VMware

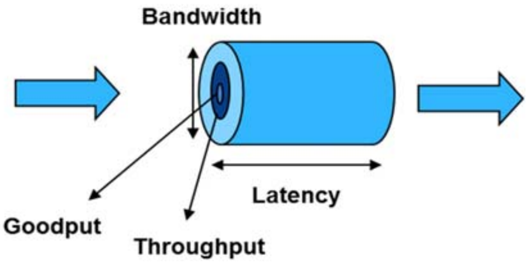
Performance Measurements

Tien-Fu Chen

Dept. of Computer Science
National Chiao Tung Univ.

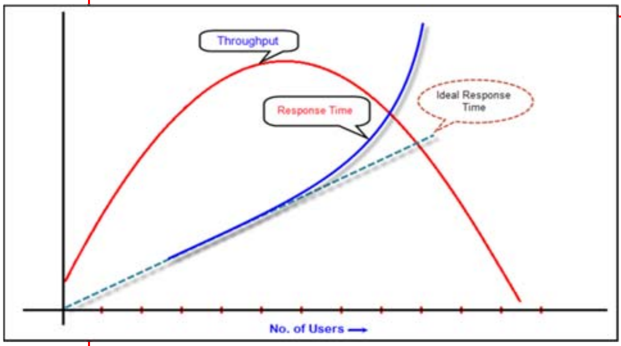
Performance Metrics

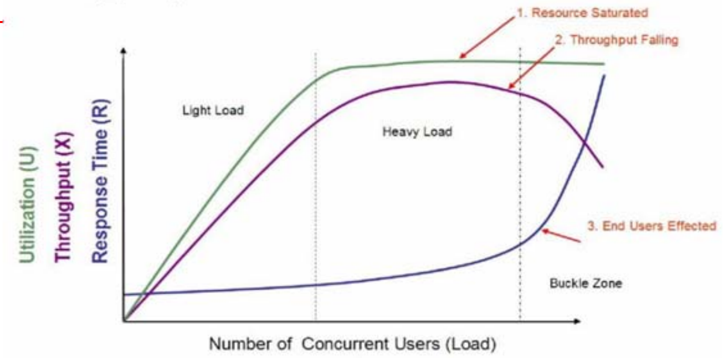
- CPU
  - Throughput: MIPS (%used), Goodput: useful instructions
  - Latency: Instruction Latency (cache latency, cache miss)
- Memory
  - Throughput: MB/Sec, Goodput: useful data
  - Latency: nanosecs
- Storage
  - Throughput: MB/Sec, IOPS/Sec, Goodput: useful data
  - Latency: Seek time
- Networking
  - Throughput: MB/Sec, IO/Sec, Goodput: useful traffic
  - Latency: microseconds

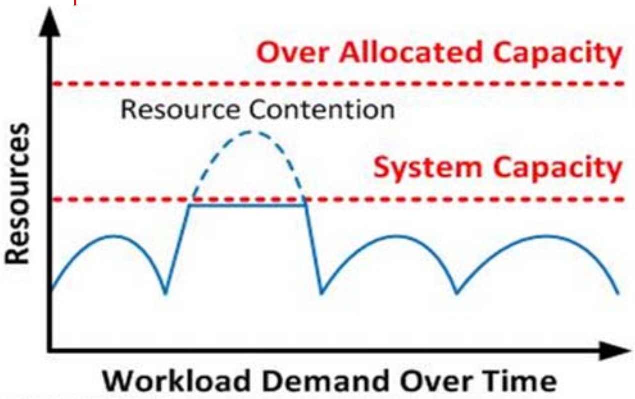


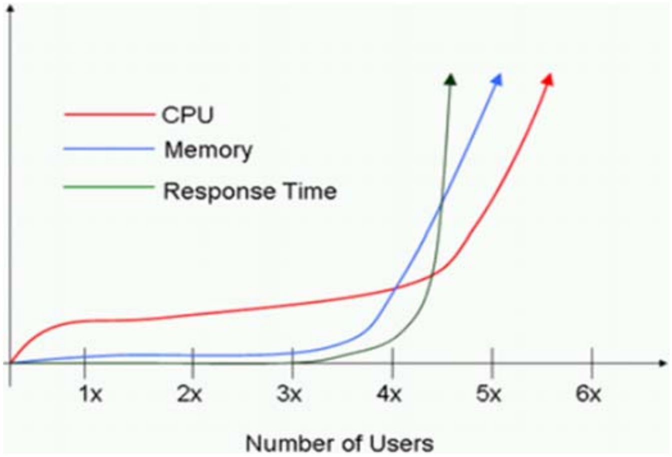
VM Provisioning

Service Quality vs throughput







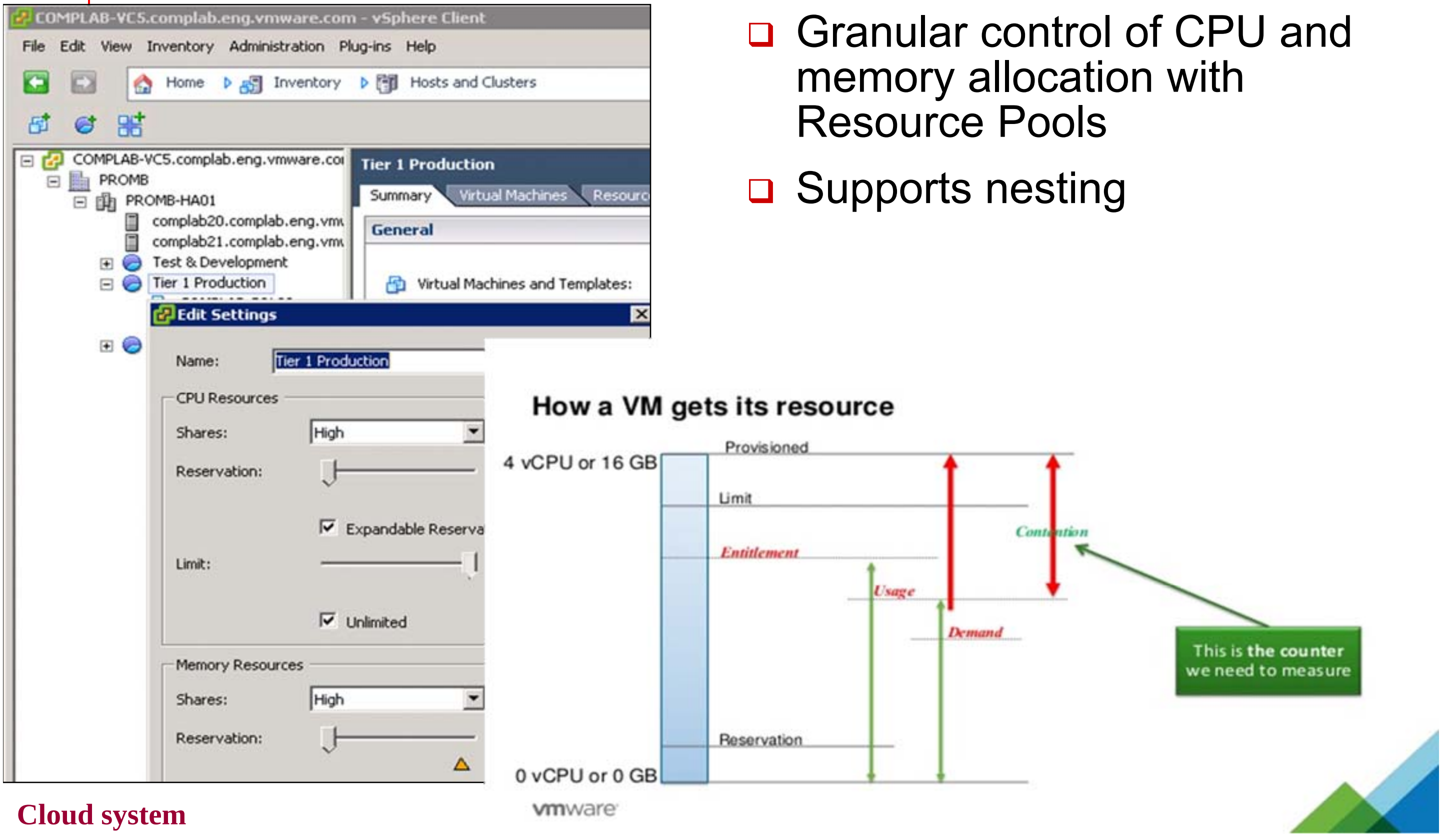


VMware resource: Reservations, Limits, and Shares

- Reservation
  - a guarantee on either memory or cpu for a virtual machine. You define the reservation in MB or MHZ.
  - Memory: swap file = configured memory – memory reservation
  - CPU reservations ensure that a VM always get physical cpu
- Limit
  - the max amount of physical memory the virtual machine can use.
  - Setting the limit lower than the configured memory for a VM it will cause swapping and balloon activity for the VM.
- Share
  - how much access to a resource compared to something else.
  - Every VM has 1000 shares configured per. vCPU as a default.
- A **resource pool** is a logical abstraction for flexible management of resources. Resource pools can be grouped into hierarchies and used to hierarchically partition available CPU and memory resources.

