

# Memory Performance Analysis and Monitoring

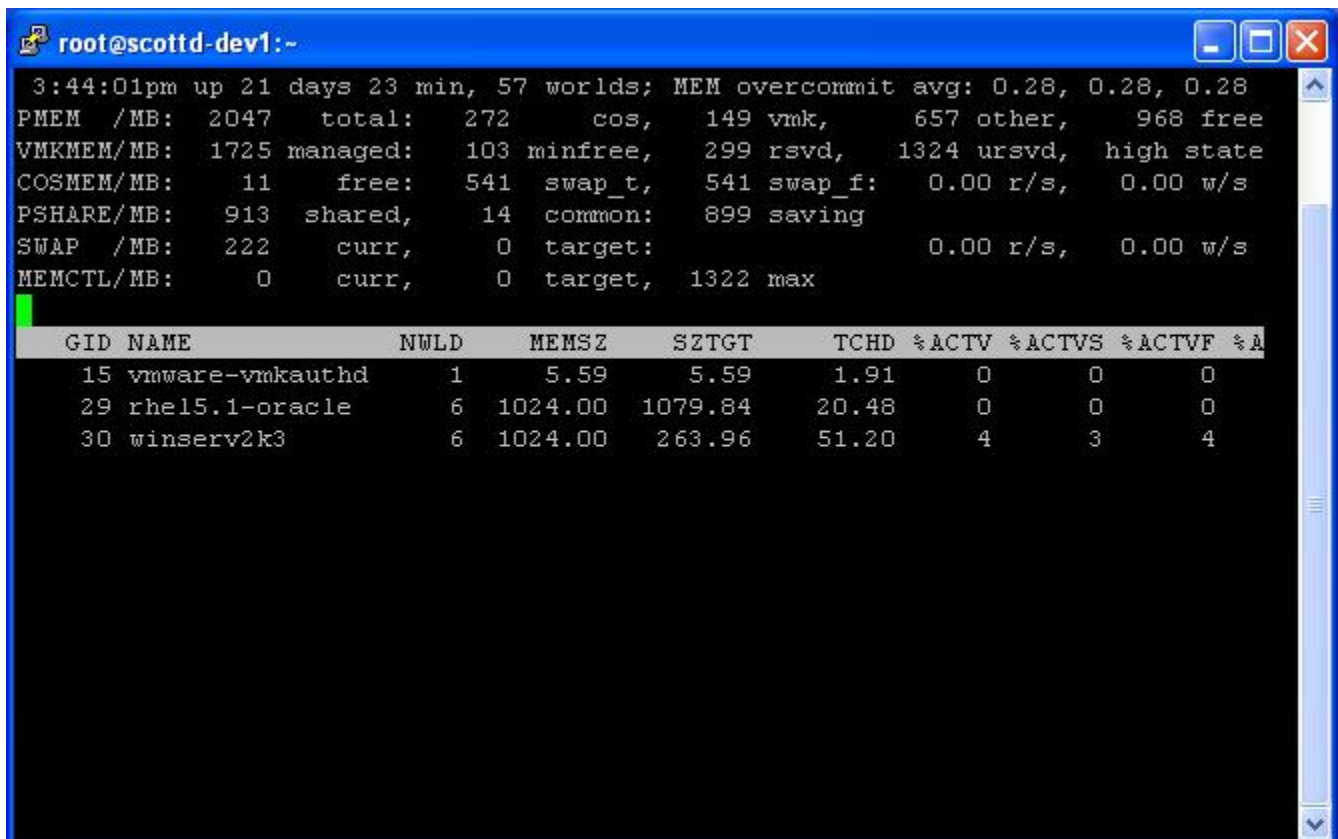
## Introduction

This document is a living, up-to-date version of the [performance analysis methods whitepaper](#).

Host memory utilization represents the entirety of memory usage due to the VM and all tasks required by ESX Server to manage and provide control of the VMs. Using ESX Server's monitoring capabilities there is no visibility into improper usage of configuration of memory within the guest. Continue to use traditional monitoring tools in the guest to identify memory-hungry applications or shortages that lead to in-guest swapping.

