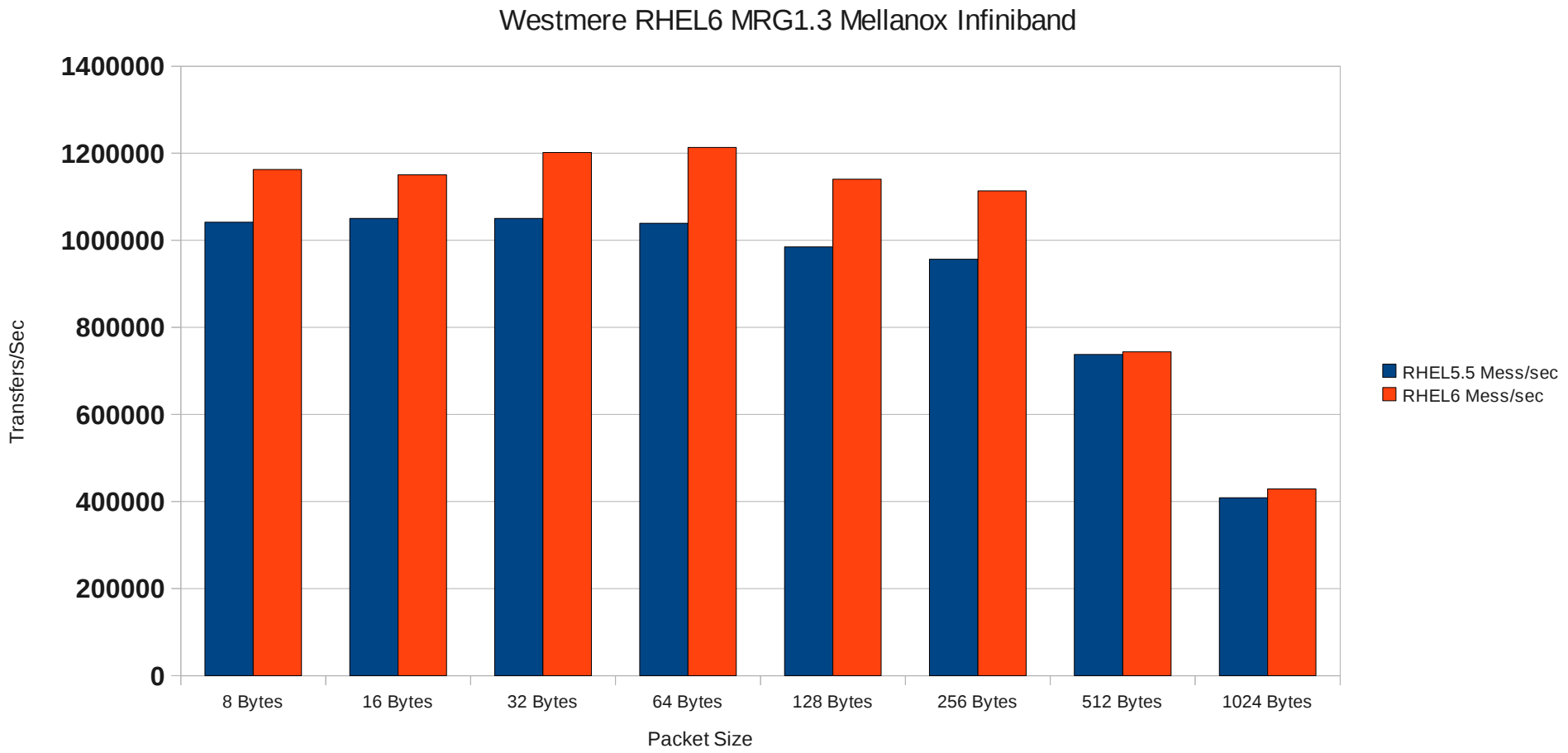
- **Networking**
  - Multi-queue
  - Tools to monitor dropped packets – tc, dropwatch.
  - RCU adoption in stack
  - Multi-CPU receive to pull in from the wire faster.
  - 10GbE driver improvements.
  - Data center bridging in ixbge driver.
  - FcoE performance improvements throughout the stack.

RHEL6 vs RHEL5 10Gbit AMQP TCP/IP Perftest (Messages / Sec - Bigger=Better)

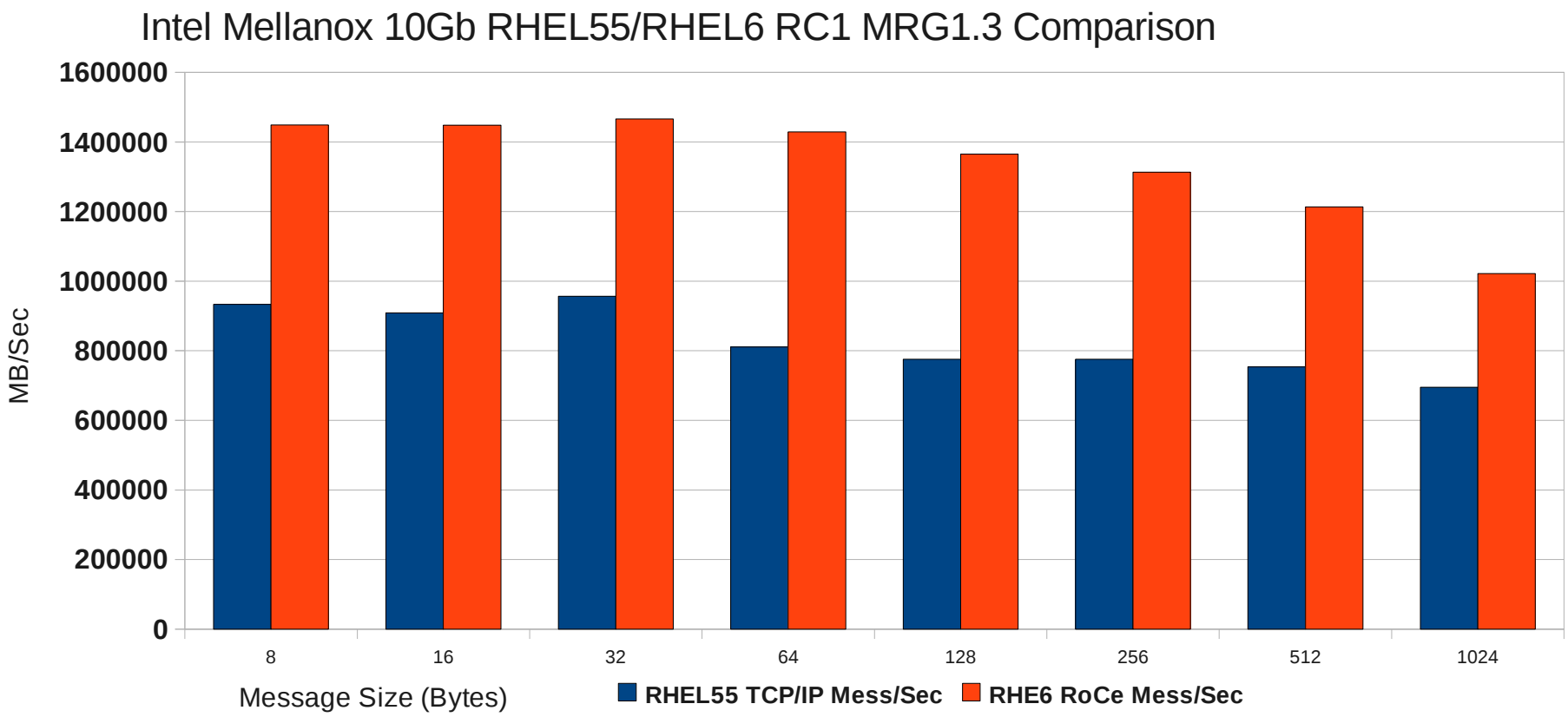


10 Gbit Ethernet (Mellanox)

RHEL6 vs RHEL5.5 IB RDMA w/ AMQP Perftest (Messages / Sec - Bigger = Better)



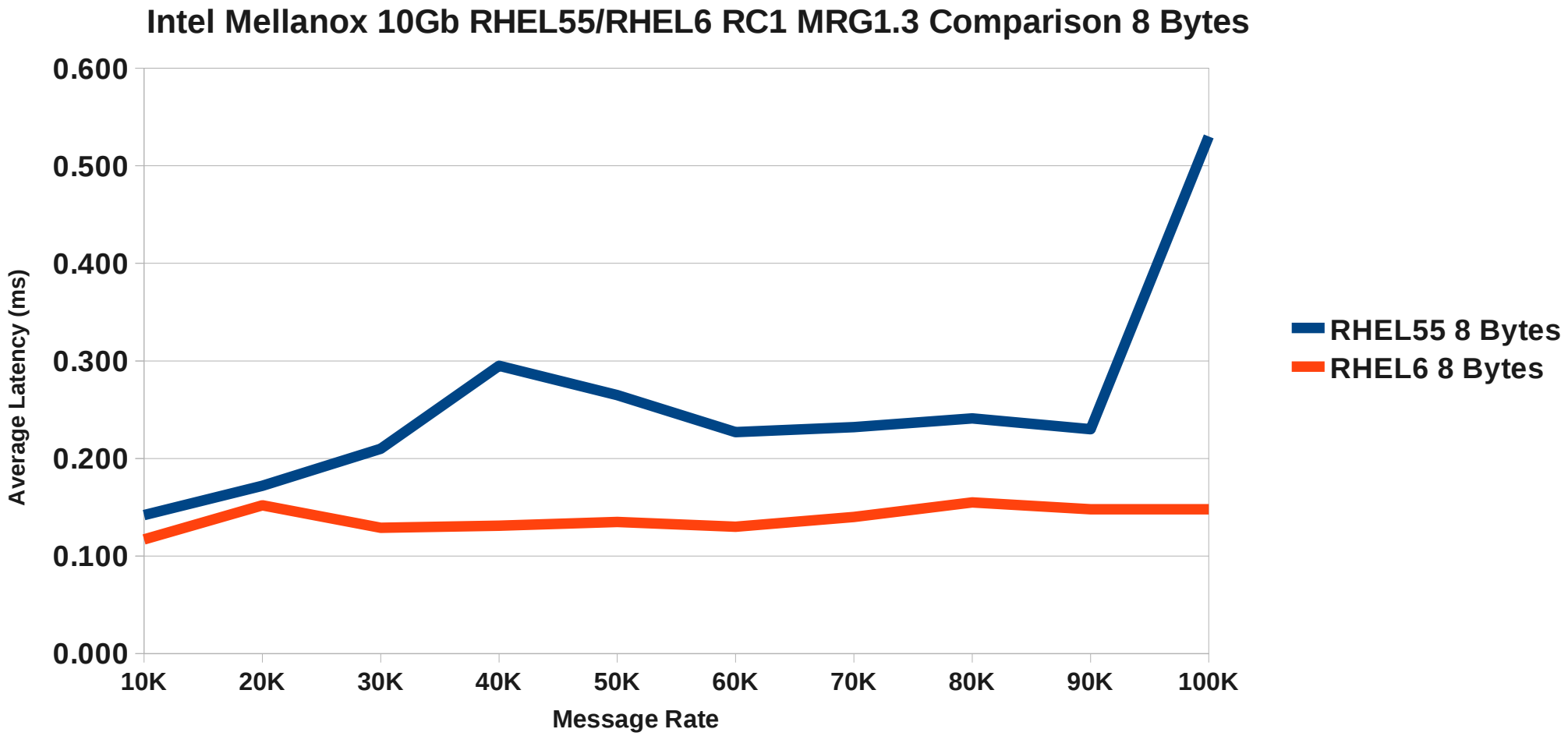
RHEL5.5 IB to RHEL6 AMQP w/ RoCE mess/sec and MB/sec (Bigger=Better)



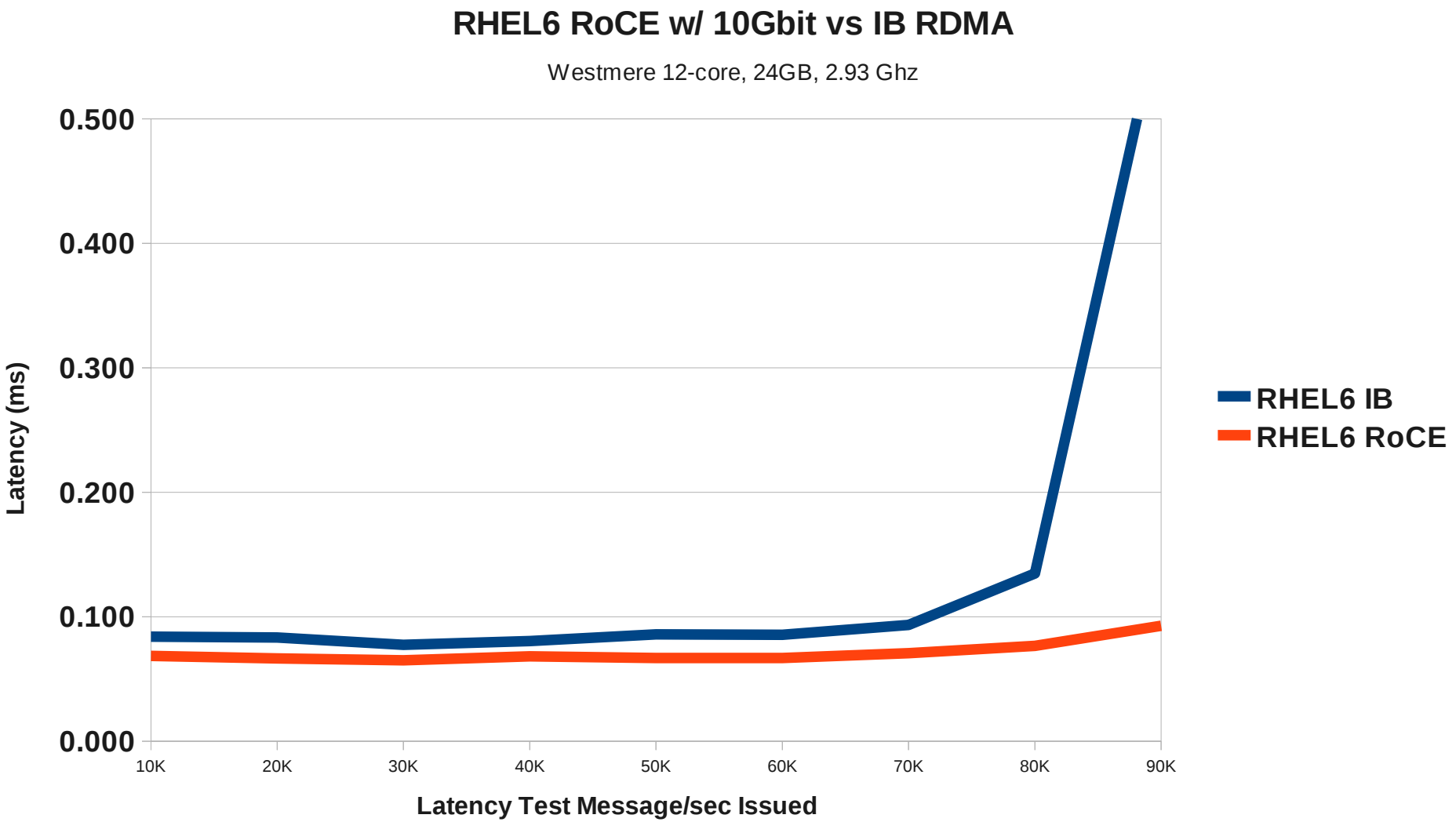
10 Gbit Ethernet (Mellanox)

RHEL5.5 to RHEL6 AMQP TCP Latency

(smaller=better, 8 byte packet sizes)



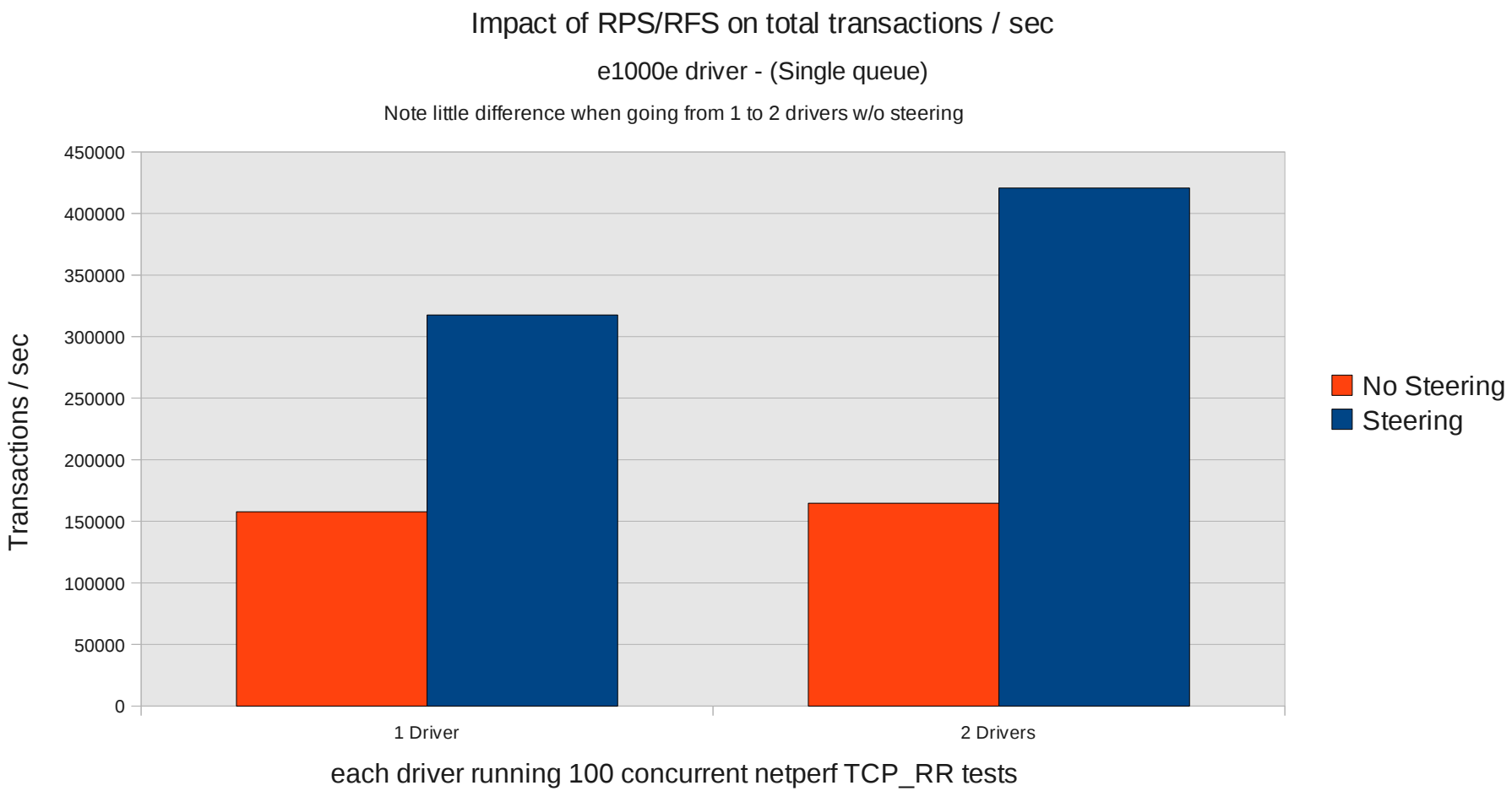
RHEL6 IB vs 10Gb RDMA IP vs Converged Ethernet (RoCE) (Latency msec - Smaller = Better)



Networking

- Receive Packet Steering (RPS) breaks the bottleneck of having to receive network traffic for a NIC on one CPU
- Receive Flow Steering (RFS) allows the optimal CPU to receive network data intended for a specific application
- Add getsockopt support for TCP thin-streams to reduce latency from retransmission of lost packets in time-sensitive applications
- Add Transparent Proxy (TProxy) support for non-locally bound IPv4 TCP and UDP sockets (similar to Linux 2.2)
  - Allows packet interception and serving of response without client reconfiguration (transparent to client)

RHEL6.1 Network packet / flow steering - Greatly improves messages / sec rate



Tuning

- Tuning can provide excellent improvements
  - Steps are different for throughput vs latency, goal is the same.
    - Try to maximize CPU cache hits and localize memory
    - Use NUMA if possible
      - *numactl -c1 -m1 /root/qpid/cpp/src/qpidd --auth no -m no --pid-dir /var/run/qpidd --data-dir /var/lib/qpidd --load-module /root/qpid/cpp/src/.libs/rdma.so -P rdma*
    - Move IRQ handlers as needed
    - Understand the NIC parameters, tune as necessary

Section 2: System Overview

CPU support

NUMA support

Physical memory

Memory management

I/O

Processors Supported/Tested

RHEL4

- x86 – 32
- x86_64 – 8, 64(LargeSMP)
- ia64 – 64, 512(SGI)

RHEL5

- x86 – 32
- x86_64 – 255
- ia64 – 64, 1024(SGI)

RHEL6

- x86 - 32
- x86_64 - 4096

Processor types & locations

```
[root@intel-s3e36-01 node1]# cat /proc/cpuinfo
processor       : 0  <logical cpu #>

physical id     : 0  <socket #>

siblings        : 16 <logical cpus per socket>

core id         : 0  <core # in socket>

cpu cores       : 8  <physical cores per socket>


# cat /sys/devices/system/node/node*/cpulist
node0: 0-3
node1: 4-7
```