VMware vSphere Provides Advanced Resource Management

- Granular control of CPU and memory allocation with Resource Pools
- Supports nesting



Name: Tier 1 Production

CPU Resources

Shares: High

Reservation:

☑ Expandable Reserva

Limit:

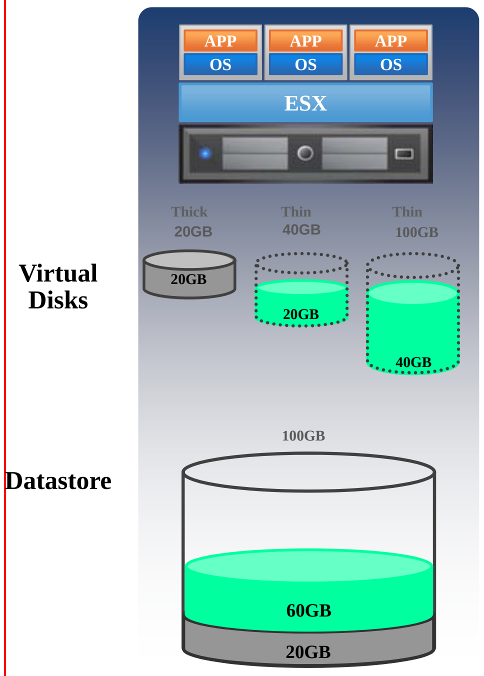
☑ Unlimited

Memory Resources

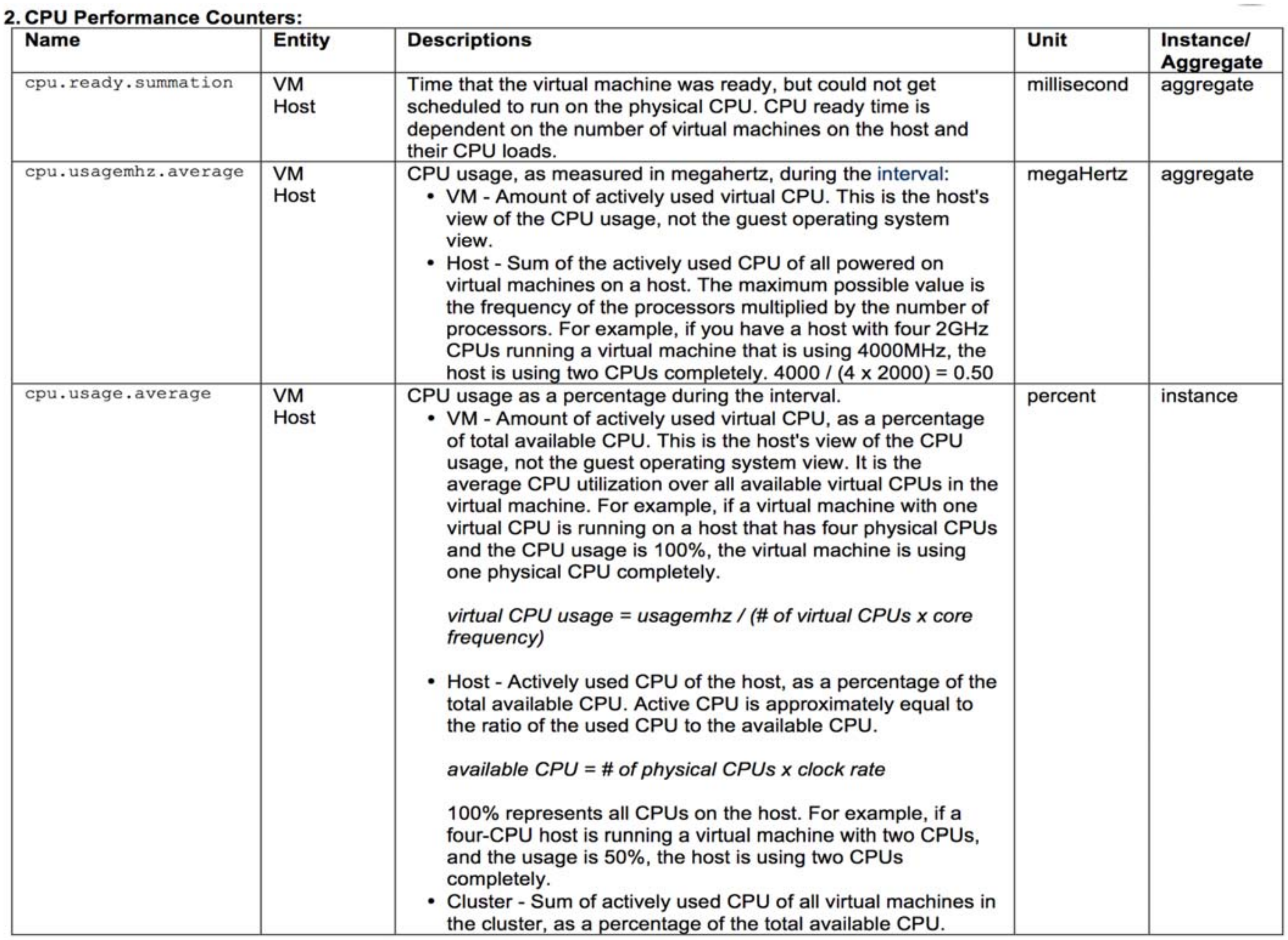
Shares: High

Reservation:

vStorage Thin Provisioning

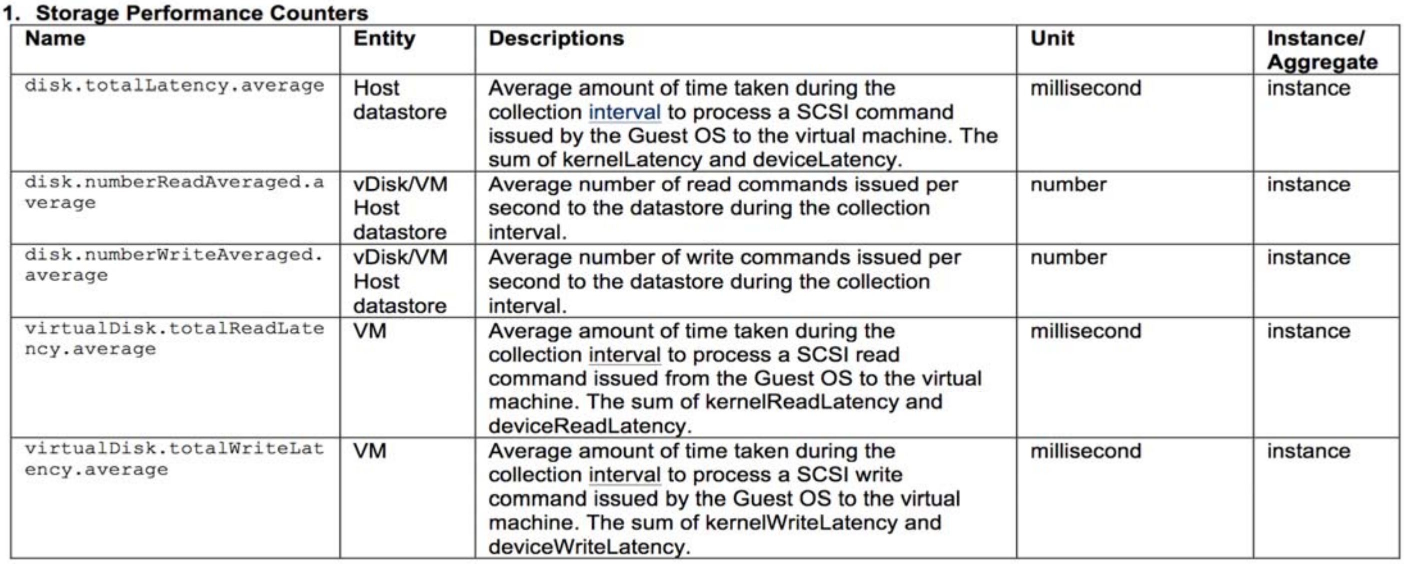


- Virtual machine disks consume only the amount of physical space in use
  - Virtual machine sees full logical disk size at all times
  - Full reporting and alerting on allocation and consumption
- Significantly improve storage utilization
- Eliminate need to over-provision virtual disks
- Reduce storage costs by up to 50%



2. CPU Performance Counters:

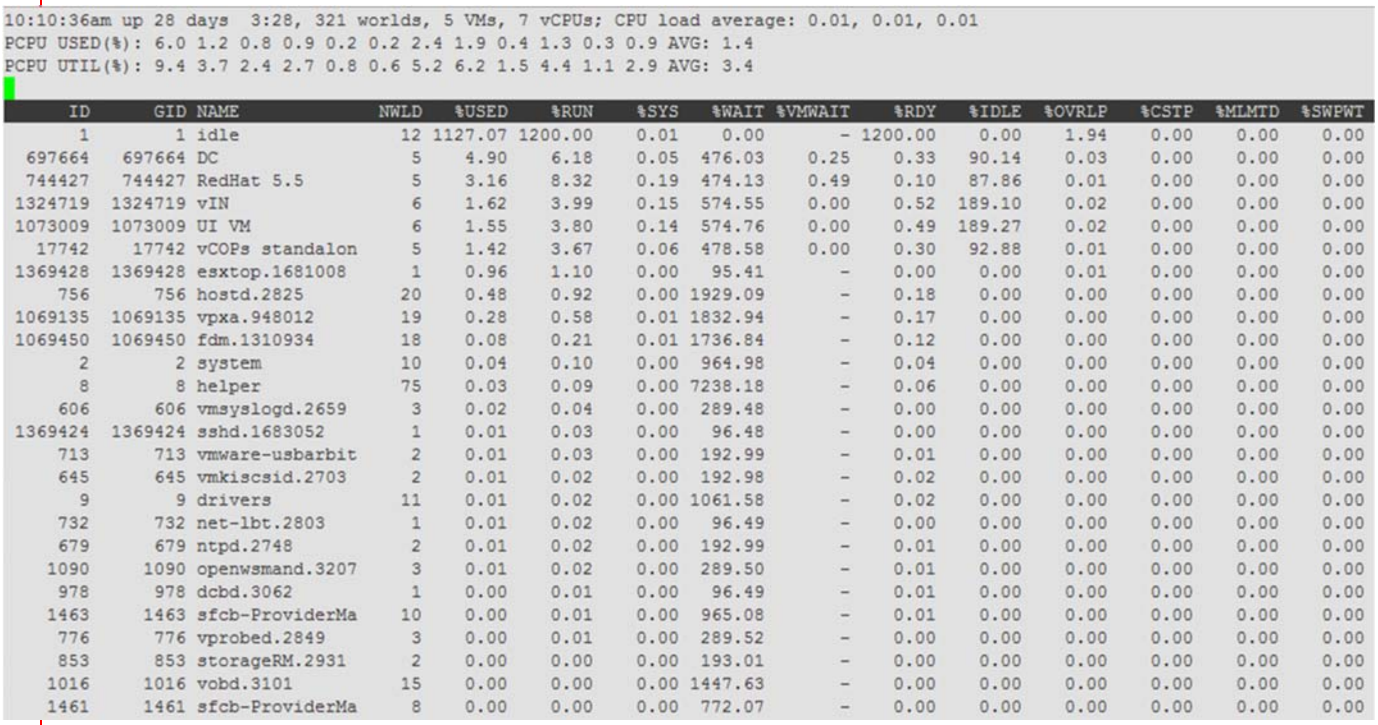
Name	Entity	Descriptions	Unit	Instance/ Aggregate
`cpu.ready.summation`	VM Host	Time that the virtual machine was ready, but could not get scheduled to run on the physical CPU. CPU ready time is dependent on the number of virtual machines on the host and their CPU loads.	millisecond	aggregate
`cpu.usagemhz.average`	VM Host	CPU usage, as measured in megahertz, during the interval: • VM - Amount of actively used virtual CPU. This is the host's view of the CPU usage, not the guest operating system view. • Host - Sum of the actively used CPU of all powered on virtual machines on a host. The maximum possible value is the frequency of the processors multiplied by the number of processors. For example, if you have a host with four 2GHz CPUs running a virtual machine that is using 4000MHz, the host is using two CPUs completely. 4000 / (4 x 2000) = 0.50	megaHertz	aggregate
`cpu.usage.average`	VM Host	CPU usage as a percentage during the interval. • VM - Amount of actively used virtual CPU, as a percentage of total available CPU. This is the host's view of the CPU usage, not the guest operating system view. It is the average CPU utilization over all available virtual CPUs in the virtual machine. For example, if a virtual machine with one virtual CPU is running on a host that has four physical CPUs and the CPU usage is 100%, the virtual machine is using one physical CPU completely. *virtual CPU usage = usagemhz / (# of virtual CPUs x core frequency)* • Host - Actively used CPU of the host, as a percentage of the total available CPU. Active CPU is approximately equal to the ratio of the used CPU to the available CPU. *available CPU = # of physical CPUs x clock rate* 100% represents all CPUs on the host. For example, if a four-CPU host is running a virtual machine with two CPUs, and the usage is 50%, the host is using two CPUs completely. • Cluster - Sum of actively used CPU of all virtual machines in the cluster, as a percentage of the total available CPU.	percent	instance



