Virtualization Performance Analysis:

Has the Game Changed?

John Paul

Cerner Heath Services, Managed Services Cloud R&D

April 7, 2015

Key Points to be Discussed

- Storage solutions have made things both better and worse for the end user and IT staff
- Storage vendors still focus on the overall capacity and frame and not on the individual paths
- Storage solutions that help production may not help development or load testing
- Storage and virtualization tools are really split into trending and element management
- Our virtualization goal is to move the roadblock from storage back to the compute node
  - We can return to our standard virtualization performance analysis techniques
  - We can better plan for an upgrade at the compute node level
- It is critical to know how an anomaly in the workload affects the performance for an end user
  - How does a surge in IOPS affect performance?
  - Which points in the storage path are affected?
  - Do we have the tools to see the queueing in those points?
  - How we set queue target depths in each point along the storage path?
- How closely does development configuration need to look like production?
- How closely does your load testing match your production loads?

Trends to Consider

- Anything, Anywhere, Anytime
  - The social adoption of mobile computing with smart phones, tablets, watches, glasses has taken virtualization to a whole new level
  - Mobile computing has made it more necessary to understand the individual parts of the solution
  - Mobile computing has weakened the stronghold of Intel which could splinter some of the de facto standards
  - Consumers and not business are driving digitalization
- Hybrid Cloud – Abstraction – This is the key to implementation of a hybrid cloud strategy so workloads can be moved between and inside clouds:
  - Compute Node
  - Networking
  - Storage
  - Infrastructure (ex., Active Directory)

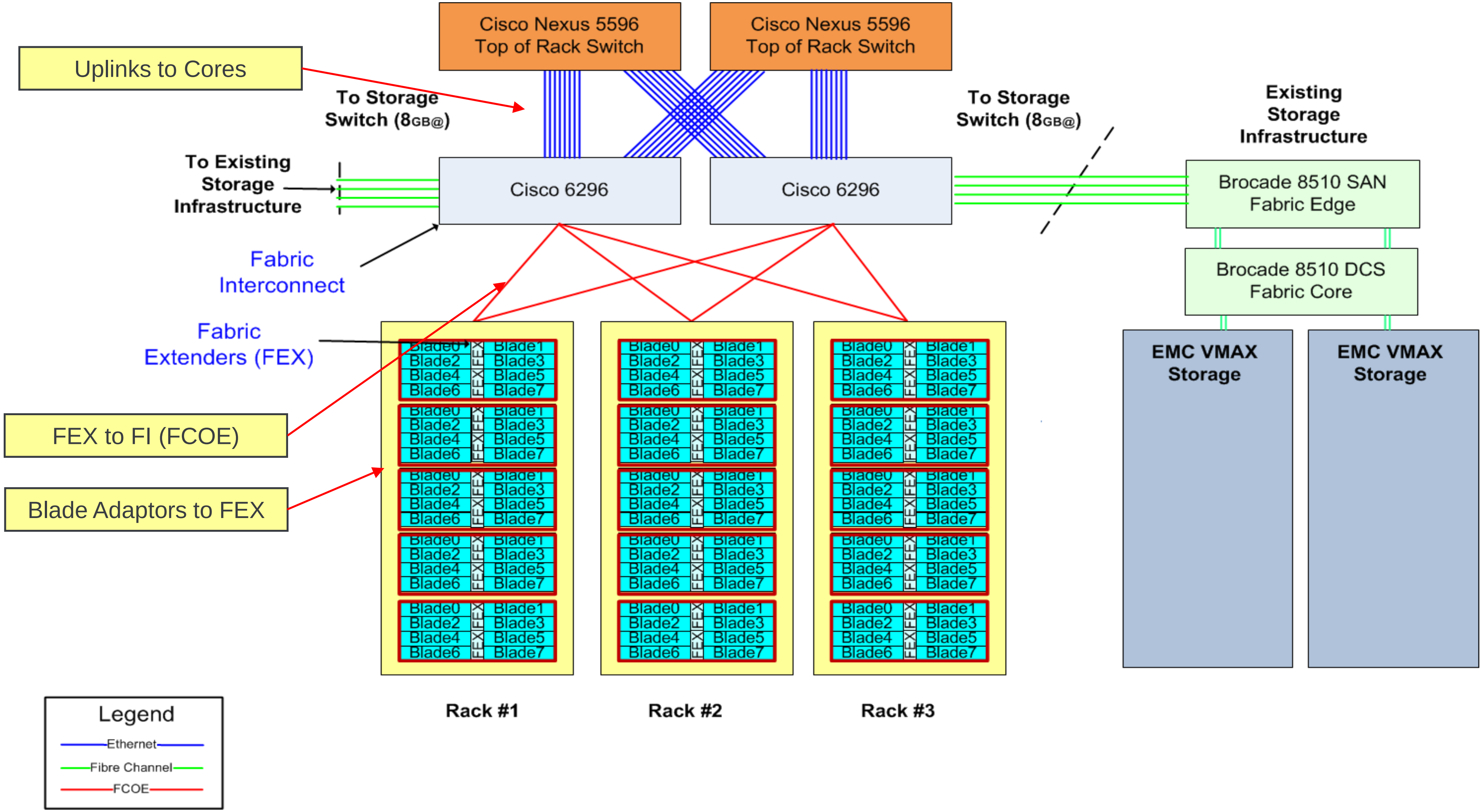
Trends to Consider

- Intel Strategies – Intel necessarily had to move to a horizontal, multi-core strategy due to the physical restrictions of what could be done on the current technology base. This resulted in:
  - Increasing number of cores per socket
  - Stabilization of processor speed (i.e., processor speeds no longer are increasing according to Moore’s Law)
  - Focus on new architectures that allow for more efficient movement of data between memory and the processors, external sources and the processors, and larger and faster caches associated with different components
- Storage Strategies – The speed (RPM) of the individual disks are becoming less important while the blending of multiple type of storage into tiers is more of the norm. Moving high speed storage caching close/on to the server is more of an option but not a panacea

Trends to Consider: Changing Compute Density and Architecture

- Low density computing
  - Standard rack mount servers
- Middle density computing
  - Chassis centric blade servers
- High density computing
  - Domain centric blade servers
- Local SSD Disks/Cards
  - Movement of I/O from remote storage to server resident
- The performance roadblock has moved from the compute node to storage (for now)

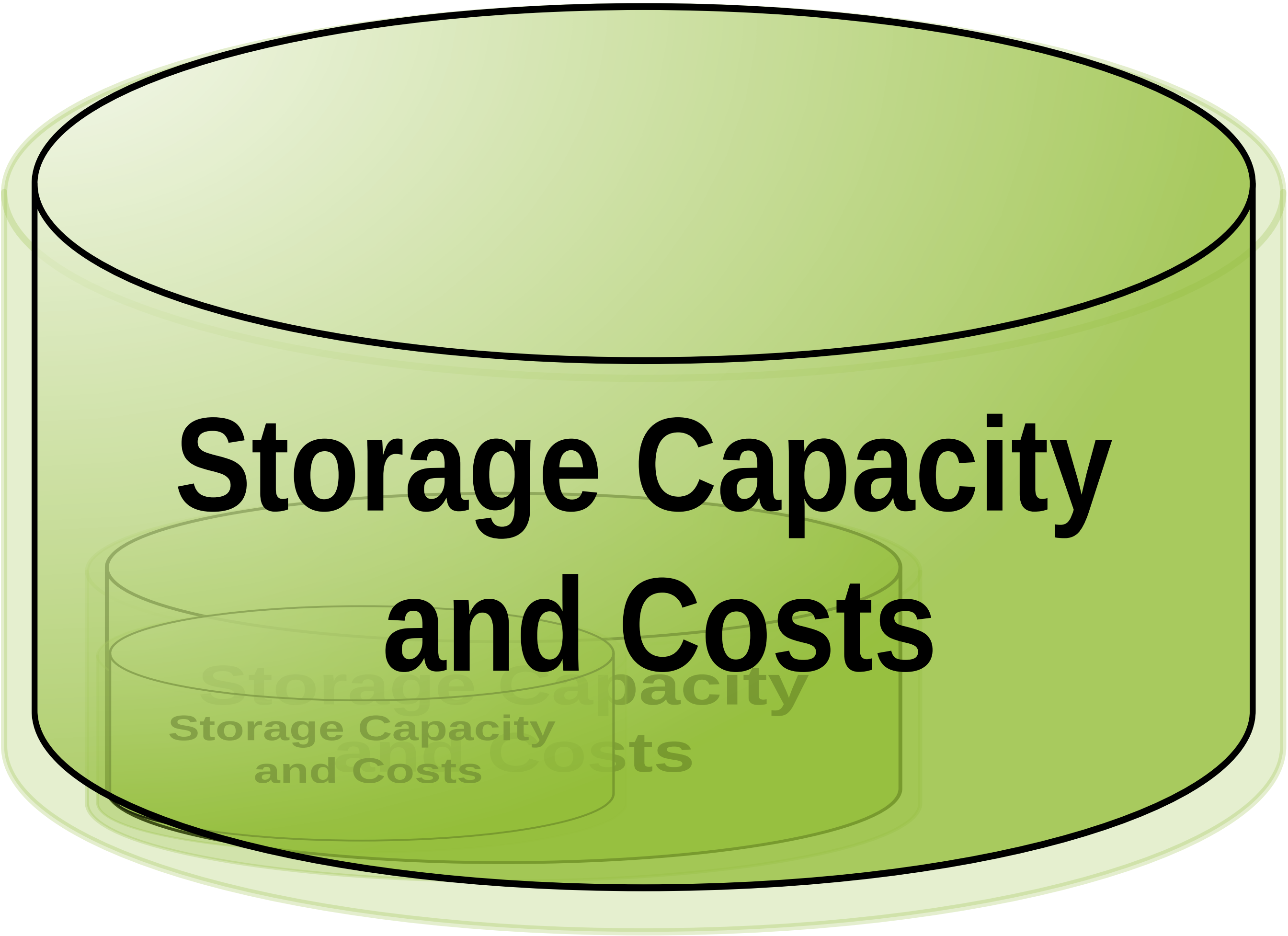
High Density Computing (ex., Cisco UCS)


Cisco Nexus 5596 Top of Rack Switch
Cisco Nexus 5596 Top of Rack Switch
Uplinks to Cores
To Storage Switch (8GB@)
To Storage Switch (8GB@)
Existing Storage Infrastructure
To Existing Storage Infrastructure
Cisco 6296
Cisco 6296
Brocade 8510 SAN Fabric Edge
Brocade 8510 DCS Fabric Core
Fabric Interconnect
Fabric Extenders (FEX)
FEX to FI (FCOE)
Blade Adaptors to FEX
EMC VMAX Storage
EMC VMAX Storage
Blade0 FEX Blade1 Blade2 Blade3 Blade4 FEX Blade5 Blade6 Blade7
Blade0 FEX Blade1 Blade2 Blade3 Blade4 FEX Blade5 Blade6 Blade7
Blade0 FEX Blade1 Blade2 Blade3 Blade4 FEX Blade5 Blade6 Blade7
Blade0 FEX Blade1 Blade2 Blade3 Blade4 FEX Blade5 Blade6 Blade7
Blade0 FEX Blade1 Blade2 Blade3 Blade4 FEX Blade5 Blade6 Blade7
Blade0 FEX Blade1 Blade2 Blade3 Blade4 FEX Blade5 Blade6 Blade7
Blade0 FEX Blade1 Blade2 Blade3 Blade4 FEX Blade5 Blade6 Blade7
Blade0 FEX Blade1 Blade2 Blade3 Blade4 FEX Blade5 Blade6 Blade7
Blade0 FEX Blade1 Blade2 Blade3 Blade4 FEX Blade5 Blade6 Blade7
Blade0 FEX Blade1 Blade2 Blade3 Blade4 FEX Blade5 Blade6 Blade7
Blade0 FEX Blade1 Blade2 Blade3 Blade4 FEX Blade5 Blade6 Blade7
Blade0 FEX Blade1 Blade2 Blade3 Blade4 FEX Blade5 Blade6 Blade7
Blade0 FEX Blade1 Blade2 Blade3 Blade4 FEX Blade5 Blade6 Blade7
Blade0 FEX Blade1 Blade2 Blade3 Blade4 FEX Blade5 Blade6 Blade7
Blade0 FEX Blade1 Blade2 Blade3 Blade4 FEX Blade5 Blade6 Blade7
Rack #1
Rack #2
Rack #3
Legend
Ethernet
Fibre Channel
FCOE

Trends to Consider: Frame Based Storage Movement – Lots of Options

- Data progression (Dell/Compellent)
- Fully Automated Storage Tiering (EMC)
- Adaptive Automation (HP/3PAR)
- Easy Tier (IBM)
- Dynamic Tiering (Hitachi Data Systems)
- Flash Cache (Netapp - Pka Performance Application Module)

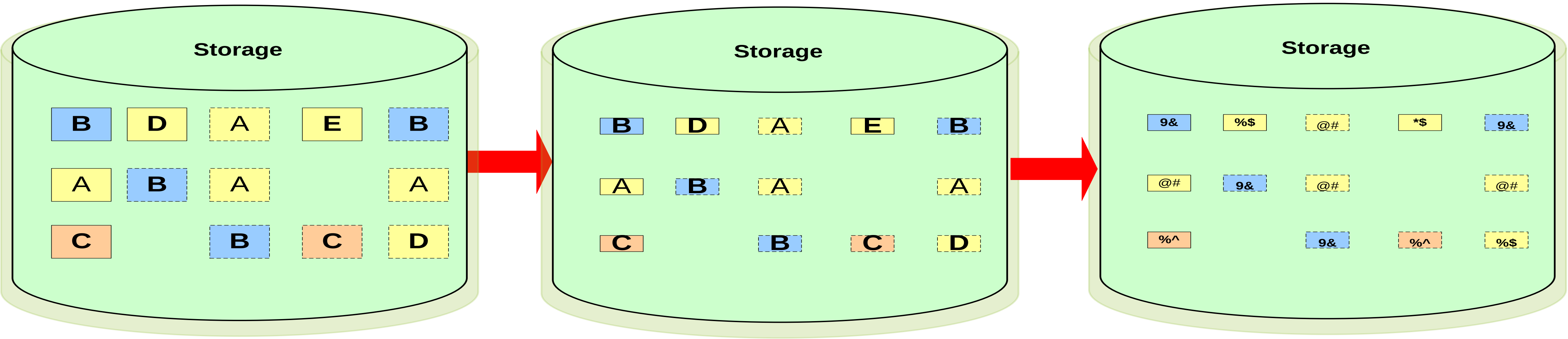
Trends to Consider: Storage Growth


Storage Capacity and Costs
Storage Capacity and Costs
Storage Capacity and Costs

- Duplicate data
- Some data no longer referenced/accessed
- Much longer data retention requirements
- Multiple copies of data in different locations for DR needs
- Continuous availability and speed of access needed
- Tiered storage pricing

Trends to Consider: Compression, Encryption, De-Duplication

- Compression, encryption, de-duplication, and replication can all coexist, you just need to do it in the right order.
- You first de-duplicate, then you compress, then you encrypt, and last of all, you replicate.