Physical Memory Supported/Tested

RHEL4

- x86 – 4GB, 16GB, 64GB
- x86_64 – 512GB
- ia64 – 1TB

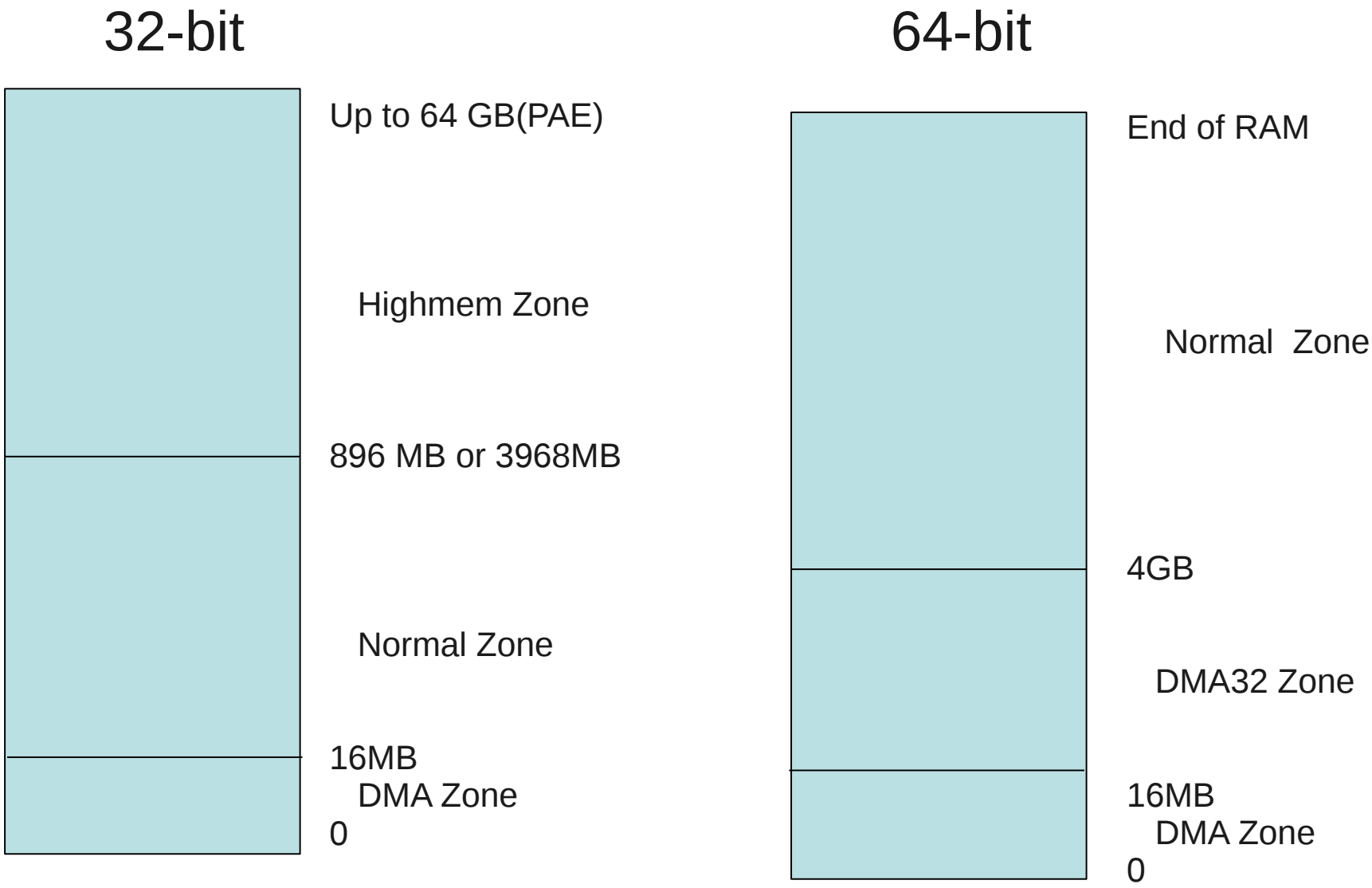
RHEL5

- x86 – 4GB, 16GB
- x86_64 – 1TB
- ia64 – 2TB

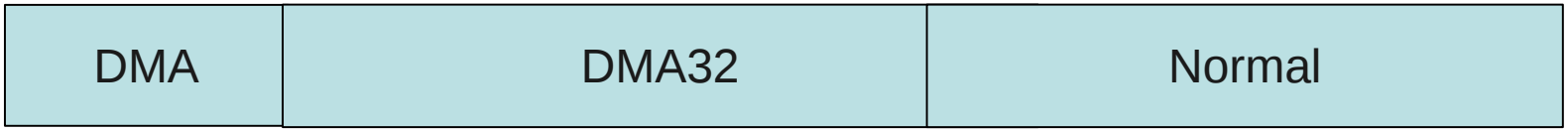
RHEL6

- x86 – 16GB
- x86_64 – 64TB/8TB(in progress)

Memory Zones



Memory Zone Utilization(x86_64)



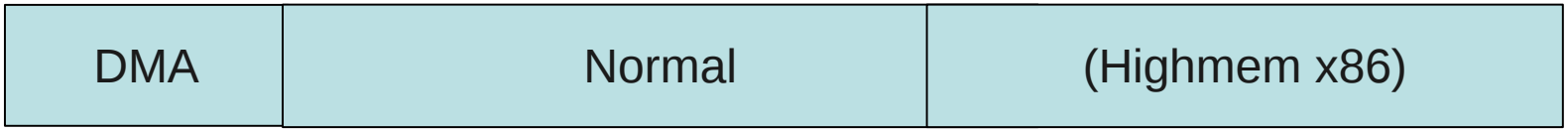
24bit I/O

32bit I/O
Normal overflow

Kernel Static
Kernel Dynamic
- **slabcache**
- **bounce buffers**
- **driver allocations**

User
- **Anonymous**
- **Pagecache**
- **Pagetables**

Memory Zone Utilization(x86)



24bit I/O

Kernel Static
Kernel Dynamic
- **slabcache**
- **bounce buffers**
- **driver allocations**

User Overflow

User
- **Anonymous**
- **Pagecache**
- **Pagetables**

Per-zone page lists

Active List - most recently referenced

- Anonymous-stack, heap, bss
- Pagecache-filesystem data/meta-data

Inactive List - least recently referenced

- Dirty-modified
- writeback in progress
- Clean-ready to free

Free

- Coalesced buddy allocator

Per zone Free list/buddy allocator lists

Kernel maintains per-zone free list

Buddy allocator coalesces free pages into larger physically contiguous pieces

DMA

1*4kB 4*8kB 6*16kB 4*32kB 3*64kB 1*128kB 1*256kB 1*512kB 0*1024kB 1*2048kB 2*4096kB = 11588kB)

Normal

217*4kB 207*8kB 1*16kB 1*32kB 0*64kB 1*128kB 1*256kB 1*512kB 0*1024kB 0*2048kB 0*4096kB = 3468kB)

HighMem

847*4kB 409*8kB 17*16kB 1*32kB 1*64kB 1*128kB 1*256kB 1*512kB 0*1024kB 0*2048kB 0*4096kB = 7924kB)

Memory allocation failures

Freelist exhaustion.

Freelist fragmentation.

Per NUMA-Node Resources

Memory zones(DMA & Normal zones)

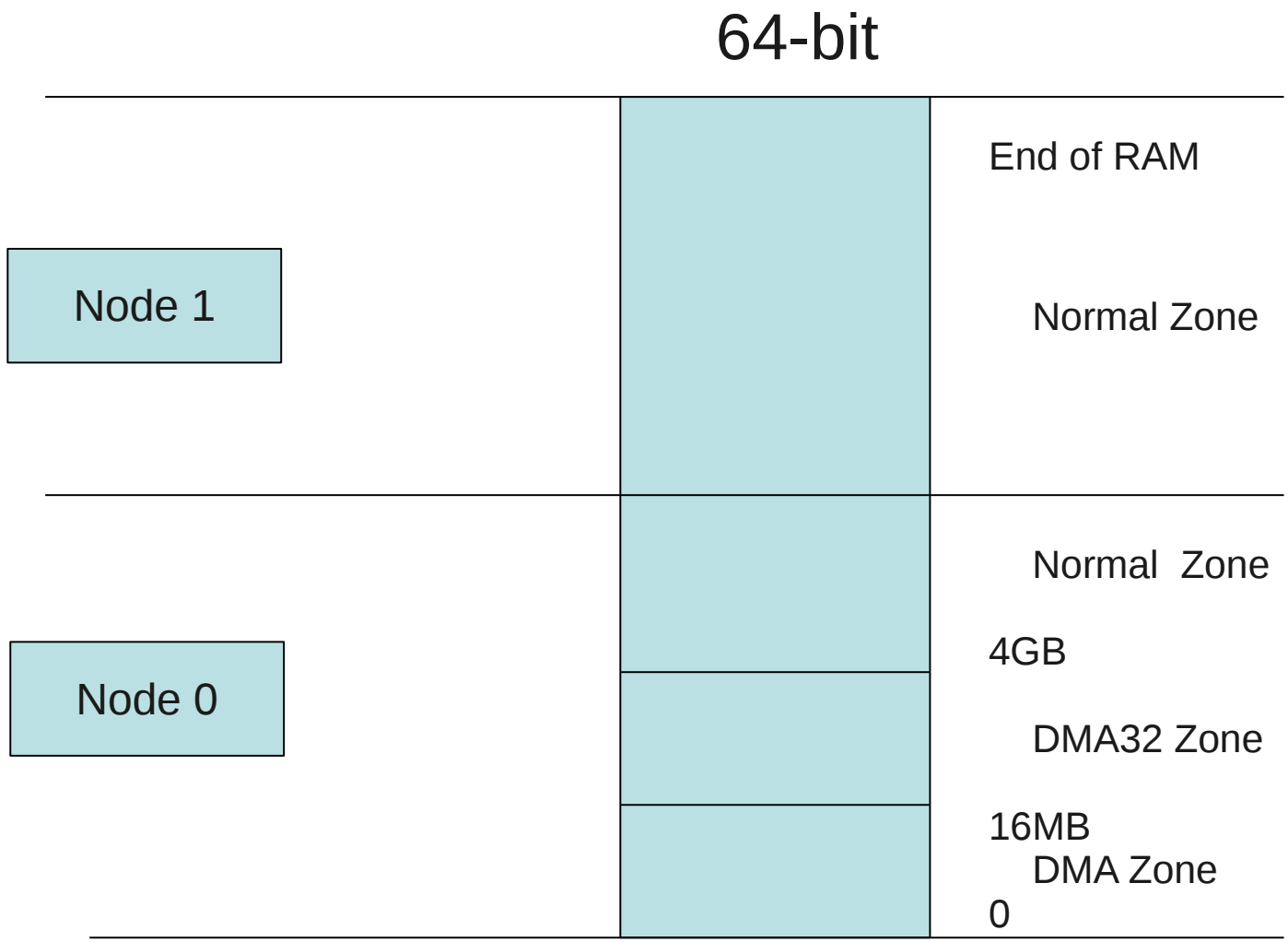
CPUs

IO/DMA capacity

Interrupt processing

Page reclamation kernel thread(kswapd#)

NUMA Nodes and Zones



Virtual Address Space Maps

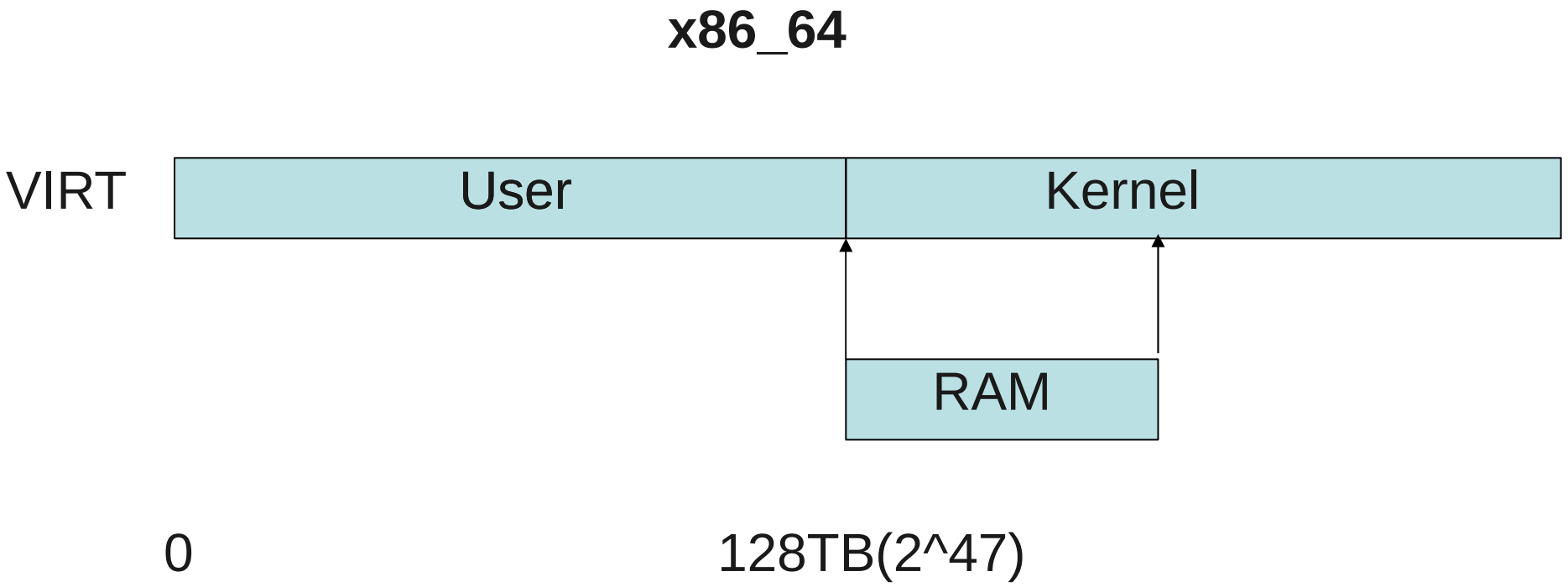
64-bit

- X86_64

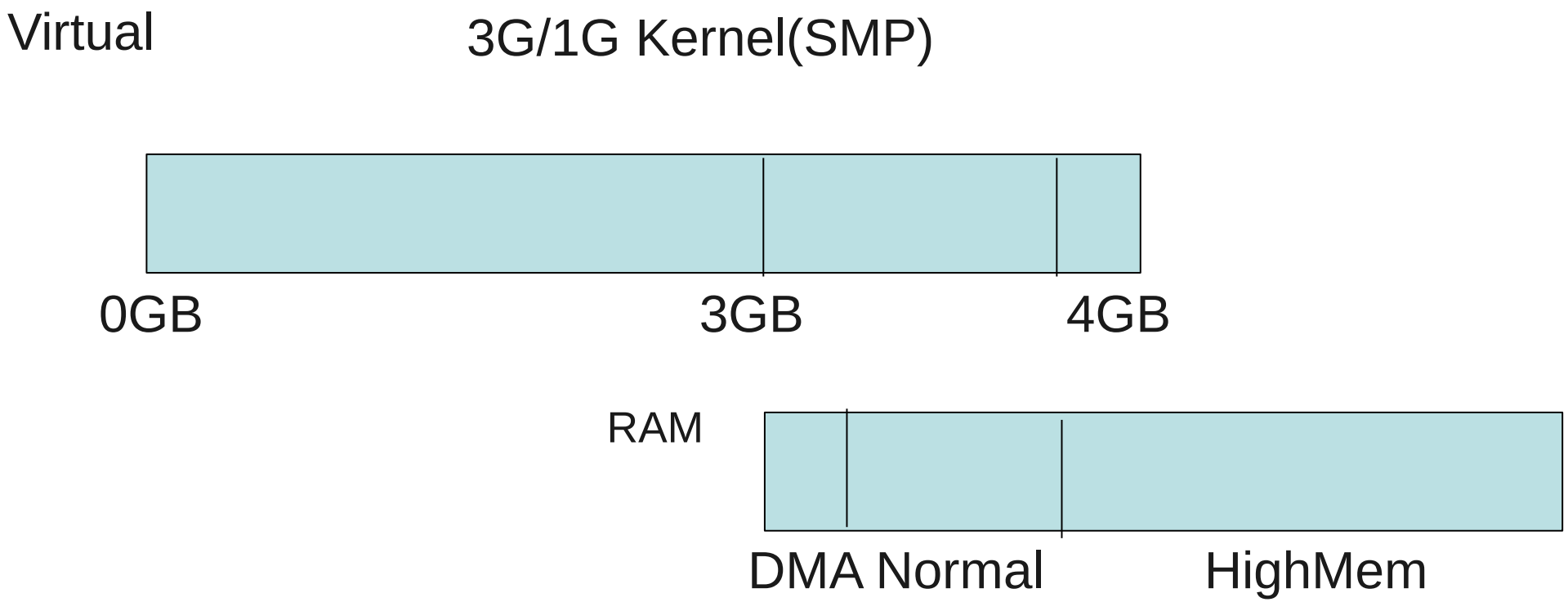
32-bit

- 3G/1G address space
- 4G/4G address space(RHEL4 only)

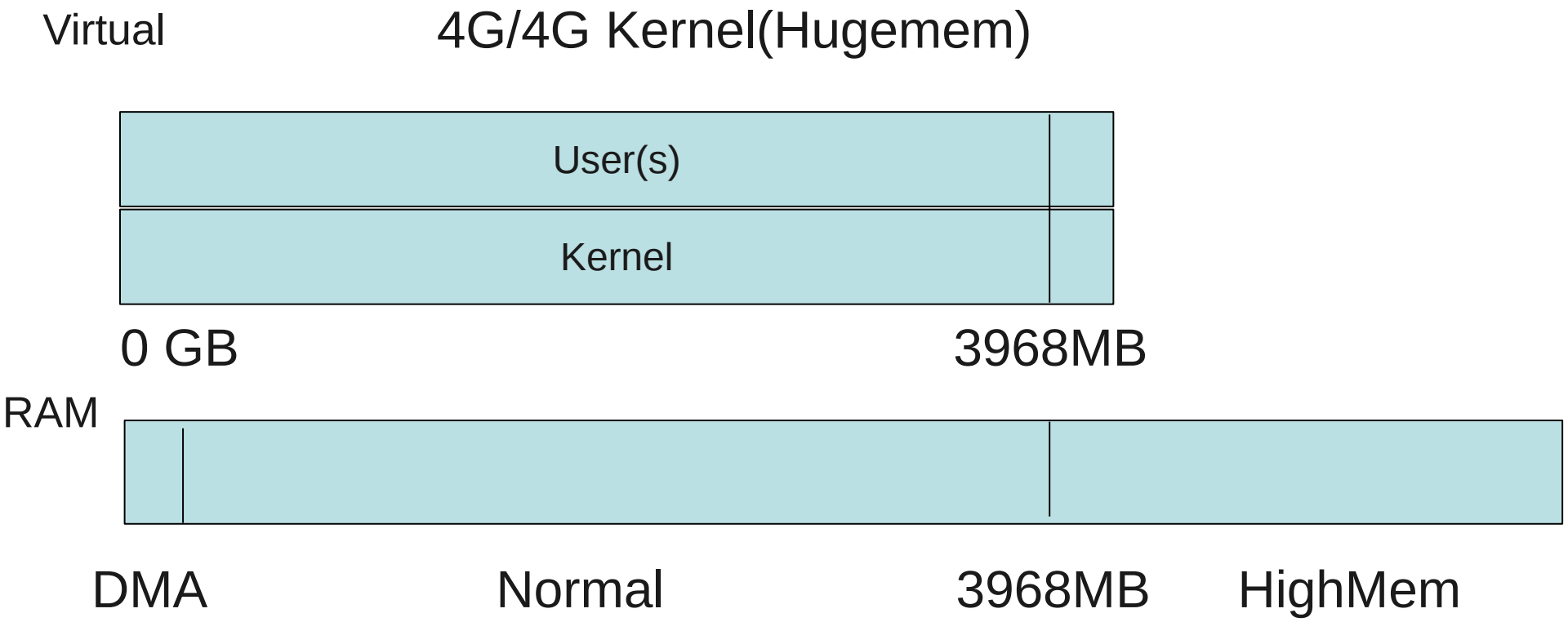
Linux 64-bit Address Space



Linux 32-bit Address Spaces(SMP)



RHEL4 32-bit Address Space(Hugemem)

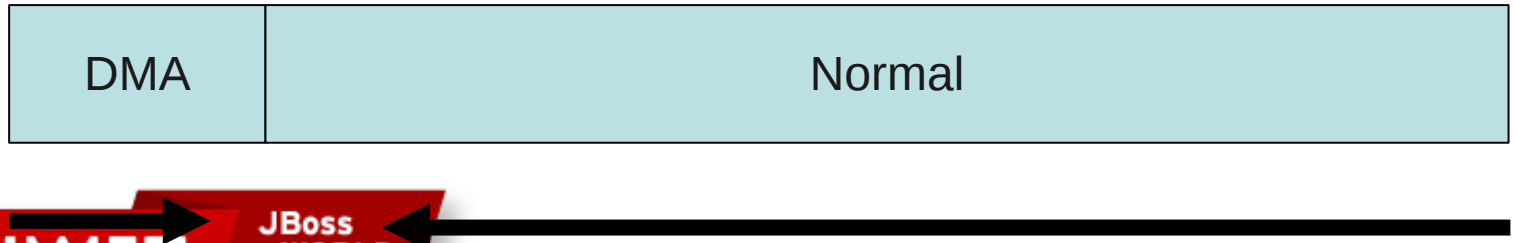


Memory Pressure

32- bit

Kernel Allocations **User Allocations**

64- bit



Kernel Memory Pressure

Static – Boot-time(DMA and Normal zones)

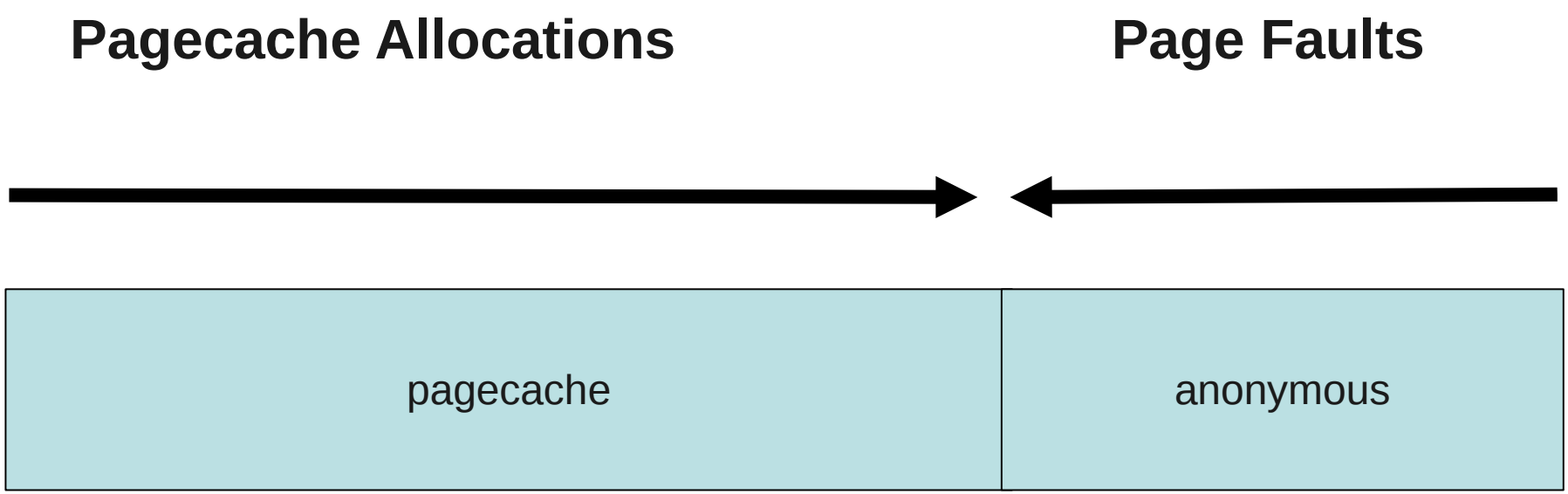
- Kernel text, data, BSS
- Bootmem allocator, tables and hashes(mem_map)

Dynamic

- Slabcache(Normal zone)
- Kernel data structs
- Inode cache, dentry cache and buffer header dynamics
- Pagetables(Highmem/Normal zone)

HughTLBfs(Highmem/Normal zone)

User Memory Pressure
Anonymous/pagecache split



PageCache/Anonymous memory split

Pagecache memory is global and grows when filesystem data is accessed until memory is exhausted.