1. **Storage Performance Counters**

Name	Entity	Descriptions	Unit	Instance/ Aggregate
`disk.totalLatency.average`	Host datastore	Average amount of time taken during the collection interval to process a SCSI command issued by the Guest OS to the virtual machine. The sum of kernelLatency and deviceLatency.	millisecond	instance
`disk.numberReadAveraged.average`	vDisk/VM Host datastore	Average number of read commands issued per second to the datastore during the collection interval.	number	instance
`disk.numberWriteAveraged.average`	vDisk/VM Host datastore	Average number of write commands issued per second to the datastore during the collection interval.	number	instance
`virtualDisk.totalReadLatency.average`	VM	Average amount of time taken during the collection interval to process a SCSI read command issued from the Guest OS to the virtual machine. The sum of kernelReadLatency and deviceReadLatency.	millisecond	instance
`virtualDisk.totalWriteLatency.average`	VM	Average amount of time taken during the collection interval to process a SCSI write command issued by the Guest OS to the virtual machine. The sum of kernelWriteLatency and deviceWriteLatency.	millisecond	instance

CPU – esxtop



```
10:10:36am up 28 days  3:28, 321 worlds, 5 VMs, 7 vCPUs; CPU load average: 0.01, 0.01, 0.01
PCPU USED(%): 6.0 1.2 0.8 0.9 0.2 0.2 2.4 1.9 0.4 1.3 0.3 0.9 AVG: 1.4
PCPU UTIL(%): 9.4 3.7 2.4 2.7 0.8 0.6 5.2 6.2 1.5 4.4 1.1 2.9 AVG: 3.4

       ID       GID NAME             NWLD    %USED     %RUN     %SYS    %WAIT  %VMWAIT     %RDY    %IDLE   %OVRLP    %CSTP   %MLMTD   %SWPWT
        1         1 idle               12  1127.07  1200.00     0.01     0.00        -  1200.00     0.00     1.94     0.00     0.00     0.00
   697664    697664 DC                  5     4.90     6.18     0.05   476.03     0.25     0.33    90.14     0.03     0.00     0.00     0.00
   744427    744427 RedHat 5.5          5     3.16     8.32     0.19   474.13     0.49     0.10    87.86     0.01     0.00     0.00     0.00
  1324719   1324719 vIN                 6     1.62     3.99     0.15   574.55     0.00     0.52   189.10     0.02     0.00     0.00     0.00
  1073009   1073009 UI VM               6     1.55     3.80     0.14   574.76     0.00     0.49   189.27     0.02     0.00     0.00     0.00
    17742     17742 vCOPs standalon     5     1.42     3.67     0.06   478.58     0.00     0.30    92.88     0.01     0.00     0.00     0.00
  1369428   1369428 esxtop.1681008      1     0.96     1.10     0.00    95.41        -     0.00     0.00     0.01     0.00     0.00     0.00
      756       756 hostd.2825         20     0.48     0.92     0.00  1929.09        -     0.18     0.00     0.00     0.00     0.00     0.00
  1069135   1069135 vpxa.948012        19     0.28     0.58     0.01  1832.94        -     0.17     0.00     0.00     0.00     0.00     0.00
  1069450   1069450 fdm.1310934        18     0.08     0.21     0.01  1736.84        -     0.12     0.00     0.00     0.00     0.00     0.00
        2         2 system             10     0.04     0.10     0.00   964.98        -     0.04     0.00     0.00     0.00     0.00     0.00
        8         8 helper             75     0.03     0.09     0.00  7238.18        -     0.06     0.00     0.00     0.00     0.00     0.00
      606       606 vmsyslogd.2659      3     0.02     0.04     0.00   289.48        -     0.00     0.00     0.00     0.00     0.00     0.00
  1369424   1369424 sshd.1683052        1     0.01     0.03     0.00    96.48        -     0.00     0.00     0.00     0.00     0.00     0.00
      713       713 vmware-usbarbit     2     0.01     0.03     0.00   192.99        -     0.01     0.00     0.00     0.00     0.00     0.00
      645       645 vmkiscsid.2703      2     0.01     0.02     0.00   192.98        -     0.02     0.00     0.00     0.00     0.00     0.00
        9         9 drivers            11     0.01     0.02     0.00  1061.58        -     0.02     0.00     0.00     0.00     0.00     0.00
      732       732 net-lbt.2803        1     0.01     0.02     0.00    96.49        -     0.00     0.00     0.00     0.00     0.00     0.00
      679       679 ntpd.2748           2     0.01     0.02     0.00   192.99        -     0.01     0.00     0.00     0.00     0.00     0.00
     1090      1090 openwsmand.3207     3     0.01     0.02     0.00   289.50        -     0.01     0.00     0.00     0.00     0.00     0.00
      978       978 dcbd.3062           1     0.00     0.01     0.00    96.49        -     0.01     0.00     0.00     0.00     0.00     0.00
     1463      1463 sfcb-ProviderMa    10     0.00     0.01     0.00   965.08        -     0.01     0.00     0.00     0.00     0.00     0.00
      776       776 vprobed.2849        3     0.00     0.01     0.00   289.52        -     0.00     0.00     0.00     0.00     0.00     0.00
      853       853 storageRM.2931      2     0.00     0.00     0.00   193.01        -     0.00     0.00     0.00     0.00     0.00     0.00
     1016      1016 vobd.3101          15     0.00     0.00     0.00  1447.63        -     0.00     0.00     0.00     0.00     0.00     0.00
     1461      1461 sfcb-ProviderMa     8     0.00     0.00     0.00   772.07        -     0.00     0.00     0.00     0.00     0.00     0.00
```

Running esxtop

- **Select different view**
  c = cpu
  m = memory
  n = network
  i = interrupts
  d = disk adapter
  u = disk device
  v = disk VM
  p = power states

- **Select fields**
  f: Add/Remove fields
  o: Changing the order
  W: write to config

- **Limiting scope**
  - V = only show VM worlds
  - e = Expand/Rollup CPU statistics, show details of all worlds associated with group (GID)
  - k = kill world
  - l = limit display to a single group (GID)
  - # = limiting the number of entitites

Batch esxtop

- esxtop -b -d 2 -n 100 > esxtop-out.csv

“-b” stands for batch mode, “-d 2″ is a delay of 2s and “-n 100″ are 100 iterations.

- esxtop -b -a -d 2 -n 100 | gzip -9c > esxtop-out.csv.gz
- esxtop -b -d 30 -n 480 -c esxtop-cpu-config > esxtop_cpu_stats.csv
- Display config:

```
abcDeFghij
abcDefgHijKlmnopq
abcdefghijkl
abcdefghijklmnop
abcdefgh
abcdefghijklmno
abcdef
abcde
5m
```

View esxtop output

- esxtop utility outputs to a CSV formatted file
- esxtop utility creates huge amount of data. Some hosts have reported up to 120,000 columns.

```
$cat filename | cut -d "," -f 1,`head -1 filename | tr "," "\12" | egrep -n "regex" | cut -d ":" -f 1 | tr "\12" "," | sed "s/,$//"` | head
```

- Tools to analyze the captured data.
  - VisualEsxtop
  - perfmon
  - excel
  - esxplot

Virtualization Performance I

CPU

CPU – Overview

- Raw processing power of a given host or VM
  - Hosts provide CPU resources
  - VMs and Resource Pools consume CPU resources
- CPU cores/threads need to be shared between VMs
- Fair scheduling vCPU time
  - Hardware interrupts for a VM
  - Parallel processing for SMP VMs
  - I/O

CPU – esxtop

- Interpret the esxtop columns correctly
- %RDY - The percentage of time a VM is ready to run, but no physical processor is ready to run it which may result in decreased performance
- %USED – Physical CPU usage
- %SYS – Percentage of time in the VMkernel
- %RUN – Percentage of total scheduled time to run
- %WAIT – Percentage of time in blocked or busy wait states
- %IDLE – %WAIT- %IDLE can be used to estimate I/O wait time

CPU – Performance Overhead & Utilization

- Different workloads have different overhead costs (%SYS) even for the same utilization (%USED)

- CPU virtualization adds varying amounts of system overhead
  - Direct execution vs. privileged execution
  - Non-paravirtual adapters vs. emulated adaptors
  - Virtual hardware (Interrupts!)
  - Network and storage I/O

CPU – vSMP

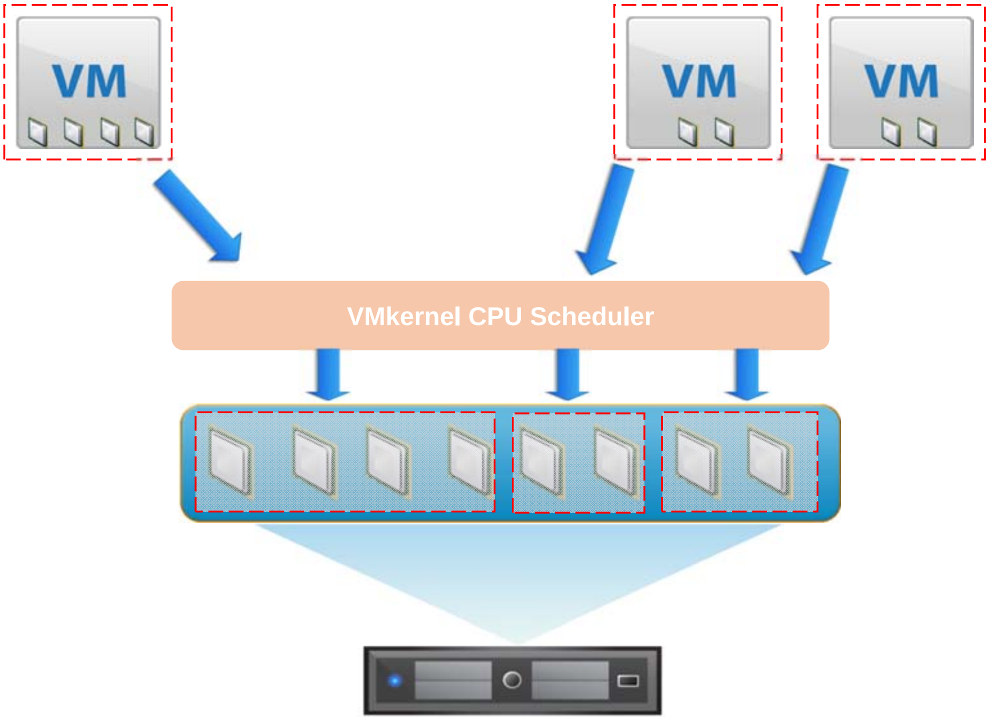
- Relaxed Co-Scheduling: vCPUs can run out-of-sync

