

VirtualCenter Memory Statistic Definitions

Kit Colbert

October, 2008

Background Reading

The definitions below assume some basic knowledge of memory virtualization concepts. Please read the following before diving into the definitions:

1. Resource Management Guide, Memory Virtualization Basics, http://vmware.com/pdf/vi3_35/esx_3/r35u2/vi3_35_25_u2_resource_mgmt.pdf, pp 40 – 42
2. Resource Management Guide, Advanced Resource Management, http://vmware.com/pdf/vi3_35/esx_3/r35u2/vi3_35_25_u2_resource_mgmt.pdf, pp 139 – 156
3. Understanding Host and Guest Memory Usage and Related Memory Management Concepts, <http://vmworld.com/vmworld/static/sessions/2007/TA21.html>
4. Carl A. Waldspurger. "Memory Resource Management in VMware ESX Server", Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI '02), Boston, Massachusetts, December 2002, http://www.vmware.com/pdf/usenix_resource_mgmt.pdf

Some definitions below will reference topics covered in the above documents. When they do, they will refer to the documents by “[number]”, where the number is the listed above. For example, [1] would refer to Resource Management Guide, Memory Virtualization Basics.

It is important to note that some definitions refer to guest physical memory while others talk about machine memory. “Guest physical memory” is really virtual-hardware “physical” memory presented to the VM. “Machine memory” is the actual physical RAM in the ESX host. Each definition below will explicitly specify the type of memory to which it applies. See [2] pages 139 – 141 for more information about guest physical memory versus machine memory.

This distinction plays out in interesting and potentially confusing ways. The final section, “Differences between Memory Granted/Consumed and Memory Shared/Shared Common”, expands on the differences between guest physical memory and machine memory metrics.

VM metrics

Memory Active

Units: KB Internal Name: active

Amount of guest physical memory recently used by the VM, which is estimated by VMkernel statistical sampling.

See [4] section 5.3 for details on how the VMkernel estimates active memory.

Memory Balloon

Units: KB Internal Name: vmmemctl

Amount of guest physical memory that is currently reclaimed from the VM through ballooning.

This is the amount of guest physical memory that has been allocated and pinned by the balloon driver. See [3] slides 25 – 27 and [4] section 3.2 for details on how memory ballooning works.

Memory Balloon Target

Units: KB Internal Name: vmmemctltarget

Desired value for the VM's `Memory Balloon`, as determined by the VMkernel.

The VMkernel sets a target for the level of ballooned memory for each VM, based on a number of factors (see [2] page 144, *How ESX Server Hosts Allocate Memory*). If `Memory Balloon Target` is greater than `Memory Balloon`, the VMkernel will inflate the balloon, causing more VM memory to be reclaimed and `Memory Balloon` to increase. If `Memory Balloon Target` is less than `Memory Balloon`, the VMkernel will deflate the balloon, allowing the VM to map/consume additional memory if it needs it. `Memory Balloon` will decrease as the VMkernel deflates the balloon.

Memory Consumed

Units: KB Internal Name: consumed

Amount of machine memory allocated to the VM, accounting for savings from memory sharing with other VMs. `Memory Overhead` is not included.

When multiple VMs are sharing a single region of machine memory, each VM is “charged” for the memory proportionally based on the total references to that region of shared memory. For instance, if a VM has 100 MB of memory that is shared equally with three other VMs, then that VM's portion of the shared memory is only 25 MB (100 MB / 4 VMs) for the `Memory Consumed` statistic.

Because `Memory Consumed` measures a VM's use of machine memory, it should be used for issues such as chargeback.

See the “Differences between `Memory Granted/Shared` and `Memory Shared/Shared Common`” section for more information on this statistic.

Memory Granted

Units: KB Internal Name: granted

Amount of guest physical memory currently mapped to (i.e., backed by) machine memory. `Memory Overhead` is not included, but `Memory Shared` is.

Note that this is *not* the amount of machine memory used by the VM; `Memory Consumed` measures a VM's machine memory usage.