Pagecache is freed:

- Underlying files are deleted.
- Unmount of the filesystem.
- Kswapd reclaims pagecache pages when memory is exhausted.
- /proc/sys/vm/drop_caches

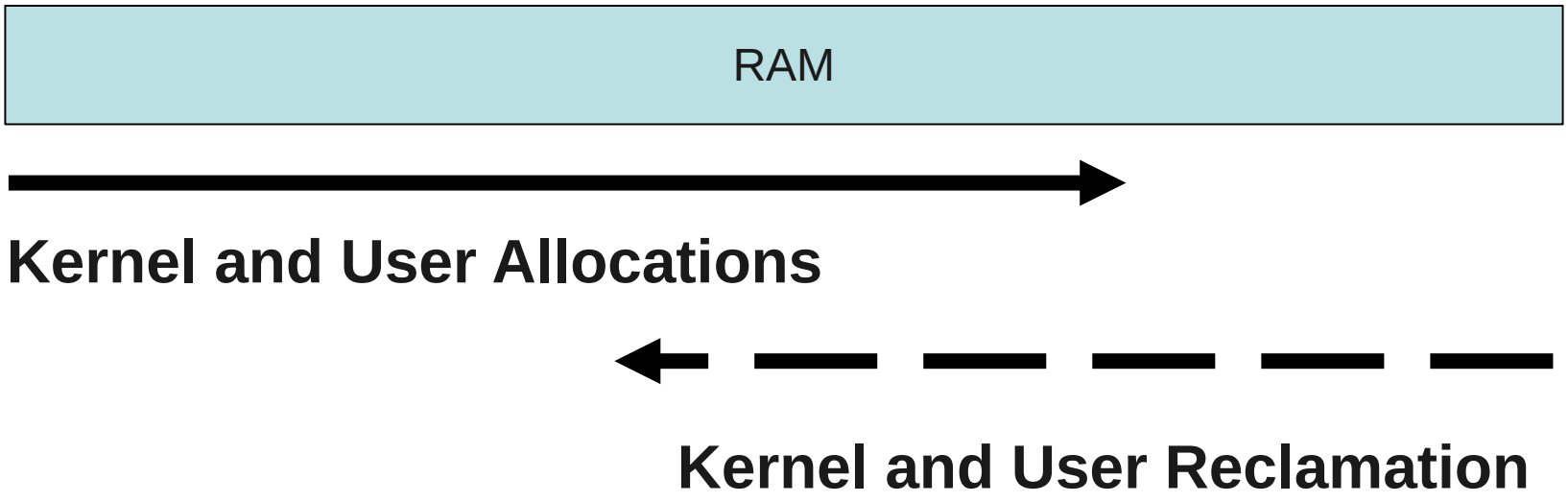
Anonymous memory is private and grows on user demmand

- Allocation followed by pagefault.
- Swapin.

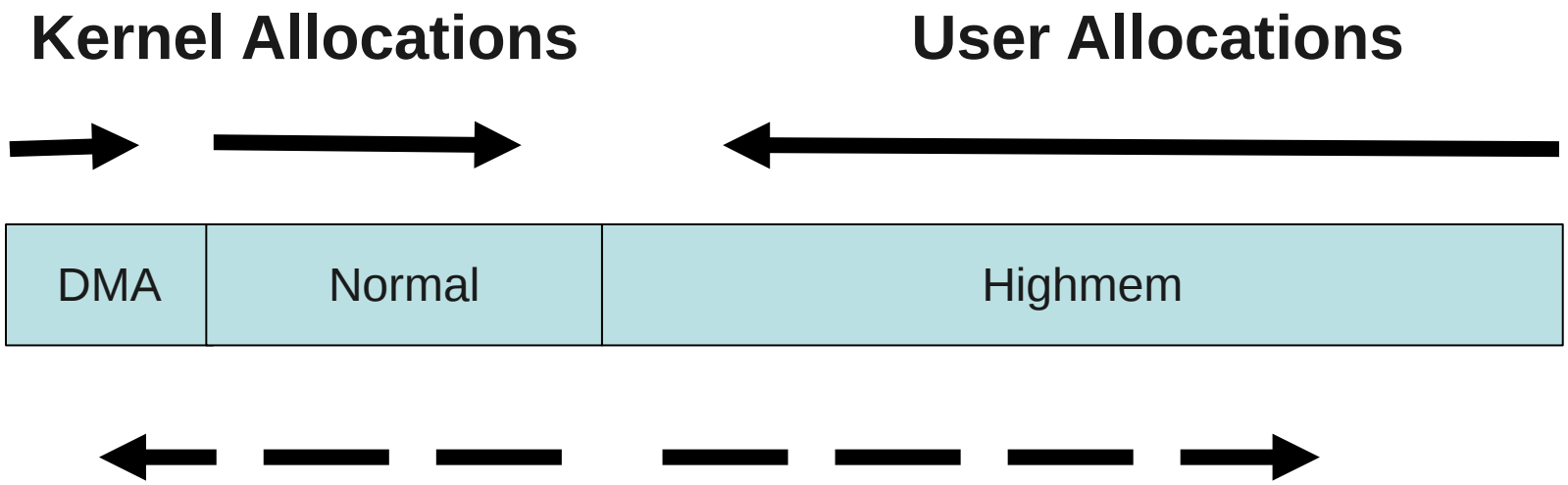
Anonymous memory is freed:

- Process unmaps anonymous region or exits.
- Kswapd reclaims anonymous pages(swapout) when memory is exhausted

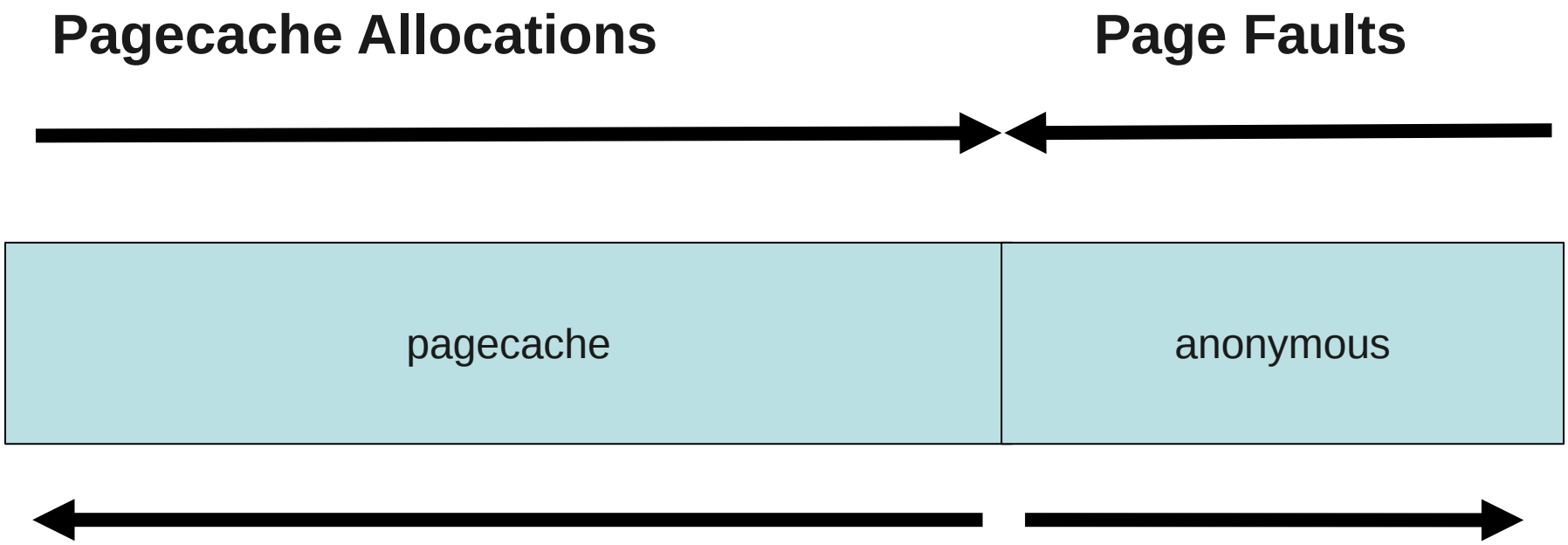
64-bit Memory Reclamation



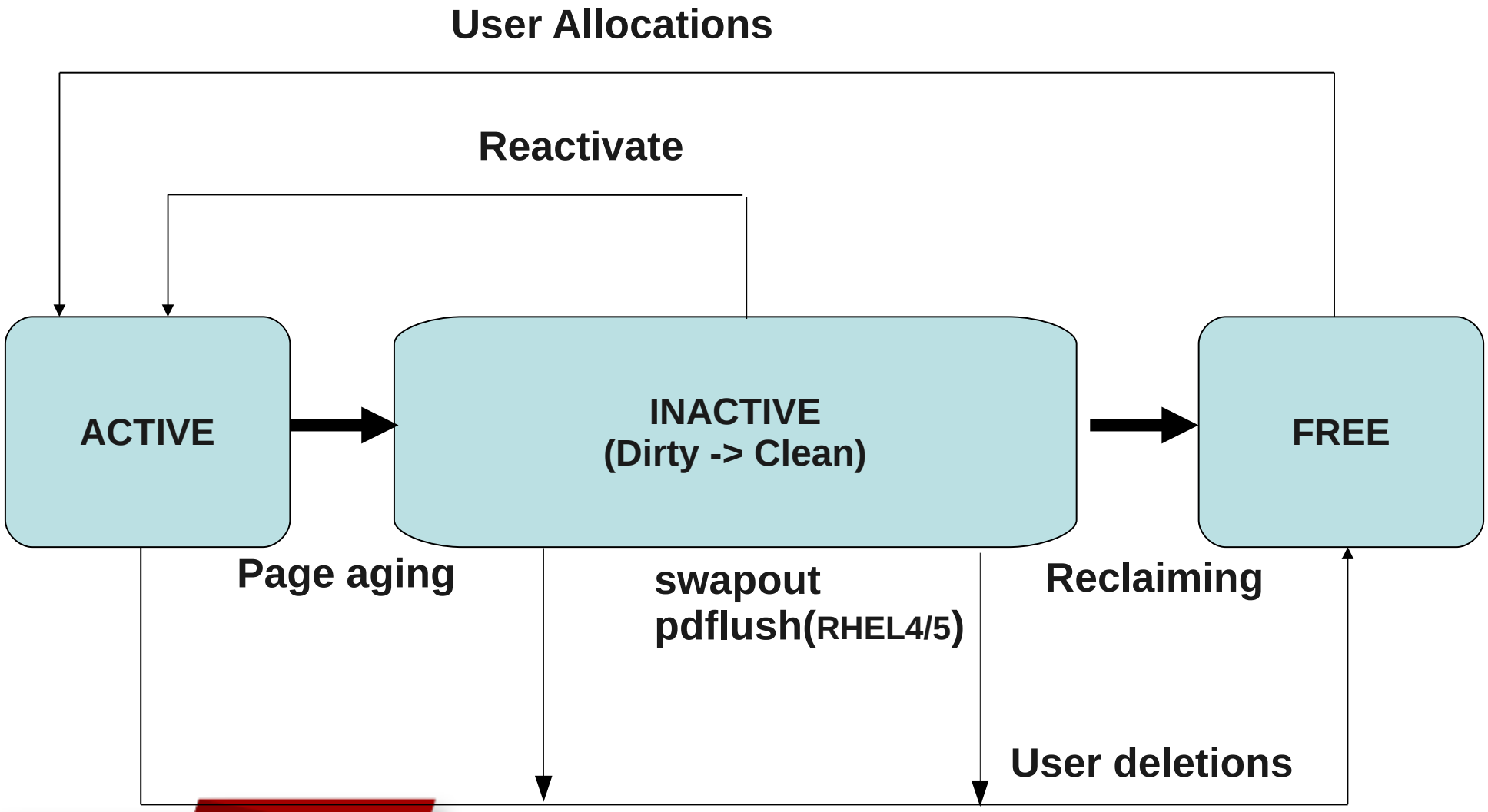
32-bit Memory Reclamation



Anonymous/pagecache reclaiming

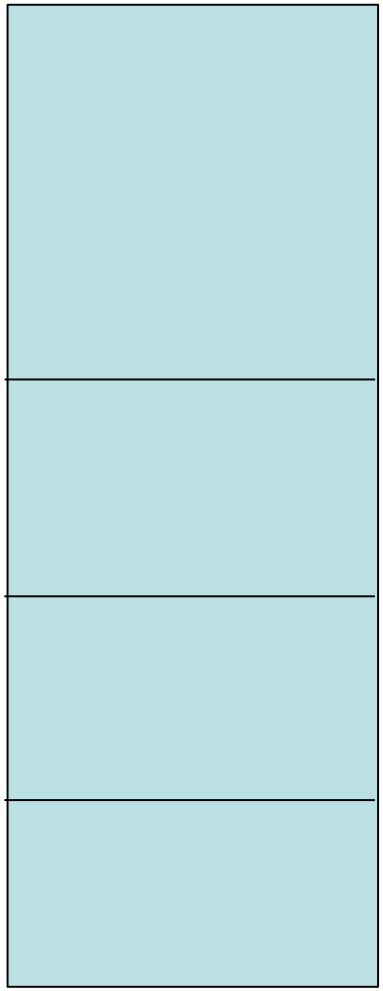


Per Node/Zone Paging Dynamics



Memory reclaim Watermarks

Free List



All of RAM

Do nothing

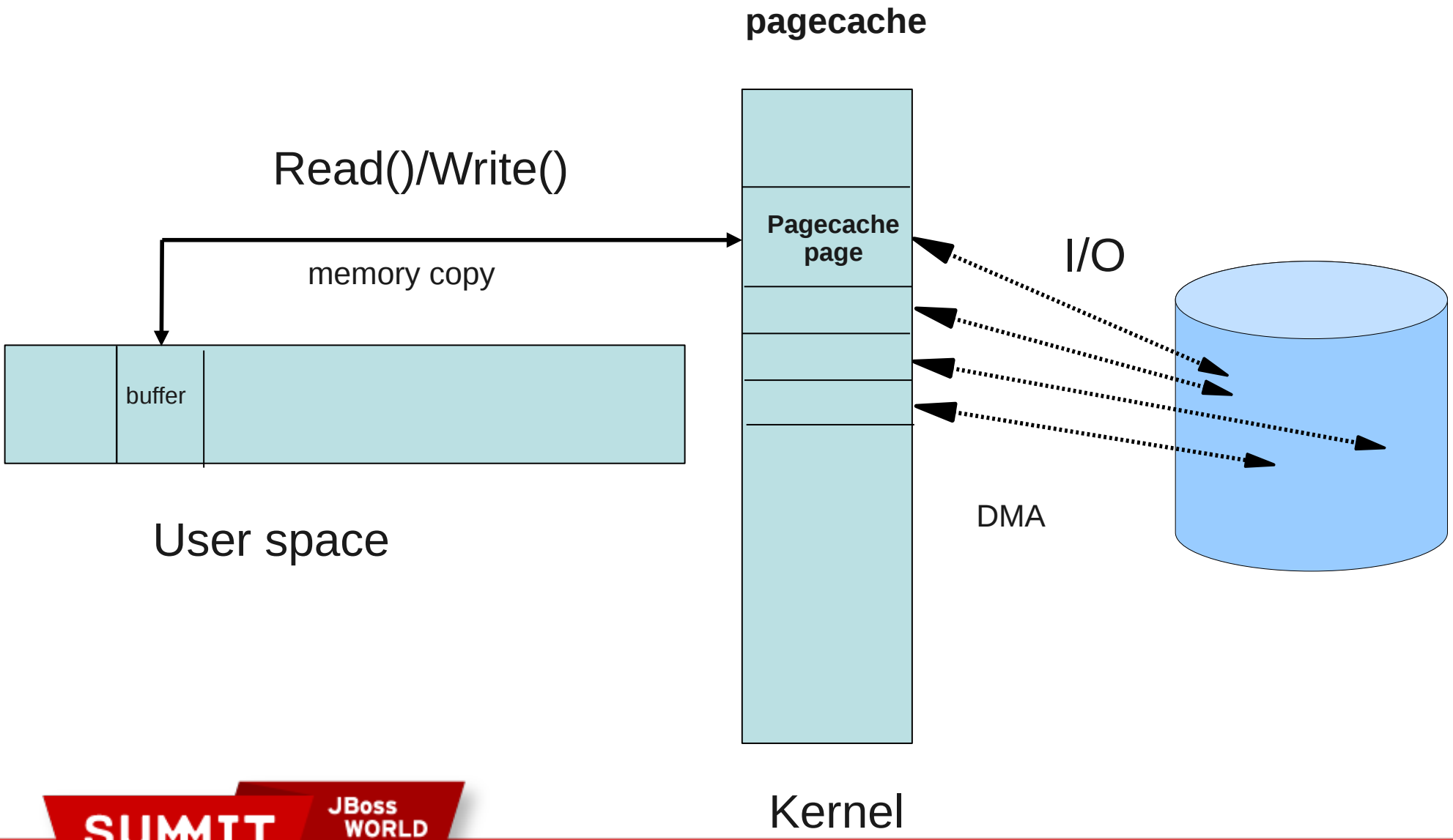
Pages High – kswapd sleeps above High
kswapd reclaims memory

Pages Low – kswapd wakesup at Low
kswapd reclaims memory

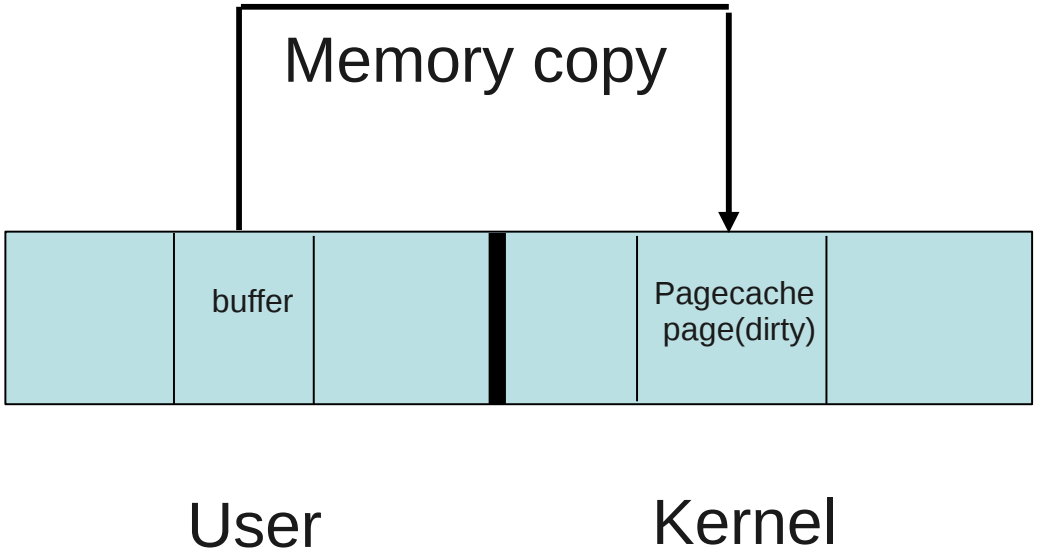
Pages Min – all memory allocators reclaim at Min
user processes/kswapd reclaim memory

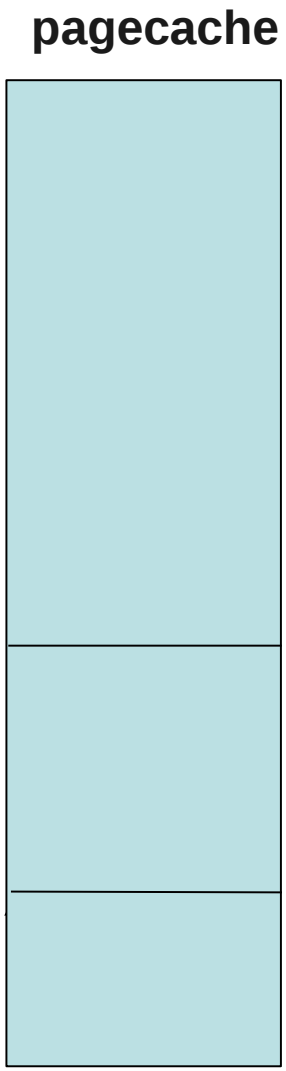
0

File System & Disk IO



Buffered file system write





100% of pagecache RAM dirty

pdflushd and
write()'ng processes
write dirty buffers

40% dirty) – processes start
synchronous writes

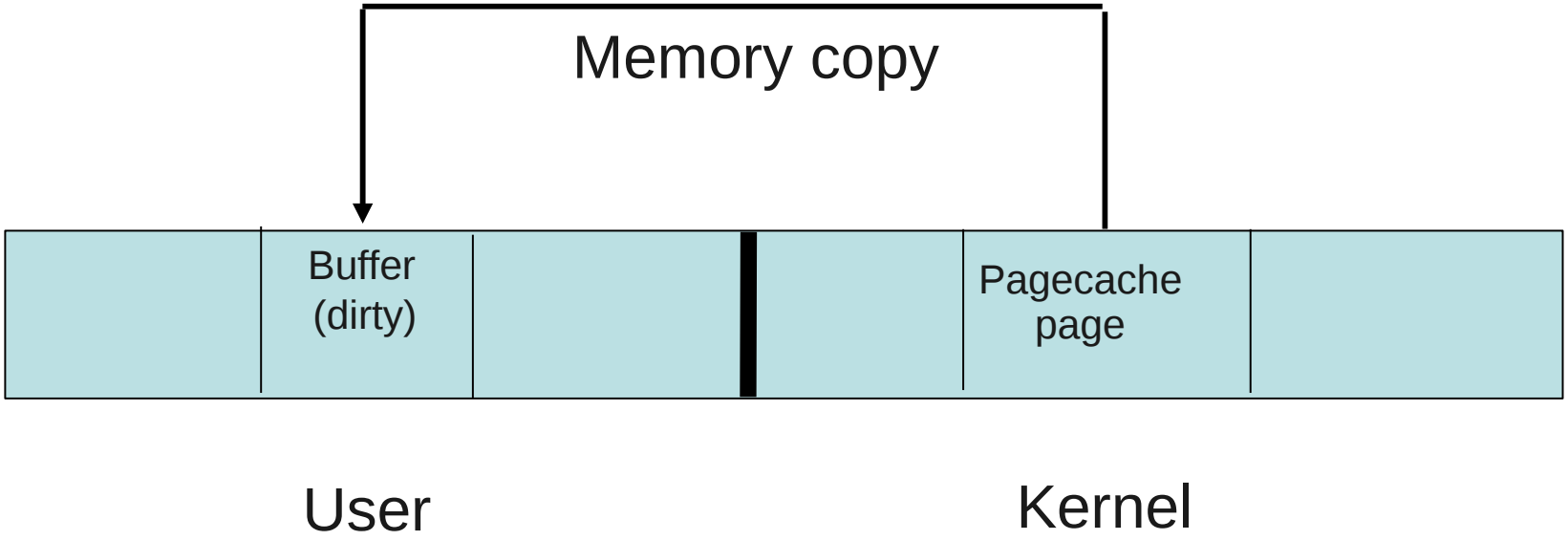
pdflushd writes dirty
buffers in
background