- Idle vCPUs incur a scheduling penalty
  - configure only as many vCPUs as needed
  - Imposes unnecessary scheduling constraints

- Use Uniprocessor VMs for single-threaded applications

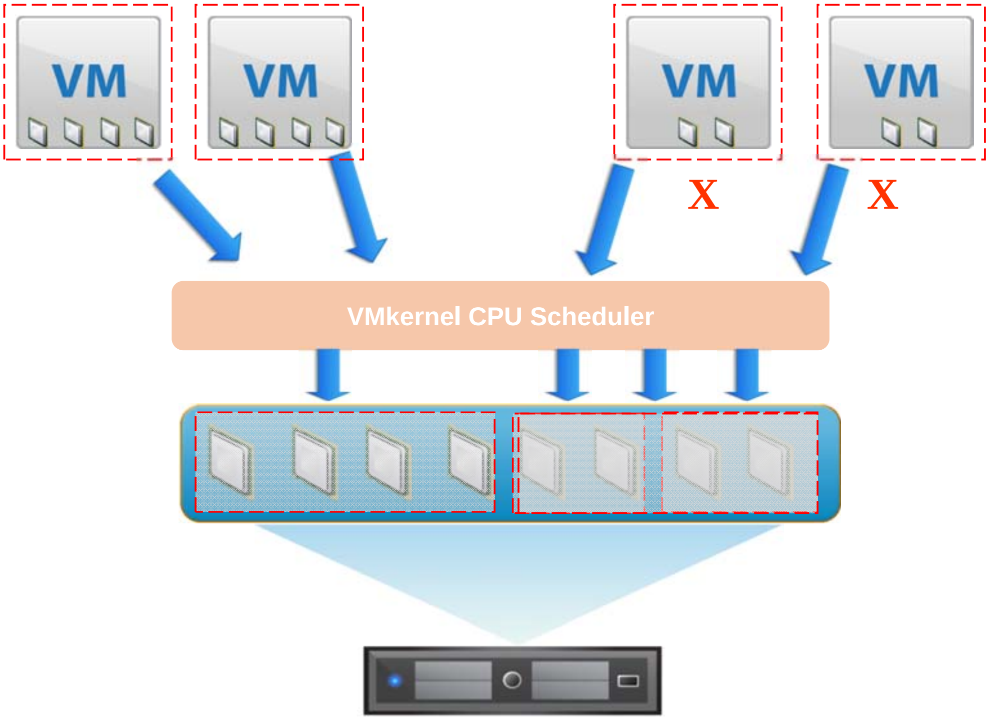
CPU– Scheduling

Over committing physical CPUs



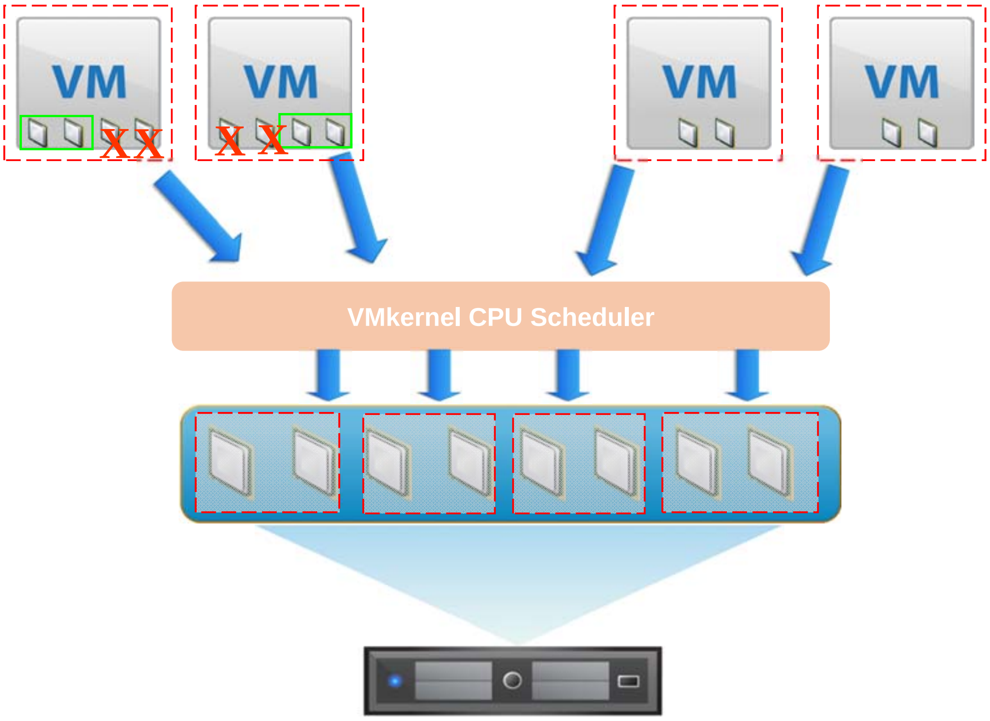
CPU– Scheduling

Over committing physical CPUs



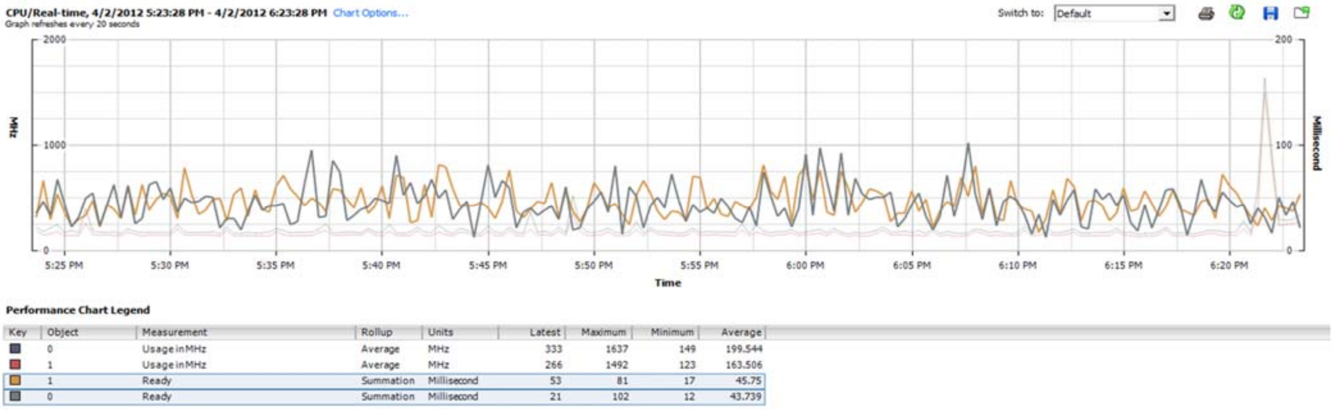
CPU– Scheduling

Over committing physical CPUs



CPU – Ready Time

- The percentage of time that a vCPU is ready to execute, but waiting for physical CPU time
- Does not necessarily indicate a problem
  - Indicates possible CPU contention or limits



CPU/Real-time, 4/2/2012 5:23:28 PM - 4/2/2012 6:23:28 PM Chart Options...

Graph refreshes every 20 seconds

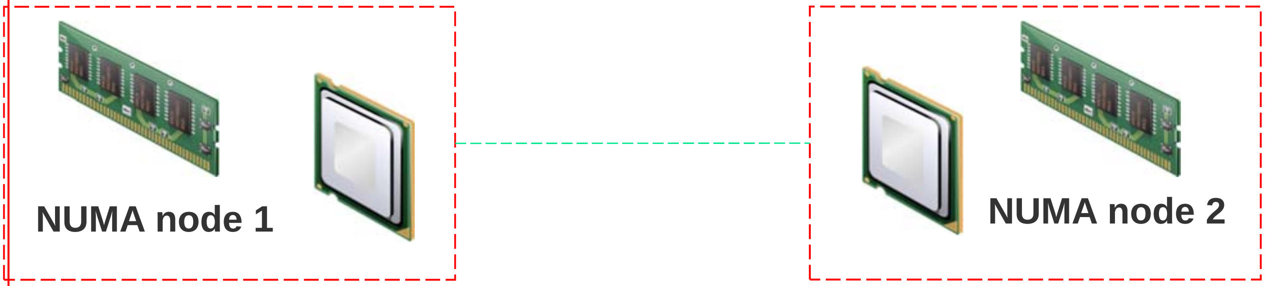
Switch to: Default

Performance Chart Legend

Key	Object	Measurement	Rollup	Units	Latest	Maximum	Minimum	Average
■	0	Usage in MHz	Average	MHz	333	1637	149	199.544
🟥	1	Usage in MHz	Average	MHz	266	1492	123	163.506
🟧	1	Ready	Summation	Millisecond	53	81	17	45.75
■	0	Ready	Summation	Millisecond	21	102	12	43.739

CPU – NUMA nodes

- Non-Uniform Memory Access system architecture
- Each node consists of CPU cores and memory
- A CPU core in one NUMA node can access memory in another node, but at a small performance cost



Demystifying "Ready" time

- Powered on VM could be either running, halted or in a ready state
- Ready time signifies the time spent by a VM on the run queue waiting to be scheduled
- Ready time accrues when more than one world wants to run at the same time on the same CPU
  - PCPU, VCPU over-commitment with CPU intensive workloads
  - Scheduler constraints - CPU affinity settings
- Higher ready time reduces response times or increases job completion time
- Total accrued ready time is not useful
  - VM could have accrued ready time during their runtime without incurring performance loss (for example during boot)
- %ready = ready time accrual rate

CPU – Troubleshooting

- vCPU to pCPU over allocation
  - HyperThreading does not double CPU capacity!
- Limits or too many reservations
  - can create artificial limits.
- Expecting the same consolidation ratios with different workloads
  - Virtualizing “easy” systems first, then expanding to heavier systems
- Compare Apples to Apples
  - Frequency, turbo, cache sizes, cache sharing, core count, instruction set…