Deduped

Compressed

Encrypted

Storage

Storage – Key Question and Considerations

Is the bandwidth and configuration of the storage subsystem sufficient to meet the desired latency (a.k.a. response time) for the target workloads? If the latency target is not being met then further analysis may be very time consuming.

Storage Frames specifications refer to the aggregate bandwidth of the frame or components, not the single path capacity of those components.

- Queuing - Queuing can happen at any point along the storage path, but is not necessarily a bad thing if the latency meets requirements.
- Storage Path Configuration and Capacity – It is critical to know the configuration of the storage path and the capacity of each component along that path. The number of active vmkernel commands must be less then or equal to the queue depth max of any of the storage path components while processing the target storage workload.

Storage Trends That Affect Performance

- RPM of disk drives less and less important for overall performance
- Blending of storage devices (SDD, SAS, SATA) more the norm
- Storage devices automatic migration of data to avoid/correct hot spots
- Running the same simulation twice in a row could have substantially different results
- Queuing in several different points of the infrastructure

Storage – Aggregate versus Single Paths

Storage Frames specifications refer to the aggregate bandwidth of the frame or components, not the single path capacity of those components*

- DMX Message Bandwidth: 4-6.4 GB/s
- DMX Data Bandwidth: 32-128 GB/s
- Global Memory: 32-512 GB
- Concurrent Memory Transfers: 16-32 (4 per Global Memory Director)

Performance Measurement for storage is all about individual paths and the performance of the components contained in that path

(* Source – EMC Symmetrix DMX-4 Specification Sheet c1166-dmx4-ss.pdf)

Storage – More Questions

- Virtual Machines per LUN - The number of outstanding active vmkernel commands per virtual machine times the number of virtual machines on a specific LUN must be less then the queue depth of that adapter
- How fast can the individual disk drive process a request?
- Based upon the block-size and type of I/O (sequential read, sequential write, random read, random write) what type of configuration (RAID, number of physical spindles, cache) is required to match the I/O characteristics and workload demands for average and peak throughput?
- Does the network storage (SAN frame) handle the I/O rate down each path and aggregated across the internal bus, frame adaptors, and front end processors?

In order to answer these questions we need to better understand the underlying design, considerations, and basics of the storage subsystem

Back-end Storage Design Considerations

Capacity - What is the storage capacity needed for this workload/cluster?

- Disk drive size (ex., 144GB, 300GB)
- Number of disk drives needed within a single logical unit (ex., LUN)

IOPS Rate – How many I/Os per second are required with the needed latency?

- Number of physical spindles per LUN (traditional)
- Impact of sharing of physical disk drives between LUNs
- Configuration (ex., cache) and speed of the disk drive

Availability – How many disk drives, storage components can fail at one time?

- Type of RAID chosen (traditional), number of parity drives per grouping
- Amount of redundancy built into the storage solution

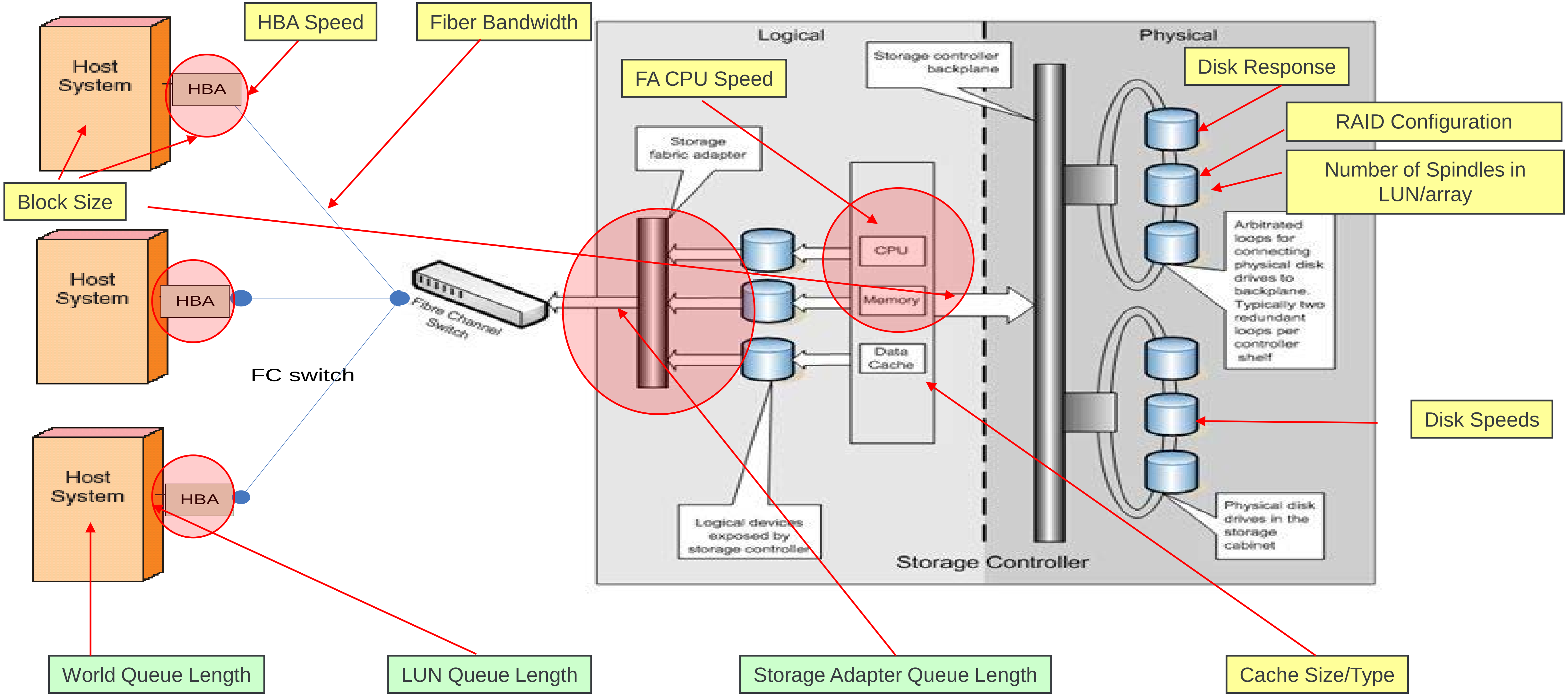
Cost – Delivered cost per byte at the required speed and availability

- Many options are available for each design consideration
- Final decisions on the choice for each component
- The cumulative amount of capacity, IOPS rate, and availability often dictate the overall solution

Storage: Components That Can Affect Performance/Availability

- Size and use of cache (i.e., % dedicated to reads versus writes)
- Number of independent internal data paths and buses
- Number of front-end interfaces and processors
- Types of interfaces supported (ex. Fiber channel and iSCSI)
- Number and type of physical disk drives available
- MetaLUN Expansion
  - MetaLUNs allow for the aggregation of LUNs
  - System typically re-stripes data when MetaLUN is changed
  - Some performance degradation during re-striping
- Storage Virtualization
  - Aggregation of storage arrays behind a presented mount point/LUN
  - Movements between disk drives and tiers control by storage management
  - Change of physical drives and configuration may be transient and severe

SAN Storage Infrastructure – Areas to Watch/Consider



HBA Speed

Fiber Bandwidth

FA CPU Speed

Disk Response

RAID Configuration

Number of Spindles in LUN/array

Block Size

Disk Speeds

World Queue Length

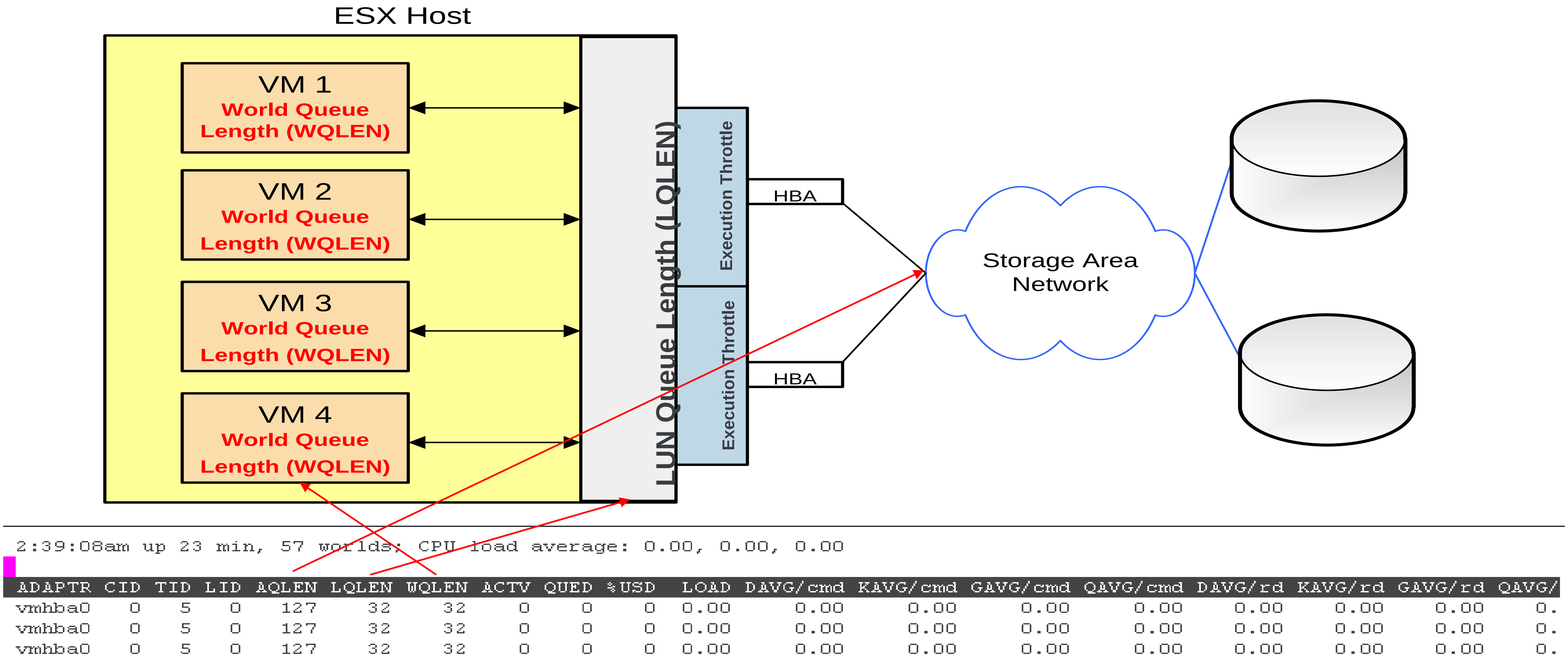
LUN Queue Length

Storage Adapter Queue Length

Cache Size/Type

Host System
HBA
Host System
HBA
Host System
HBA
FC switch
Fibre Channel Switch
Logical
Physical
Storage controller backplane
Storage fabric adapter
CPU
Memory
Data Cache
Arbitrated loops for connecting physical disk drives to backplane. Typically two redundant loops per controller shelf
Physical disk drives in the storage cabinet
Logical devices exposed by storage controller
Storage Controller

Storage Queuing – The Key Throttle Points with ESX Virtualization



```
 2:39:08am up 23 min, 57 worlds; CPU load average: 0.00, 0.00, 0.00

 ADAPTR CID TID LID AQLEN LQLEN WQLEN ACTV QUED %USD  LOAD DAVG/cmd KAVG/cmd GAVG/cmd QAVG/cmd DAVG/rd KAVG/rd GAVG/rd QAVG/
 vmhba0   0   5   0   127    32    32    0    0    0  0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00    0.
 vmhba0   0   5   0   127    32    32    0    0    0  0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00    0.
 vmhba0   0   5   0   127    32    32    0    0    0  0.00     0.00     0.00     0.00     0.00    0.00    0.00    0.00    0.
```

Storage I/O – The Key Throttle Point Definitions for ESX Virtualization

- Storage Adapter Queue Length (AQLEN)
  - The number of outstanding vmkernel active commands that the adapter is configured to support.
  - This is not settable. It is a parameter passed from the adapter to the kernel.
- LUN Queue Length (LQLEN)
  - The maximum number of permitted outstanding vmkernel active commands to a LUN. (This would be the HBA queue depth setting for an HBA.)
  - This is set in the storage adapter configuration via the command line.
- World Queue Length (WQLEN) VMware Recommends Not to Change This!!!!
  - The maximum number of permitted outstanding vmkernel active requests to a LUN from any singular virtual machine (min:1, max:256: default: 32)
  - Configuration->Advanced Settings->Disk-> Disk.SchedNumReqOutstanding
- Execution Throttle (this is not a displayed counter)
  - The maximum number of permitted outstanding vmkernel active commands that can be executed on any **one** HBA port (min:1, max:256: default: ~16, depending on vendor)
  - This is set in the HBA driver configuration.

Storage: Queue Length Rules of Thumb

- For a lightly-loaded system, average queue length should be less than 1 per spindle with occasional spikes up to 10. If the workload is write-heavy, the average queue length above a mirrored controller should be less than 0.6 per spindle and less than 0.3 per spindle above a RAID-5 controller.
- For a heavily-loaded system that isn’t saturated, average queue length should be less than 2.5 per spindle with infrequent spikes up to 20. If the workload is write-heavy, the average queue length above a mirrored controller should be less than 1.5 per spindle and less than 1 above a RAID-5 controller.