## Navigating esxtop

As before, bring up esxtop to inspect system specifics. Hitting the 'm' key will display the memory counters.



```
root@scott-d-dev1:-
3:44:01pm up 21 days 23 min, 57 worlds; MEM overcommit avg: 0.28, 0.28, 0.28
PMEM /MB: 2047 total: 272 cos, 149 vmk, 657 other, 968 free
VMKMEM/MB: 1725 managed: 103 minfree, 299 rsvd, 1324 ursvd, high state
COSMEM/MB: 11 free: 541 swap_t, 541 swap_f: 0.00 r/s, 0.00 w/s
PSHARE/MB: 913 shared, 14 common: 899 saving
SWAP /MB: 222 curr, 0 target: 0.00 r/s, 0.00 w/s
MEMCTL/MB: 0 curr, 0 target, 1322 max

  PID  NAME           NPLD  MEMSZ  S2TGT  TCHD  %ACTV  %ACTVS  %ACTVF  %A
  ---  ---           ---   ---    ---    ---    ---    ---    ---
    15  vmware-vmkauthd  1     5.59   5.59   1.91   0      0      0      0
    29  rhel5.1-oracle  6  1024.00  1079.84  20.48  0      0      0      0
    30  winserv2k3      6  1024.00  263.96  51.20  4      3      4      4
```

Once running, the following can be observed from the esxtop report:

- The header data contains host data that impacts all VMs running on the host. The physical memory row (PMEM) contains the total RAM installed on the system, the amount used by the console operating system (COS), the memory used by the kernel (VMK), and other statistics.
- The next few rows contains host-level memory statistics for various ESX subsystems:
  - VMKMEM: shows memory statistics for the ESX Server VMkernel
  - COSMEM: displays the memory statistics as reported by the ESX Server service console.
  - PSHARE: displays the ESX Server page-sharing statistics.
  - SWAP: displays the ESX Server swap usage statistics.
  - MEMCTL: displays the memory balloon driver statistics.

## Relevant Counters

Type	VirtualCenter	esxtop	Details
Total memory size		MEMSZ	The is the amount of memory that the VM has been sized to. The VM will never get more than this but most of the time will be using far less than this amount due to sharing, ballooning, and swapping.
Memory target		SZTGT	The amount of memory that the kernel would like to provide to the VM. This number is calculated by on the guest's memory usage. When memory is over-committed, it may not equal the amount of memory that is actually provided due to ballooning and swapping.
Granted memory	mem.granted.average		The amount of memory that has been provided to the VM. Memory is not granted to the VM until it has been touched once. In the case of Linux, which does not zero out pages upon boot, a 4G VM will only be granted the small portion (100M or so) needed to run the OS until the OS or applications start to access more.
Touched memory		TCHD	The amount of memory (in MB) that has been "touched" (read from or

			written to) in the past X minutes.
Consumed memory	mem.consumed.average		The amount of machine memory allocated to the VM. For instance, a Linux VM might have been sized to 4G. Half of the pages may not yet have been used by the OS. Perhaps 1G of this remaining 2G can be shared. That leaves a consumed memory of only 1G.
Shared memory	mem.shared.average		Shared memory represents the entire pool of shareable memory. For instance, if two VMs each have 500M of identical memory, the shared memory is 1G.
Shared common memory	mem.sharedcommon.average		Shared common memory represents the footprint in machine memory as a result of memory sharing. For instance, if two VMs each have 500M of identical memory, the shared common memory is 500M.
Active memory	mem.active.average	%ACTV, %ACTVS, %ACTVF	The amount of memory (as a percentage of the entire host's memory) that has been used by the VM in the past sample period. %ACTVS and %ACTVF are slow and fast counters showing recent and long-term averages.
Ballon driver usage	mem.vmmemctl.average	MCTLSZ	The amount of memory claimed by the balloon driver for us in other VMs.
Swap rate		SWW/s SWR/s	The rates at which memory is swapped out (written) or in (read).
Swap Totals	mem.swapout.average, mem.swapin.average		These are cumulative amounts of swapping that has occurred since the VM was powered on. It's important to check if

			swpin and swapout are increasing, rather than just seeing if they are nonzero. Because if they are non-zero, it could be the result of swapping in the past, and not swapping at the present time.
NUMA migrations		NMIG	The number of NUMA migrations that have occurred since the VM's creation.
NUMA memory		NLMEM, NRMEM	The amount of the VM's memory that is on the local and remote NUMA nodes.
Overhead	mem.overhead.average	OVHD	The amount of memory required by the VMkernel to maintain and execute the VM.

## Evaluate the Data

Memory analysis on an ESX Server means not just investigation of server-side statistics but also a solid understanding of the application that is running in the VM. When memory is short on the host, ballooning and swapping may be visible in esxtop, with swapping having a great impact on performance. When memory is short within the VM the guest will swap.