Resource Over-Commitment

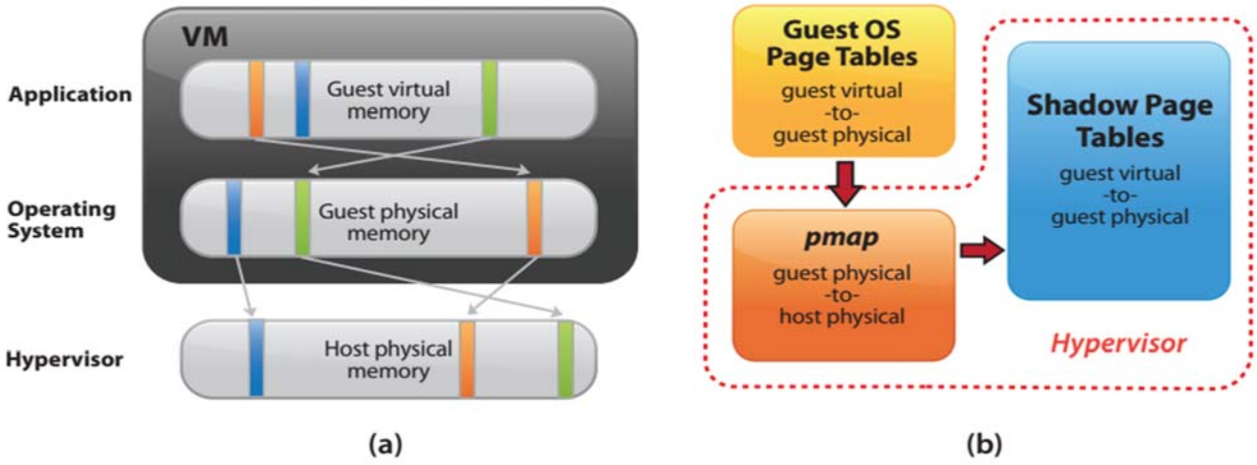
❑ **CPU Over-Commitment**

- Higher CPU utilization does not necessarily mean lesser performance.
  - ❑ Application’s progress is not affected by higher CPU utilization
  - ❑ However if higher CPU utilization is due to monitor overheads then it may impact performance by increasing latency
  - ❑ When there is no headroom (100% CPU), performance degrades
- 100% CPU utilization and %ready are almost identical – both delay application progress
- CPU Over-Commitment could lead to other performance problems
  - ❑ Dropped network packets
  - ❑ Poor I/O throughput
  - ❑ Higher latency, poor response time

Virtualization Performance II

Memory

Memory under virtualization



- The hypervisor adds an extra level of address translation that maps the guest physical address to the host physical address.
- Hypervisor provides the mapping from guest to host physical memory

Memory – Overhead

- A VM's RAM is not necessarily machine RAM
  - vRAM + overhead = maximum machine RAM

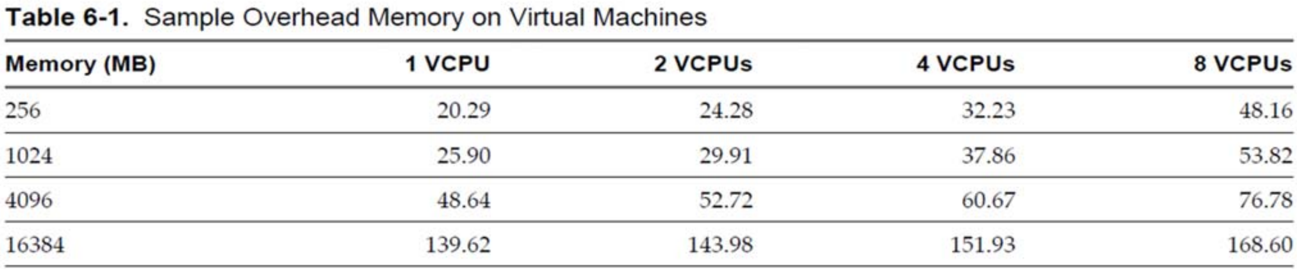


Table 6-1. Sample Overhead Memory on Virtual Machines

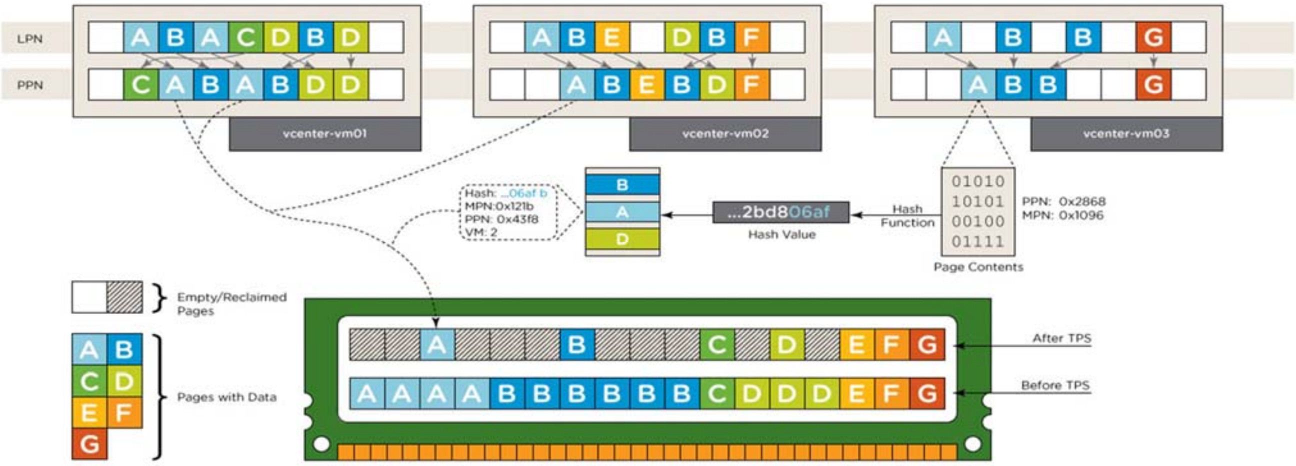
Memory (MB)	1 VCPU	2 VCPUs	4 VCPUs	8 VCPUs
256	20.29	24.28	32.23	48.16
1024	25.90	29.91	37.86	53.82
4096	48.64	52.72	60.67	76.78
16384	139.62	143.98	151.93	168.60

Source: vSphere 5.1 Resource Management Guide

- **Note: These are estimated values**

Memory Management in esxi

- Transparent Page Sharing



- Occurs when memory is under contention
  - **Ballooning**
  - **Compression**
  - **Swapping**

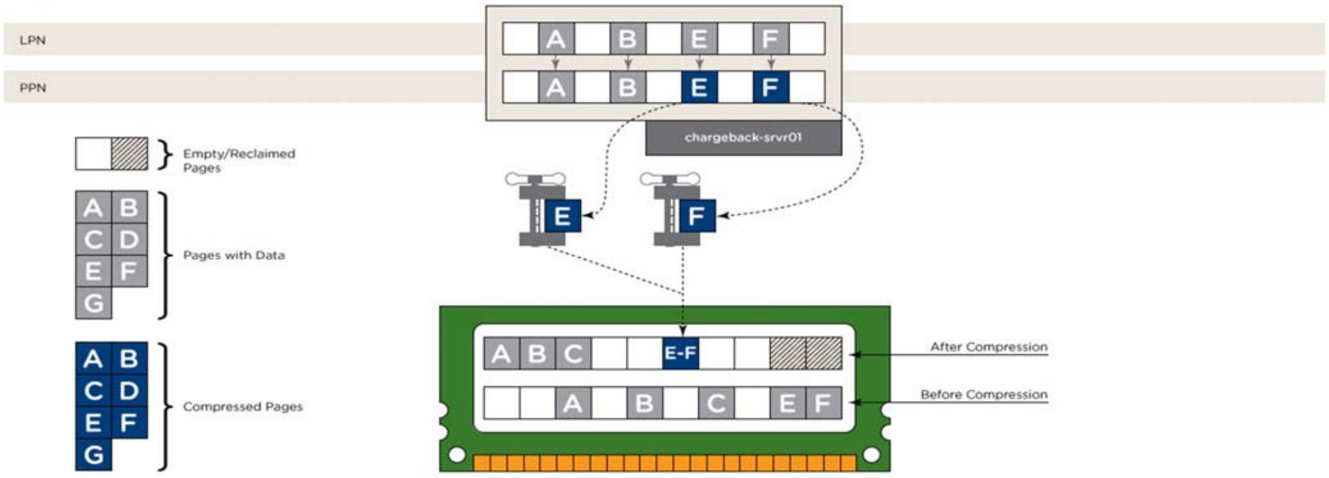
Memory Over-Commitment

- ❑ Guest Swapping - Warning
  - – Guest page faults while swapping.
  - – Performance is affected by both guest swapping and due to monitor overhead handling page faults.
  - – Additional disk I/O
- ❑ Ballooning – Serious
- ❑ VMkernel Swapping - Critical
- ❑ COS Swapping - Critical
  - – VMX process could stall and affect the progress of the VM
  - – VMX could be a victim of random process killed by the kernel
  - – COS requires additional CPU cycles, for handling frequent page faults and disk I/O
- ❑ Memory shares determine the rate of ballooning/swapping