Performance Analysis Basics

Three Different Areas for Virtualization Performance Analysis

- Production environments – general performance trends
  - Sampling rates are typically in the 5-60 minute range
  - Tools monitor the end to end path, with single pane of glass a possibility
    - Collects data from many element managers
  - Exception reporting outside of established thresholds available
- Production environments – specific performance problem
  - Move from the general to the specific
- Load and regression testing
  - Sampling rates are typically in the 3-30 second range
  - Focus may be on a single transaction with a duration of 1-30 seconds
  - Has the software performance degraded across fixes/releases?
  - Element managers usually needed to identify root cause
  - Consistency of tests require consistency of infrastructure

Terms Defined

- Load Testing – The intent of this testing is to simulate the expected concurrent average and peak user loads to determine the impact to the processing resources and capacity.
- Stress Testing – The intent of this testing is to stress targeted levels of the application or infrastructure to determine how the performance/function degrades as it enters the stress level.
- Performance Testing - The intent of this testing is to simulate the expected concurrent average and peak user loads to measure the end user or background task response time.
- Scalability Testing – The intent of this testing is to simulate the expected concurrent average and peak user loads against a vertically and horizontally scaled deployment infrastructure to measure the linearity of scaling.

Testing Methodologies and Strategies

Methodologies

- Random – An example of this is pulling data from different users in the database to ensure randomness across the database tables and security system.
- Cumulative – This approach builds scripts that can be run individually or together to help isolate problems. Typically read-only processing is done first, followed by update scripts (which cause the test to have to be fully reset).
- Repeatable – The environment should be reset so the same tests should produce the same result unless there is an environmental issue

Strategies

- Expect to run iterative tests
- A simple tool can be very effective in identifying initial roadblocks
- Monitoring should not overly impact the actual tests
- Analyze details after the test runs, analyze trends during the test runs

Types of Resources – The Core Four (Plus One)

Though the Core Four resources exist at both the host and virtual machine levels, they are not the same in how they are instantiated and reported against.

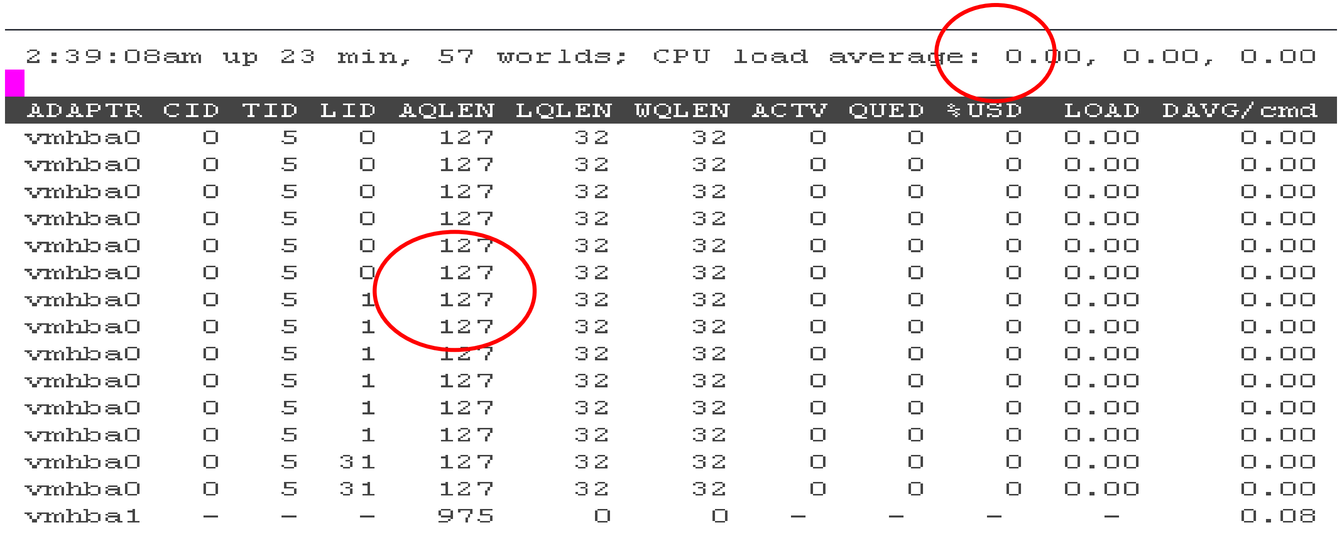
- CPU – processor cycles (vertical), multi-processing (horizontal)
- Memory – allocation and sharing
- Disk (a.k.a. storage) – throughput, size, latencies, queuing
- Network - throughput, latencies, queuing

Though all resources are limited, hypervisors handle the resources differently. CPU is more strictly scheduled, memory is adjusted and reclaimed (more fluid) if based on shares, disk and network are fixed bandwidth (except for queue depths) resources.

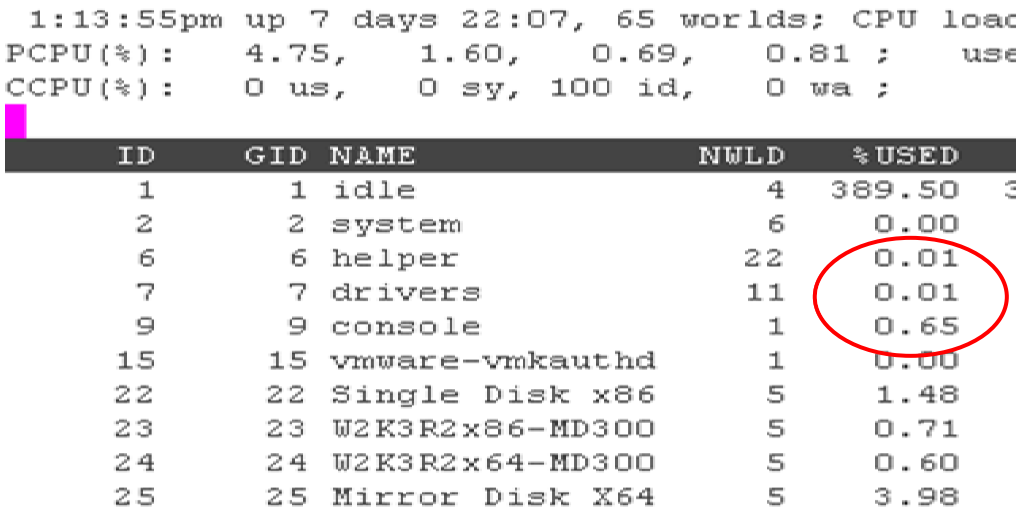
The Fifth Core Four resource is virtualization overhead!

Types of Performance Counters (a.k.a. statistics)

- **Static** – Counters that don’t change during runtime, for example MEMSZ (memsize), Adapter queue depth, VM Name. The static counters are informational and may not be essential during performance problem analysis.
- **Dynamic** – Counters that are computed dynamically, for example CPU load average, memory over-commitment load average.
- **Calculated** - Some are calculated from the delta between two successive snapshots. Refresh interval (-d) determines the time between successive snapshots. For example %CPU used = (CPU used time at snapshot 2 - CPU used time at snapshot 1) / time elapsed between snapshots



```
 2:39:08am up 23 min, 57 worlds; CPU load average: 0.00, 0.00, 0.00

 ADAPTR CID TID LID AQLEN LQLEN WQLEN ACTV QUED %USD   LOAD DAVG/cmd
 vmhba0   0   5   0   127    32    32    0    0    0   0.00     0.00
 vmhba0   0   5   0   127    32    32    0    0    0   0.00     0.00
 vmhba0   0   5   0   127    32    32    0    0    0   0.00     0.00
 vmhba0   0   5   0   127    32    32    0    0    0   0.00     0.00
 vmhba0   0   5   0   127    32    32    0    0    0   0.00     0.00
 vmhba0   0   5   0   127    32    32    0    0    0   0.00     0.00
 vmhba0   0   5   1   127    32    32    0    0    0   0.00     0.00
 vmhba0   0   5   1   127    32    32    0    0    0   0.00     0.00
 vmhba0   0   5   1   127    32    32    0    0    0   0.00     0.00
 vmhba0   0   5   1   127    32    32    0    0    0   0.00     0.00
 vmhba0   0   5   1   127    32    32    0    0    0   0.00     0.00
 vmhba0   0   5   1   127    32    32    0    0    0   0.00     0.00
 vmhba0   0   5  31   127    32    32    0    0    0   0.00     0.00
 vmhba0   0   5  31   127    32    32    0    0    0   0.00     0.00
 vmhba1   -   -   -   975     0     0    -    -    -      -     0.08
```



```
 1:13:55pm up 7 days 22:07, 65 worlds; CPU loac
PCPU(%):   4.75,   1.60,   0.69,   0.81 ;   use
CCPU(%):   0 us,   0 sy, 100 id,   0 wa ;

   ID     GID NAME                  NWLD   %USED
    1       1 idle                     4  389.50  3
    2       2 system                   6    0.00
    6       6 helper                  22    0.01
    7       7 drivers                 11    0.01
    9       9 console                  1    0.65
   15      15 vmware-vmkauthd          1    0.00
   22      22 Single Disk x86          5    1.48
   23      23 W2K3R2x86-MD300          5    0.71
   24      24 W2K3R2x64-MD300          5    0.60
   25      25 Mirror Disk X64          5    3.98
```

A Review of the Basic Performance Analysis Approach

Identify the virtual context of the reported performance problem

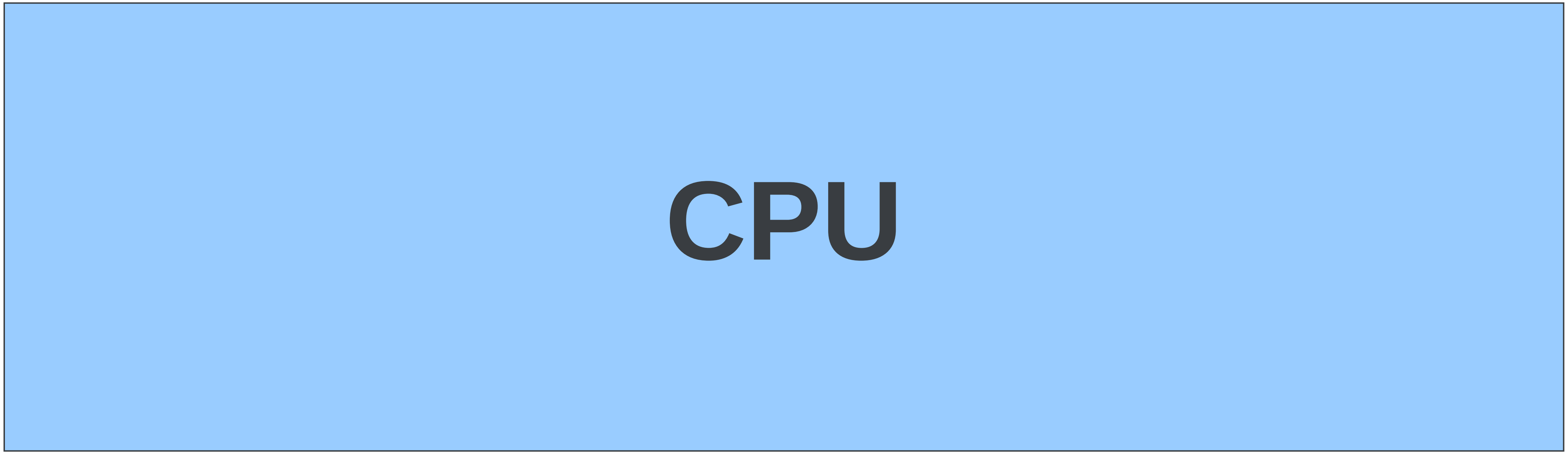
- Where is the problem being seen? (“When I do this here, I get that”)
- How is the problem being quantified? (“My function is 25% slower)
- Apply a reasonability check (“Has something changed from the status quo?”)

Monitor the performance from within that virtual context

- View the performance counters in the same context as the problem
- Look at the ESX cluster level performance counters
- Look for atypical behavior (“Is the amount of resources consumed characteristic of this particular application or task for the server processing tier?”)
- Look for repeat offenders! This happens often.

Expand the performance monitoring to each virtual context as needed

- Are other workloads influencing the virtual context of this particular application and causing a shortage of a particular resource?
- Consider how a shortage is instantiated for each of the Core Four resources



CPU

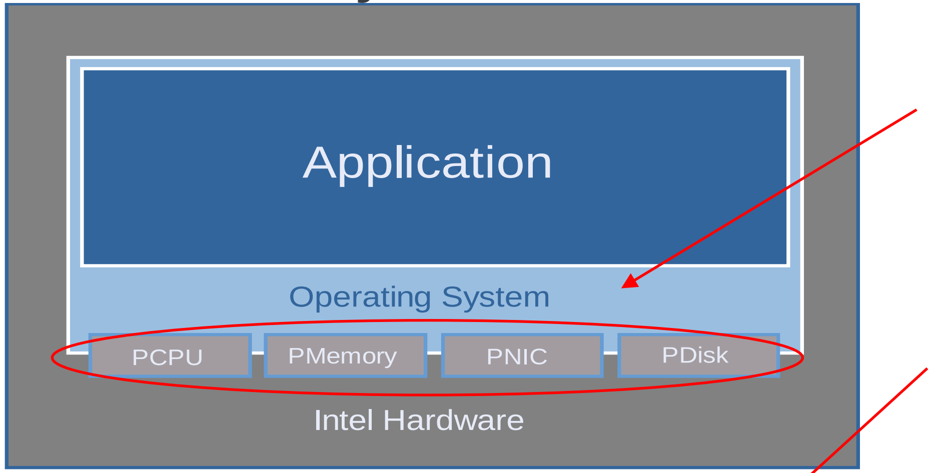
CPU – Understanding PCPU versus VCPU

It is important to separate the physical CPU (PCPU) resources of the ESX host from the virtual CPU (VCPU) resources that are presented by ESX to the virtual machine.