10% dirty – wakeup pdflushd

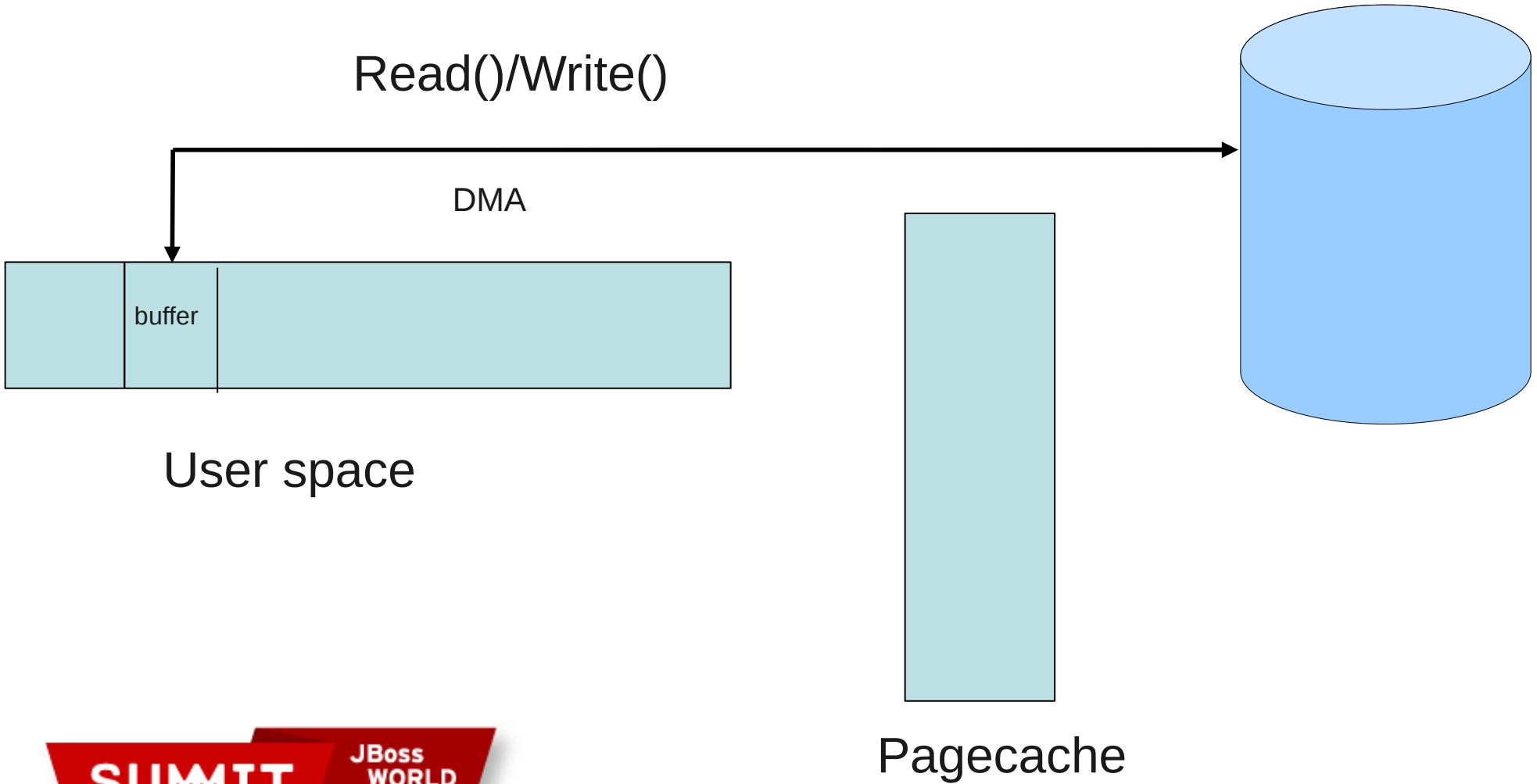
do_nothing

0% dirty

Buffered file system read



DirectIO file system read()/write()



Section 3: Analyzing System Performance

- Performance monitoring tools
- Profiling
- Event tracing
- Performance tools new in RHEL6

Performance Monitoring Tools

- Standard Unix OS tools
  - Monitoring - cpu, memory, process, disk
  - oprofile
- Kernel Tools
  - /proc, info (cpu, mem, slab), dmesg, AltSysrq
- Networking
  - netstat, sar, ethtool, tcpdump, iptraf
- Profiling
  - nmi_watchdog=1, profile=2
  - Tracing strace, ltrace
  - dprobe, kprobe
- 3rd party profiling/ capacity monitoring
  - Perfmon, Caliper, vtune
  - SARcheck, KDE, BEA Patrol, HP Openview

Red Hat Top Tools

- CPU Tools
  - 1 – top
  - 2 – vmstat
  - 3 – ps aux
  - 4 – mpstat -P all
  - 5 – sar -u
  - 6 – iostat
  - 7 – oprofile
  - 8 – gnome-
  - system-monitor
  - 9 – KDE-monitor
  - 10 – /proc
- Memory Tools
  - 1 – top
  - 2 – vmstat -s
  - 3 – ps aur
  - 4 – ipcs
  - 5 – sar -r -B -W
  - 6 – free
  - 7 – oprofile
  - 8 – gnome-
  - system-monitor
  - 9 – KDE-monitor
  - 10 – /proc
- Process Tools
  - 1 – top
  - 2 – ps -o pmem
  - 3 – gprof
  - 4 – strace,ltrace
  - 5 – sar
- Disk Tools
  - 1 – iostat -x
  - 2 – vmstat - D
  - 3 – sar -DEV #
  - 4 – nfsstat
  - 5 – NEED MORE!

ps

```
[root@localhost root]# ps aux
[root@localhost root]# ps -aux | more
USER       PID %CPU %MEM   VSZ  RSS TTY      STAT START   TIME COMMAND
root         1  0.1  0.1  1528  516 ?        S    23:18   0:04 init
root         2  0.0  0.0     0    0 ?        SW   23:18   0:00 [keventd]
root         3  0.0  0.0     0    0 ?        SW   23:18   0:00 [kapmd]
root         4  0.0  0.0     0    0 ?        SWN  23:18   0:00 [ksoftirqd/0]
root         7  0.0  0.0     0    0 ?        SW   23:18   0:00 [bdflush]
root         5  0.0  0.0     0    0 ?        SW   23:18   0:00 [kswapd]
root         6  0.0  0.0     0    0 ?        SW   23:18   0:00 [kscand]
```


vmstat(paging vs swapping)

Vmstat 10

procs			memory			swap		io		system		cpu			
r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	wa	id
2	0	0	5483524	200524	234576	0	0	54	63	152	513	0	3	0	96
0	2	0	1697840	200524	3931440	0	0	578	50482	1085	3994	1	22	14	63
3	0	0	7844	200524	5784109	0	0	59330	58946	3243	14430	7	32	18	42

Vmstat 10

procs			memory			swap		io		system		cpu			
r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	wa	id
2	0	0	5483524	200524	234576	0	0	54	63	152	513	0	3	0	96
0	2	0	1662340	200524	234576	0	0	578	50482	1085	3994	1	22	14	63
3	0	235678	7384	200524	234576	18754	23745	193	58946	3243	14430	7	32	18	42

iostat -x of same IOzone EXT3 file system

Iostat metrics

rates perf sec	sizes and response time
r|w rqm/s – request merged/s	averq-sz – average request sz
r|w sec/s – 512 byte sectors/s	avequ-sz – average queue sz
r|w KB/s – Kilobyte/s	await – average wait time ms
r|w /s – operations/s	svcm – ave service time m

```
Linux 2.4.21-27.0.2.ELsmp (node1)

avg-cpu:  %user   %nice    %sys %iowait   %idle
           0.40    0.00    2.63    0.91   96.06

Device:    rrqm/s wrqm/s   r/s   w/s  rsec/s  wsec/s    rkB/s    wkB/s avgrq-sz avgqu-sz   await  svctm  %util
sdi       16164.60   0.00 523.40  0.00 133504.00    0.00 66752.00     0.00   255.07     1.00    1.91   1.88  98.40
sdi       17110.10   0.00 553.90  0.00 141312.00    0.00 70656.00     0.00   255.12     0.99    1.80   1.78  98.40
sdi       16153.50   0.00 522.50  0.00 133408.00    0.00 66704.00     0.00   255.33     0.98    1.88   1.86  97.00
sdi       17561.90   0.00 568.10  0.00 145040.00    0.00 72520.00     0.00   255.31     1.01    1.78   1.76 100.00
```


SAR

```
[root@localhost redhat]# sar -u 3 3
Linux 2.4.21-20.EL (localhost.localdomain)      05/16/2005

10:32:28 PM       CPU     %user     %nice   %system     %idle
10:32:31 PM       all      0.00      0.00      0.00    100.00
10:32:34 PM       all      1.33      0.00      0.33     98.33
10:32:37 PM       all      1.34      0.00      0.00     98.66
Average:          all      0.89      0.00      0.11     99.00

[root] sar -n DEV
Linux 2.4.21-20.EL (localhost.localdomain)      03/16/2005

01:10:01 PM     IFACE   rxpck/s   txpck/s   rxbyt/s   txbyt/s   rxcmp/s
txcmp/s  rxmcst/s
01:20:00 PM        lo      3.49      3.49    306.16    306.16      0.00
0.00      0.00
01:20:00 PM      eth0      3.89      3.53   2395.34    484.70      0.00
0.00      0.00
01:20:00 PM      eth1      0.00      0.00      0.00      0.00      0.00
0.00      0.00
```


Networking tools

ethtool – View and change Ethernet card settings

sysctl – View and set */proc/sys* settings

ifconfig – View and set ethX variables

setpci – View and set pci bus params for device

netperf – Can run a bunch of different network tests

/proc – OS info, place for changing device tunables

ethtool -S – provides HW level stats

Counters since boot time, create scripts to calculate diffs

ethtool -c - Interrupt coalescing

ethtool -g - provides ring buffer information

ethtool -k - provides hw assist information

ethtool -i - provides the driver information

MPSTAT CPU Utilization
Raw vs. Tuned IRQ, NAPI

Not Tuned

CPU	%user	%nice	%system	%iowait	%irq	%soft	%idle	intr/s
all	0.23	0.00	8.01	0.02	0.00	10.78	80.96	21034.49
0	0.00	0.00	0.00	0.01	0.00	52.16	47.83	20158.58
1	0.00	0.00	0.00	0.02	0.00	0.00	100.00	125.14
2	0.00	0.00	0.00	0.08	0.00	0.00	99.93	125.14
3	0.00	0.00	0.00	0.03	0.00	0.00	99.99	125.13
4	1.79	0.00	64.11	0.00	0.00	34.11	0.01	125.14
5	0.01	0.00	0.00	0.02	0.00	0.00	99.99	125.14
6	0.00	0.00	0.00	0.00	0.00	0.00	100.01	125.14
7	0.00	0.00	0.00	0.02	0.00	0.00	99.99	125.14

With IRQ affinity Tuning

CPU	%user	%nice	%system	%iowait	%irq	%soft	%idle	intr/s
all	0.26	0.00	10.44	0.00	0.00	12.50	76.79	1118.61
0	0.00	0.00	0.00	0.00	0.00	0.00	100.00	1.12
1	0.01	0.00	0.00	0.00	0.00	0.00	99.99	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00
4	2.08	0.00	83.54	0.00	0.00	0.00	14.38	0.00
5	0.00	0.00	0.01	0.00	0.00	100.00	0.00	1.95
6	0.00	0.00	0.00	0.00	0.00	0.02	99.98	0.68
7	0.00	0.00	0.00	0.00	0.03	0.00	99.98	1114.86

free/numastat – memory allocation

```
[root@localhost redhat]# free -l
             total       used       free     shared    buffers     cached
Mem:        511368     342336     169032          0      29712     167408
Low:        511368     342336     169032          0          0          0
High:            0          0          0          0          0          0
-/+ buffers/cache:     145216     366152
Swap:      1043240          0    1043240

numastat (on 2-node x86_64 based system)
                         node1          node0
numa_hit                 9803332       10905630
numa_miss                2049018        1609361
numa_foreign             1609361        2049018
interleave_hit             58689          54749
local_node               9770927       10880901
other_node               2081423        1634090
```


NUMAstat and NUMActl

NUMAstat to display system NUMA characteristics on a numasystem

```
[root@perf5 ~]# numastat
                           node3           node2           node1           node0
numa_hit                   72684           82215          157244          325444
numa_miss                      0               0               0               0
numa_foreign                   0               0               0               0
interleave_hit              2668            2431            2763            2699
local_node                 67306           77456          152115          324733
other_node                  5378            4759            5129             711
```

numactl [--interleave nodes] [--preferred node] [--membind nodes] [--cpubind nodes] [--localalloc] command {arguments ...}

NUMActl to control process and memory”

TIP

- **App < memory single NUMA zone**
- **Numactl use –cpubind cpus within same socket**
- **App > memory of a single NUMA zone**
- **Numactl –interleave XY and –cpubind XY**

NUMAstat and NUMActl

EXAMPLES

numactl --interleave=all bigdatabase arguments Run big database with its memory interleaved on all CPUs.

numactl --cpubind=0--membind=0,1 process Run process on node 0 with memory allocated on node 0 and 1.

numactl --preferred=1 numactl --show Set preferred node 1 and show the resulting state.

numactl --interleave=all --shmkeyfile /tmp/shmkey Interleave all of the sysv shared memory regiion specified by /tmp/shmkey over all nodes.

numactl --offset=1G --length=1G --membind=1 --file /dev/shm/A --touch Bind the second gigabyte in the tmpfs file /dev/shm/A to node 1.

numactl --localalloc /dev/shm/file Reset the policy for the shared memory file to the default localalloc policy.

The /proc filesystem

- /proc/meminfo
- /proc/slabinfo
- /proc/cpuinfo
- /proc/<pid#>/maps
- /proc/vmstat
- /proc/zoneinfo(RHEL5&RHEL6)
- /proc/sysrq-trigger

/proc/meminfo

RHEL4> cat /proc/meminfo
MemTotal: 32749568 kB
MemFree: 31313344 kB
Buffers: 29992 kB
Cached: 1250584 kB
SwapCached: 0 kB
Active: 235284 kB
Inactive: 1124168 kB
HighTotal: 0 kB
HighFree: 0 kB
LowTotal: 32749568 kB
LowFree: 31313344 kB
SwapTotal: 4095992 kB
SwapFree: 4095992 kB
Dirty: 0 kB
Writeback: 0 kB
Mapped: 1124080 kB
Slab: 38460 kB
CommitLimit: 20470776 kB
Committed_AS: 1158556 kB
PageTables: 5096 kB
VmallocTotal: 536870911 kB
VmallocUsed: 2984 kB
VmallocChunk: 536867627 kB
HugePages_Total: 0
HugePages_Free: 0
Hugepagesize: 2048 kB

RHEL5> cat /proc/meminfo
MemTotal: 1025220 kB
MemFree: 11048 kB
Buffers: 141944 kB
Cached: 342664 kB
SwapCached: 4 kB
Active: 715304 kB
Inactive: 164780 kB
HighTotal: 0 kB
HighFree: 0 kB
LowTotal: 1025220 kB
LowFree: 11048 kB
SwapTotal: 2031608 kB
SwapFree: 2031472 kB
Dirty: 84 kB
Writeback: 0 kB
AnonPages: 395572 kB
Mapped: 82860 kB
Slab: 92296 kB
PageTables: 23884 kB
NFS_Unstable: 0 kB
Bounce: 0 kB
CommitLimit: 2544216 kB
Committed_AS: 804656 kB
VmallocTotal: 34359738367 kB
VmallocUsed: 263472 kB
VmallocChunk: 34359474711 kB
HugePages_Total: 0
HugePages_Free: 0
HugePages_Rsvd: 0
Hugepagesize: 2048 kB

RHEL6> cat /proc/meminfo
MemTotal: 2053304 kB
MemFree: 140884 kB
Buffers: 113292 kB
Cached: 653624 kB
SwapCached: 21956 kB
Active: 1003052 kB
Inactive: 593332 kB
Active(anon): 620408 kB
Inactive(anon): 213340 kB
Active(file): 382644 kB
Inactive(file): 379992 kB
Unevictable: 0 kB
Mlocked: 0 kB
SwapTotal: 4128760 kB
SwapFree: 3972280 kB
Dirty: 36 kB
Writeback: 0 kB
AnonPages: 819108 kB
Mapped: 76768 kB
Shmem: 4272 kB
Slab: 249088 kB
SReclaimable: 144304 kB
SUnreclaim: 104784 kB
KernelStack: 2600 kB
PageTables: 34804 kB
NFS_Unstable: 0 kB
Bounce: 0 kB
WritebackTmp: 0 kB
CommitLimit: 5155412 kB
Committed_AS: 2074048 kB
VmallocTotal: 34359738367 kB
VmallocUsed: 85196 kB
VmallocChunk: 34359634364 kB
HardwareCorrupted: 0 kB
AnonHugePages: 393216 kB
HugePages_Total: 0
HugePages_Free: 0
HugePages_Rsvd: 0
HugePages_Surp: 0
Hugepagesize: 2048 kB
DirectMap4k: 6640 kB
DirectMap2M: 2088960 kB

/proc/slabinfo

```
slabinfo - version: 2.1
# name            <active_objs> <num_objs> <objsize> <objperslab> <pagesperslab> : tunables <limit>
<batchcount> <sharedfactor>: slabdata <active_slabs> <num_slabs> <sharedavail>
nfsd4_delegations      0      0    656    6    1 : tunables   54   27    8 : slabdata      0      0      0
nfsd4_stateids         0      0    128   30    1 : tunables  120   60    8 : slabdata      0      0      0
nfsd4_files            0      0     72   53    1 : tunables  120   60    8 : slabdata      0      0      0
nfsd4_stateowners      0      0    424    9    1 : tunables   54   27    8 : slabdata      0      0      0
nfs_direct_cache       0      0    128   30    1 : tunables  120   60    8 : slabdata      0      0      0
nfs_write_data        36     36    832    9    2 : tunables   54   27    8 : slabdata      4      4      0
nfs_read_data         32     35    768    5    1 : tunables   54   27    8 : slabdata      7      7      0
nfs_inode_cache     1383   1389   1040    3    1 : tunables   24   12    8 : slabdata    463    463      0
nfs_page               0      0    128   30    1 : tunables  120   60    8 : slabdata      0      0      0
fscache_cookie_jar     3     53     72   53    1 : tunables  120   60    8 : slabdata      1      1      0
ip_conntrack_expect    0      0    136   28    1 : tunables  120   60    8 : slabdata      0      0      0
ip_conntrack          75    130    304   13    1 : tunables   54   27    8 : slabdata     10     10      0
bridge_fdb_cache       0      0     64   59    1 : tunables  120   60    8 : slabdata      0      0      0
rpc_buffers            8      8   2048    2    1 : tunables   24   12    8 : slabdata      4      4      0
rpc_tasks             30     30    384   10    1 : tunables   54   27    8 : slabdata      3      3      0
```


/proc/cpuinfo

```
processor   : 7
vendor_id   : AuthenticAMD
cpu family  : 16
model               : 2
model name          : Quad-Core AMD Opteron(tm) Processor 8356
stepping    : 3
cpu MHz             : 1150.000
cache size  : 512 KB
physical id : 1
siblings    : 4
core id             : 3
cpu cores   : 4
apicid              : 7
initial apicid      : 7
fpu         : yes
fpu_exception       : yes
cpuid level : 5
wp          : yes
flags       : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2
ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm 3dnowext 3dnow constant_tsc rep_good nonstop_tsc
extd_apicid pni monitor cx16 popcnt lahf_lm cmp_legacy svm extapic cr8_legacy abm sse4a misalignsse
3dnowprefetch osvw ibs npt lbrv svm_lock
bogomips    : 4600.00
TLB size    : 1024 4K pages
clflush size : 64
cache_alignment : 64
address sizes       : 48 bits physical, 48 bits virtual
power management: ts ttp tm stc 100mhzsteps hwpstate
```


Alt Sysrq

RHEL5# echo 1 > /proc/sys/kernel/sysrq
RHEL5# echo ? > /proc/sysrq-trigger
RHEL5# dmesg

SysRq : HELP : loglevel0-8 reBoot Crashdump tErm Full kIll thaw-filesystems(J) saK showMem Nice powerOff showPc unRaw Sync showTasks Unmount shoWcpus

RHEL6# echo 1 > /proc/sys/kernel/sysrq
RHEL6# echo ? > /proc/sysrq-trigger
RHEL6# dmesg

SysRq : HELP : loglevel(0-9) reBoot Crash terminate-all-tasks(E) memory-full-oom-kill(F) kill-all-tasks(I) thaw-filesystems(J) saK show-backtrace-all-active-cpus(L) show-memory-usage(M) nice-all-RT-tasks(N) powerOff show-registers(P) show-all-timers(Q) unRaw Sync show-task-states(T) Unmount force-fb(V) show-blocked-tasks(W) dump-ftrace-buffer(Z)