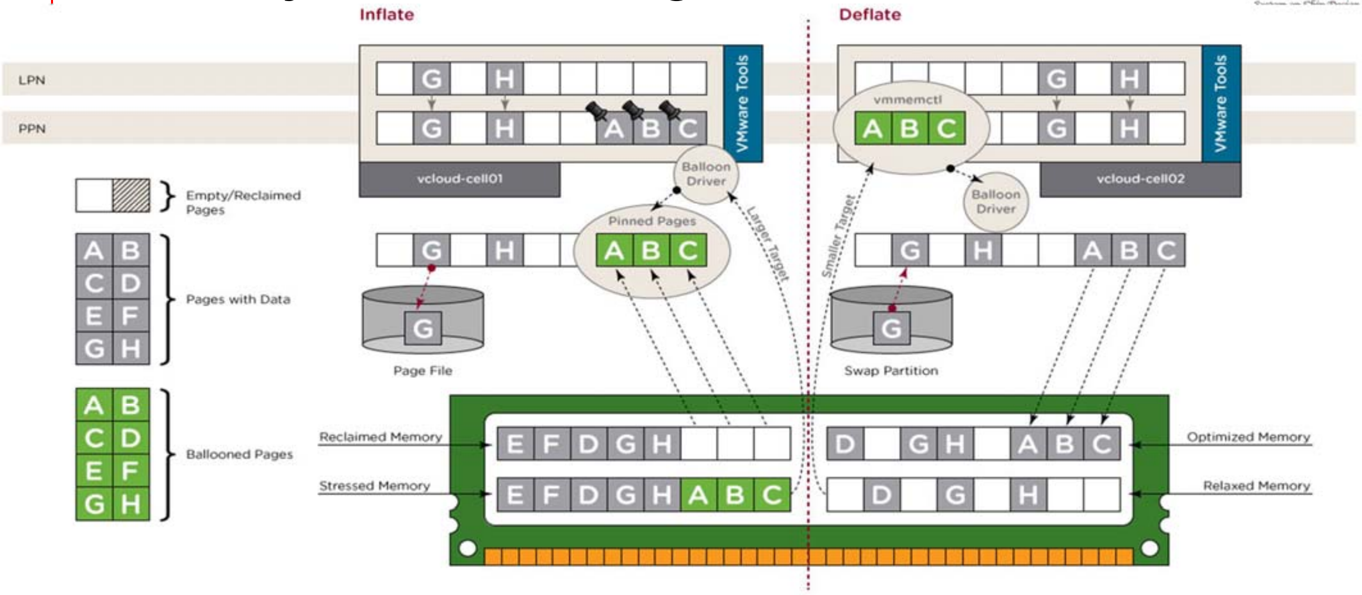
Memory Over-Commitment

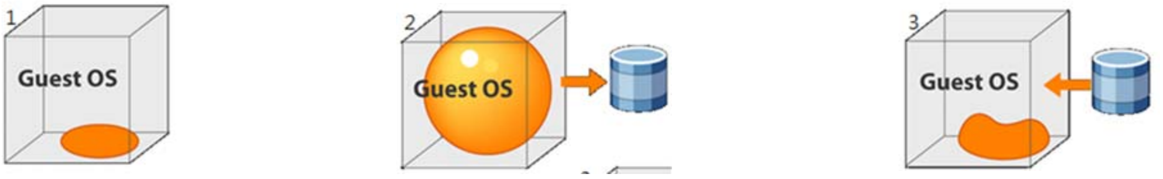
- ❑ Ballooning
  - – Ballooning/swapping stalls processor, increases delay
  - – Windows VMs touches all allocated memory pages during boot. Memory pages touched by the guest could be reclaimed only by ballooning
  - – Linux guest touches memory pages on demand. Ballooning kicks in only when the guest is under complete memory pressure
  - – Ballooning could be avoided by using min=max
  - – /proc/vmware/sched/mem
    - ❑ size <>sizetgt indicates memory pressure
    - ❑ mctl > mctlgt – ballooning out (giving away pages)
    - ❑ mctl < mctlgt – ballooning in (taking in pages)
  - – Memory shares affect ballooning rate

Memory – Compression



Memory – Ballooning

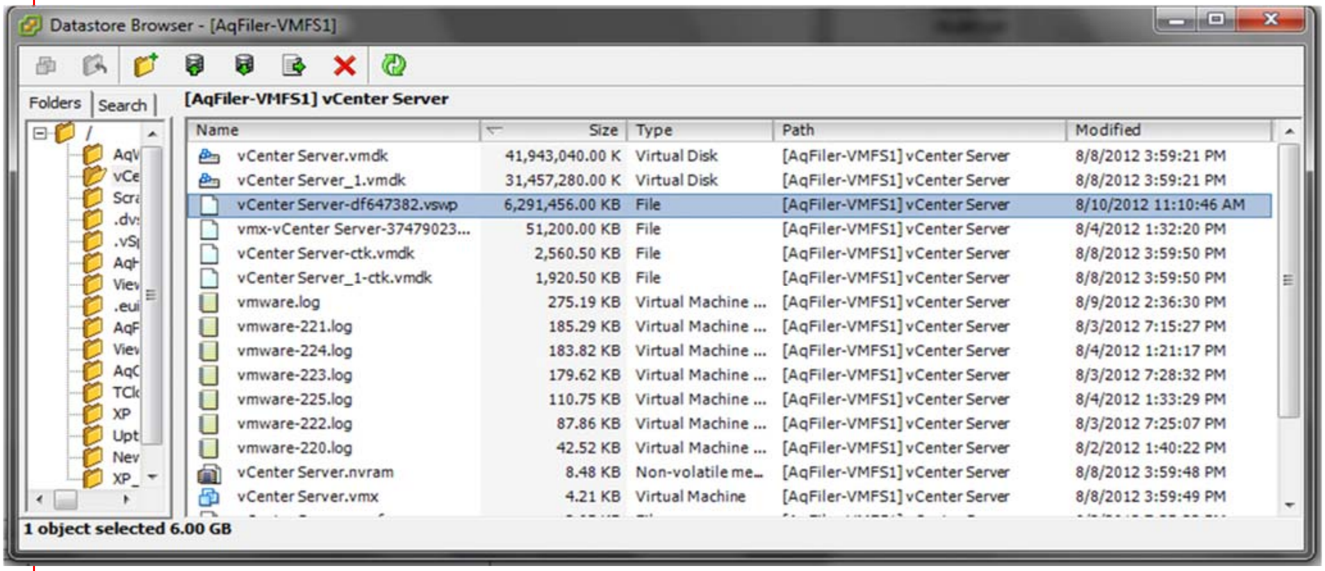




Memory Over-Commitment

- VMKernel Swapping
  - Processor stalls due to VMkernel swapping are more expensive than ballooning (due to disk I/O)
  - Do not confuse this with
    - Swap reservation: Swap is always reserved for worst case scenario if min<> max, reservation = max – min
    - Total swapped pages: Only current swap I/O affects performance
  - /proc/vmware/sched/mem-verbose
    - swpd – total pages swapped
    - swapin, swapout – swap I/O activity
  - SCSI I/O delays during VMKernel I/O swapping could result in system reliability issues

Memory – Swapping

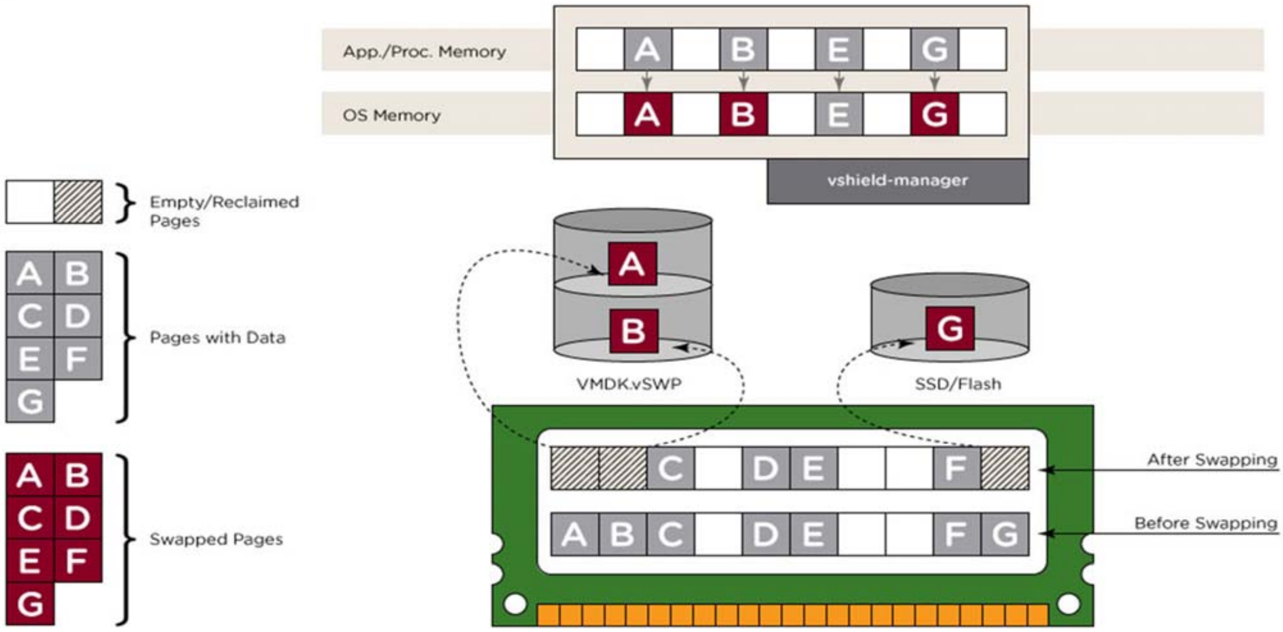


[AqFiler-VMFS1] vCenter Server

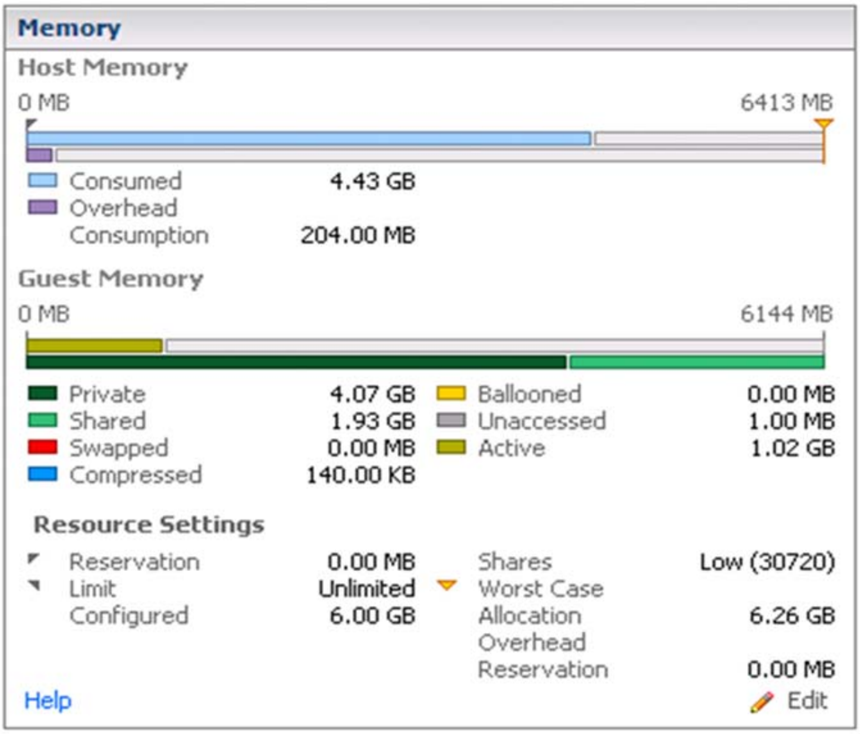
Name	Size	Type	Path	Modified
vCenter Server.vmdk	41,943,040.00 K	Virtual Disk	[AqFiler-VMFS1] vCenter Server	8/8/2012 3:59:21 PM
vCenter Server_1.vmdk	31,457,280.00 K	Virtual Disk	[AqFiler-VMFS1] vCenter Server	8/8/2012 3:59:21 PM
vCenter Server-df647382.vswp	6,291,456.00 KB	File	[AqFiler-VMFS1] vCenter Server	8/10/2012 11:10:46 AM
vmx-vCenter Server-37479023...	51,200.00 KB	File	[AqFiler-VMFS1] vCenter Server	8/4/2012 1:32:20 PM
vCenter Server-ctk.vmdk	2,560.50 KB	File	[AqFiler-VMFS1] vCenter Server	8/8/2012 3:59:50 PM
vCenter Server_1-ctk.vmdk	1,920.50 KB	File	[AqFiler-VMFS1] vCenter Server	8/8/2012 3:59:50 PM
vmware.log	275.19 KB	Virtual Machine ...	[AqFiler-VMFS1] vCenter Server	8/9/2012 2:36:30 PM
vmware-221.log	185.29 KB	Virtual Machine ...	[AqFiler-VMFS1] vCenter Server	8/3/2012 7:15:27 PM
vmware-224.log	183.82 KB	Virtual Machine ...	[AqFiler-VMFS1] vCenter Server	8/4/2012 1:21:17 PM
vmware-223.log	179.62 KB	Virtual Machine ...	[AqFiler-VMFS1] vCenter Server	8/3/2012 7:28:32 PM
vmware-225.log	110.75 KB	Virtual Machine ...	[AqFiler-VMFS1] vCenter Server	8/4/2012 1:33:29 PM
vmware-222.log	87.86 KB	Virtual Machine ...	[AqFiler-VMFS1] vCenter Server	8/3/2012 7:25:07 PM
vmware-220.log	42.52 KB	Virtual Machine ...	[AqFiler-VMFS1] vCenter Server	8/2/2012 1:40:22 PM
vCenter Server.nvram	8.48 KB	Non-volatile me...	[AqFiler-VMFS1] vCenter Server	8/8/2012 3:59:48 PM
vCenter Server.vmx	4.21 KB	Virtual Machine	[AqFiler-VMFS1] vCenter Server	8/8/2012 3:59:49 PM