- How much memory are the VMs actually using? While they may have been allocated large amounts of memory, its likely that the OS and applications are only using a small percentage of what the VM was assigned. Check the active and touched memory counters for accurate numbers on guest memory usage.
- Is memory short in the host? Swapping (SWW/s and SWR/s) is a certain sign of this problem. Heavy use of the balloon driver may also suggest this but ballooning has a very slight impact to guest performance.
- Can memory deficiencies be addressed through VM resizing? Checking memory usage of critical apps within the VMs can help inform decisions to decrease the amount of RAM provided to those VMs. Some operating systems will expand to utilize all available memory at little or no value to the application. Reducing the memory space and correcting over-sized caches frees up memory for other VMs.
- Is the collection of all VMs' active memory (TCHD or %ACTV) sustaining at an amount that exceeds the total available memory? If so, then either more memory must be added to the host or VMs must be migrated to another DRS cluster.
- Are the guests swapping? If the VM has been sized with too little memory then the guest OS will swap inside the VM. This will appear to ESX Server as any other disk activity but should be investigated and solved with traditional OS analysis tools.
- Can NUMA migrations (NMIG) be seen on the system? NMIG reports total migrations since the VM has been powered on. If this number continues to climb then the VM is being migrated from node to node which most certainly degrades performance.
- Does the amount of memory located on a remote NUMA node (NRMEM) remain at a non-zero number? This may be a sign that the VM has been sized to exceed the memory of a single NUMA node. If the VM is using more memory than fits on a single node, some of its memory is certain to be located on a remote node. Remote memory access is quite slow relative to local memory access.

## Correct the System

The prescriptive advice for memory shortages is fairly simple: use less memory or buy more. The following recommendations are variations on this theme:

1. Verify that VMware Tools has been installed on every VM on the system and that the memory balloon driver has not been disabled. (The balloon driver is always on by default and disabled manually through text-based advanced configuration in extremely rare cases.) When provide the ability to balloon memory within the guests, ESX Server is able to take memory from VMs that are not using it and make it available to those that do need it.
2. Provide more memory to the DRS cluster. As total resources go up, VirtualCenter will balance VMs across the cluster so VMs that need the memory are able to get it.
3. Set memory reservations to minimally provide the amount of memory required of the OS and critical applications. This will allow for sustained, fast access for critical code and provide hints to VirtualCenter for optimal VM positioning across the DRS cluster.
4. Make sure the amount of memory used by the VMkernel to maintain the VMs is acceptable. This value, reported for each VM with the overhead counter (OVHD), is dependent on the memory size of the VM, the number of vCPUs provided to it, and whether or not it is executing a 64-bit OS. Fewer VMs on the host, fewer aggregate vCPUs, and lower precision OSEs (32-bit as opposed to 64-bit) will lower this number. Reducing any of these in the cluster will free up resources for every VM in the cluster.
5. Size VMs on NUMA systems to guarantee that each VM's memory will fit on a single node. This means either decreasing the memory allocated to a VM or increasing the node memory size.
6. Size guests appropriately according to their needs. For example:
  1. Depending on the access pattern of the data, databases may not benefit from the last doubling of cache size. Experiment with smaller cache sizes and see if performance drops. If not, decrease the VM's available memory so it can be used by other VMs.
  2. Check the guest OS's statistics for in-guest swapping. Provide memory as its needed and pay attention to esxstop statistics to see if the additional memory provided generates a new bottleneck in the host.

## Understanding Page Sharing

One cannot fully optimize an ESX Server's memory without understanding the performance implications of page sharing. VMware's page sharing algorithm was presented at EMC World 2008 as resulting in a 2% increase in CPU load. But the benefits of page sharing have been demonstrated to provide overcommitment of memory safely to 2X and beyond.

The value of page sharing can be seen in the following counters:

esxstop	VirtualCenter	Description
SHRD	memory.shared	The amount of memory in the VM that is sharable.
SHRDSVD	<i>No equivalent.</i>	The amount of memory saved due to page sharing.
<i>No equivalent.</i>	memory.sharedcommon	The size of the memory after redundant pages have been removed.

Note that missing counters can be calculated using the other two. Shared memory minus shared common memory equals shared savings.

## References

The top-level [Performance Monitoring and Analysis](#) paper.

The [esxtop Performance Counters](#) index.

The [Understanding VirtualCenter Performance Statistics](#) page.