Alt Sysrq M - NUMA

active_anon:42449 inactive_anon:1110 isolated_anon:0
 active_file:78845 inactive_file:144076 isolated_file:0
 unevictable:0 dirty:1 writeback:0 unstable:0
 free:3708129 slab_reclaimable:34950 slab_unreclaimable:83870
 mapped:14447 shmem:1139 pagetables:6512 bounce:0
Node 0 DMA free:15636kB min:80kB low:100kB high:120kB active_anon:0kB inactive_anon:0kB active_file:0kB inactive_file:0kB unevictable:0kB isolated(anon):0kB isolated(file):0kB present:15220kB mlocked:0kB dirty:0kB writeback:0kB mapped:0kB shmem:0kB slab_reclaimable:0kB slab_unreclaimable:0kB kernel_stack:0kB pagetables:0kB unstable:0kB bounce:0kB writeback_tmp:0kB pages_scanned:0 all_unreclaimable? no
lowmem_reserve[]: 0 2743 8045 8045
Node 0 DMA32 free:2669600kB min:15312kB low:19140kB high:22968kB active_anon:0kB inactive_anon:0kB active_file:0kB inactive_file:0kB unevictable:0kB isolated(anon):0kB isolated(file):0kB present:2808992kB mlocked:0kB dirty:0kB writeback:0kB mapped:0kB shmem:0kB slab_reclaimable:0kB slab_unreclaimable:0kB kernel_stack:0kB pagetables:0kB unstable:0kB bounce:0kB writeback_tmp:0kB pages_scanned:0 all_unreclaimable? no
lowmem_reserve[]: 0 0 5302 5302
Node 0 Normal free:4345696kB min:29600kB low:37000kB high:44400kB active_anon:109920kB inactive_anon:2656kB active_file:285768kB inactive_file:419868kB unevictable:0kB isolated(anon):0kB isolated(file):0kB present:5429760kB mlocked:0kB dirty:4kB writeback:0kB mapped:35592kB shmem:2768kB slab_reclaimable:124912kB slab_unreclaimable:165260kB kernel_stack:1776kB pagetables:13636kB unstable:0kB bounce:0kB writeback_tmp:0kB pages_scanned:0 all_unreclaimable? no
lowmem_reserve[]: 0 0 0 0
Node 1 Normal free:7801584kB min:45108kB low:56384kB high:67660kB active_anon:59876kB inactive_anon:1784kB active_file:29612kB inactive_file:156436kB unevictable:0kB isolated(anon):0kB isolated(file):0kB present:8273920kB mlocked:0kB dirty:0kB writeback:0kB mapped:22196kB shmem:1788kB slab_reclaimable:14888kB slab_unreclaimable:170220kB kernel_stack:1408kB pagetables:12412kB unstable:0kB bounce:0kB writeback_tmp:0kB pages_scanned:0 all_unreclaimable? no
lowmem_reserve[]: 0 0 0 0
Node 0 DMA: 1*4kB 0*8kB 1*16kB 2*32kB 1*64kB 1*128kB 0*256kB 0*512kB 1*1024kB 1*2048kB 3*4096kB = 15636kB
Node 0 DMA32: 8*4kB 10*8kB 5*16kB 7*32kB 10*64kB 10*128kB 11*256kB 8*512kB 6*1024kB 4*2048kB 646*4096kB = 2669600kB
Node 0 Normal: 1666*4kB 437*8kB 125*16kB 73*32kB 41*64kB 17*128kB 12*256kB 8*512kB 6*1024kB 8*2048kB 1049*4096kB = 4345696kB
Node 1 Normal: 510*4kB 399*8kB 178*16kB 87*32kB 258*64kB 148*128kB 64*256kB 7*512kB 2*1024kB 2*2048kB 1887*4096kB = 7801584kB
224059 total pagecache pages
0 pages in swap cache
Swap cache stats: add 0, delete 0, find 0/0
Free swap = 2097144kB
Total swap = 2097144kB
4194288 pages RAM
77971 pages reserved
134109 pages shared
261013 pages non-shared

/sys/devices/system/node/node*/meminfo

```
Node 0 MemTotal:       8387044 kB
Node 0 MemFree:        7027512 kB
Node 0 MemUsed:        1359532 kB
Node 0 Active:          397548 kB
Node 0 Inactive:        423960 kB
Node 0 Active(anon):    111080 kB
Node 0 Inactive(anon):    2768 kB
Node 0 Active(file):    286468 kB
Node 0 Inactive(file):  421192 kB
Node 0 Unevictable:          0 kB
Node 0 Mlocked:              0 kB
Node 0 Dirty:                0 kB
Node 0 Writeback:            0 kB
Node 0 FilePages:       710536 kB
Node 0 Mapped:           35704 kB
Node 0 AnonPages:        80248 kB
Node 0 Shmem:             2880 kB
Node 0 KernelStack:       1776 kB
Node 0 PageTables:       13628 kB
Node 0 NFS_Unstable:         0 kB
Node 0 Bounce:               0 kB
Node 0 WritebackTmp:         0 kB
Node 0 Slab:           290328 kB
Node 0 SReclaimable:   124948 kB
Node 0 SUnreclaim:     165380 kB
Node 0 HugePages_Total:     0
Node 0 HugePages_Free:      0
Node 0 HugePages_Surp:      0
```

```
Node 1 MemTotal:       8388608 kB
Node 1 MemFree:        7801380 kB
Node 1 MemUsed:         587228 kB
Node 1 Active:           89684 kB
Node 1 Inactive:        158244 kB
Node 1 Active(anon):     60056 kB
Node 1 Inactive(anon):    1784 kB
Node 1 Active(file):     29628 kB
Node 1 Inactive(file):  156460 kB
Node 1 Unevictable:          0 kB
Node 1 Mlocked:              0 kB
Node 1 Dirty:                0 kB
Node 1 Writeback:            0 kB
Node 1 FilePages:       187876 kB
Node 1 Mapped:           22196 kB
Node 1 AnonPages:        49820 kB
Node 1 Shmem:             1788 kB
Node 1 KernelStack:       1408 kB
Node 1 PageTables:       12412 kB
Node 1 NFS_Unstable:         0 kB
Node 1 Bounce:               0 kB
Node 1 WritebackTmp:         0 kB
Node 1 Slab:           185092 kB
Node 1 SReclaimable:    14896 kB
Node 1 SUnreclaim:     170196 kB
Node 1 HugePages_Total:     0
Node 1 HugePages_Free:      0
Node 1 HugePages_Surp:      0
```


Alt Sysrq T

```
gdmgreeter    S ffff810009036800     0  7511   7483                7489 (NOTLB)
 ffff81044ae05b38 0000000000000082 0000000000000080 0000000000000000
 0000000000000000 000000000000000a ffff810432ed97a0 ffff81010f387080
 0000002a3a0d4398 0000000000003b57 ffff810432ed9988 0000000600000000
Call Trace:
 [<ffffffff8006380f>] schedule_timeout+0x1e/0xad
 [<ffffffff80049b33>] add_wait_queue+0x24/0x34
 [<ffffffff8002db7e>] pipe_poll+0x2d/0x90
 [<ffffffff8002f764>] do_sys_poll+0x277/0x360
 [<ffffffff8001e99c>] __pollwait+0x0/0xe2
 [<ffffffff8008be44>] default_wake_function+0x0/0xe
 [<ffffffff8008be44>] default_wake_function+0x0/0xe
 [<ffffffff8008be44>] default_wake_function+0x0/0xe
 [<ffffffff80012f1a>] sock_def_readable+0x34/0x5f
 [<ffffffff8004a81a>] unix_stream_sendmsg+0x281/0x346
 [<ffffffff80037c3a>] do_sock_write+0xc6/0x102
 [<ffffffff801277da>] avc_has_perm+0x43/0x55
 [<ffffffff80276a6e>] unix_ioctl+0xc7/0xd0
 [<ffffffff8021f48f>] sock_ioctl+0x1c1/0x1e5
 [<ffffffff800420a7>] do_ioctl+0x21/0x6b
 [<ffffffff800302a0>] vfs_ioctl+0x457/0x4b9
 [<ffffffff800b6193>] audit_syscall_entry+0x180/0x1b3
 [<ffffffff8004c4f6>] sys_poll+0x2d/0x34
 [<ffffffff8005d28d>] tracesys+0xd5/0xe0
```


Alt Sysrq L(W) and P

```
SysRq : Show CPUs
CPU2:
 ffff81010f30bf48 0000000000000000 ffff81010f305e20 ffffffff801ae69e
 0000000000000000 0000000000000200 ffffffff803ea2a0 ffffffff801ae6cd
 ffffffff801ae69e ffffffff80022d85 ffffffff80197393 00000000000000ff
Call Trace:
 <IRQ>  [<ffffffff801ae69e>] showacpu+0x0/0x3b
 [<ffffffff801ae6cd>] showacpu+0x2f/0x3b
 [<ffffffff801ae69e>] showacpu+0x0/0x3b
 [<ffffffff80022d85>] smp_call_function_interrupt+0x57/0x75
 [<ffffffff80197393>] acpi_processor_idle+0x0/0x463
 [<ffffffff8005dc22>] call_function_interrupt+0x66/0x6c
 <EOI>  [<ffffffff80197324>] acpi_safe_halt+0x25/0x36
 [<ffffffff8019751a>] acpi_processor_idle+0x187/0x463
 [<ffffffff80197395>] acpi_processor_idle+0x2/0x463
 [<ffffffff80197393>] acpi_processor_idle+0x0/0x463
 [<ffffffff80197393>] acpi_processor_idle+0x0/0x463
 [<ffffffff80049399>] cpu_idle+0x95/0xb8
 [<ffffffff80076e12>] start_secondary+0x45a/0x469
```


64-bit /proc/<pid>/maps

```
# cat /proc/2345/maps
00400000-0100b000 r-xp 00000000 fd:00 1933328                            /usr/sybase/ASE-12_5/bin/dataserver.esd3
0110b000-01433000 rw-p 00c0b000 fd:00 1933328                            /usr/sybase/ASE-12_5/bin/dataserver.esd3
01433000-014eb000 rwxp 01433000 00:00 0
40000000-40001000 ---p 40000000 00:00 0
40001000-40a01000 rwxp 40001000 00:00 0
2a95f73000-2a96073000 ---p 0012b000 fd:00 819273                         /lib64/tls/libc-2.3.4.so
2a96073000-2a96075000 r--p 0012b000 fd:00 819273                         /lib64/tls/libc-2.3.4.so
2a96075000-2a96078000 rw-p 0012d000 fd:00 819273                         /lib64/tls/libc-2.3.4.so
2a96078000-2a9607e000 rw-p 2a96078000 00:00 0
2a9607e000-2a98c3e000 rw-s 00000000 00:06 360450                         /SYSV0100401e (deleted)
2a98c3e000-2a98c47000 rw-p 2a98c3e000 00:00 0
2a98c47000-2a98c51000 r-xp 00000000 fd:00 819227                         /lib64/libnss_files-2.3.4.so
2a98c51000-2a98d51000 ---p 0000a000 fd:00 819227                         /lib64/libnss_files-2.3.4.so
2a98d51000-2a98d53000 rw-p 0000a000 fd:00 819227                         /lib64/libnss_files-2.3.4.so
2a98d53000-2a98d57000 r-xp 00000000 fd:00 819225                         /lib64/libnss_dns-2.3.4.so
2a98d57000-2a98e56000 ---p 00004000 fd:00 819225                         /lib64/libnss_dns-2.3.4.so
2a98e56000-2a98e58000 rw-p 00003000 fd:00 819225                         /lib64/libnss_dns-2.3.4.so
2a98e58000-2a98e69000 r-xp 00000000 fd:00 819237                         /lib64/libresolv-2.3.4.so
2a98e69000-2a98f69000 ---p 00011000 fd:00 819237                         /lib64/libresolv-2.3.4.so
2a98f69000-2a98f6b000 rw-p 00011000 fd:00 819237                         /lib64/libresolv-2.3.4.so
2a98f6b000-2a98f6d000 rw-p 2a98f6b000 00:00 0
35c7e00000-35c7e08000 r-xp 00000000 fd:00 819469                         /lib64/libpam.so.0.77
35c7e08000-35c7f08000 ---p 00008000 fd:00 819469                         /lib64/libpam.so.0.77
35c7f08000-35c7f09000 rw-p 00008000 fd:00 819469                         /lib64/libpam.so.0.77
35c8000000-35c8011000 r-xp 00000000 fd:00 819468                         /lib64/libaudit.so.0.0.0
35c8011000-35c8110000 ---p 00011000 fd:00 819468                         /lib64/libaudit.so.0.0.0
35c8110000-35c8118000 rw-p 00010000 fd:00 819468                         /lib64/libaudit.so.0.0.0
35c9000000-35c900b000 r-xp 00000000 fd:00 819457                         /lib64/libgcc_s-3.4.4-20050721.so.1
35c900b000-35c910a000 ---p 0000b000 fd:00 819457                         /lib64/libgcc_s-3.4.4-20050721.so.1
35c910a000-35c910b000 rw-p 0000a000 fd:00 819457                         /lib64/libgcc_s-3.4.4-20050721.so.1
7fbfff1000-7fc0000000 rwxp 7fbfff1000 00:00 0
ffffffffff600000-ffffffffffe00000 ---p 00000000 00:00 0
```


32-bit /proc/<pid>/maps

```
0022e000-0023b000 r-xp 00000000 03:03 4137068    /lib/tls/libpthread-0.60.so
0023b000-0023c000 rw-p 0000c000 03:03 4137068    /lib/tls/libpthread-0.60.so
0023c000-0023e000 rw-p 00000000 00:00 0
0037f000-00391000 r-xp 00000000 03:03 523285     /lib/libnsl-2.3.2.so
00391000-00392000 rw-p 00011000 03:03 523285     /lib/libnsl-2.3.2.so
00392000-00394000 rw-p 00000000 00:00 0
00c45000-00c5a000 r-xp 00000000 03:03 523268     /lib/ld-2.3.2.so
00c5a000-00c5b000 rw-p 00015000 03:03 523268     /lib/ld-2.3.2.so
00e5c000-00f8e000 r-xp 00000000 03:03 4137064    /lib/tls/libc-2.3.2.so
00f8e000-00f91000 rw-p 00131000 03:03 4137064    /lib/tls/libc-2.3.2.so
00f91000-00f94000 rw-p 00000000 00:00 0
08048000-0804f000 r-xp 00000000 03:03 1046791    /sbin/ypbind
0804f000-08050000 rw-p 00007000 03:03 1046791    /sbin/ypbind
```


Profiling Tools: OProfile

Open source project – http://oprofile.sourceforge.net

Upstream; Red Hat contributes

Originally modeled after DEC Continuous Profiling Infrastructure (DCPI)

System-wide profiler (both kernel and user code)

Sample-based profiler with SMP machine support

Performance monitoring hardware support

Relatively low overhead, typically <10%

Designed to run for long times

Included in base Red Hat Enterprise Linux product

Events to measure with Oprofile:

Initially time-based samples most useful:

PPro/PII/PIII/AMD: CPU_CLK_UNHALTED

P4: GLOBAL_POWER_EVENTS

IA64: CPU_CYCLES

TIMER_INT (fall-back profiling mechanism) default

Processor specific performance monitoring hardware can provide additional kinds of sampling

Many events to choose from

Branch mispredictions

Cache misses - TLB misses

Pipeline stalls/serializing instructions

oprofile – builtin to RHEL4 & 5 (smp)

opcontrol – on/off data

--start start collection

--stop stop collection

--dump output to disk

--event=:name:count

Example:

```
# opcontrol –start
# /bin/time test1 &
# sleep 60
# opcontrol –stop
# opcontrol dump
```

opreport – analyze profile

-r reverse order sort

-t [percentage] theshold to view

-f /path/filename

-d details

opannotate

-s /path/source

-a /path/assembly

oprofile – opcontrol and opreport cpu_cycles

```
# CPU: Core 2, speed 2666.72 MHz (estimated)
Counted CPU_CLK_UNHALTED events (Clock cycles when not halted) with a unit mask of 0x00 (Unhalted core
c
ycles) count 100000
CPU_CLK_UNHALT...|
  samples|      %|
------------------
397435971 84.6702 vmlinux
 19703064  4.1976 zeus.web
 16914317  3.6034 e1000
 12208514  2.6009 ld-2.5.so
 11711746  2.4951 libc-2.5.so
  5164664  1.1003 sim.cgi
  2333427  0.4971 oprofiled
  1295161  0.2759 oprofile
  1099731  0.2343 zeus.cgi
   968623  0.2064 ext3
   270163  0.0576 jbd
```


RHEL6 (new) PERF TOP – 64 kvm guests

```
------------------------------------------------------------------------------
   PerfTop:    1381 irqs/sec  kernel:24.9% [1000Hz cycles],  (all, 8 CPUs)
------------------------------------------------------------------------------

             samples  pcnt function                    DSO
             _______ _____ ____________________________ ____________

             1994.00 70.9% daxpy                       linpackd
              139.00  4.9% dgefa                       linpackd
               54.00  1.9% find_busiest_group          [kernel]
               50.00  1.8% tick_nohz_stop_sched_tick   [kernel]
               37.00  1.3% native_read_tsc             [kernel]
               30.00  1.1% ktime_get                   [kernel]
               26.00  0.9% rebalance_domains           [kernel]
               24.00  0.9% matgen                      linpackd
               23.00  0.8% find_next_bit               [kernel]
               22.00  0.8% cpumask_next_and            [kernel]
               21.00  0.7% _spin_lock                  [kernel]
```


RHEL6 (new) PERF STATS - linpackd

```
Performance counter stats for './linpackd':

  15516.960937  task-clock-msecs         #      0.996 CPUs
           33  context-switches         #      0.000 M/sec
            0  CPU-migrations           #      0.000 M/sec
         4060  page-faults              #      0.000 M/sec
   30972629189  cycles                   #   1996.050 M/sec
   47178860731  instructions             #      1.523 IPC
    2432058056  branches                 #    156.735 M/sec
      24045591  branch-misses            #      0.989 %
     509453960  cache-references         #     32.832 M/sec
       1014589  cache-misses             #      0.065 M/sec

  15.586072255  seconds time elapsed
```


Profiling Tools: SystemTap

Technology: Kprobes:

In current 2.6 kernels

Upstream 2.6.12, backported to RHEL4 kernel

Kernel instrumentation without recompile/reboot

Uses software int and trap handler for instrumentation

Debug information:

Provides map between executable and source code

Generated as part of RPM builds

Available at: ftp://ftp.redhat.com

Safety: Instrumentation scripting language:

No dynamic memory allocation or assembly/C code

Types and type conversions limited

Restrict access through pointers

Script compiler checks:

Infinite loops and recursion – Invalid variable access

Profiling Tools: SystemTap

Red Hat, Intel, IBM & Hitachi collaboration

Linux answer to Solaris Dtrace

Dynamic instrumentation

Tool to take a deep look into a running system:

Assists in identifying causes of performance problems

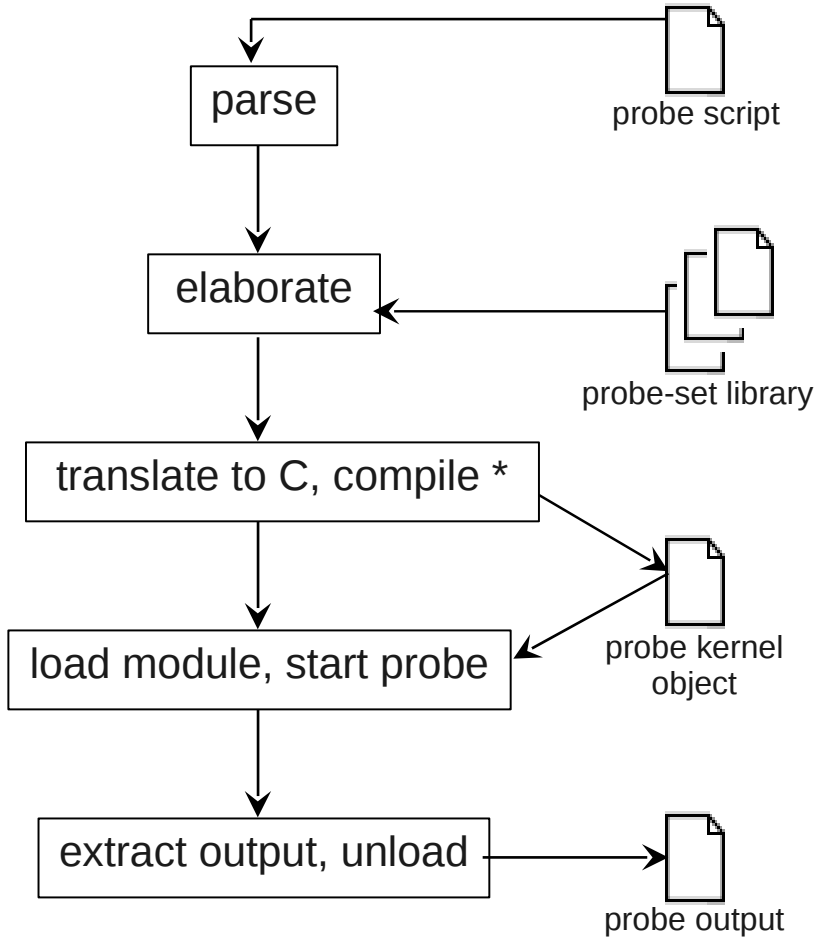
Simplifies building instrumentation

Current snapshots available from:
http://sources.redhat.com/systemtap

Source for presentations/papers

Kernel space tracing today, user space tracing under development

Technology preview status until 5.1



* Solaris Dtrace is interpretive

SystemTap: Kernel debugging

Several hundred tracepoints were added to the kernel

```
trace_mm_filemap_fault(area->vm_mm, address, page);
trace_mm_anon_userfree(mm, addr, page);
trace_mm_filemap_userunmap(mm, addr, page);
trace_mm_filemap_cow(mm, address, new_page);
trace_mm_anon_cow(mm, address, new_page);
trace_mm_anon_pgin(mm, address, page);
trace_mm_anon_fault(mm, address, page);
trace_mm_page_free(page);
trace_mm_page_allocation(page, zone->free_pages);
trace_mm_pdflush_bgwriteout(_min_pages);
trace_mm_pdflush_kupdate(nr_to_write);
trace_mm_anon_unmap(page, ret == SWAP_SUCCESS);
trace_mm_filemap_unmap(page, ret == SWAP_SUCCESS);
trace_mm_pagereclaim_pgout(page, PageAnon(page));
trace_mm_pagereclaim_free(page, PageAnon(page));
trace_mm_pagereclaim_shrinkinactive_i2a(page);
trace_mm_pagereclaim_shrinkinactive_i2i(page);
trace_mm_pagereclaim_shrinkinactive(nr_reclaimed);
trace_mm_pagereclaim_shrinkactive_a2a(page);
trace_mm_pagereclaim_shrinkactive_a2i(page);
trace_mm_pagereclaim_shrinkactive(pgscanned);
trace_mm_pagereclaim_shrinkzone(nr_reclaimed);
trace_mm_directreclaim_reclaimall(priority);
trace_mm_kswapd_runs(sc.nr_reclaimed);
```


SystemTap: Kernel debugging

Several custom scripts enable/use tracepoints

(/usr/local/share/doc/systemtap/examples)

```
#! /usr/local/bin/stap
global traced_pid
function log_event:long ()
{
        return (!traced_pid ||traced_pid == (task_pid(task_current())))
}
probe kernel.trace("mm_pagereclaim_shrinkinactive") {
        if (!log_event()) next
        reclaims[pid()]++
        command[pid()]=execname()
}
//MM kernel tracepoints prolog and epilog routines
probe begin {
        printf("Starting mm tracepoints\n");
        traced_pid = target();
        if (traced_pid) {
                printf("mode Specific Pid, traced pid: %d\n", traced_pid);
        } else {
                printf("mode - All Pids\n");
        }
        printf("\n");
}
probe end {
        printf("Terminating mm tracepoints\n");
        printf("Command              Pid       Direct    Activate   Deactivate Reclaims   Freed\n");
        printf("-------              ---       ------    --------   ---------- --------   -----\n");
        foreach (pid in reclaims-)
```


SystemTap: Kernel debugging

Command	Pid	Direct	Activate	Deactivate	Reclaims	Freed
kswapd0	544	0	1503767	919437	15157	430730
kswapd1	545	0	1806788	824347	12117	341408
memory	25435	997	569757	308360	4621	115837
mixer_applet2	7687	6	4180	1013	33	981
Xorg	7491	5	1906	2839	20	382
gnome-terminal	7161	2	1038	695	12	320
gnome-terminal	7701	5	2614	2245	7	172
cupsd	7100	1	927	0	4	128

SystemTap: Kernel debugging

Command	Pid	Alloc	Free	A_fault	A_ufree	A_pgin	A_cow	A_unmap
memory	25685	2842784	4064408	2834840	3989816	14	0	48185
kswapd1	545	3007	53257	0	0	0	0	49884
kswapd0	544	620	25241	0	0	0	0	17568
mixer_applet2	7687	302	2827	0	0	1	0	1241
sshd	25051	227	0	0	0	6	0	0
kjournald	863	207	283	0	0	0	0	2149
Xorg	7491	169	898	0	0	0	0	310
gnome-power-man	7653	152	0	0	0	18	0	0
avahi-daemon	7252	150	1280	0	0	48	0	160
irqbalance	6725	126	364	13	13	18	0	190
bash	25053	122	0	0	0	13	0	0
hald	7264	89	0	0	0	83	0	0
gconfd-2	7163	82	526	0	0	68	0	116

Section 4: Tuning RHEL

How to tune Linux

Capacity tuning

Fix problems by adding resources

Performance Tuning

Throughput versus Latency

Methodology

1) Document config
2) Baseline results
3) While results non-optimal
    a) Monitor/Instrument system/workload
    b) Apply tuning 1 change at a time
    c) Analyze results, exit or loop
4) Document final config

Tuning - setting kernel parameters

/proc

[root@foobar fs]# cat /proc/sys/kernel/sysrq (see “0”)

[root@foobar fs]# echo 1 > /proc/sys/kernel/sysrq

[root@foobar fs]# cat /proc/sys/kernel/sysrq (see “1”)

Sysctl command

[root@foobar fs]# sysctl kernel.sysrq

kernel.sysrq = 0

[root@foobar fs]# sysctl -w kernel.sysrq=1

kernel.sysrq = 1

[root@foobar fs]# sysctl kernel.sysrq

kernel.sysrq = 1

Edit the /etc/sysctl.conf file

Kernel sysctl configuration file for Red Hat Linux

Controls the System Request debugging functionality of the kernel

kernel.sysrq = 1

CPUspeed and performance:

Enabled = governor set to “ondemand”

- Looks at cpu usage to regulate power
- Within 3-5% of performance for cpu loads
- IO loads can keep cpu stepped down -15-30%
- Supported in RHEL5/6 virtualization

Disable service cpuspeed off, or in BIOS

Tune for performance :

```
# echo performance > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

Then check to see if it stuck:

```
# cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

Check /proc/cpuinfo to make sure your seeing the expected CPU freq.