See the “Differences between `Memory Granted/Shared` and `Memory Shared/Shared Common`” section for more information on this statistic.

Memory Overhead

Units: KB Internal Name: overhead

Extra machine memory used by the VMkernel to run the VM. For more information on memory overhead, see [2], page 142.

Since `Memory Overhead` always resides in machine memory, the VMkernel must reserve memory for the VM's overhead memory. Thus a VM's memory reservation has two components: a user-specified memory reservation and the overhead memory reservation (i.e. `Memory Overhead`). For example, if the user specified a 1 GB reservation and the `Memory Overhead` for the VM is 100 MB, then the VM's memory reservation when powered on would be 1.1 GB.

Memory Shared

Units: KB Internal Name: shared

Amount of guest physical memory shared through transparent page sharing. It includes `Memory Zero`.

Transparent page sharing is a method of RAM de-duplication. See [4] section 4 for details on how transparent page sharing works.

Memory Swap In

Units: KB Internal Name: swapin

Total amount of memory data that has been read in from the VM's swap file to machine memory by the VMkernel since the VM powered on. This statistic refers to VMkernel swapping and not to guest OS swapping.

Memory Swap Out

Units: KB Internal Name: swapout

Total amount of memory data that has been written out to the VM's swap file from machine memory by the VMkernel since the VM powered on. This statistic refers to VMkernel swapping and not to guest OS swapping.

Memory Swap Target

Units: KB Internal Name: swaptarget

Desired value for the VM's `Memory Swap`, as determined by the VMkernel.

The VMkernel sets a target for the level of swapping for each VM, based on a number of factors (see [2], page 144, How ESX Server Hosts Allocate Memory). If `Memory Swap Target` is greater than `Memory Swap`, then the VMkernel will start swapping, causing more VM memory to be swapped out. This will generally happen quickly. If `Memory Swap Target` is less than `Memory Swap`, then the VMkernel will stop swapping. Since swapped memory stays swapped until the VM accesses it, `Memory Swapped` can be greater than `Memory Swap Target`, possibly for a prolonged period of time. This simply means that the swapped memory is not currently needed by the VM and is not a cause for concern.

Memory Swapped

Units: KB Internal Name: swapped

Current amount of guest physical memory swapped out to the VM's swap file by the VMkernel. This statistic refers to VMkernel swapping and not to guest OS swapping.

Memory Usage

Units: Percent Internal Name: usage

`Memory Active` / VM memory configured size.

Memory Zero

Units: KB Internal Name: zero

Amount of guest physical "zero memory" that is shared through transparent page sharing.

"Zero memory" is simply memory that is all zeroes. This statistic is included in `Memory Shared`. Transparent page sharing is a method of RAM de-duplication. See [4] section 4 for details on how transparent page sharing works.

Host metrics

Memory Active

Units: KB Internal Name: active

Sum of `Memory Active` of all powered on VMs and VMware Infrastructure services on the host.

Memory Balloon

Units: KB Internal Name: vmmemctl

Sum of `Memory Balloon` of all powered on VMs and VMware Infrastructure services on the host.

Memory Consumed

Units: KB Internal Name: consumed

Amount of machine memory used on the host. It includes Service Console memory, VMkernel memory, VMware Infrastructure services memory and VM memory.

Memory Granted

Units: KB Internal Name: granted

Sum of `Memory Granted` of all powered on VMs and VMware Infrastructure services on the host.

Since each VM's `Memory Granted` measures its guest physical memory, the host's `Memory Granted` may be larger than the amount of machine memory if memory is overcommitted (i.e. aggregate VM configured memory is greater than machine memory).

Memory Heap

Units: KB Internal Name: heap

Amount of VMkernel virtual address space dedicated to VMkernel main heap and related data.

`Memory Heap` is hardcoded at 30 MB for ESX 3.5U2. This space is used by the VMkernel for some of its memory plus memory for modules and device drivers. Users should not worry about this statistic.

Memory Heap Free

Units: KB Internal Name: heapfree

Amount of free machine memory on the VMkernel's main heap.