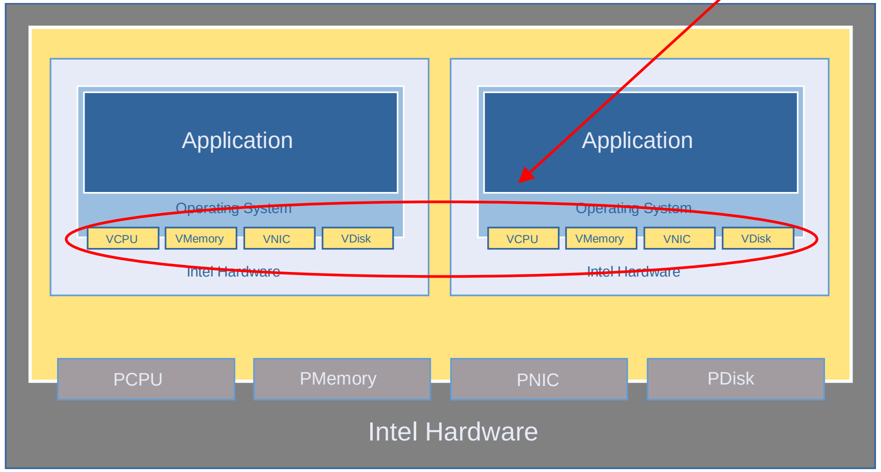
- PCPU – The ESX host’s processor resources are exposed only to ESX. The virtual machines are not aware and cannot report on those physical resources.
- VCPU – ESX effectively assembles a virtual CPU(s) for each virtual machine from the physical machine’s processors/cores, based upon the type of resource allocation (ex. shares, guarantees, minimums).
- Scheduling - The virtual machine is scheduled to run inside the VCPU(s), with the virtual machine’s reporting mechanism (such as W2K’s System Monitor) reporting on the virtual machine’s allocated VCPU(s) and remaining Core Four resources.

PCPU and VCPU Example – Two Virtual Machines

Physical Host



Two Virtual Machines



Physical Resources

2 socket, four core @2.1Ghz = 16.8Ghz CPU
8 GB RAM

Virtual Machine Logical Resources

Each virtual machine defined as a uniprocessor
VCPU = 2.1GHZ since uniprocessor
Memory allocation of 1GB per virtual machine

Allocated Physical Resources

2 cores @2.1Ghz = 4.2Ghz CPU
2 GB RAM

Remaining Physical Resources

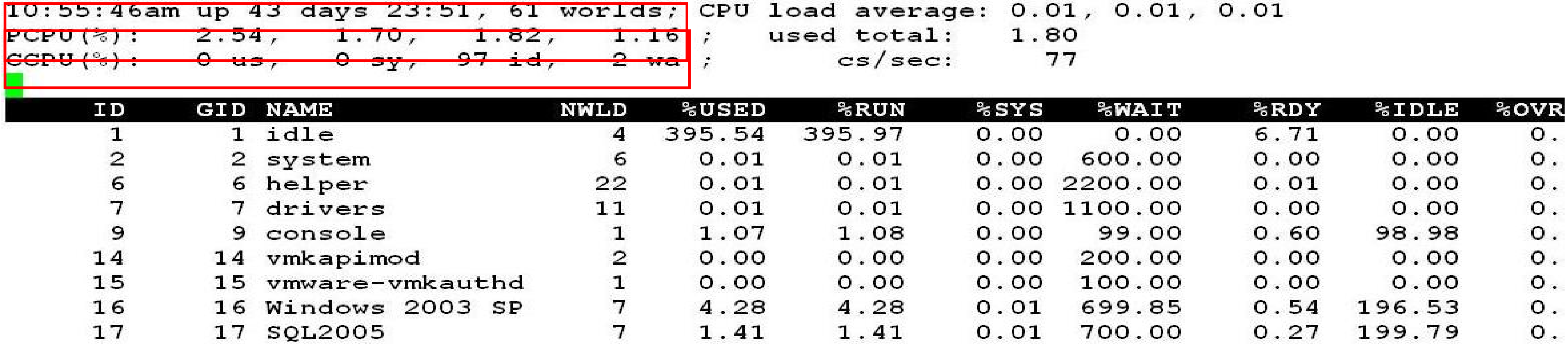
6 cores @2.1Ghz = 12.6Ghz CPU (minus virt. overhead)
6 GB RAM (minus virt. overhead)
Limits, maximums, shares all affect real resources

CPU – Key Question and Considerations

Is there a lack of CPU resources for the VCPU(s) of the virtual machine or for the PCPU(s) of the ESX host?

- Allocation – The CPU allocation for a specific workload can be constrained due to the resource settings or number of CPUs, amount of shares, or limits. The key field at the virtual machine level is CPU queuing and at the ESX level it is Ready to Run (%RDY in Esxtop).
- Capacity - The virtual machine's CPU can be constrained due to a lack of sufficient capacity at the ESX host level as evidenced by the PCPU/LCPU utilization.
- Contention – The specific workload may be constrained by the consumption of workloads operating outside of their typical patterns
- SMP CPU Skewing – The movement towards lazy scheduling of SMP CPUs can cause delays if one CPU gets too far "ahead" of the other. Look for higher %CSTP (co-schedule pending)

Esxtop CPU screen (c)

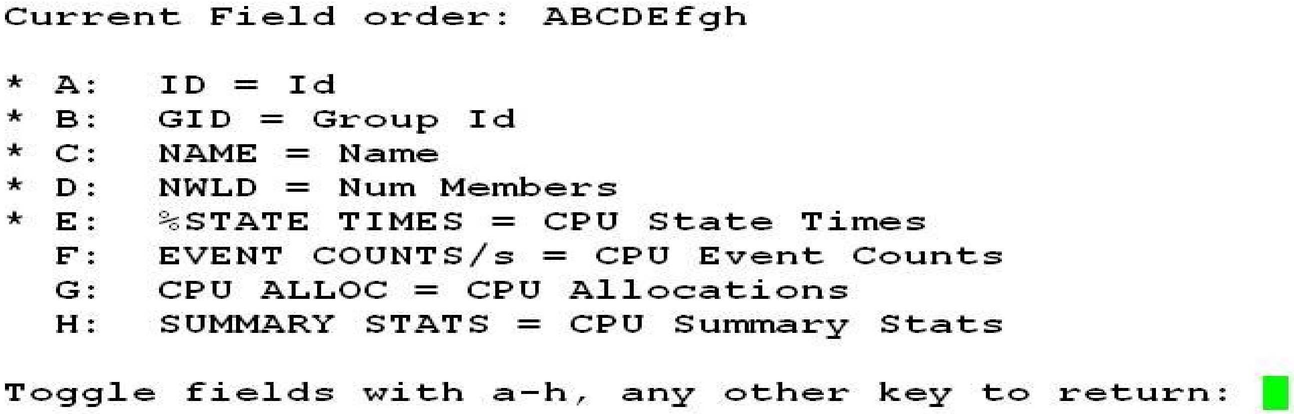


```
10:55:46am up 43 days 23:51, 61 worlds; CPU load average: 0.01, 0.01, 0.01
PCPU(%):    2.54,    1.70,    1.82,    1.16 ;   used total:   1.80
CCPU(%):    0 us,    0 sy,   97 id,    2 wa ;       cs/sec:     77

       ID      GID NAME               NWLD   %USED    %RUN    %SYS   %WAIT    %RDY   %IDLE  %OVR
        1        1 idle                  4  395.54  395.97    0.00    0.00    6.71    0.00    0.
        2        2 system                6    0.01    0.01    0.00  600.00    0.00    0.00    0.
        6        6 helper               22    0.01    0.01    0.00 2200.00    0.01    0.00    0.
        7        7 drivers              11    0.01    0.01    0.00 1100.00    0.00    0.00    0.
        9        9 console               1    1.07    1.08    0.00   99.00    0.60   98.98    0.
       14       14 vmkapimod             2    0.00    0.00    0.00  200.00    0.00    0.00    0.
       15       15 vmware-vmkauthd       1    0.00    0.00    0.00  100.00    0.00    0.00    0.
       16       16 Windows 2003 SP       7    4.28    4.28    0.01  699.85    0.54  196.53    0.
       17       17 SQL2005               7    1.41    1.41    0.01  700.00    0.27  199.79    0.
```

PCPU = Physical CPU/core

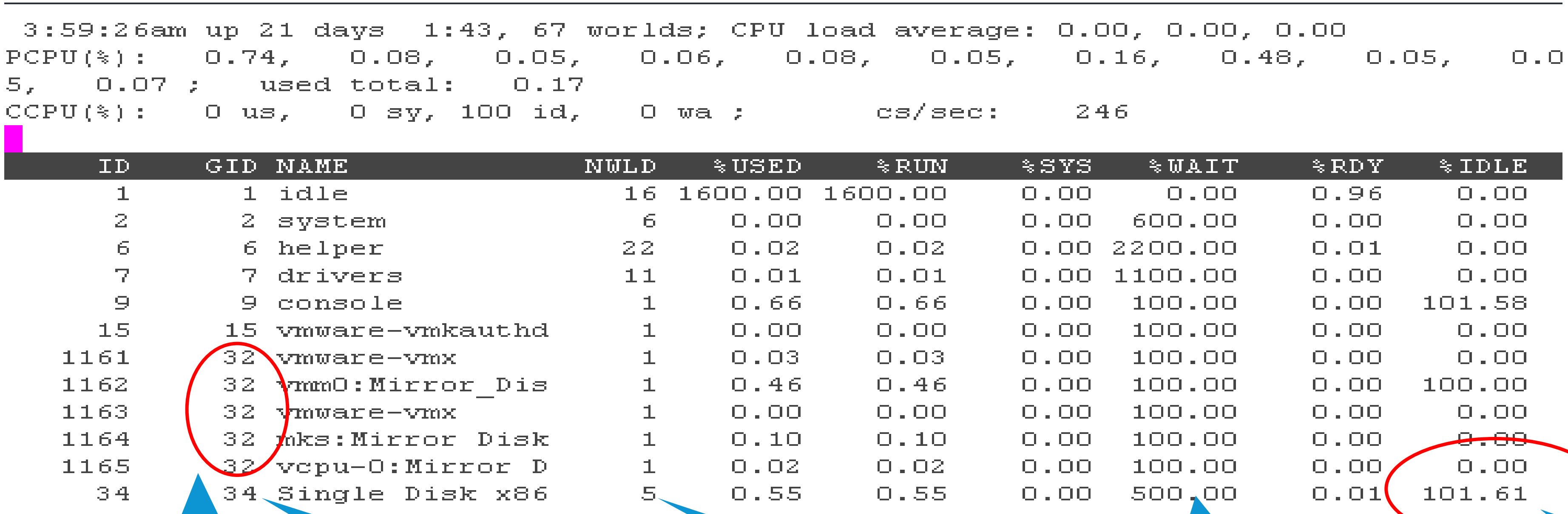
CCPU = Console CPU (CPU 0)

Press ‘f’ key to choose fields



```
Current Field order: ABCDEfgh

* A:  ID = Id
* B:  GID = Group Id
* C:  NAME = Name
* D:  NWLD = Num Members
* E:  %STATE TIMES = CPU State Times
  F:  EVENT COUNTS/s = CPU Event Counts
  G:  CPU ALLOC = CPU Allocations
  H:  SUMMARY STATS = CPU Summary Stats

Toggle fields with a-h, any other key to return:
```

Idle State on Test Bed – GID 32 Expanded



```
 3:59:26am up 21 days  1:43, 67 worlds; CPU load average: 0.00, 0.00, 0.00
PCPU(%):   0.74,   0.08,   0.05,   0.06,   0.08,   0.05,   0.16,   0.48,   0.05,   0.0
5,   0.07 ;   used total:   0.17
CCPU(%):   0 us,   0 sy, 100 id,   0 wa ;       cs/sec:    246

     ID      GID NAME             NWLD  %USED    %RUN    %SYS   %WAIT    %RDY   %IDLE
      1        1 idle               16 1600.00 1600.00    0.00    0.00    0.96    0.00
      2        2 system              6    0.00    0.00    0.00  600.00    0.00    0.00
      6        6 helper             22    0.02    0.02    0.00 2200.00    0.01    0.00
      7        7 drivers            11    0.01    0.01    0.00 1100.00    0.00    0.00
      9        9 console             1    0.66    0.66    0.00  100.00    0.00  101.58
     15       15 vmware-vmkauthd     1    0.00    0.00    0.00  100.00    0.00    0.00
   1161       32 vmware-vmx          1    0.03    0.03    0.00  100.00    0.00    0.00
   1162       32 vmm0:Mirror_Dis     1    0.46    0.46    0.00  100.00    0.00  100.00
   1163       32 vmware-vmx          1    0.00    0.00    0.00  100.00    0.00    0.00
   1164       32 mks:Mirror Disk     1    0.10    0.10    0.00  100.00    0.00    0.00
   1165       32 vcpu-0:Mirror D     1    0.02    0.02    0.00  100.00    0.00    0.00
     34       34 Single Disk x86     5    0.55    0.55    0.00  500.00    0.01  101.61
```

Wait includes idle

Expanded GID

Rolled Up GID

Five Worlds

Cumulative Wait %

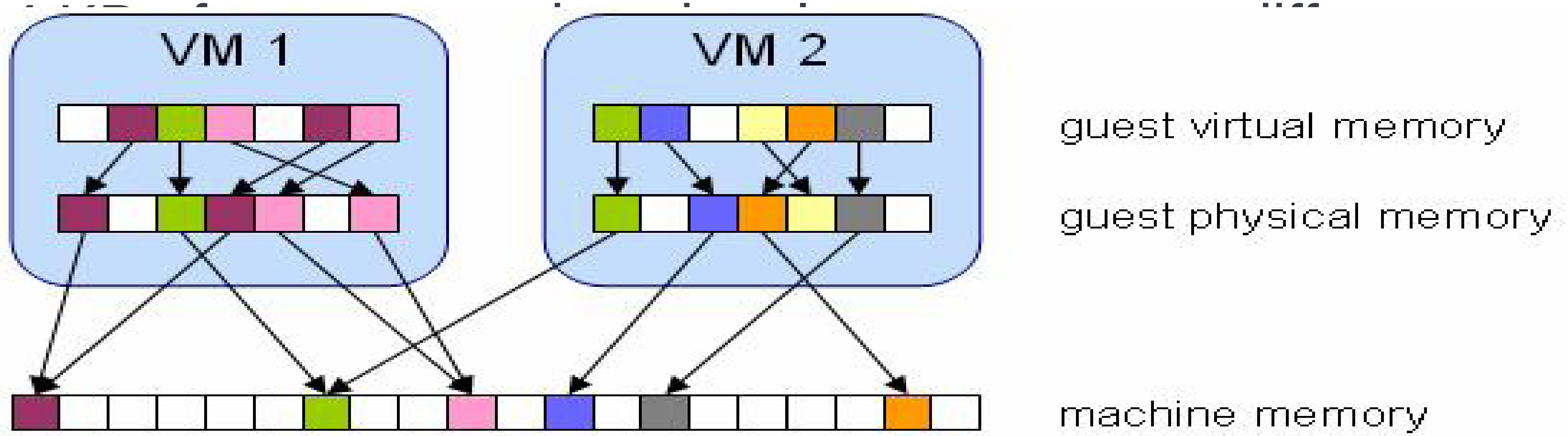
Total Idle %

Memory

Memory – Separating the machine and guest memory

It is important to note that some statistics refer to guest physical memory while others refer to machine memory. “ **Guest physical memory**" is the virtual-hardware physical memory presented to the VM. " **Machine memory**" is actual physical RAM in the ESX host.