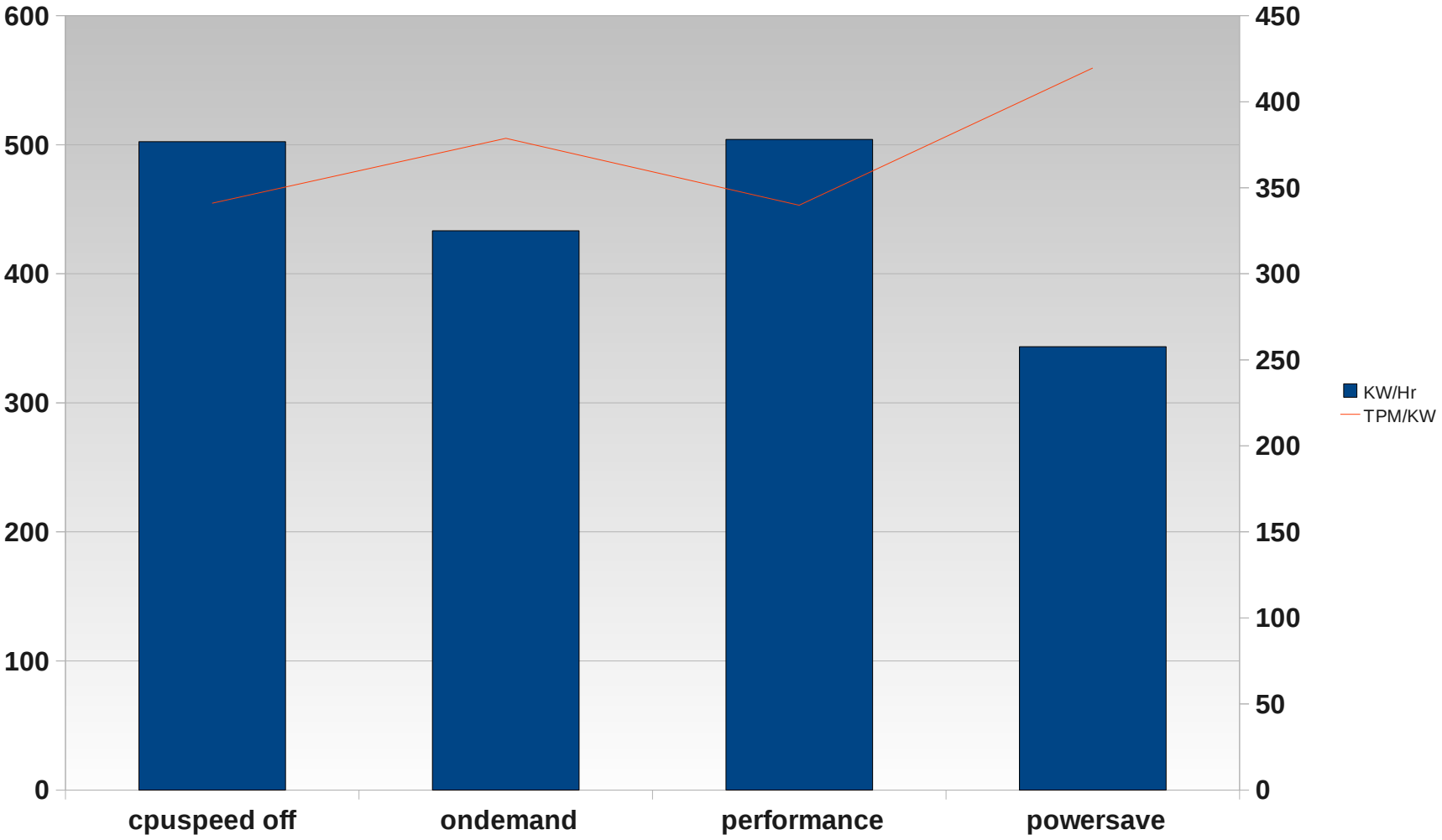
Proceed to “normal” service disable

Turbo Mode needs cpuspeed ENABLED

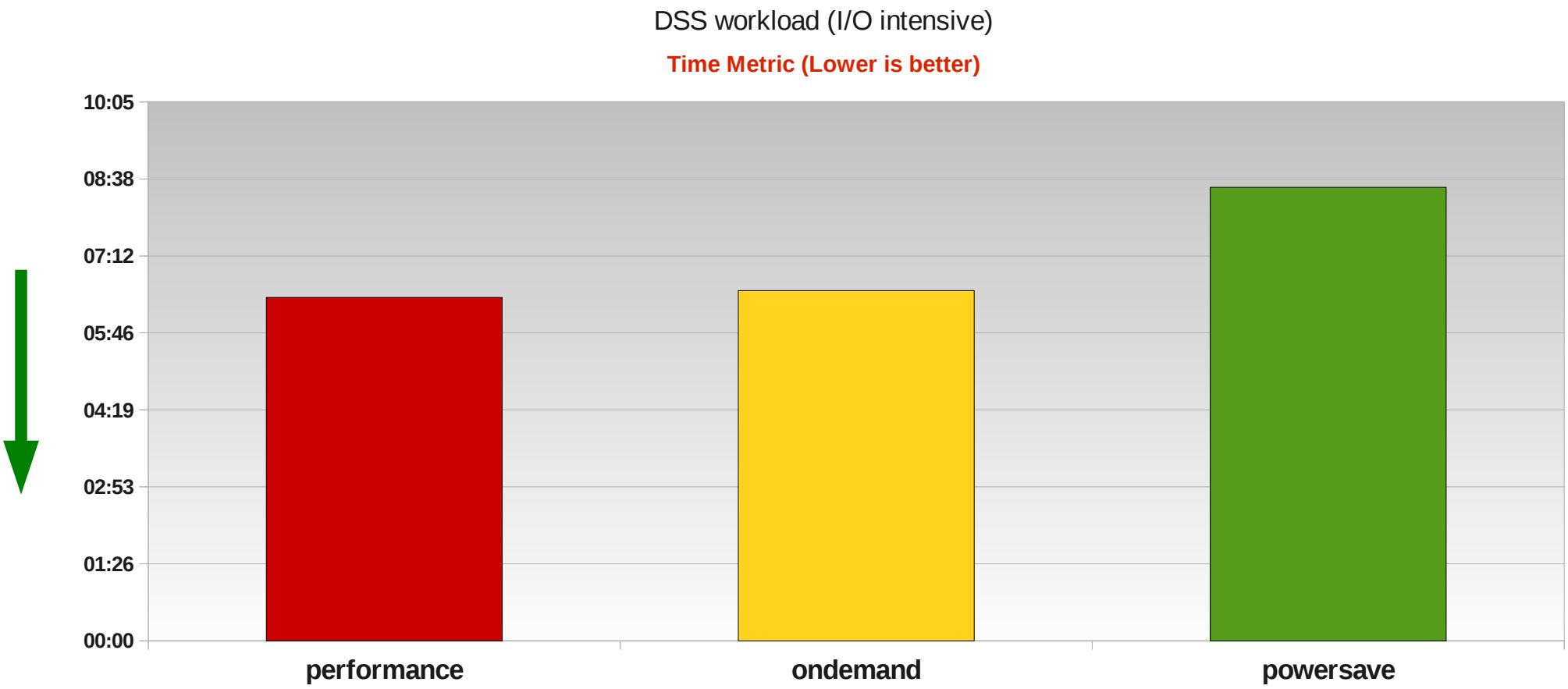
CPU Tuning

- CPU performance
  - Clock speed
  - Multiple cores
  - Power savings mode
    - cpuspeed off
    - performance
    - ondemand
    - powersave
- How to
  - echo "performance" > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
  - Best of both worlds – cron jobs to configure the governor mode
  - tuned-adm profile server-powersave

Performance / Power consumption (OLTP)



Tuning CPU – effect of power settings - DSS



Vmstat output during the run

7	12	5884	122884416	485900	734376	0	0	184848	39721	9175	37669	4	1	89	6	0
7	12	5884	122885024	485900	734376	0	0	217766	27468	9904	42807	4	2	87	6	0
2	0	5884	122884928	485908	734376	0	0	168496	45375	6294	27759	4	1	90	5	0
7	11	5884	122885056	485912	734372	0	0	178790	40969	9433	38140	4	1	90	5	0
1	15	5884	122885176	485920	734376	0	0	248283	19807	7710	37788	5	2	86	7	0

CPU Scheduler

Recognizes differences between logical and physical processors

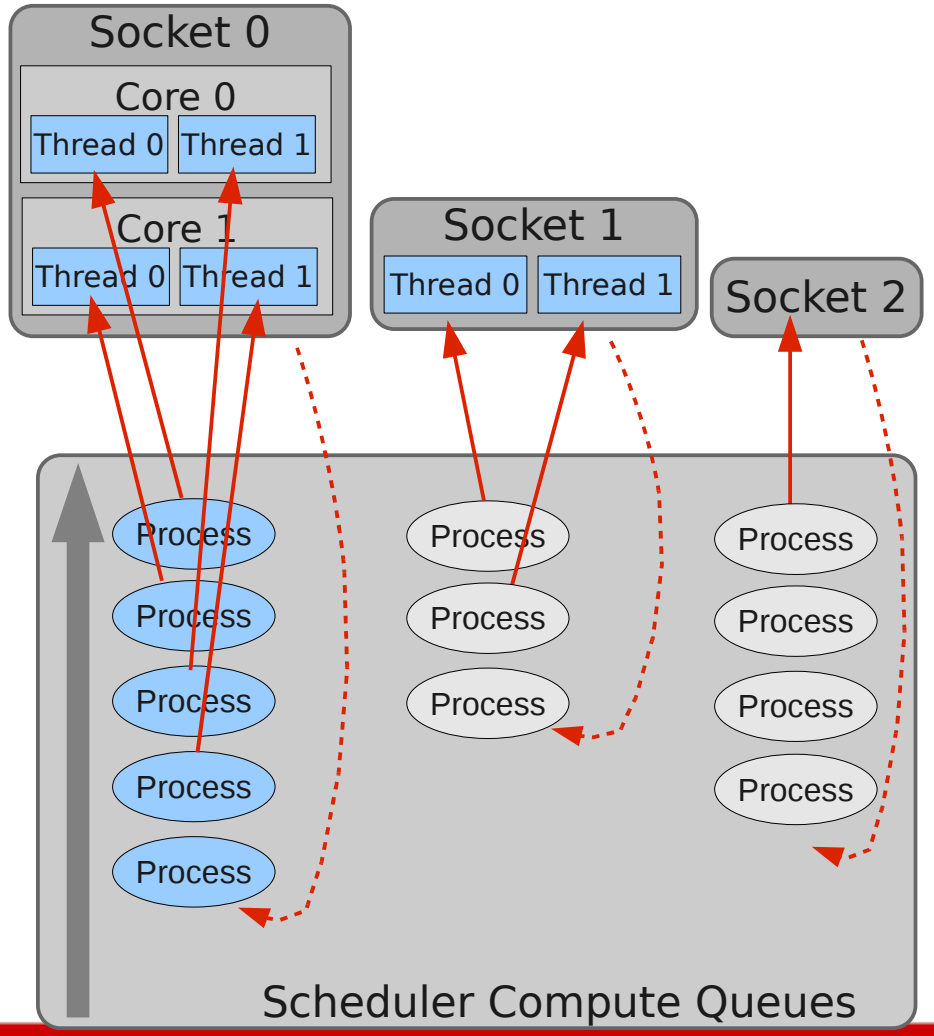
I.E. Multi-core, hyperthreaded & chips/sockets

Optimizes process scheduling to take advantage of shared on-chip cache, and NUMA memory nodes

Implements multilevel run queues for sockets and cores (as opposed to one run queue per processor or per system)

Strong CPU affinity avoids task bouncing

Requires system BIOS to report CPU topology correctly



Finer grained scheduler tuning

- /proc/sys/kernel/sched_*
- increase quantum on par with RHEL5 (used in tuned-adm)
  - echo 10000000 > /proc/sys/kernel/sched_min_granularity_ns
  - echo 15000000 > /proc/sys/kernel/sched_wakeup_granularity_ns

Capacity Tuning

- Memory
  - /proc/sys/vm/overcommit_memory
  - /proc/sys/vm/overcommit_ratio
  - /proc/sys/vm/max_map_count
  - /proc/sys/vm/nr_hugepages
- Kernel
  - /proc/sys/kernel/msgmax
  - /proc/sys/kernel/msgmnb
  - /proc/sys/kernel/msgmni
  - /proc/sys/kernel/shmall
  - /proc/sys/kernel/shmmax
  - /proc/sys/kernel/shmmni
  - /proc/sys/kernel/threads-max
- Filesystems
  - /proc/sys/fs/aio_max_nr
  - /proc/sys/fs/file_max
- OOM kills

OOM kills – lowmem consumption

```
Free pages:     9003696kB (8990400kB HighMem)
Active:323264 inactive:346882 dirty:327575 writeback:3686 unstable:0 free:2250924 slab:177094
mapped:15855 pagetables:987
DMA free:12640kB min:16kB low:32kB high:48kB active:0kB inactive:0kB present:16384kB
pages_scanned:149 all_unreclaimable? yes
protections[]: 0 0 0
Normal free:656kB min:928kB low:1856kB high:2784kB active:6976kB inactive:9976kB present:901120kB
pages_scanned:28281 all_unreclaimable? yes
protections[]: 0 0 0
HighMem free:8990400kB min:512kB low:1024kB high:1536kB active:1286080kB inactive:1377552kB
present:12451840kB pages_scanned:0 all_unreclaimable? no
protections[]: 0 0 0
DMA: 4*4kB 4*8kB 3*16kB 4*32kB 4*64kB 1*128kB 1*256kB 1*512kB 1*1024kB 1*2048kB 2*4096kB =
12640kB
Normal: 0*4kB 2*8kB 0*16kB 0*32kB 0*64kB 1*128kB 0*256kB 1*512kB 0*1024kB 0*2048kB 0*4096kB =
656kB
HighMem: 15994*4kB 17663*8kB 11584*16kB 8561*32kB 8193*64kB 1543*128kB 69*256kB 2101*512kB
1328*1024kB 765*2048kB 875*4096kB = 8990400kB
Swap cache: add 0, delete 0, find 0/0, race 0+0
Free swap:       8385912kB
3342336 pages of RAM
2916288 pages of HIGHMEM
224303 reserved pages
666061 pages shared
0 pages swap cached
Out of Memory: Killed process 22248 (httpd).
oom-killer: gfp_mask=0xd0
```


OOM kills – IO system stall

```
Free pages: 15096kB (1664kB HighMem) Active:34146 inactive:1995536 dirty:255
writeback:314829 unstable:0 free:3774 slab:39266 mapped:31803 pagetables:820
DMA free:12552kB min:16kB low:32kB high:48kB active:0kB inactive:0kB present:16384kB
pages_scanned:2023 all_unreclaimable? yes
protections[]: 0 0 0
Normal free:880kB min:928kB low:1856kB high:2784kB active:744kB inactive:660296kB
present:901120kB pages_scanned:726099 all_unreclaimable? yes
protections[]: 0 0 0
HighMem free:1664kB min:512kB low:1024kB high:1536kB active:135840kB inactive:7321848kB
present:7995388kB pages_scanned:0 all_unreclaimable? no
protections[]: 0 0 0
DMA: 2*4kB 4*8kB 2*16kB 4*32kB 3*64kB 1*128kB 1*256kB 1*512kB 1*1024kB 1*2048kB 2*4096kB =
12552kB
Normal: 0*4kB 18*8kB 14*16kB 0*32kB 0*64kB 0*128kB 0*256kB 1*512kB 0*1024kB 0*2048kB
0*4096kB = 880kB
HighMem: 6*4kB 9*8kB 66*16kB 0*32kB 0*64kB 0*128kB 0*256kB 1*512kB 0*1024kB 0*2048kB
0*4096kB = 1664kB
Swap cache: add 856, delete 599, find 341/403, race 0+0
0 bounce buffer pages
Free swap:       4193264kB
2228223 pages of RAM
1867481 pages of HIGHMEM
150341 reserved pages
343042 pages shared
257 pages swap cached
kernel: Out of Memory: Killed process 3450 (hpsmhd).
```

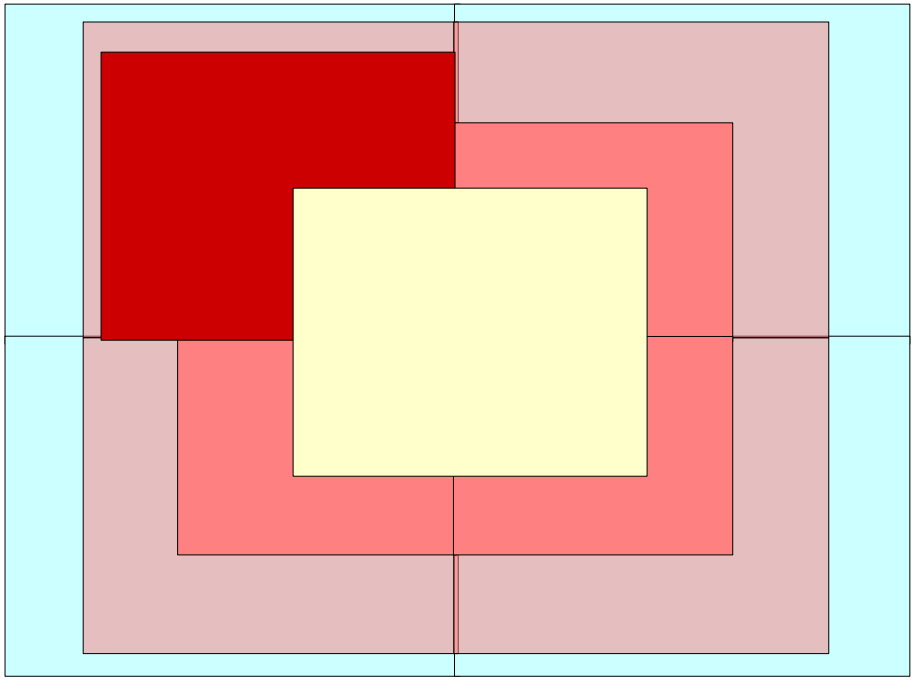

Eliminating OOMkills

- RHEL4
  - /proc/sys/vm/oom-kill – oom kill enable/disable flag(default 1).

- RHEL5 & RHEL6
  - /proc/<pid>/oom_adj – per-process OOM adjustment(-17 to +15)
  - Set to -17 to disable that process from being OOM killed
  - Decrease to decrease OOM kill likelyhood.
  - Increase to increase OOM kill likelyhood.
  - /proc/<pid>/oom_score – current OOM kill priority.