1 object selected 6.00 GB

Memory – Swapping



Memory – VM Resource Allocation



Memory

Host Memory

0 MB 6413 MB

🟦 Consumed	4.43 GB
🟪 Overhead Consumption	204.00 MB

Guest Memory

0 MB 6144 MB

🟩 Private	4.07 GB	🟨 Ballooned	0.00 MB
🟩 Shared	1.93 GB	⬜ Unaccessed	1.00 MB
🟥 Swapped	0.00 MB	🟨 Active	1.02 GB
🟦 Compressed	140.00 KB		

Resource Settings

Reservation	0.00 MB	Shares	Low (30720)
Limit	Unlimited	Worst Case Allocation	6.26 GB
Configured	6.00 GB	Overhead Reservation	0.00 MB

Help Edit

Memory – Ballooning vs. Swapping

- Ballooning is better than swapping
  - Guest can surrender unused/free pages
  - Guest chooses what to swap, can avoid swapping “hot” pages
- Generally it is better to OVER-commit than UNDER-commit
- If the running VMs are consuming too much host/pool memory…
  - Some VMs may not get physical memory
  - Ballooning or host swapping
  - Higher disk IO
  - All VMs slow down

Memory – Rightsizing

- If a VM has too little vRAM…
  - Applications suffer from lack of RAM
  - The guest OS swaps
  - Increased disk traffic, thrashing
  - SAN slow down as a result of increased disk traffic
- If a VM has too much vRAM…
  - Higher overhead memory
  - Possible decreased failover capacity
  - Longer vMotion time
  - Larger VSWP file
  - Wasted resources

Memory – Troubleshooting and practices

- Wrong resource allocation
  - May not notice a limit, e.g. VM or template with a limit gets cloned

Ballooning or swapping at the host level

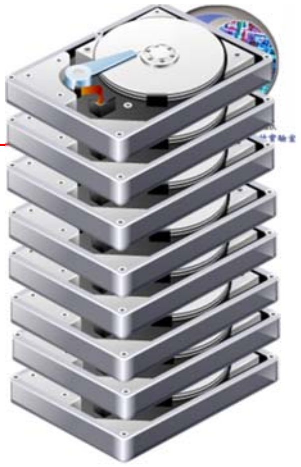
  - Ballooning is a warning sign, not a problem
  - Swapping is a performance issue if seen over an extended period

- Swapping/paging at the guest level
  - Under-provisioned guest memory
- Avoid high active host memory over-commitment
  - No host swapping occurs when total memory demand is less than the physical memory (Assuming no limits)
- Right-size guest memory
  - Avoid guest OS swapping
- Use a fully automated DRS cluster
  - Use Resource Pools with High/Normal/Low shares

Virtualization Performance III

Storage

Hard Disk Performance



- Areas of Concern
  - Disk subsystem bottlenecks
  - Spikes and sustained latency
- Performance versus Capacity
  - Disk performance does not scale with drive size
  - Larger drives generally equate lower performance
- For example: 1 TB of space is required for an app
  - ❑ 2 x 500GB 15K RPM SAS drives = ~300 IOPS
    - Capacity needs satisfied, Performance low
  - ❑ 8 x 146GB 15K RPM SAS drives = ~1,240 IOPS
    - Capacity needs satisfied, Performance high
- More spindles generally equals greater performance

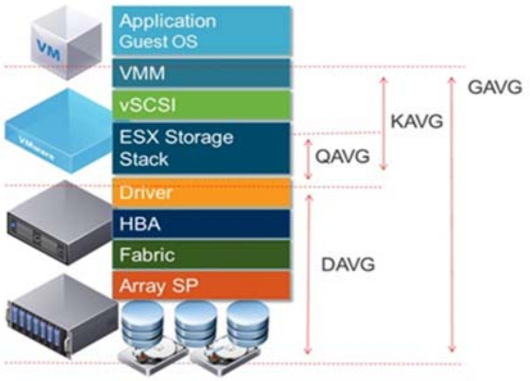
Storage – esxtop Counters

Different esxtop storage views

- Adapter (d)
- VM (v)
- Disk Device (u)

Key Fields:

- DAVG + KAVG = GAVG
- QUED/USD – Command Queue Depth
- CMDS/s – Commands Per Second
- MBREADS/s
- MBWRTN/s



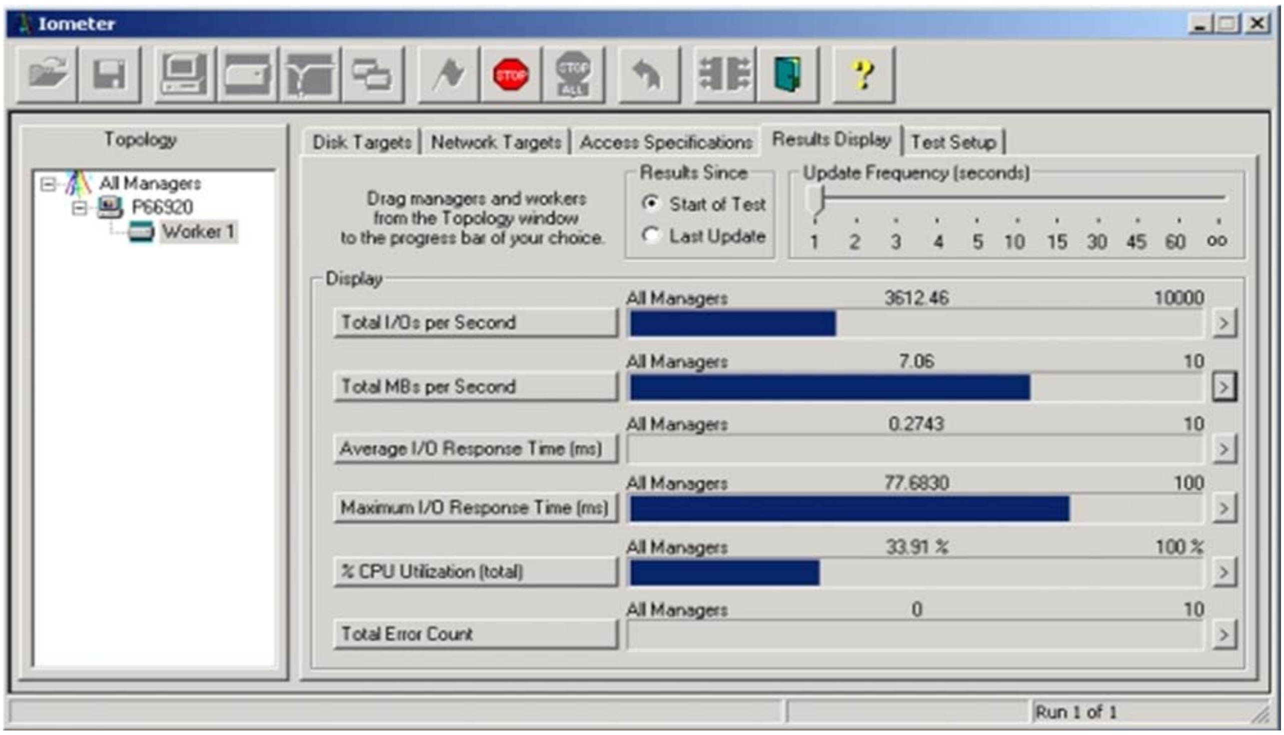
Performance Monitoring – vCenter/vSphere Client

- Performance Charts
  - Performance →Advanced →Chart Options → Datastore/Disk
  - Latency and IOPS statistic counters for disks
  - Historical availability depends on vCenter Statistics level (vCenter Advanced Settings)
  - Statistics available for ESX(i) host objects
  - Datastore
  - ❑ Read/Write Latency = device average in milliseconds
  - Disk
  - ❑ Physical Device Command/Write/Read Latency = device average in milliseconds

Performance Monitoring – Latency Values

- Good latency values are subjective based on application performance needs and storage environment capabilities
- Lower is better!
- General rules of thumb:
  - 10 milliseconds or less is adequate
  - 20 milliseconds or higher, sustained, should be investigated
  - Sustained spikes in latency may need to be investigated
- Commands not acknowledged by the SAN within 5000 ms will be aborted
  - This may lead to performance issues as the commands are retried
  - Multiple aborts should be investigated

Storage – Benchmarking with iometer



Results Since: ● Start of Test ○ Last Update

Update Frequency (seconds): 1 2 3 4 5 10 15 30 45 60 oo

Display	Manager	Value	Scale
Total I/Os per Second	All Managers	3612.46	10000
Total MBs per Second	All Managers	7.06	10
Average I/O Response Time (ms)	All Managers	0.2743	10
Maximum I/O Response Time (ms)	All Managers	77.6830	100
% CPU Utilization (total)	All Managers	33.91 %	100 %
Total Error Count	All Managers	0	10