In the figure below, two VMs are running on an ESX host, where each block represents of data on a block.



Inside each VM, the guest OS maps the virtual memory to its physical memory. ESX Kernel maps the guest physical memory to machine memory. Due to ESX Page Sharing technology, guest physical pages with the same content can
be mapped to the same machine page.

A Brief Look at Ballooning

- The W2K balloon driver is located in VMtools
- ESX sets a balloon target for each workload at start-up and as workloads are introduced/removed
- The balloon driver expands memory consumption, requiring the Virtual Machine operating system to reclaim memory based on its algorithms
- Ballooning routinely takes 10-20 minutes to reach the target
- The returned memory is now available for ESX to use
- Key ballooning fields:

SZTGT: determined by reservation, limit and memory shares
SWCUR = 0 : no swapping in the past
SWTGT = 0 : no swapping pressure
SWR/S, SWR/W = 0 : No swapping activity currently

ESX Memory Sharing - The "Water Bed Effect"

ESX handles memory shares on an ESX host and across an ESX cluster with a result similar to a single water bed, or room full of water beds, depending upon the action and the memory allocation type:

- **Initial ESX boot** (i.e., "lying down on the water bed") – ESX sets a target working size for each virtual machine, based upon the memory allocations or shares, and uses ballooning to pare back the initial allocations until those targets are reached (if possible).
- **Steady State** (i.e., "minor position changes") - The host gets into a steady state with small adjustments made to memory allocation targets. Memory "ripples" occur during steady state, with the amplitude dependent upon the workload characteristics and consumption by the virtual machines.
- **New Event** (i.e., "second person on the bed") – The host receives additional workload via a newly started virtual machine or VMotion moves a virtual machine to the host through a manual step, maintenance mode, or DRS. ESX pares back the target working size of that virtual machine while the other virtual machines lose CPU cycles that are directed to the new workload.
- **Large Event** (i.e., "jumping across water beds") – The cluster has a major event that causes a substantial movement of workloads to or between multiple hosts. Each of the hosts has to reach a steady state, or to have DRS determine that the workload is not a current candidate for the existing host, moving to another host that has reached a steady state with available capacity.   Maintenance mode is another major event.

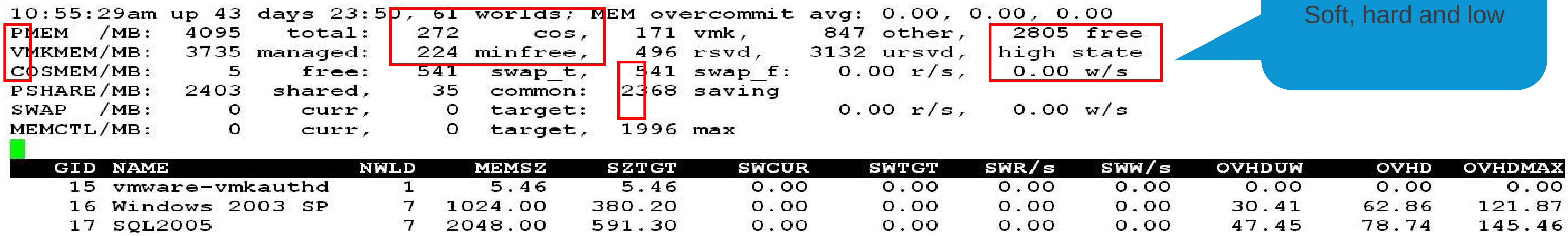
Memory – Key Question and Considerations

Is the memory allocation for each workload optimum to prevent swapping at the Virtual Machine level, yet low enough not to constrain other workloads or the ESX host?

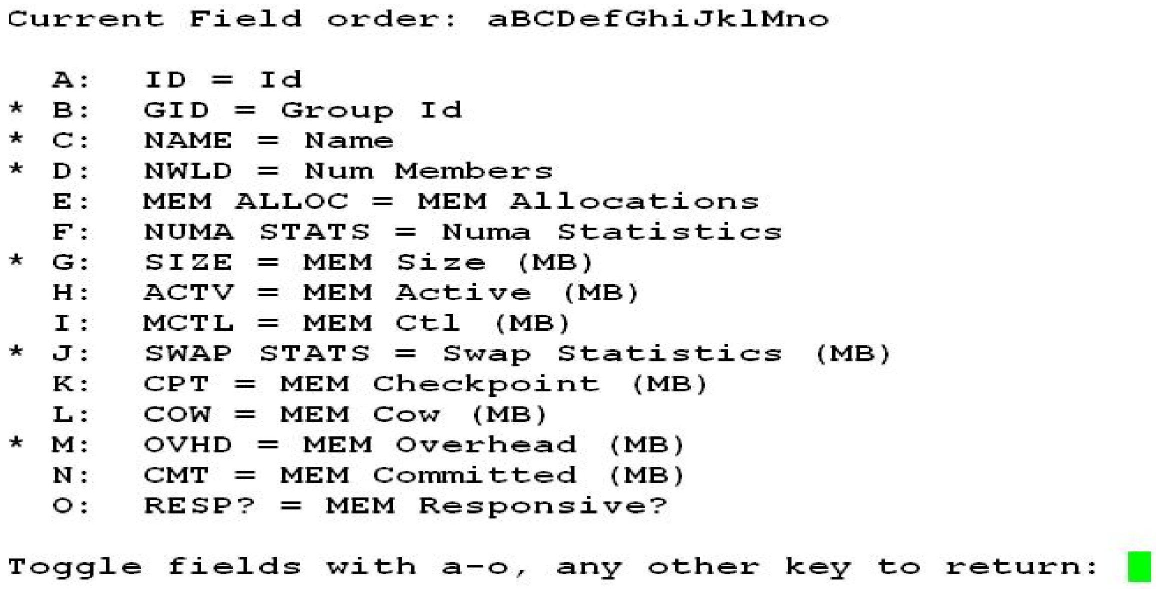
- HA/DRS/Maintenance Mode Regularity – How often do the workloads in the cluster get moved between hosts? Each movement causes an impact on the receiving (negative) and sending (positive) hosts with maintenance mode causing a rolling wave of impact across the cluster, depending upon the timing.
- Allocation Type – Each of the allocation types have their drawbacks so tread carefully when choosing the allocation type. One size seldom is right for all needs.
- Capacity/Swapping - The virtual machine’s CPU can be constrained due to a lack of sufficient capacity at the ESX host level. Look for regular swapping at the ESX host level as an indicator of a memory capacity issue but be sure to notice memory leaks that artificially force a memory shortage situation.

Esxtop memory screen (m)

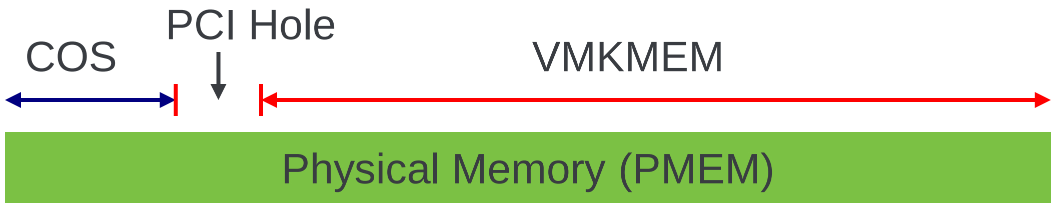
Possible states: High, Soft, hard and low



```
10:55:29am up 43 days 23:50, 61 worlds; MEM overcommit avg: 0.00, 0.00, 0.00
PMEM   /MB:  4095    total:   272      cos,    171 vmk,      847 other,   2805 free
VMKMEM/MB:  3735 managed:   224 minfree,   496 rsvd,   3132 ursvd,  high state
COSMEM/MB:     5     free:   541   swap_t,   541 swap_f:   0.00 r/s,   0.00 w/s
PSHARE/MB:  2403   shared,    35   common:  2368 saving
SWAP   /MB:     0     curr,     0   target:                0.00 r/s,   0.00 w/s
MEMCTL/MB:     0     curr,     0   target,  1996 max

    GID NAME                NWLD     MEMSZ    SZTGT    SWCUR    SWTGT   SWR/s   SWW/s   OVHDUW     OVHD  OVHDMAX
     15 vmware-vmkauthd        1      5.46     5.46     0.00     0.00    0.00    0.00     0.00     0.00     0.00
     16 Windows 2003 SP        7   1024.00   380.20     0.00     0.00    0.00    0.00    30.41    62.86   121.87
     17 SQL2005                7   2048.00   591.30     0.00     0.00    0.00    0.00    47.45    78.74   145.46
```

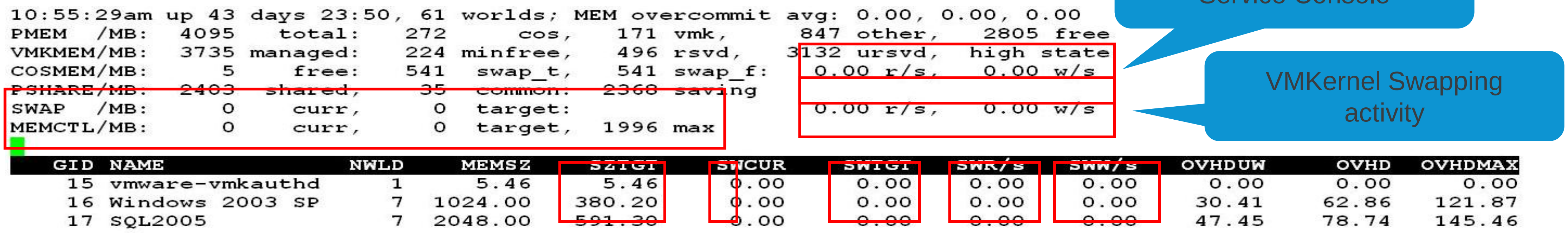


```
Current Field order: aBCDefGhiJklMno

  A:  ID = Id
* B:  GID = Group Id
* C:  NAME = Name
* D:  NWLD = Num Members
  E:  MEM ALLOC = MEM Allocations
  F:  NUMA STATS = Numa Statistics
* G:  SIZE = MEM Size (MB)
  H:  ACTV = MEM Active (MB)
  I:  MCTL = MEM Ctl (MB)
* J:  SWAP STATS = Swap Statistics (MB)
  K:  CPT = MEM Checkpoint (MB)
  L:  COW = MEM Cow (MB)
* M:  OVHD = MEM Overhead (MB)
  N:  CMT = MEM Committed (MB)
  O:  RESP? = MEM Responsive?

Toggle fields with a-o, any other key to return:
```



VMKMEM - Memory managed by VMKernel
COSMEM - Memory used by Service Console

Esxtop memory screen (m)



```
10:55:29am up 43 days 23:50, 61 worlds; MEM overcommit avg: 0.00, 0.00, 0.00
PMEM   /MB:  4095   total:    272       cos,    171 vmk,     847 other,   2805 free
VMKMEM/MB:  3735 managed:    224 minfree,    496 rsvd,    3132 ursvd,   high state
COSMEM/MB:     5    free:    541   swap_t,    541 swap_f:   0.00 r/s,   0.00 w/s
PSHARE/MB:  2403   shared,     35   common:   2368 saving
SWAP   /MB:     0     curr,      0   target:                0.00 r/s,   0.00 w/s
MEMCTL/MB:     0     curr,      0   target,   1996 max

    GID NAME              NWLD     MEMSZ     SZTGT     SWCUR     SWTGT     SWR/s     SWW/s    OVHDUW      OVHD   OVHDMAX
     15 vmware-vmkauthd      1      5.46      5.46      0.00      0.00      0.00      0.00      0.00      0.00      0.00
     16 Windows 2003 SP      7   1024.00    380.20      0.00      0.00      0.00      0.00     30.41     62.86    121.87
     17 SQL2005              7   2048.00    591.30      0.00      0.00      0.00      0.00     47.45     78.74    145.46
```

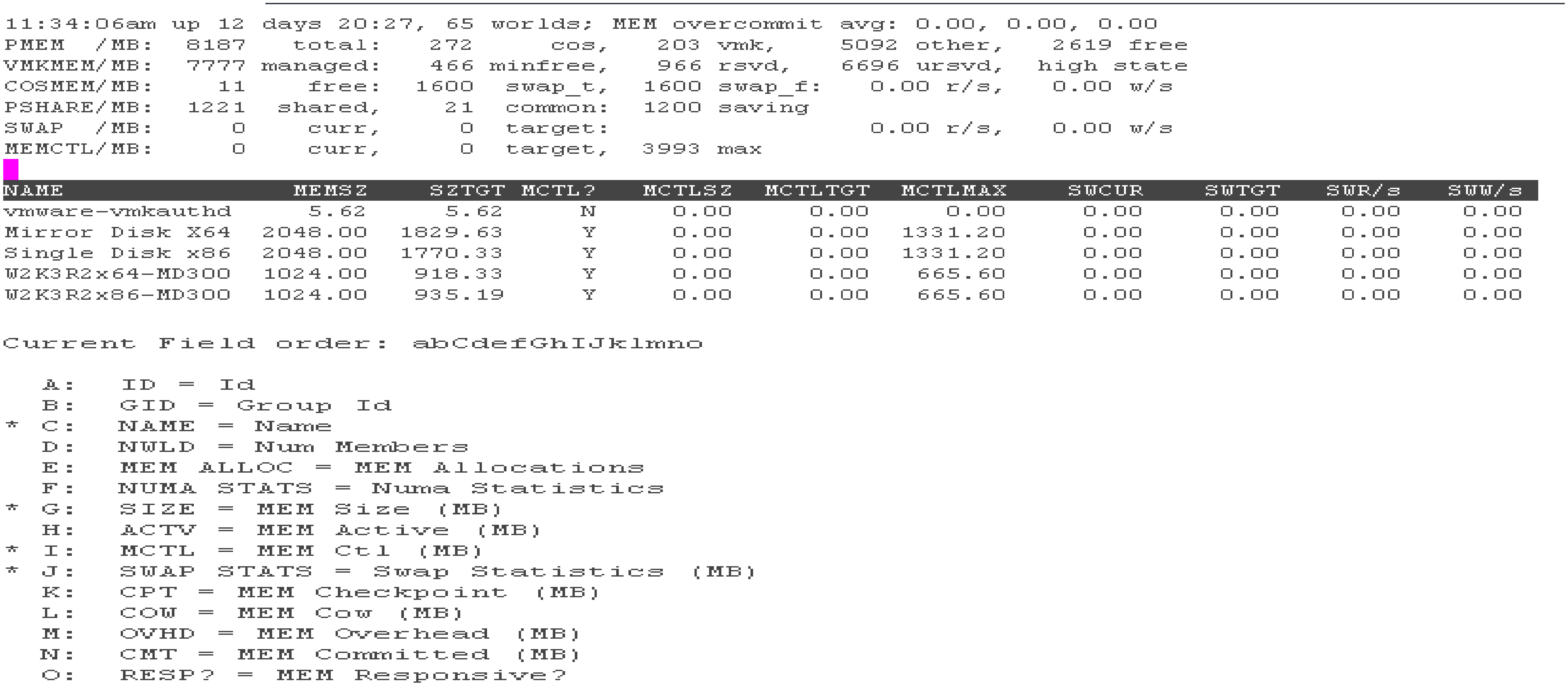
Swapping activity in Service Console

VMKernel Swapping activity

SZTGT : determined by reservation, limit and memory shares
SWCUR = 0 : no swapping in the past
SWTGT = 0 : no swapping pressure
SWR/S, SWR/W = 0 : No swapping activity currently