NUMA and Huge Pages

- Huge page allocation takes place uniformly across NUMA nodes
- Make sure that database shared segments are sized to fit
- Workaround – Allocate Huge pages / Start DB / De-allocate Huge pages



Physical Memory
128G – 4 NUMA nodes

Huge Pages – 80G
20G in each NUMA node

24G DB Shared Segment using Huge Pages

24G DB Shared Segment using NUMA and Huge Pages

Huge Pages – 100G
25G in each NUMA node

Hugepages - before

```
$vmstat
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu------
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0      0 15623656  31044 401120    0    0   187    14  163   75  1  0 97  2  0


$cat /proc/meminfo
MemTotal:     16301368 kB
MemFree:      15623604 kB
...
HugePages_Total:     0
HugePages_Free:      0
HugePages_Rsvd:      0
Hugepagesize:     2048 kB
```


Hugepages reserving

```
$echo 2000 > /proc/sys/vm/nr_hugepages
$vmstat
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu------
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0      0 11526632  31168 401780    0    0   129    10  156   63  1  0 98  1  0


$cat /proc/meminfo
MemTotal:     16301368 kB
MemFree:      11526520 kB
...
HugePages_Total:  2000
HugePages_Free:   2000
HugePages_Rsvd:      0
Hugepagesize:     2048 kB
```


Hugepages - using

```
$mount -t hugetlbfs hugetlbfs /huge
$cp 1GB-file /huge/junk


$vmstat
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu------
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0      0 10526632  31168 1401780    0    0   129    10  156   63  1  0 98  1  0


$cat /proc/meminfo
LowTotal:     16301368 kB
LowFree:      11524756 kB
...
HugePages_Total:  2000
HugePages_Free:   1488
HugePages_Rsvd:      0
Hugepagesize:     2048 kB
```


Hugepages - releasing

```
$rm /huge/junk
$cat /proc/meminfo
MemTotal:     16301368 kB
MemFree:      11524776 kB
...
HugePages_Total:  2000
HugePages_Free:   2000
HugePages_Rsvd:      0
Hugepagesize:     2048 kB


$echo 0 > /proc/sys/vm/nr_hugepages
$vmstat
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu------
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0      0 15620488  31512 401944    0    0    71     6  149   59  1  0 98  1  0


$cat /proc/meminfo
MemTotal:     16301368 kB
MemFree:      15620500 kB
...
HugePages_Total:     0
HugePages_Free:      0
HugePages_Rsvd:      0
Hugepagesize:     2048 kB
```


NUMA Hugepages - reserving

[root@dhcp-100-19-50 ~]# cat /sys/devices/system/node/*/meminfo | grep Huge
Node 0 HugePages_Total: 0
Node 0 HugePages_Free: 0
Node 1 HugePages_Total: 0
Node 1 HugePages_Free: 0
[root@dhcp-100-19-50 ~]# echo 6000 > /proc/sys/vm/nr_hugepages
[root@dhcp-100-19-50 ~]# cat /sys/devices/system/node/*/meminfo | grep Huge
Node 0 HugePages_Total: 2980
Node 0 HugePages_Free: 2980
Node 1 HugePages_Total: 3020
Node 1 HugePages_Free: 3020

NUMA Hugepages - using

[root@dhcp-100-19-50 ~]# mount -t hugetlbfs hugetlbfs /huge

[root@dhcp-100-19-50 ~]# /usr/tmp/mmapwrite /huge/junk 32 &

[1] 18804

[root@dhcp-100-19-50 ~]# Writing 1048576 pages of random junk to file /huge/junk

wrote 4294967296 bytes to file /huge/junk

[root@dhcp-100-19-50 ~]# cat /sys/devices/system/node/*/meminfo | grep Huge

Node 0 HugePages_Total: 2980

Node 0 HugePages_Free: 2980

Node 1 HugePages_Total: 3020

Node 1 HugePages_Free: 972

NUMA Hugepages - using(overcommit)

[root@dhcp-100-19-50 ~]# /usr/tmp/mmapwrite /huge/junk 33 &

[1] 18815

[root@dhcp-100-19-50 ~]# Writing 2097152 pages of random junk to file /huge/junk

wrote 8589934592 bytes to file /huge/junk

[root@dhcp-100-19-50 ~]# cat /sys/devices/system/node/*/meminfo | grep Huge

Node 0 HugePages_Total: 2980

Node 0 HugePages_Free: 1904

Node 1 HugePages_Total: 3020

Node 1 HugePages_Free: 0

NUMA Hugepages - reducing

[root@dhcp-100-19-50 ~]# cat /sys/devices/system/node/*/meminfo | grep Huge

Node 0 HugePages_Total: 2980

Node 0 HugePages_Free: 2980

Node 1 HugePages_Total: 3020

Node 1 HugePages_Free: 3020

[root@dhcp-100-19-50 ~]# echo 3000 > /proc/sys/vm/nr_hugepages

[root@dhcp-100-19-50 ~]# cat /sys/devices/system/node/*/meminfo | grep Huge

Node 0 HugePages_Total: 0

Node 0 HugePages_Free: 0

Node 1 HugePages_Total: 3000

Node 1 HugePages_Free: 3000

NUMA Hugepages - freeing/reserving

[root@dhcp-100-19-50 ~]# echo 6000 > /proc/sys/vm/nr_hugepages

[root@dhcp-100-19-50 ~]# cat /sys/devices/system/node/*/meminfo | grep Huge

Node 0 HugePages_Total: 2982

Node 0 HugePages_Free: 2982

Node 1 HugePages_Total: 3018

Node 1 HugePages_Free: 3018

[root@dhcp-100-19-50 ~]# echo 0 > /proc/sys/vm/nr_hugepages

[root@dhcp-100-19-50 ~]# echo 3000 > /proc/sys/vm/nr_hugepages

[root@dhcp-100-19-50 ~]# cat /sys/devices/system/node/*/meminfo | grep Huge

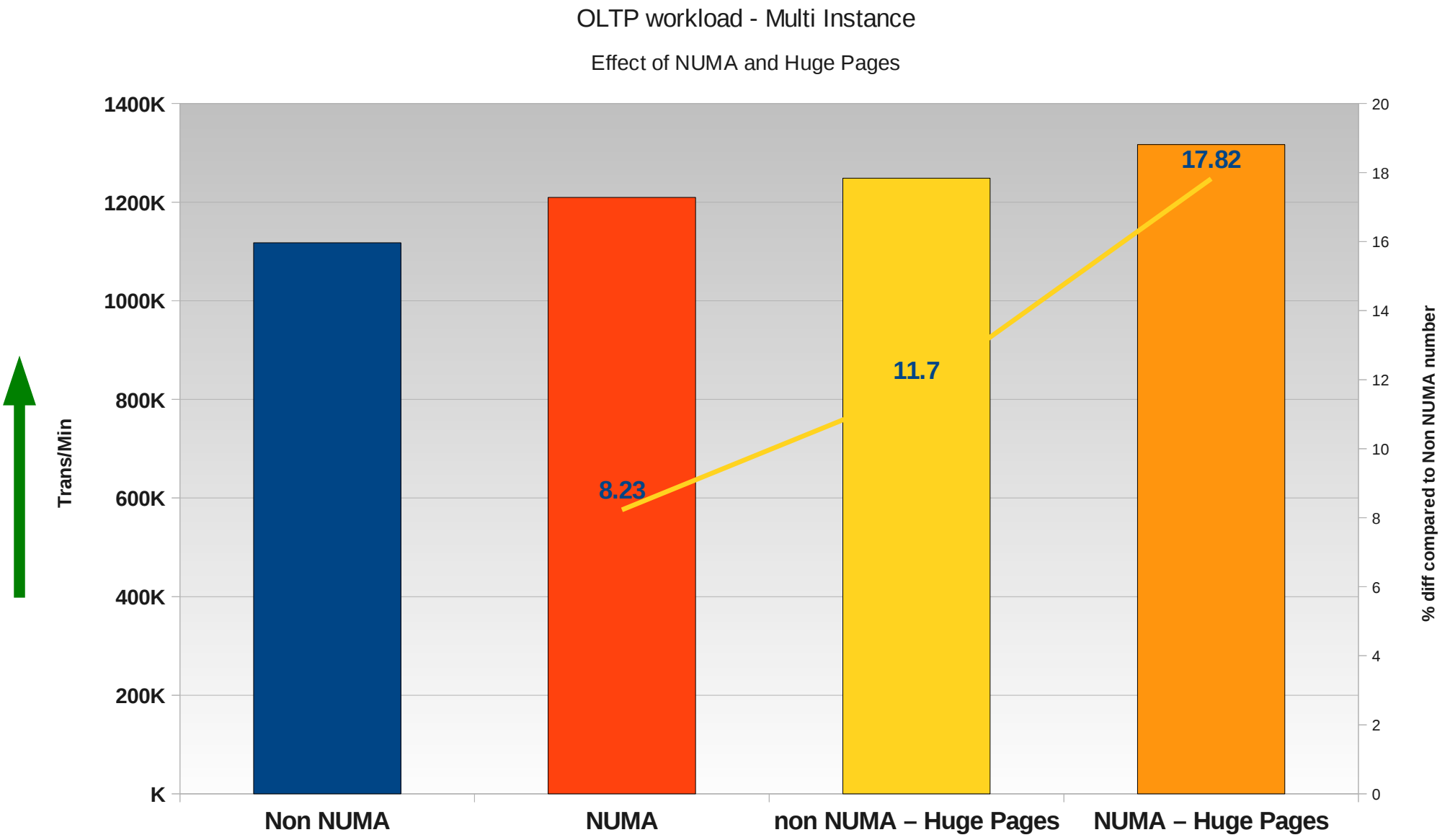
Node 0 HugePages_Total: 1500

Node 0 HugePages_Free: 1500

Node 1 HugePages_Total: 1500

Node 1 HugePages_Free: 1500

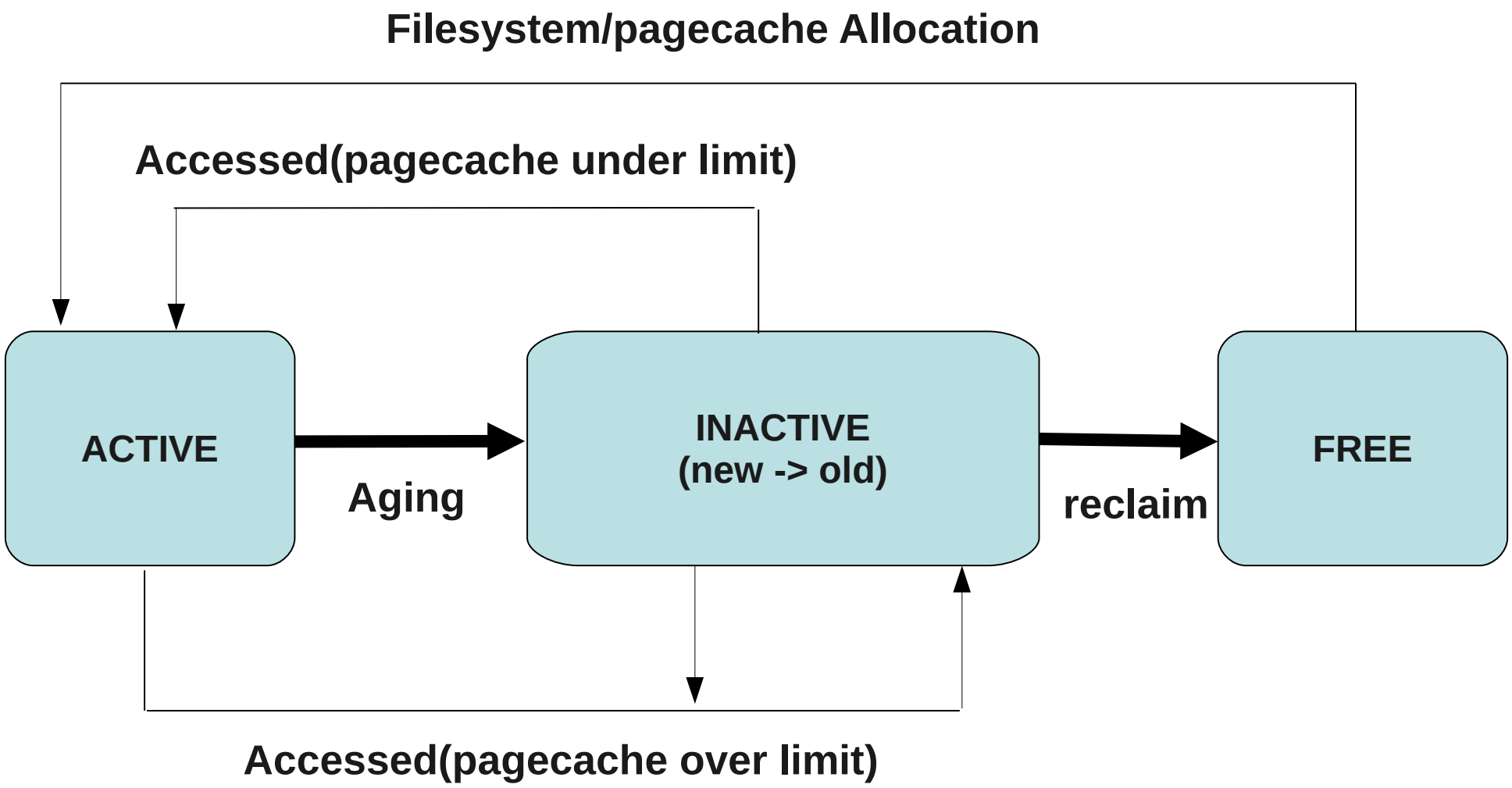
OLTP Workload – Effect of NUMA and Huge Pages



/proc/sys/vm/pagecache(RHEL4&5)

- Controls when pagecache memory is deactivated.
- Default is 100%
- Lower
  - Prevents swapping out anonymous memory
- Higher
  - Favors pagecache pages
  - Disabled at 100%

Pagecache Tuning (RHEL)



swappiness

- Not needed as much in RHEL6
- Controls how aggressively the system reclaims “mapped” memory:
- Anonymous memory - swapping
- Mapped file pages – writing if dirty and freeing
- System V shared memory - swapping
- Decreasing: more aggressive reclaiming of unmapped pagecache memory
- Increasing: more aggressive swapping of mapped memory

/proc/sys/vm/swappiness

Database server with /proc/sys/vm/swappiness set to 60(default)

```
procs -----------memory---------- ---swap--  -----io---- --system-- ----cpu----
r  b   swpd   free   buff  cache   si   so    bi    bo    in    cs  us sy id wa
5  1  643644 26788   3544 32341788 880 120    4044 7496  1302 20846 25 34 25 16
```

Database server with /proc/sys/vm/swappiness set to 10

```
procs -----------memory---------- ---swap--  -----io---- --system-- ----cpu----
r  b   swpd   free   buff  cache   si   so    bi    bo    in    cs  us sy id wa
8  3      0   24228   6724 32280696  0    0    23888 63776 1286 20020 24 38 13 26
```


zone_reclaim_mode

- Controls NUMA specific memory allocation policy
- When set and node memory is exhausted:
  - Reclaim memory from local node rather than allocating from next node
  - Slower allocation, higher NUMA hit ratio
- When clear and node memory is exhausted:
  - Allocate from all nodes before reclaiming memory
  - Faster allocation, higher NUMA miss ratio
- Default is set at boot time based on NUMA factor

/proc/sys/vm/min_free_kbytes

Directly controls the page reclaim watermarks in KB

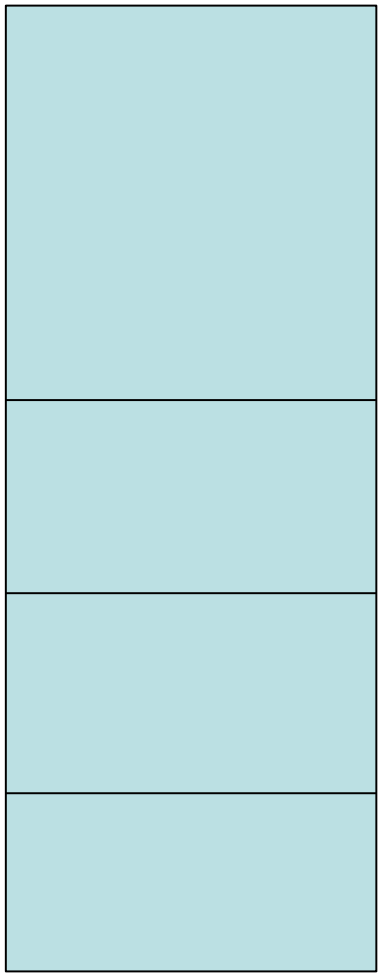
Defaults are higher when THP is enabled

```
# echo 1024 > /proc/sys/vm/min_free_kbytes
----------------------------------------------------------
Node 0 DMA free:4420kB min:8kB low:8kB high:12kB
Node 0 DMA32 free:14456kB min:1012kB low:1264kB high:1516kB
----------------------------------------------------------

echo 2048 > /proc/sys/vm/min_free_kbytes
----------------------------------------------------------
Node 0 DMA free:4420kB min:20kB low:24kB high:28kB
Node 0 DMA32 free:14456kB min:2024kB low:2528kB high:3036kB
----------------------------------------------------------
```


Memory reclaim Watermarks - min_free_kbytes

Free List



All of RAM

Do nothing

Pages High – kswapd sleeps above High

kswapd reclaims memory

Pages Low – kswapd wakesup at Low

kswapd reclaims memory

Pages Min – all memory allocators reclaim at Min

user processes/kswapd reclaim memory

0

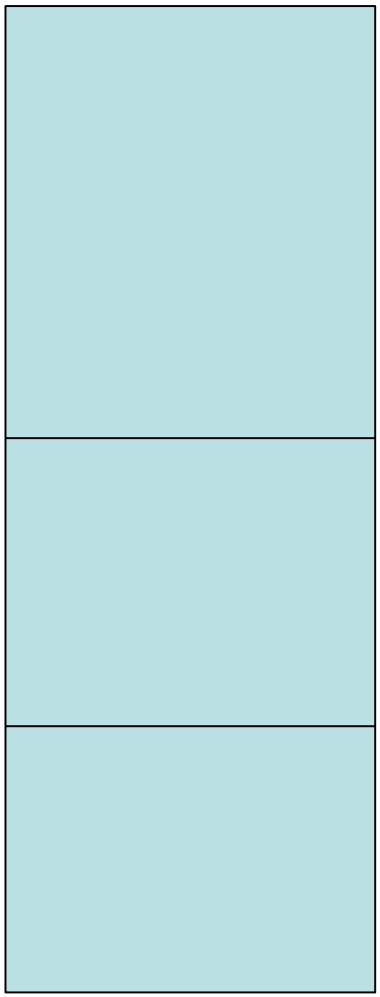
/proc/sys/vm/dirty_background_ratio
/proc/sys/vm/dirty_background_bytes

- Controls when dirty pagecache memory starts getting written.
- Default is 10%
- Lower
  - flushing starts earlier
  - less dirty pagecache and smaller IO streams
- Higher
  - flushing starts later
  - more dirty pagecache and larger IO streams
- dirty_background_bytes over-rides when you want < 1%

/proc/sys/vm/dirty_ratio
/proc/sys/vm/dirty_bytes

- Absolute limit to percentage of dirty pagecache memory
- Default is 20%
- Lower means clean pagecache and smaller IO streams
- Higher means dirty pagecache and larger IO streams
- dirty_bytes overrides when you want < 1%

dirty_ratio and dirty_background_ratio



100% of pagecache RAM dirty

flushd and write()'ng processes write dirty buffers

dirty_ratio(20% of RAM dirty) – processes start synchronous writes

flushd writes dirty buffers in background

dirty_background_ratio(10% of RAM dirty) – wakeup flushd

do_nothing

0% of pagecache RAM dirty

(Hint)flushing the pagecache

```
# sync
# echo 1 > /proc/sys/vm/drop_caches
```

```
procs -----------memory---------- ---swap-- -----io---- --system-- ----cpu----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in    cs us sy id wa
 0  0    224  57184 107808 3350196    0    0     0    56 1136   212  0  0 83 17
 0  0    224  57184 107808 3350196    0    0     0     0 1039   198  0  0 100  0
 0  0    224  57184 107808 3350196    0    0     0     0 1021   188  0  0 100  0
 0  0    224  57184 107808 3350196    0    0     0     0 1035   204  0  0 100  0
 0  0    224  57248 107808 3350196    0    0     0     0 1008   164  0  0 100  0
 3  0    224 2128160    176 1438636    0    0     0     0 1030   197  0 15 85  0
 0  0    224 3610656    204  34408    0    0    28    36 1027   177  0 32 67  2
 0  0    224 3610656    204  34408    0    0     0     0 1026   180  0  0 100  0
 0  0    224 3610720    212  34400    0    0     8     0 1010   183  0  0 99  1
```


(Hint)flushing the slabcache

echo 2 > /proc/sys/vm/drop_caches

```
[tmp]# cat /proc/meminfo
MemTotal:       3907444 kB
MemFree:        3104576 kB

Slab:            415420 kB

Hugepagesize:      2048 kB
```

```
tmp]# cat /proc/meminfo
MemTotal:       3907444 kB
MemFree:        3301788 kB

Slab:            218208 kB

Hugepagesize:      2048 kB
```


Section 5: Examples

RHEL 6 Filesystem (Wed/Thu)

- Ext3 / 4 / XFS / GFS2
- Large Scale IO w/ Fusion IO
- > 2 Million IO/s, 12.5 GB/sec
- KVM Virtualization(Wed/Thus)

RHEL 6 File Systems

ext4

- Scales to 16TB; default in Red Hat Enterprise Linux 6

XFS

- Support for extremely large file-sizes and high-end arrays.