Run 1 of 1

Storage – Storage I/O Control

- Allows the use of Shares per VMDK
- Throttling occurs when datastore reaches latency threshold
  - Higher share VMDKs perform IO first
- vCenter monitors latency across all hosts
  - Not effective if datastore shared with other vCenters

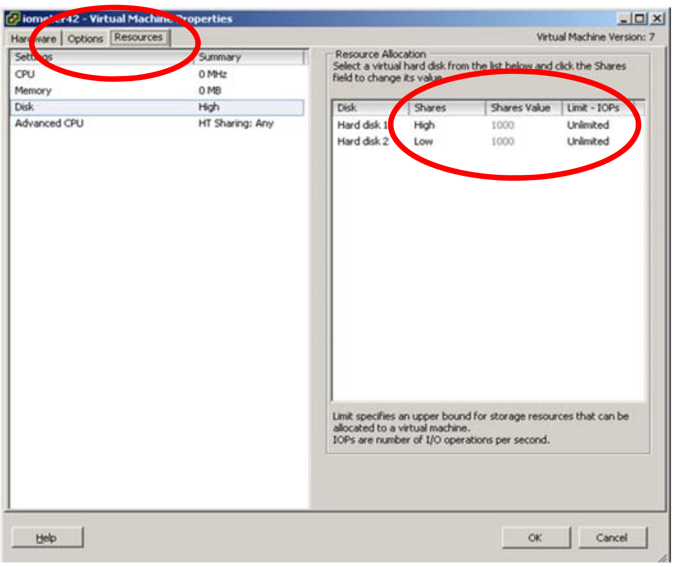
Storage DRS

- Datastore clusters
  - Maintenance mode
  - Anti-affinity rules
- vCenter monitors for latency and disk space
  - Migrate VMDKs for better performance or utilization
- Not effective with automated tiering SANs

Storage I/O Control - Overview

- ❑ Storage I/O Control
  - – Monitors I/O latency to datastores at each ESX host sharing a physical device.
  - – When the average normalized latency exceeds a set threshold (30ms by default), the datastore is considered to be congested.

- **If congested, SIOC distributes available storage resources to virtual machines in proportion to their configured shares.**
- **Used to determine migration needs with Storage DRS in ESXi 5.0**



iometer42 - Virtual Machine Properties

Virtual Machine Version: 7

Settings	Summary
CPU	0 MHz
Memory	0 MB
Disk	High
Advanced CPU	HT Sharing: Any

Resource Allocation

Select a virtual hard disk from the list below and click the Shares field to change its value.

Disk	Shares	Shares Value	Limit - IOPs
Hard disk 1	High	1000	Unlimited
Hard disk 2	Low	1000	Unlimited

Limit specifies an upper bound for storage resources that can be allocated to a virtual machine.
IOPs are number of I/O operations per second.

Storage I/O Control – Usage Scenario

- ❑ Without SIOC, Disk shares do not provide a fair distributed

Figure 3a. Without Storage I/O Control
Actual Disk Resources utilized by each VM are not in the correct ratio

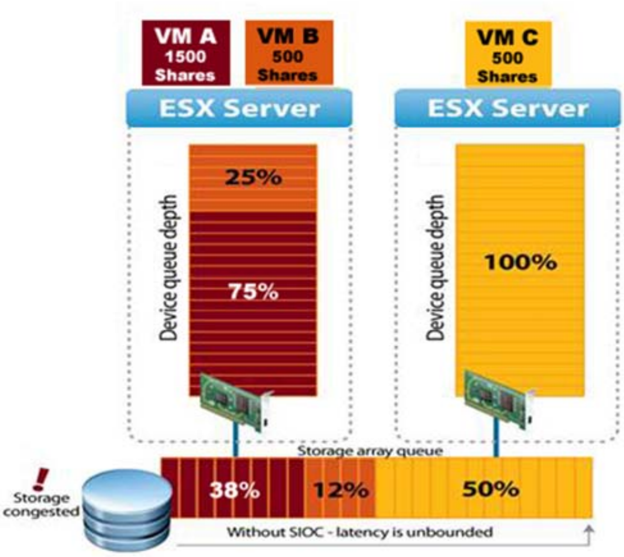
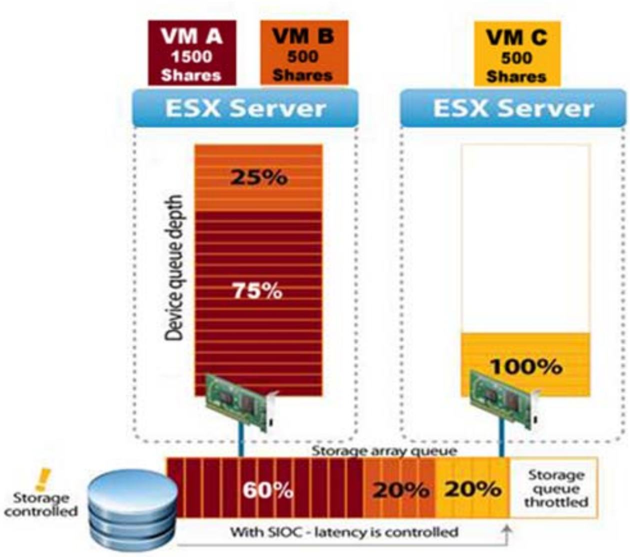


Figure 3b. With Storage I/O Control
Actual Disk Resources utilized by each VM are in the correct ratio even across ESX Hosts



Network – Load Balancing

Load balancing defines which uplink is used

- Route based on Port ID
- Route based on IP hash
- Route based on MAC hash
- Route based on NIC load (Load Based Teaming)

Probability of high-bandwidth VMs being on the same physical NIC

Traffic will stay on elected uplink until an event occurs

- NIC link state change, adding/removing NIC from a team, beacon probe timeout…

How to check network performance?

VM – VM on same ESXi host. This will exclude physical network problems

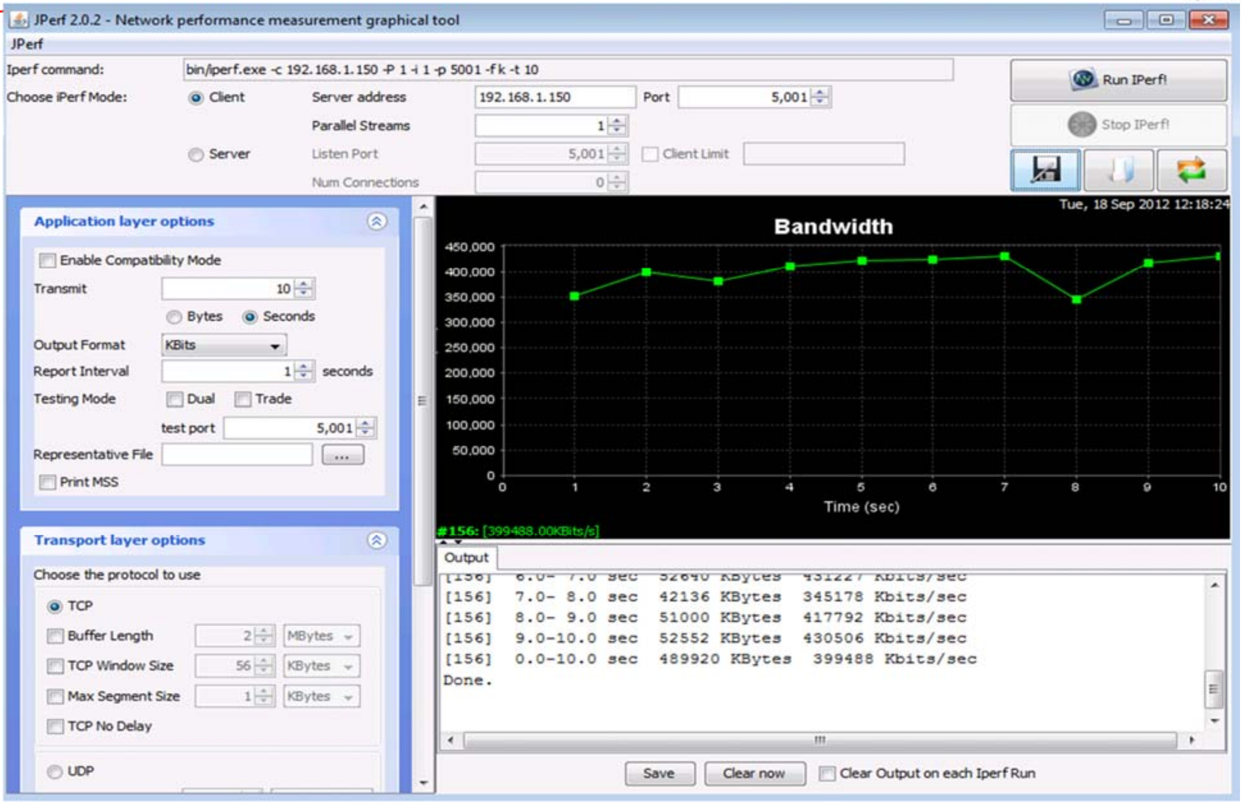
VM –VM on different ESXi host. This will involve physical NICs and switch as well

Physical – VM. Will also test physical devices but we can focus on one VM

Physical – Physical: this will give us some number about what to expect

Use iperf/jperf/netperf. Free tool for network test

Iperf



JPerf 2.0.2 - Network performance measurement graphical tool

JPerf

Iperf command: bin/iperf.exe -c 192.168.1.150 -P 1 -i 1 -p 5001 -f k -t 10

Choose iPerf Mode: ● Client Server address 192.168.1.150 Port 5,001

Parallel Streams 1

○ Server Listen Port 5,001 □ Client Limit

Num Connections 0

Application layer options

□ Enable Compatibility Mode

Transmit 10 ○ Bytes ● Seconds

Output Format KBits

Report Interval 1 seconds

Testing Mode □ Dual □ Trade

test port 5,001

Representative File

□ Print MSS

Transport layer options

Choose the protocol to use

● TCP

□ Buffer Length 2 MBytes

□ TCP Window Size 56 KBytes

□ Max Segment Size 1 KBytes

□ TCP No Delay

○ UDP

Output

```
[156]  6.0- 7.0 sec  52640 KBytes  431227 Kbits/sec
[156]  7.0- 8.0 sec  42136 KBytes  345178 Kbits/sec
[156]  8.0- 9.0 sec  51000 KBytes  417792 Kbits/sec
[156]  9.0-10.0 sec  52552 KBytes  430506 Kbits/sec
[156]  0.0-10.0 sec  489920 KBytes  399488 Kbits/sec
Done.
```

□ Clear Output on each Iperf Run