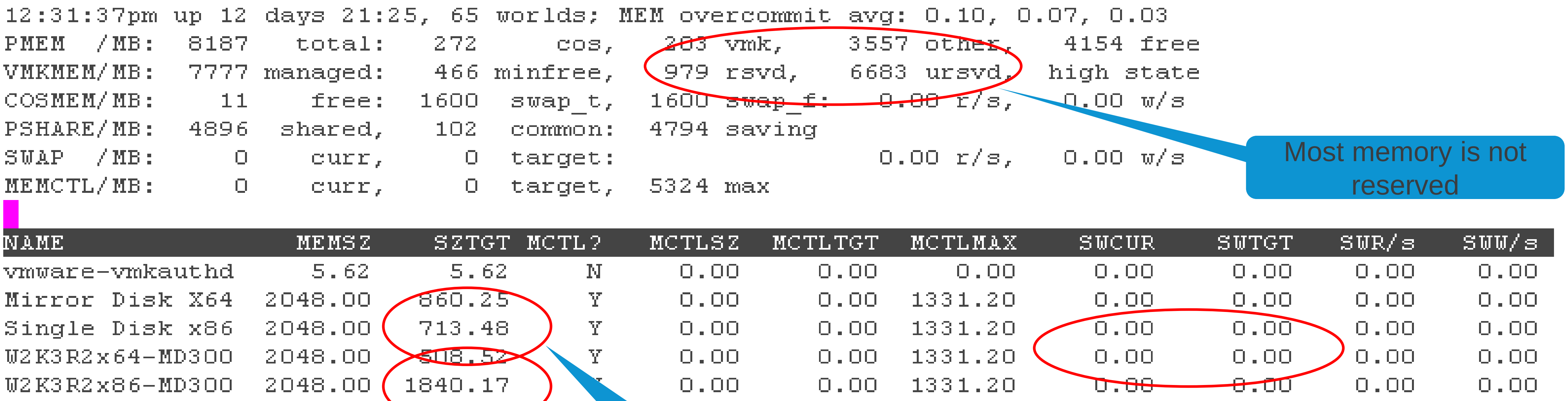
SZTGT = Size target
SWTGT = Swap target
SWCUR = Currently swapped
MEMCTL = Balloon driver
SWR/S = Swap read /sec
SWW/S = Swap write /sec

Idle State on Test Bed – Memory View



```
11:34:06am up 12 days 20:27, 65 worlds; MEM overcommit avg: 0.00, 0.00, 0.00
PMEM  /MB:  8187   total:   272     cos,   203 vmk,   5092 other,   2619 free
VMKMEM/MB:  7777 managed:   466 minfree,   966 rsvd,   6696 ursvd,  high state
COSMEM/MB:    11    free:  1600  swap_t,  1600 swap_f:   0.00 r/s,   0.00 w/s
PSHARE/MB:  1221  shared,    21  common:  1200 saving
SWAP  /MB:     0    curr,     0  target:                 0.00 r/s,   0.00 w/s
MEMCTL/MB:     0    curr,     0  target,  3993 max

NAME                   MEMSZ     SZTGT MCTL?    MCTLSZ  MCTLTGT  MCTLMAX     SWCUR     SWTGT    SWR/s    SWW/s
vmware-vmkauthd         5.62      5.62     N      0.00     0.00     0.00      0.00      0.00     0.00     0.00
Mirror Disk X64      2048.00   1829.63     Y      0.00     0.00  1331.20      0.00      0.00     0.00     0.00
Single Disk x86      2048.00   1770.33     Y      0.00     0.00  1331.20      0.00      0.00     0.00     0.00
W2K3R2x64-MD300      1024.00    918.33     Y      0.00     0.00   665.60      0.00      0.00     0.00     0.00
W2K3R2x86-MD300      1024.00    935.19     Y      0.00     0.00   665.60      0.00      0.00     0.00     0.00

Current Field order: abCdefGhIJklmno

   A:   ID = Id
   B:   GID = Group Id
*  C:   NAME = Name
   D:   NWLD = Num Members
   E:   MEM ALLOC = MEM Allocations
   F:   NUMA STATS = Numa Statistics
*  G:   SIZE = MEM Size (MB)
   H:   ACTV = MEM Active (MB)
*  I:   MCTL = MEM Ctl (MB)
*  J:   SWAP STATS = Swap Statistics (MB)
   K:   CPT = MEM Checkpoint (MB)
   L:   COW = MEM Cow (MB)
   M:   OVHD = MEM Overhead (MB)
   N:   CMT = MEM Committed (MB)
   O:   RESP? = MEM Responsive?
```

Memory View at Steady State of 3 Virtual Machines – Memory Shares



```
12:31:37pm up 12 days 21:25, 65 worlds; MEM overcommit avg: 0.10, 0.07, 0.03
PMEM   /MB:  8187   total:   272      cos,   203 vmk,   3557 other,   4154 free
VMKMEM/MB:  7777 managed:   466 minfree,   979 rsvd,   6683 ursvd,  high state
COSMEM/MB:    11    free:  1600  swap_t,  1600 swap_f:   0.00 r/s,   0.00 w/s
PSHARE/MB:  4896  shared,   102  common:  4794 saving
SWAP   /MB:     0    curr,     0  target:                0.00 r/s,   0.00 w/s
MEMCTL/MB:     0    curr,     0  target,  5324 max

NAME                 MEMSZ     SZTGT MCTL?    MCTLSZ   MCTLTGT   MCTLMAX     SWCUR     SWTGT     SWR/s     SWW/s
vmware-vmkauthd       5.62      5.62     N      0.00      0.00      0.00      0.00      0.00      0.00      0.00
Mirror Disk X64    2048.00    860.25     Y      0.00      0.00   1331.20      0.00      0.00      0.00      0.00
Single Disk x86    2048.00    713.48     Y      0.00      0.00   1331.20      0.00      0.00      0.00      0.00
W2K3R2x64-MD300    2048.00    508.52     Y      0.00      0.00   1331.20      0.00      0.00      0.00      0.00
W2K3R2x86-MD300    2048.00   1840.17     Y      0.00      0.00   1331.20      0.00      0.00      0.00      0.00
```

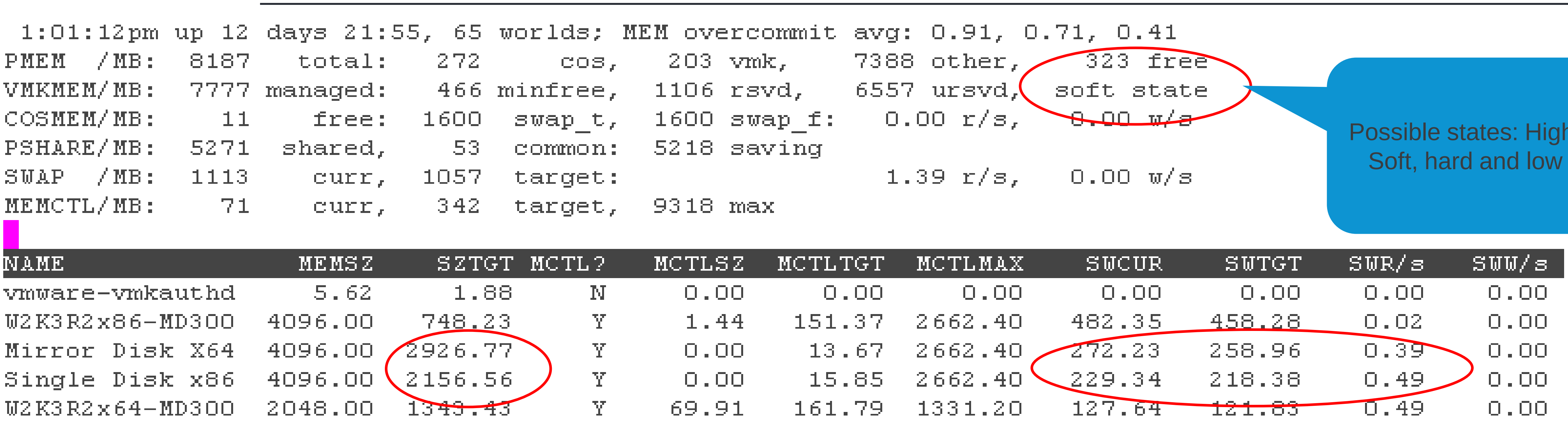
Most memory is not reserved

Virtual Machine Just Powered On

These VMs are at memory steady state

No VM Swapping or Targets

Ballooning and Swapping in Progress – Memory View



```
 1:01:12pm up 12 days 21:55, 65 worlds; MEM overcommit avg: 0.91, 0.71, 0.41
PMEM  /MB:  8187   total:   272      cos,   203 vmk,   7388 other,     323 free
VMKMEM/MB:  7777 managed:   466 minfree,  1106 rsvd,   6557 ursvd,   soft state
COSMEM/MB:    11    free:  1600  swap_t,  1600 swap_f:   0.00 r/s,   0.00 w/s
PSHARE/MB:  5271  shared,    53  common:  5218 saving
SWAP  /MB:  1113    curr,  1057  target:                 1.39 r/s,   0.00 w/s
MEMCTL/MB:    71    curr,   342  target,  9318 max

NAME                  MEMSZ     SZTGT  MCTL?    MCTLSZ   MCTLTGT   MCTLMAX     SWCUR     SWTGT    SWR/s    SWW/s
vmware-vmkauthd        5.62      1.88      N      0.00      0.00      0.00      0.00      0.00     0.00     0.00
W2K3R2x86-MD300     4096.00    748.23      Y      1.44    151.37   2662.40    482.35    458.28     0.02     0.00
Mirror Disk X64     4096.00   2926.77      Y      0.00     13.67   2662.40    272.23    258.96     0.39     0.00
Single Disk x86     4096.00   2156.56      Y      0.00     15.85   2662.40    229.34    218.38     0.49     0.00
W2K3R2x64-MD300     2048.00   1343.43      Y     69.91    161.79   1331.20    127.64    121.83     0.49     0.00
```

Possible states: High, Soft, hard and low

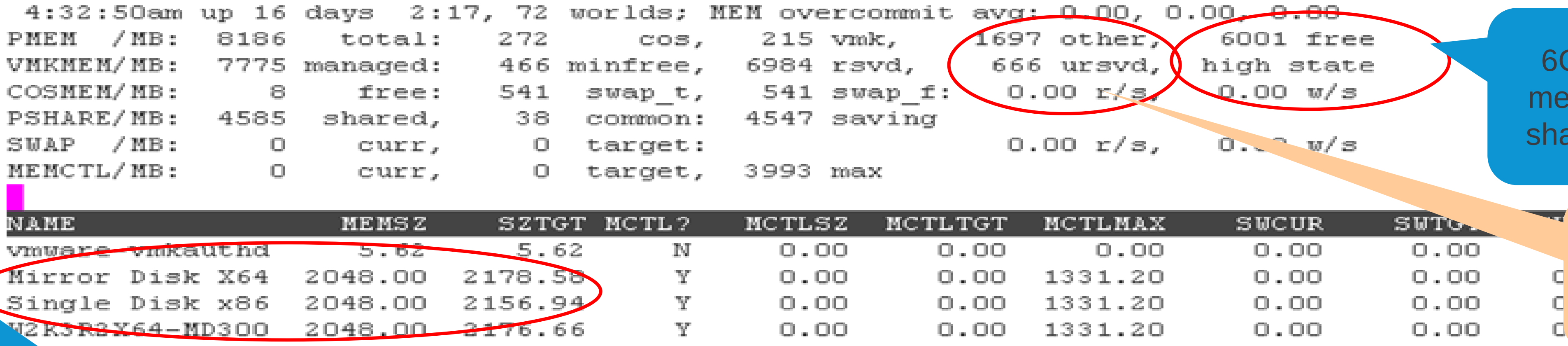
Different Size Targets Due to Different Amount of Up Time

Ballooning In Effect

Mild swapping

Memory Reservations – Effect on New Loads

What Size Virtual Machine with Reserved Memory Can Be Started?



```
 4:32:50am up 16 days  2:17, 72 worlds; MEM overcommit avg: 0.00, 0.00, 0.00
PMEM   /MB:  8186    total:   272     cos,   215 vmk,   1697 other,  6001 free
VMKMEM/MB:  7775 managed:   466 minfree,  6984 rsvd,    666 ursvd,  high state
COSMEM/MB:     8     free:   541  swap_t,   541 swap_f:  0.00 r/s,   0.00 w/s
PSHARE/MB:  4585   shared,    38  common:  4547 saving
SWAP   /MB:     0     curr,     0  target:                0.00 r/s,   0.00 w/s
MEMCTL/MB:     0     curr,     0  target,  3993 max

NAME                  MEMSZ     SZTGT  MCTL?    MCTLSZ   MCTLTGT   MCTLMAX     SWCUR     SWTGT
vmware-vmkauthd        5.62      5.62      N      0.00      0.00      0.00      0.00      0.00
Mirror Disk X64     2048.00   2178.58      Y      0.00      0.00   1331.20      0.00      0.00
Single Disk x86     2048.00   2156.94      Y      0.00      0.00   1331.20      0.00      0.00
W2K3R2X64-MD300     2048.00   2176.66      Y      0.00      0.00   1331.20      0.00      0.00
```