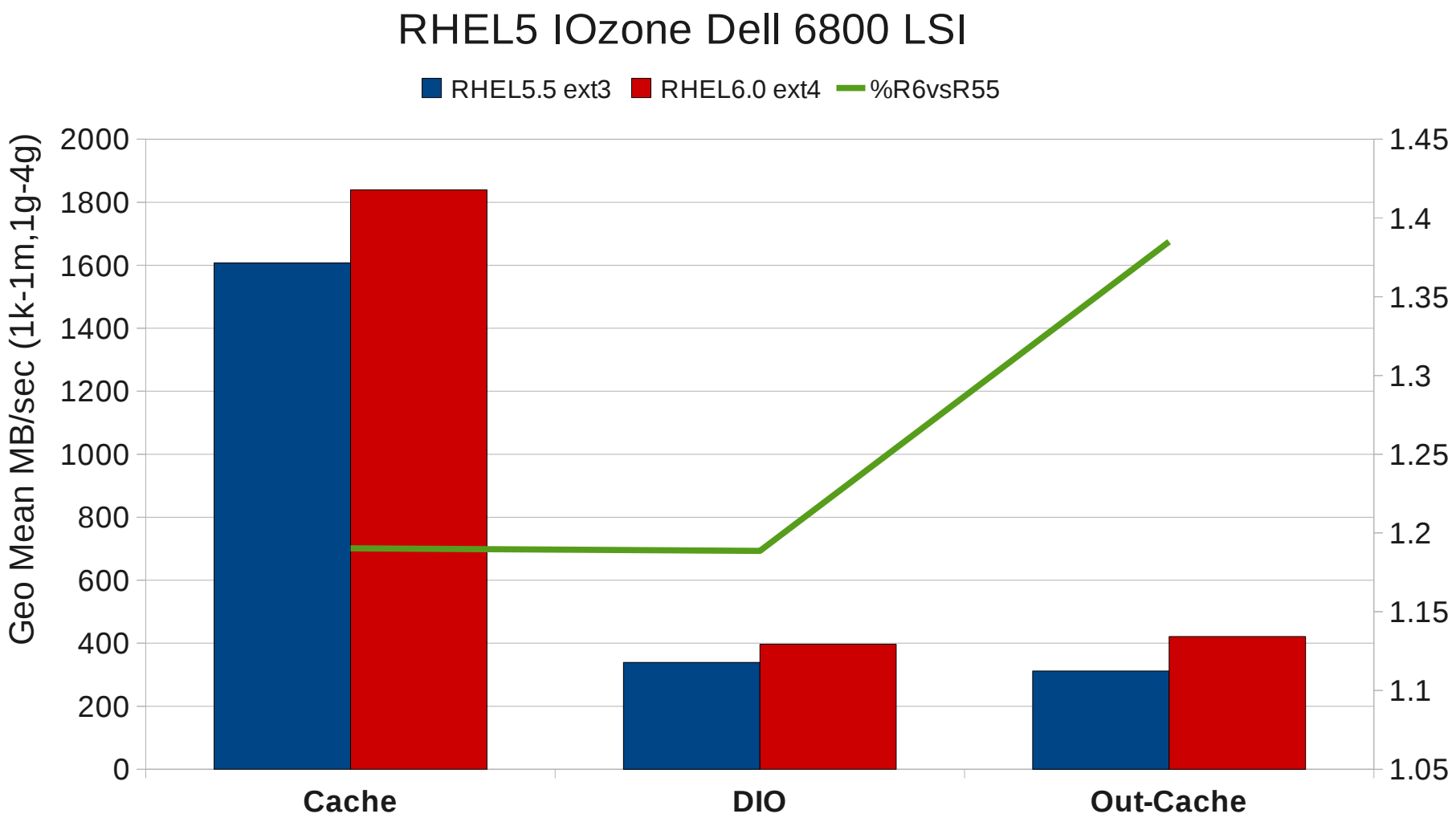
GFS2

- Supports 2 to 16 nodes

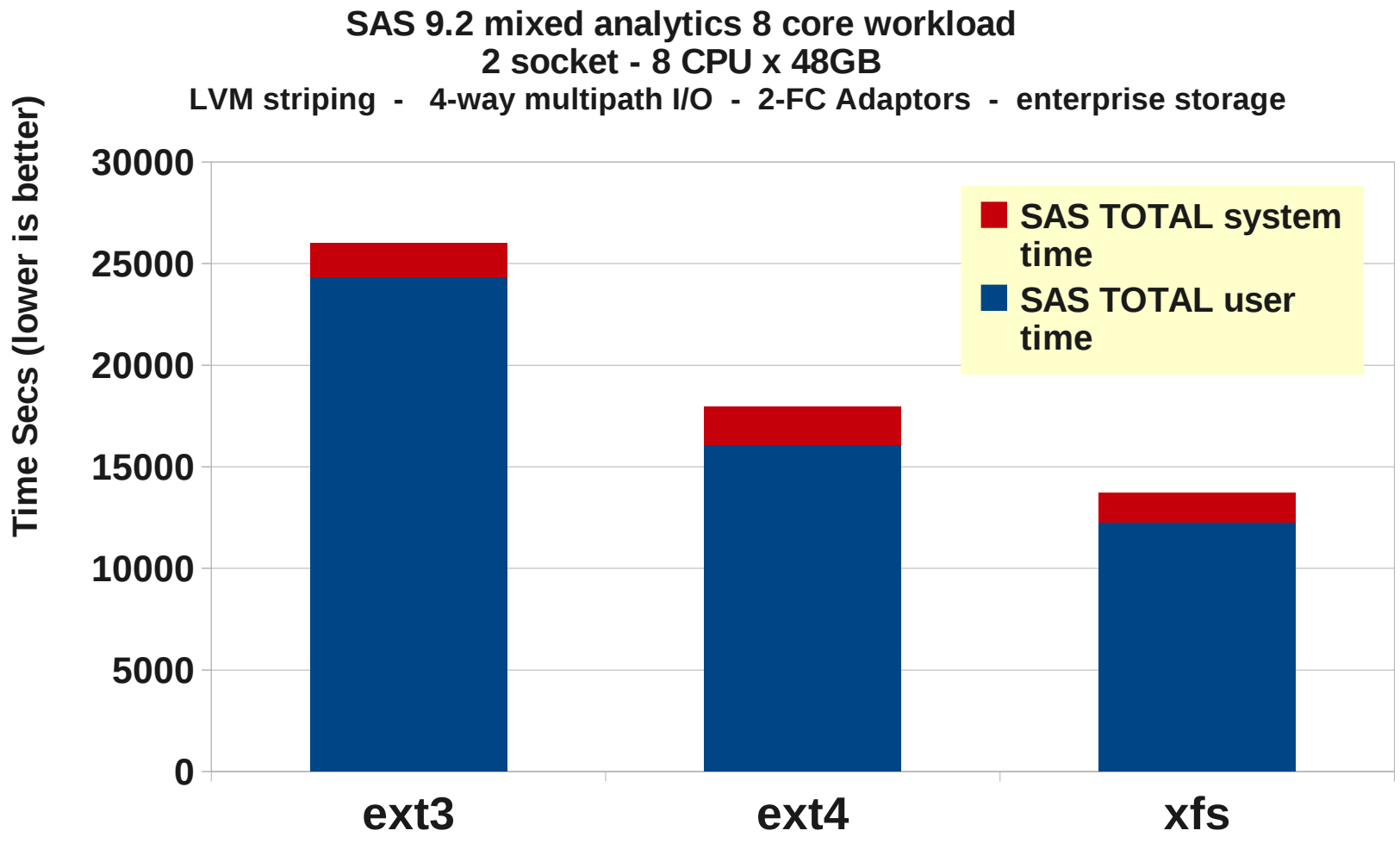
BTRFS

- New file system showing great promise. Included as Technology Preview.

RHEL6 ext4 vs RHEL5.5 ext3



RHEL 6 (2.6.32) vs RHEL 5.5 SAS



System and Fusion Device settings

Test 1 : Hardware:
 Intel Boxboro EX 32-core (64w/ HT), 2.261 Ghz,
 128GB Memory 8 x FusionIO duo
OS:
 RHEL6 (2.6.32-71), RHEL6.1 2.6.32-94

Settings for the driver module
•Enable MSI – decreases cpu utilization
 options iomemory-vsl disable_msi=0
•Coalesce interrupts – Determines how long the drive waits before sending interrupts
 options iomemory-vsl tintr_hw_wait=50

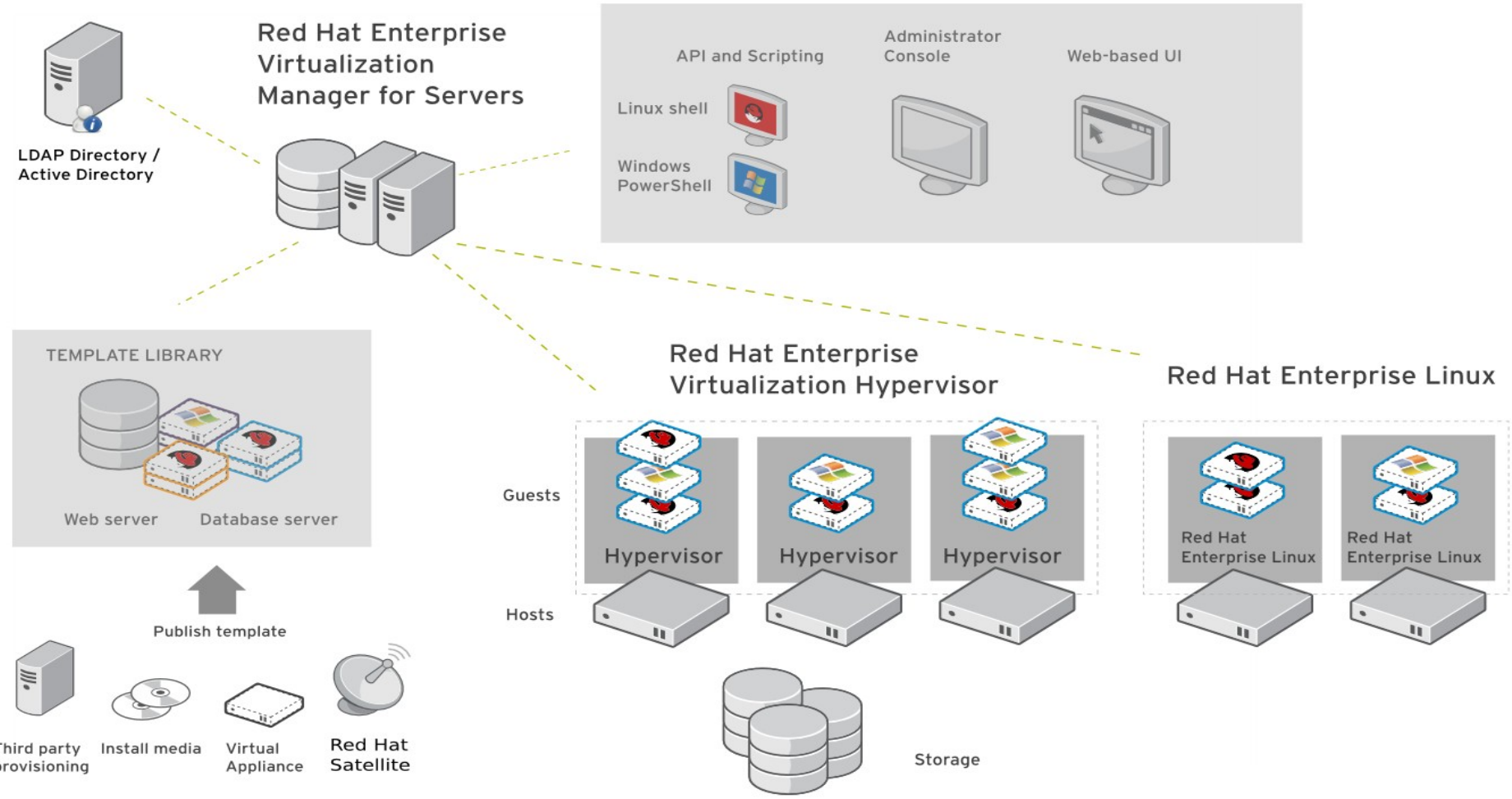
Drive information

fct0 Attached as 'fioa' (block device)
 Fusion-io ioDrive Duo 1.28TB, Product Number:FS3-202-641-CS SN:111048
 Located in 0 Upper slot of ioDrive Duo SN:111348
 PCI:0b:00.0
 Firmware v5.0.5, rev 43674
 640.00 GBytes block device size, 812 GBytes physical device size
 Sufficient power available: Unknown
 Internal temperature: avg 42.3 degC, max 47.2 degC
 Media status: Healthy; Reserves: 100.00%, warn at 10.00%

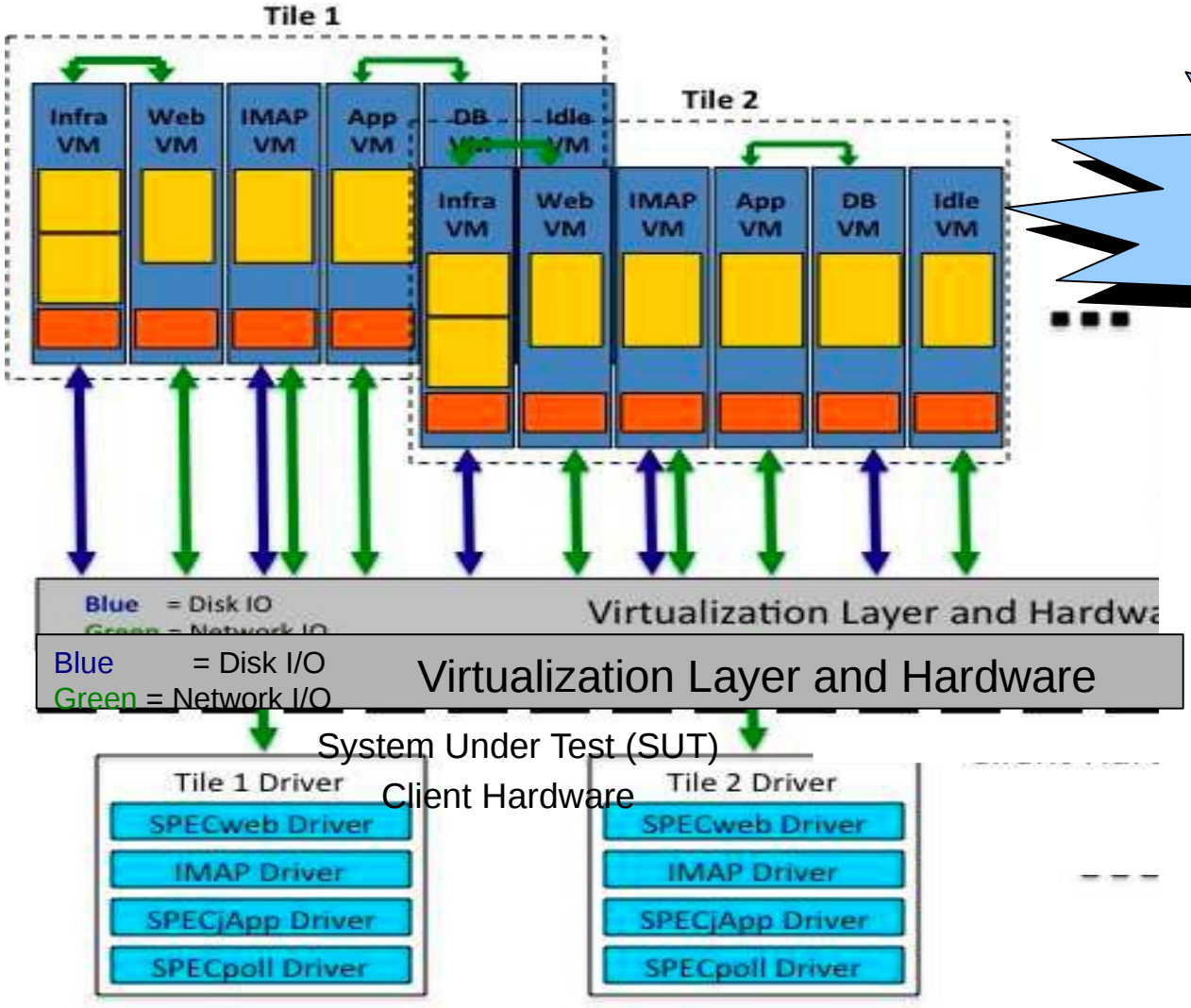
RHEL6.0 2.6.32-71 and RHEL6.1 2.6.32-94 (beta) w/ 8 FusionIO
Intel Boxboro 64-cpu, 128GB mem, 8 Fusion IO duo

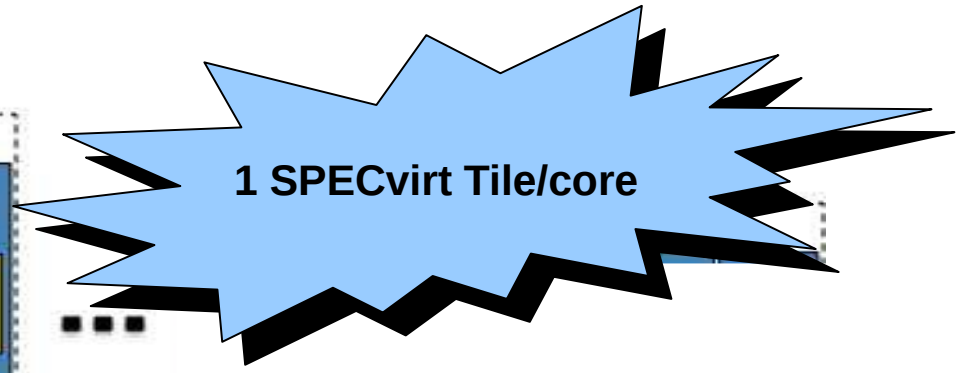
IO/secu	RHEL6.1 2.6.32-94	RHEL6.0 GA 2.6.32-71	%Gain 6.1/6.0
IO/s 1024k Seq Read	**2261223**	NA	
IO/s 8k Seq Read	**1406300**	1320188	**6.5%**
IO/s 8k Seq Write	**1253313**	1114141	**12.5%**
GB/s 1024k Seq Read	2.16	NA	
GB/s 8k Seq Read	11.2	10.07	**11.2%**
GB/s 8k Seq Write	9.56	8.5	**12.5%**

Red Hat Enterprise Virtualization



SPECvirt2010: *RHEL 5.5 and KVM Post Industry Leading Results on IBM x3640M3 w/ Xeon 5680*





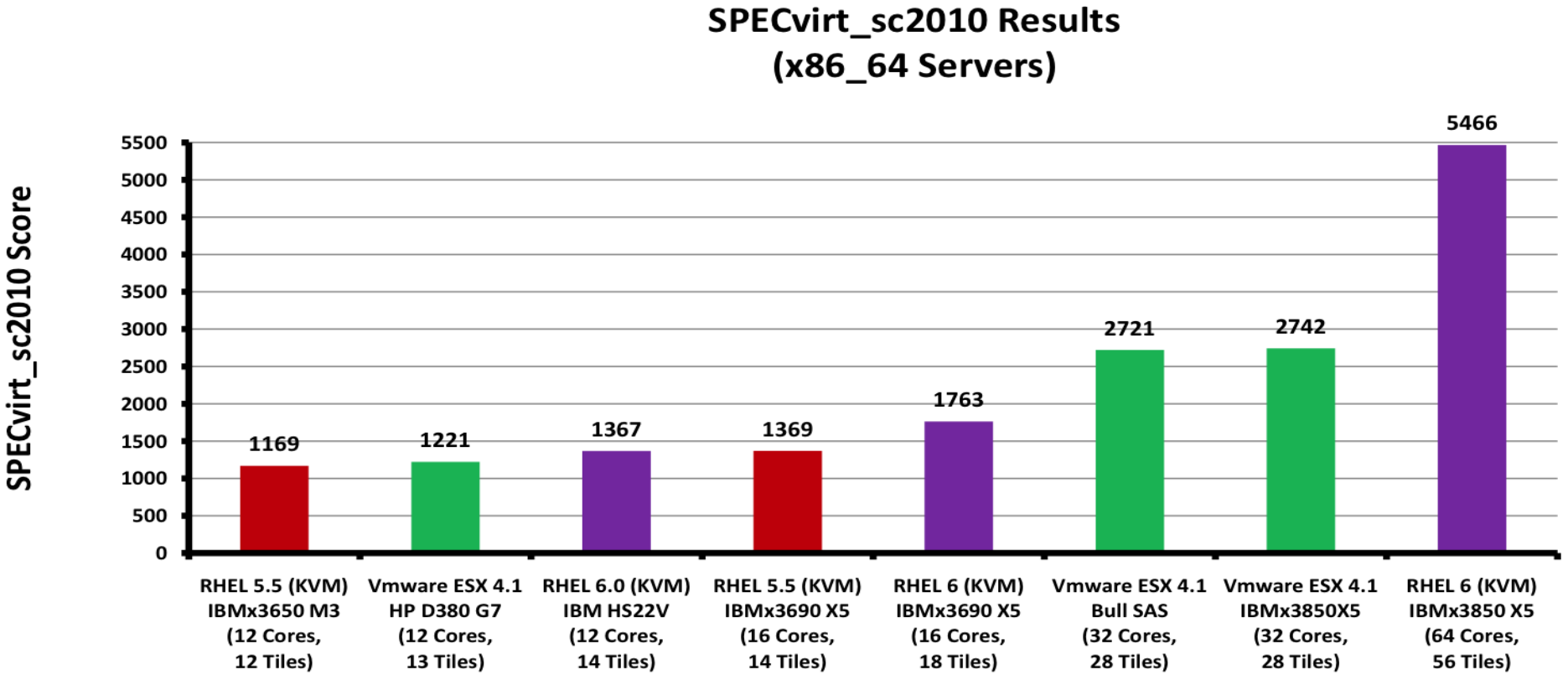
Key Enablers:

- ✓ SR-IOV
- ✓ Huge Pages
- ✓ NUMA
- ✓ Node Binding

http://www.spec.org/virt_sc2

Be More Flexible

RHEL Guest e/ all Virtualization



"SPECvirt_sc2010 Benchmark Results " March 2010

ALL SPECvirt_sc2010 results published to date use RHEL as the guest / VM Operating System!
RHEL 6 shows 29% better SPECvirt performance than RHEL 5.5 (KVM) on the same hardware!

Virtualization
Memory Enhancements

Transparent hugepages

- Efficiently manage large memory allocations as one unit

Extended Page Table (EPT) age bits

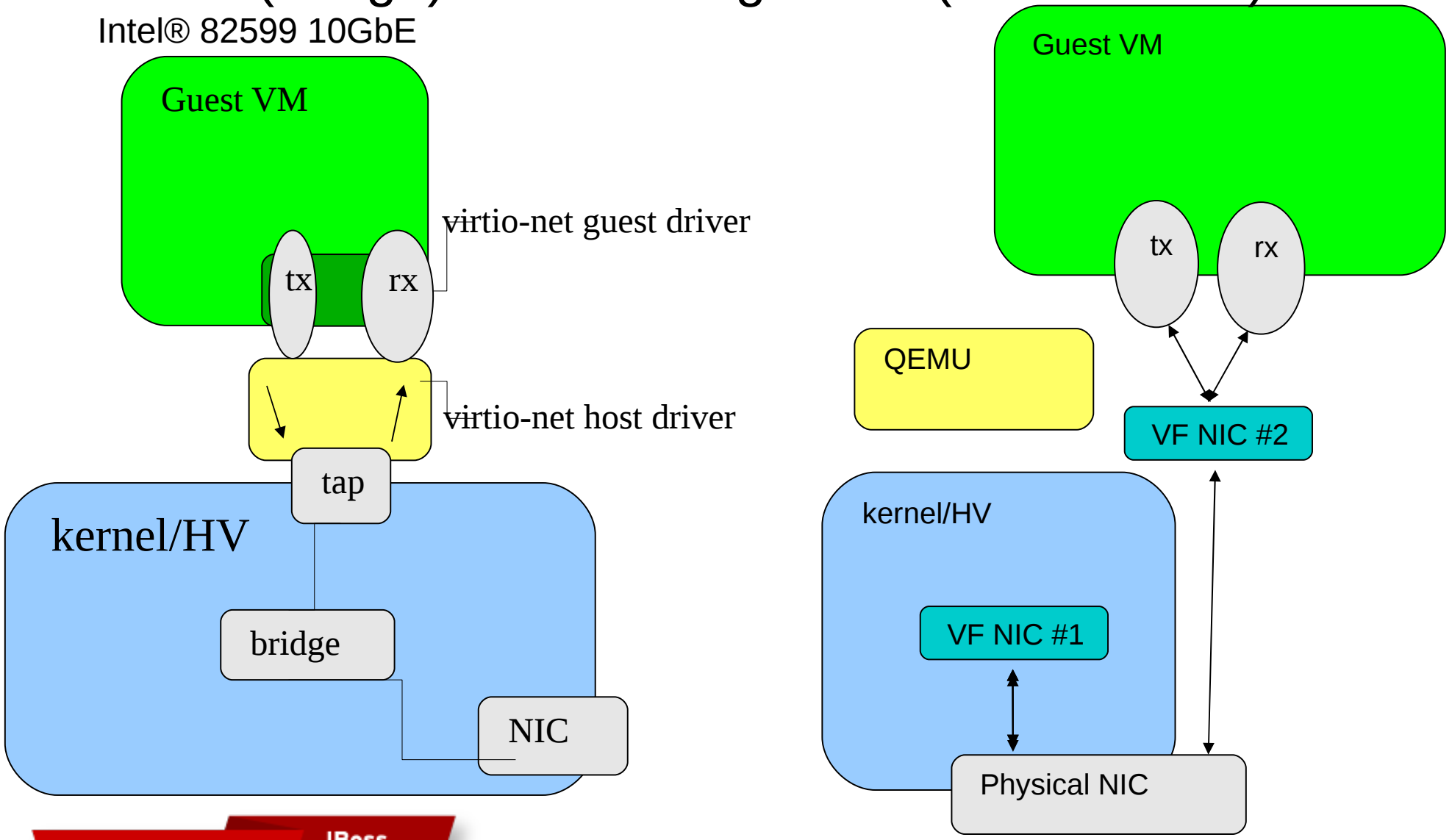
- Allow host to make smarter swap choice when under pressure.

Kernel Same-page Merging (KSM)

- Consolidate duplicate pages.
- Particularly efficient for Windows guests.

Virtio (bridge) vs PCI assignment (vt-d/SR-IOV)

Intel® 82599 10GbE



Virtualization:
RHEL6 2.6.32 SAS Intel EP (12cpu/24gb)