Memory Heap Free will vary depending on the amount of physical devices and various configuration options. There is no direct way for the user to increase or decrease this statistic. Users should not worry about this statistic.

Memory Overhead

Units: KB Internal Name: overhead

Sum of Memory Overhead of all powered on VMs and VMware Infrastructure services on the host.

Memory Reserved Capacity

Units: MB Internal Name: reservedCapacity

Total amount of memory reservation used by powered on VMs and VMware Infrastructure services on the host, which includes Memory Overhead. Since this statistic includes Memory Overhead, it will be greater than zero even when no powered on VM has a user-specified reservation (due to the fact that all powered on VMs require some overhead memory).

Memory Shared

Units: KB Internal Name: shared

Sum of Memory Shared of all powered on VMs and VMware Infrastructure services on the host.

Since each VM's Memory Shared measures guest physical memory, the host's Memory Shared may be larger than the amount of machine memory if memory is highly overcommitted (i.e. aggregate VM configured memory is much greater than machine memory). This illustrates how effective transparent page sharing and memory overcommitment are for saving machine memory.

See the "Differences between Memory Granted/Shared and Memory Shared/Shared Common" section for more information on this statistic.

Memory Shared Common

Units: KB Internal Name: sharedcommon

Amount of machine memory that is shared by powered on VMs and VMware Infrastructure services on the host.

Memory Shared - Memory Shared Common = memory savings for the host from sharing.

See the "Differences between Memory Granted/Shared and Memory Shared/Shared Common" section for more information on this statistic.

Memory State

Units: Number Internal Name: state

State (0 - 3) corresponding to amount of free memory on the host.

The VMkernel has four thresholds for the amount of free machine memory on the host: "high" (0), "soft" (1), "hard" (2) and "low" (3). As of ESX 3.5U2, the memory state is high if the free memory is greater than or equal to 6% of machine memory minus Service Console memory, it's soft at 4%, hard at 2% and low at 1%.

While the host's memory state is not used to determine whether memory should be reclaimed from VMs (that decision is made at the resource pool level), it can affect what mechanisms are used to reclaim memory if memory needs to be reclaimed. In the high and soft states, ballooning is favored over swapping. In the hard and low states, swapping is favored over ballooning.

This means that a VM's `Memory Ballooned`, `Memory Balloon Target`, `Memory Swapped` and `Memory Swap Target` may be greater than zero in any memory state, as reclamation is driven by resource pool constraints rather than host constraints.

Memory Swap In

Units: KB Internal Name: swapin

Sum of `Memory Swap In` of all powered on VMs on the host.

Memory Swap Out

Units: KB Internal Name: swapout

Sum of `Memory Swap Out` of all powered on VMs on the host.

Memory Swap Used

Units: KB Internal Name: swapused

Sum of `Memory Swapped` of all powered on VMs and VMware Infrastructure services on the host.

Memory Unreserved

Units: KB Internal Name: unreserved

Memory reservation not used by the Service Console, VMkernel, VMware Infrastructure services and other powered on VMs' user-specified memory reservations and overhead memory.

This statistic is no longer relevant to VM admission control, as reservations are now handled through resource pools.

Memory Usage

Units: Percent Internal Name: usage

Memory Consumed / size of machine memory.

Memory Used by vmkernel

Units: KB Internal Name: sysUsage

Amount of machine memory used by the VMkernel for "core" functionality (such as its own internal uses, device drivers, etc). It does not include memory used by VMs or by VMware Infrastructure services.