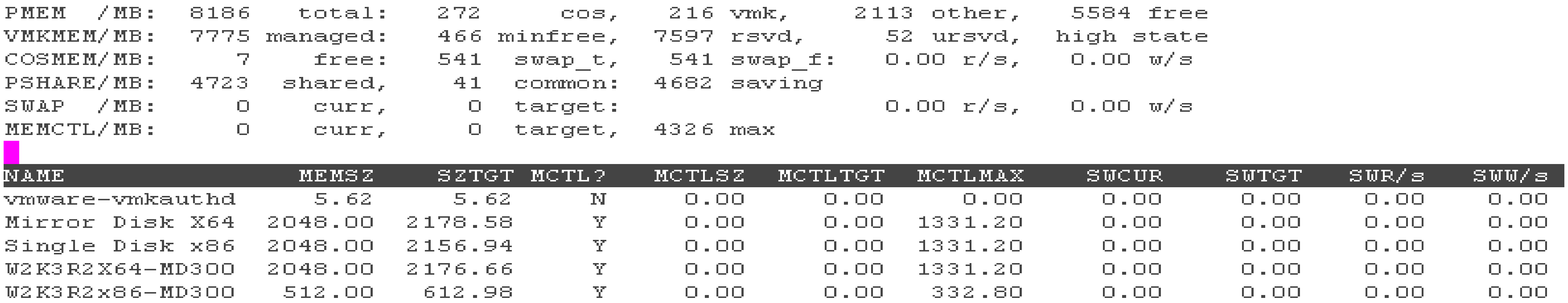
6GB of “free” physical memory due to memory sharing over 20 minutes

666MB of unreserved memory

Three VMs, each with 2GB reserved memory

Can’t start fourth virtual machine of >512MB of reserved memory

Fourth virtual machine of 512MB of reserved memory started



```
PMEM   /MB:  8186    total:   272     cos,   216 vmk,   2113 other,  5584 free
VMKMEM/MB:  7775 managed:   466 minfree,  7597 rsvd,     52 ursvd,  high state
COSMEM/MB:     7     free:   541  swap_t,   541 swap_f:  0.00 r/s,   0.00 w/s
PSHARE/MB:  4723   shared,    41  common:  4682 saving
SWAP   /MB:     0     curr,     0  target:                0.00 r/s,   0.00 w/s
MEMCTL/MB:     0     curr,     0  target,  4326 max

NAME                  MEMSZ     SZTGT  MCTL?    MCTLSZ   MCTLTGT   MCTLMAX     SWCUR     SWTGT    SWR/s    SWW/s
vmware-vmkauthd        5.62      5.62      N      0.00      0.00      0.00      0.00      0.00     0.00     0.00
Mirror Disk X64     2048.00   2178.58      Y      0.00      0.00   1331.20      0.00      0.00     0.00     0.00
Single Disk x86     2048.00   2156.94      Y      0.00      0.00   1331.20      0.00      0.00     0.00     0.00
W2K3R2X64-MD300     2048.00   2176.66      Y      0.00      0.00   1331.20      0.00      0.00     0.00     0.00
W2K3R2x86-MD300      512.00    612.98      Y      0.00      0.00    332.80      0.00      0.00     0.00     0.00
```

Tools and Application of Those Tools

Load Generators, Data Gatherers, Data Analyzers

- Load Generators
  - IOMeter – www.iometer.org
  - Consume – windows SDK
  - SQLIOSIM - http://support.microsoft.com/?id=231619
- Data Gatherers
  - ESXTOP
  - Virtual Center
  - Vscsistats
- Data Analyzers
  - ESXTOP (interactive or batch mode)
  - Windows Perfmon/Systems Monitor
  - ESXPLOT
  - RVTools: www.robware.net

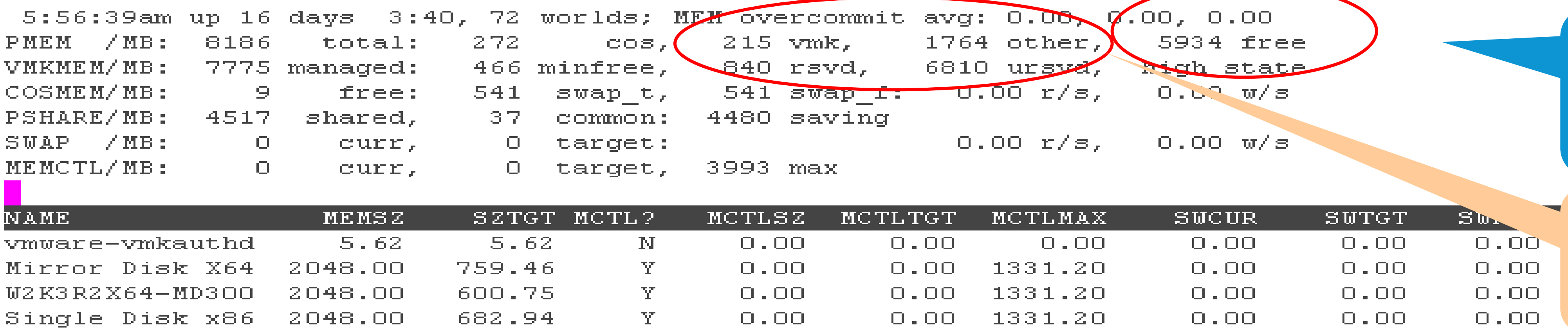
A Comparison of Esxtop and the vSphere Client

- vC gives a graphical view of both real-time and trend consumption
- vC combines real-time reporting with short term (1 hour) trending
- vC can report on the virtual machine, ESX host, or ESX cluster
- vC now has performance overview charts in vSphere 4.0
- Esxtop allows more concurrent performance counters to be shown
- Esxtop has a higher system overhead to run
- Esxtop can sample down to a 2 second sampling period
- Esxtop gives a detailed view of each of the Core Four

Recommendation – Use vC to get a general view of the system performance but use Esxtop for detailed problem analysis.

Memory Shares – Effect on New Loads

Three VMs with 2GB allocation

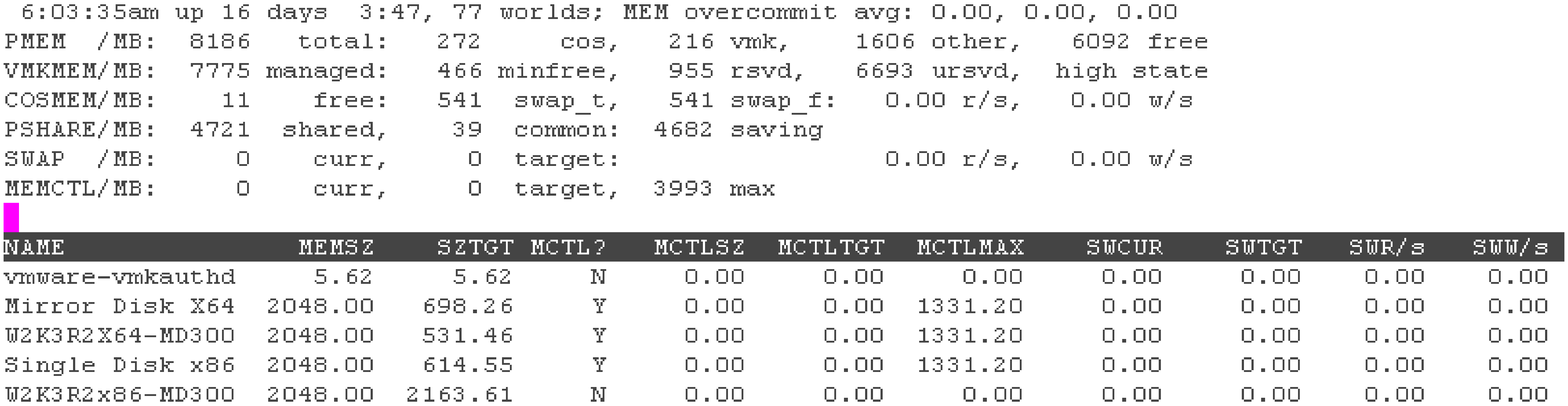


```
 5:56:39am up 16 days  3:40, 72 worlds; MEM overcommit avg: 0.00, 0.00, 0.00
PMEM  /MB:  8186   total:   272      cos,   215 vmk,    1764 other,   5934 free
VMKMEM/MB:  7775 managed:   466 minfree,   840 rsvd,   6810 ursvd,  high state
COSMEM/MB:     9    free:   541  swap_t,   541 swap_f:   0.00 r/s,   0.00 w/s
PSHARE/MB:  4517  shared,    37  common:  4480 saving
SWAP  /MB:     0    curr,     0  target:                 0.00 r/s,   0.00 w/s
MEMCTL/MB:     0    curr,     0  target,  3993 max

NAME                 MEMSZ     SZTGT  MCTL?    MCTLSZ   MCTLTGT   MCTLMAX     SWCUR     SWTGT    SW
vmware-vmkauthd       5.62      5.62      N      0.00      0.00      0.00      0.00      0.00    0.00
Mirror Disk X64    2048.00    759.46      Y      0.00      0.00   1331.20      0.00      0.00    0.00
W2K3R2X64-MD300    2048.00    600.75      Y      0.00      0.00   1331.20      0.00      0.00    0.00
Single Disk x86    2048.00    682.94      Y      0.00      0.00   1331.20      0.00      0.00    0.00
```

5.9 GB of "free" physical memory

6GB of unreserved memory

Fourth virtual machine of 2GB of memory allocation started successfully



```
 6:03:35am up 16 days  3:47, 77 worlds; MEM overcommit avg: 0.00, 0.00, 0.00
PMEM  /MB:  8186   total:   272      cos,   216 vmk,    1606 other,   6092 free
VMKMEM/MB:  7775 managed:   466 minfree,   955 rsvd,   6693 ursvd,  high state
COSMEM/MB:    11    free:   541  swap_t,   541 swap_f:   0.00 r/s,   0.00 w/s
PSHARE/MB:  4721  shared,    39  common:  4682 saving
SWAP  /MB:     0    curr,     0  target:                 0.00 r/s,   0.00 w/s
MEMCTL/MB:     0    curr,     0  target,  3993 max

NAME                 MEMSZ     SZTGT  MCTL?    MCTLSZ   MCTLTGT   MCTLMAX     SWCUR     SWTGT     SWR/s     SWW/s
vmware-vmkauthd       5.62      5.62      N      0.00      0.00      0.00      0.00      0.00      0.00      0.00
Mirror Disk X64    2048.00    698.26      Y      0.00      0.00   1331.20      0.00      0.00      0.00      0.00
W2K3R2X64-MD300    2048.00    531.46      Y      0.00      0.00   1331.20      0.00      0.00      0.00      0.00
Single Disk x86    2048.00    614.55      Y      0.00      0.00   1331.20      0.00      0.00      0.00      0.00
W2K3R2x86-MD300    2048.00   2163.61      N      0.00      0.00      0.00      0.00      0.00      0.00      0.00
```

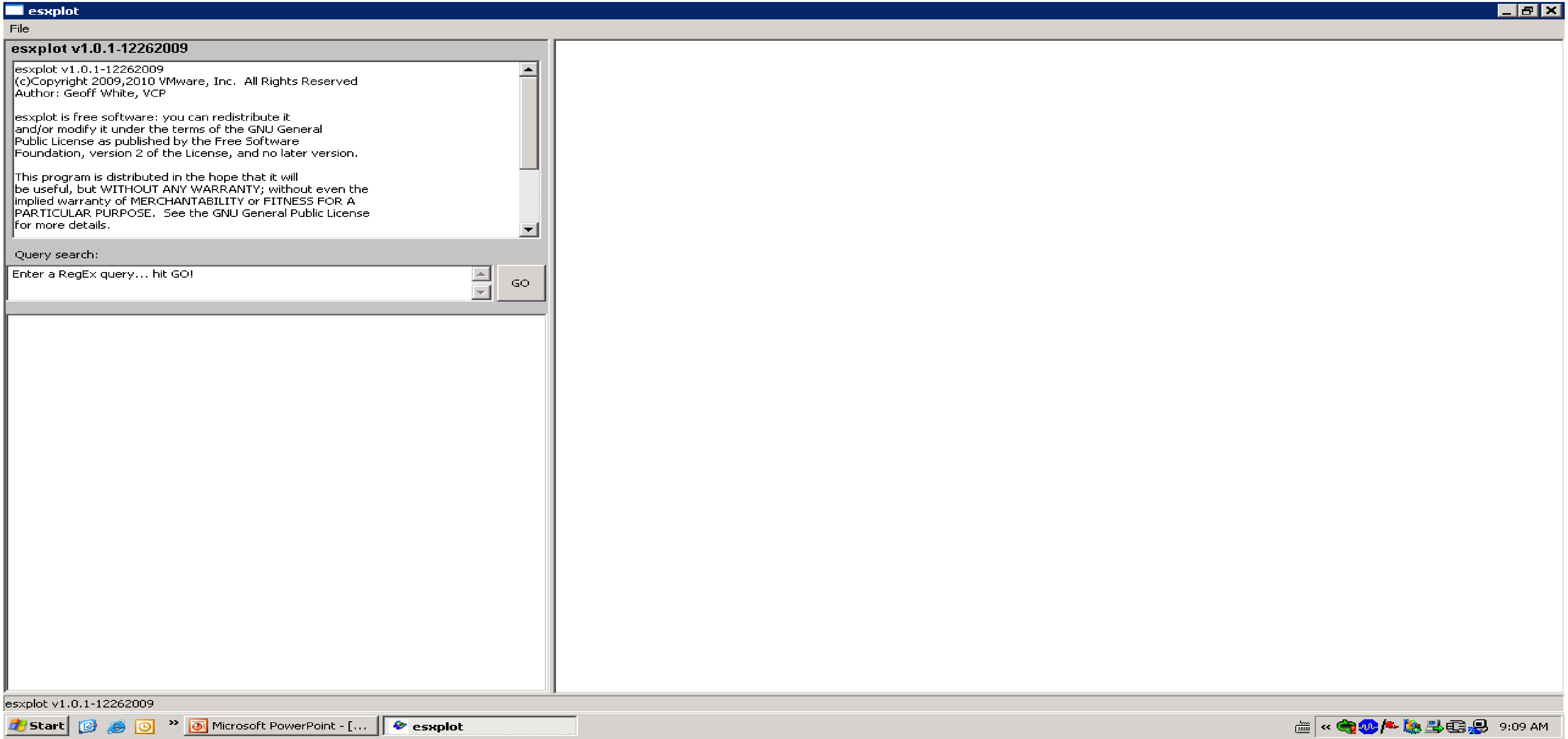
A Brief Introduction to Esxtop

- Launched at the root level of the ESX host
- Screens
  - c: cpu (default)
  - m: memory
  - n: network
  - d: disk adapter
  - u: disk device
  - v: disk VM
- Can be piped to a file and then imported in W2K System Monitor
- Horizontal and vertical screen resolution limits the number of fields and entities that could be viewed so chose your fields wisely
- Some of the rollups and counters may be confusing to the casual user