Memory Zero

Units: KB Internal Name: zero

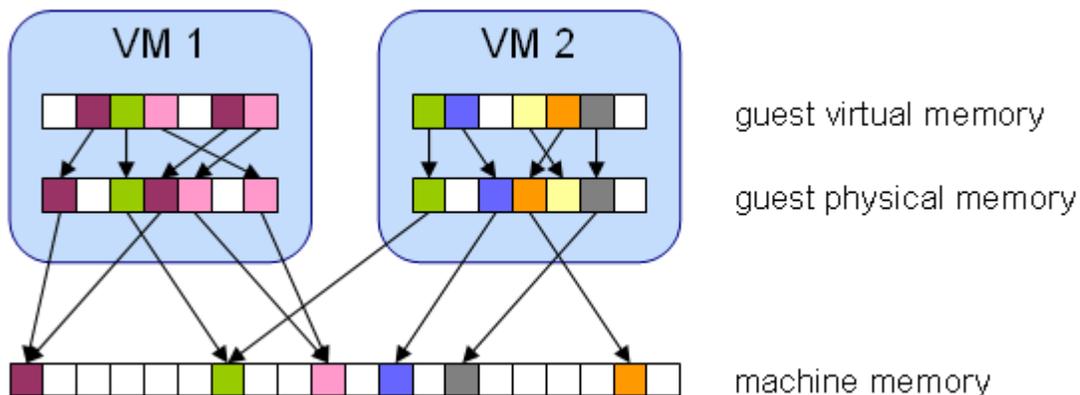
Sum of Memory Zero of all powered on VMs and VMware Infrastructure services on the host.

Difference between Memory Granted/Consumed and Memory Shared/Shared Common

A VM's Memory Granted and Memory Consumed differ in a similar way to how a host's Memory Shared and Memory Shared Common differ: Memory Granted and Memory Shared measure guest physical memory while Memory Consumed and Memory Shared Common measure machine memory.

The VMkernel maps guest physical memory to machine memory. But the VMkernel has the option of either not mapping specific regions of guest physical memory to machine memory (say in the case that the VMkernel swaps out or balloons some guest physical memory) or mapping multiple regions of guest physical memory to the same region of machine memory (in the case of memory sharing). Thus guest physical memory and machine memory are not always mapped one-to-one and differences will arise between calculations of guest physical memory usage and machine memory usage for an individual VM or an ESX system as a whole. Because of this, statistics that measure guest physical memory will differ from those that measure machine memory; this is expected and OK.

An example clearly demonstrates these differences. Consider the following figure, in which two VMs are running on an ESX host, where each block represents 4 KB of memory and each color represents a different set of data on a block:



Let's say we want to find Memory Granted and Memory Consumed for VM 1. Looking at our definitions in the previous section, we see that Memory Granted is the amount of guest physical memory that is mapped to machine memory. Looking at our figure, we simply need to count the number of blocks in VM 1's guest physical memory that have arrows to machine memory and multiply by 4 KB. Since there are five blocks with arrows, Memory Granted would be 20 KB.

As defined above, Memory Consumed is the amount of machine memory allocated to the VM, accounting for savings from shared memory. This means that we first need to count the number of blocks in machine memory that have arrows from VM 1's guest

physical memory. While there are three such blocks, one block is shared with VM 2. So we count two full blocks plus half of a third, multiply by 4 KB and get 10 KB for Memory Consumed.

The important difference between Memory Granted and Memory Consumed is that Memory Granted counts the number of blocks with arrows at the guest physical memory level and Memory Consumed counts the number of blocks with arrows at the machine memory level. And due to memory sharing, the number of blocks with arrows at each level is different, thus making Memory Granted and Memory Consumed different.

If we try to determine Memory Shared and Memory Shared Common for the host, we see the same phenomenon playing out. As defined above, Memory Shared for the host is simply the addition of each VM's Memory Shared. We can calculate this by simply looking at each VM's guest physical memory and counting the number of blocks that have arrows to machine memory blocks that themselves have more than one arrow pointing at them. We see that there are six such blocks, so Memory Shared for the host is 24 KB.

Memory Shared Common is the amount of machine memory that is shared by VMs. So to determine Memory Shared Common for our figure, we simply look at the machine memory and count the number of blocks that have more than one arrow pointing at them. We see there are three such blocks, meaning that Memory Shared Common is 12 KB.

Again we see that Memory Shared is concerned with guest physical memory while Memory Shared Common deals with machine memory. Another way to think of it is that Memory Shared looks at the origin of the arrows while Memory Shared Common looks at the destination of the arrows. In any case, understanding the conceptual difference is important in knowing what these statistics are measuring and how to interpret them.