A Brief Introduction to ESXPlot

- Launched on a Windows workstation
- Imports data from a .csv file
- Allows an in-depth analysis of an esxtop batch file session
- Capture data using esxtop batch from root using SSH utility
  - Esxtop –b > exampleout.csv
- Transfer file to Windows workstation using WinSCP

ESXPlot

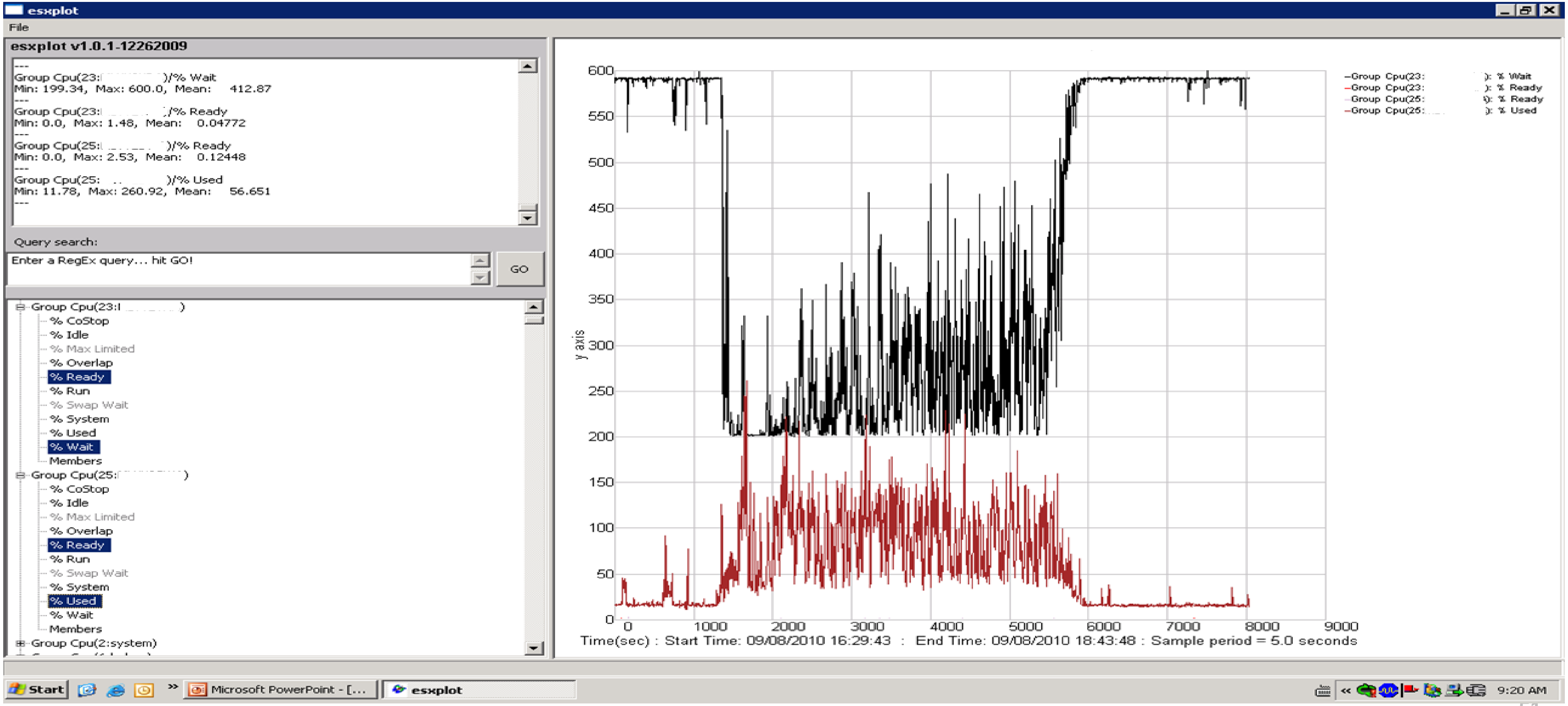


esxplot v1.0.1-12262009
(c)Copyright 2009,2010 VMware, Inc. All Rights Reserved
Author: Geoff White, VCP

esxplot is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, version 2 of the License, and no later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

ESXPlot Field Expansion: CPU



esxplot

File

esxplot v1.0.1-12262009

```
---
Group Cpu(23:          )/% Wait
Min: 199.34,  Max: 600.0,  Mean:    412.87
---
Group Cpu(23:          )/% Ready
Min: 0.0,  Max: 1.48,  Mean:   0.04772
---
Group Cpu(25:          )/% Ready
Min: 0.0,  Max: 2.53,  Mean:   0.12448
---
Group Cpu(25:          )/% Used
Min: 11.78,  Max: 260.92,  Mean:    56.651
---
```

Query search:

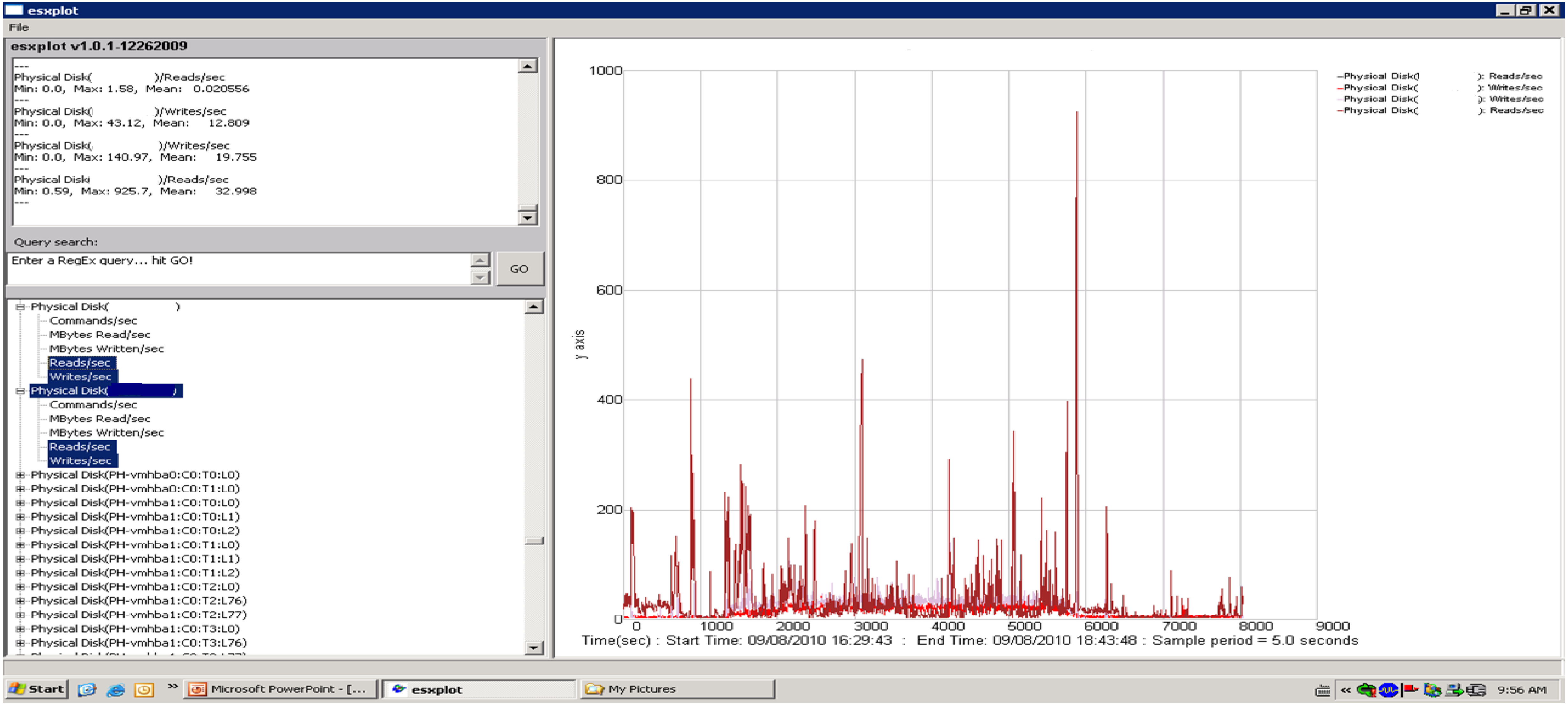
Enter a RegEx query... hit GO!

```
⊟ Group Cpu(23:          )
   % CoStop
   % Idle
   % Max Limited
   % Overlap
   % Ready
   % Run
   % Swap Wait
   % System
   % Used
   % Wait
   Members
⊟ Group Cpu(25:          )
   % CoStop
   % Idle
   % Max Limited
   % Overlap
   % Ready
   % Run
   % Swap Wait
   % System
   % Used
   % Wait
   Members
⊞ Group Cpu(2:system)
```

Time(sec) : Start Time: 09/08/2010 16:29:43 : End Time: 09/08/2010 18:43:48 : Sample period = 5.0 seconds

GO
y axis
600
550
500
450
400
350
300
250
200
150
100
50
0
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
–Group Cpu(23:   ): % Wait
–Group Cpu(23:   ): % Ready
–Group Cpu(25:   ): % Ready
–Group Cpu(25:   ): % Used
Start
Microsoft PowerPoint - [...
esxplot
9:20 AM

ESXPlot Field Expansion: Physical Disk



esxplot v1.0.1-12262009

```
---
Physical Disk(            )/Reads/sec
Min: 0.0,  Max: 1.58,  Mean:  0.020556
---
Physical Disk(            )/Writes/sec
Min: 0.0,  Max: 43.12,  Mean:   12.809
---
Physical Disk(             )/Writes/sec
Min: 0.0,  Max: 140.97,  Mean:   19.755
---
Physical Disk             )/Reads/sec
Min: 0.59,  Max: 925.7,  Mean:   32.998
---
```

Query search:

Enter a RegEx query... hit GO!

```
⊟-Physical Disk(              )
   Commands/sec
   MBytes Read/sec
   MBytes Written/sec
   Reads/sec
   Writes/sec
⊟-Physical Disk(              )
   Commands/sec
   MBytes Read/sec
   MBytes Written/sec
   Reads/sec
   Writes/sec
⊞-Physical Disk(PH-vmhba0:C0:T0:L0)
⊞-Physical Disk(PH-vmhba0:C0:T1:L0)
⊞-Physical Disk(PH-vmhba1:C0:T0:L0)
⊞-Physical Disk(PH-vmhba1:C0:T0:L1)
⊞-Physical Disk(PH-vmhba1:C0:T0:L2)
⊞-Physical Disk(PH-vmhba1:C0:T1:L0)
⊞-Physical Disk(PH-vmhba1:C0:T1:L1)
⊞-Physical Disk(PH-vmhba1:C0:T1:L2)
⊞-Physical Disk(PH-vmhba1:C0:T2:L0)
⊞-Physical Disk(PH-vmhba1:C0:T2:L76)
⊞-Physical Disk(PH-vmhba1:C0:T2:L77)
⊞-Physical Disk(PH-vmhba1:C0:T3:L0)
⊞-Physical Disk(PH-vmhba1:C0:T3:L76)
```

esxplot
File
GO
1000
800
600
400
200
0
y axis
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
Time(sec) : Start Time: 09/08/2010 16:29:43 : End Time: 09/08/2010 18:43:48 : Sample period = 5.0 seconds
Physical Disk(   ): Reads/sec
Physical Disk(   ): Writes/sec
Physical Disk(   ): Writes/sec
Physical Disk(   ): Reads/sec
Start
Microsoft PowerPoint - [...
esxplot
My Pictures
9:56 AM

Top Performance Counters to Use for Initial Problem Determination

Physical/Virtual Machine

CPU (queuing)
- Average physical CPU utilization
- Peak physical CPU utilization
- CPU Time
- Processor Queue Length

Memory (swapping)
- Average Memory Usage
- Peak Memory Usage
- Page Faults
- Page Fault Delta*

Disk (latency)
- Split IO/Sec
- Disk Read Queue Length
- Disk Write Queue Length
- Average Disk Sector Transfer Time

Network (queuing/errors)
- Total Packets/second
- Bytes Received/second
- Bytes Sent/Second
- Output queue length

ESX Host

CPU (queuing)
- PCPU%
- %SYS
- %RDY
- Average physical CPU utilization
- Peak physical CPU utilization
- Physical CPU load average

Memory (swapping)
- State (memory state)
- SWTGT (swap target)
- SWCUR (swap current)
- SWR/s (swap read/sec)
- SWW/s (swap write/sec)
- Consumed
- Active (working set)
- Swapused (instantaneous swap)
- Swapin (cumulative swap in)
- Swapout (cumulative swap out)
- VMmemctl (balloon memory)

Disk (latency, queuing)
- DiskReadLatency
- DiskWriteLatency
- CMDS/s (commands/sec)
- Bytes transferred/received/sec
- Disk bus resets
- ABRTS/s (aborts/sec)
- SPLTCMD/s (I/O split cmds/sec)

Network (queuing/errors)
- %DRPTX (packets dropped - TX)
- %DRPRX (packets dropped – RX)
- MbTX/s (mb transferred/sec – TX)
- MbRX/s (mb transferred/sec – RX)

Some of the counters previously available at the ESX Host level are now visible inside the Virtual Machine using Perfmon

Closing Thoughts

- Things HAVE changed, storage can be removed as the usual suspect for virtualization performance problems
- WHERE to look for path logjams is a key piece of the puzzle
- It is increasingly important that the operational team understands the entire storage and network path
- Don’t settle for just a dashboard to tell you if things are doing okay, ask the end user!
- Work closely with vendors, partners to better understand the details of their solutions
- Include load testing in your application plans

John Paul – johnathan.paul@cerner.com

Questions?