

# Memory Provisioning Recommendations for VMware Infrastructure 3

Operational Best Practices



## Contents

Overview .....	3
Prerequisite Knowledge .....	3
Introduction .....	3
VMware Infrastructure 3 Architecture Overview .....	3
The x86 Memory Model .....	4
Using Resource Analysis Tools to Determine Memory Bottlenecks .....	5
Memory Management in ESX Server 3 .....	6
VMkernel Memory Management Techniques .....	6
<i>Transparent Page Sharing</i> .....	7
<i>Memory Ballooning</i> .....	7
<i>Swap File Usage</i> .....	7
Memory and VMware Infrastructure Performance .....	8
Memory Reservations, Limits, and Shares .....	8
Effect of DRS on Transparent Page Sharing .....	9
Memory Headroom and VMware HA .....	9
<i>Planning and Evaluating ESX Server Memory Requirements</i> .....	10
VMkernel Memory Counters Explained.....	10
Tracking ESX Server Memory Usage .....	11
Memory RAID and VI3 .....	13
Best Practice Recommendations.....	13
Summary.....	14
About the Authors.....	15

## Overview

Physical memory is a critical component of all VMware Infrastructure 3 (VI3) deployments, and appropriate sizing is necessary to ensure the success of solutions based on VI3, such as Server Consolidation. Provisioning the ideal amount of physical memory for VI3 components is more important as systems with an ever increasing number of CPU cores and processing threads rise in popularity. This paper defines the planning process for determining the appropriate amount of memory for a VI3 deployment, dispels some common misconceptions, and describes the consequences of sub-optimal sizing.

## Prerequisite Knowledge

---

- X86 System architecture
- Memory management techniques for X86 based systems
- Intermediate knowledge of VMware ESX Server architecture

## Introduction

Memory resources are a critical component in the overall capabilities and functionality of x86 systems. The memory subsystem of x86 systems is designed to make memory resources available efficiently among threads and processes. It provides a complete address space for each process, protected from all other processes; it enables program size to be larger than physical memory, and allows efficient sharing of memory between processes.

The efficiency with which the memory subsystem performs its tasks has great implications upon the performance of the overall x86 system. A VMware Infrastructure 3 environment is comprised of many individual memory subsystems, one per virtual machine, and so a comprehensive memory provisioning and management strategy can be pivotal to the success of the virtualized datacenter. For many workloads, memory is the limiting factor, and effective memory management enables more virtual machines to share a single server, increasing ROI for consolidation. Furthermore, advances in virtualization, CPU, and memory technologies make the addition of memory one of the most effective investments for maximizing the utilization of an ESX Server host.

## VMware Infrastructure 3 Architecture Overview

A brief architectural overview of VMware Infrastructure 3 (VI3) is a prerequisite to understanding some of the more technical aspects of this paper. VMware ESX Server 3 and VirtualCenter 2 constitute the VI3 platform's major software components. VMware ESX Server is a robust, proven virtualization platform that abstracts processor, memory, storage and networking resources into multiple virtual machines; VirtualCenter is VMware's enterprise-ready VI3 management platform.

VMware ESX Server consists of three major components:

- **The Virtualization Layer** (or VMkernel) – Kernel software designed by VMware to run virtual machines, it controls the hardware resources utilized by ESX Server hosts and schedules their allocation among the virtual machines.
- **Virtual Machines** – The containers in which applications and guest operating systems run. All VMware virtual machines are isolated from one another by design. Virtual machine isolation is transparent to the guest operating system.
- **The Virtual Networking Layer** – The virtual network devices through which virtual machines and the service console interface with the physical network.

The ESX Server management interface, or *Service Console*, is a customized, enhanced-security version of Red Hat Enterprise Linux 3. The Service Console is a specialized virtual machine used to interface to and manage the VMkernel, not to be confused with the base operating system of hosted virtualization products such as VMware Server, which run on Microsoft Windows or Linux operating systems.

## The x86 Memory Model

Understanding the x86 memory model is pertinent knowledge to developing a memory provisioning strategy that leverages the advanced memory management features of VI3 while preserving or improving upon the performance of x86 workloads hosted within VI3. The x86 Virtual Memory Model allows 32-bit Intel-based architectures and operating systems to leverage their own version of memory over-commitment by providing an exclusive virtual memory space to each application, which is mapped to physical memory pages through the use of an in-memory page table map.

This model is fairly consistent across x86 platforms. Every running process receives its own 4GB virtual memory allocation, which is divided into two components; the lower portion of which is a private address space reserved for the process exclusively, and the upper portion of which is leveraged for shared and operating system components. In 32-bit implementations of Microsoft Windows, this split is 50/50, with 2GB allocated to private virtual memory and 2GB allocated to shared/kernel components, although Enterprise editions of Windows include the ability to split the virtual memory space 3:1 in favor of more process-private virtual memory. Most Linux distributions also share this 3:1 split.

Each process running on x86 platforms is comprised of a set of code, a virtual memory allocation and one or more threads of execution. A process' working set is a subset of its virtual memory allocation and is defined as the amount of memory from the process' virtual memory space that is permitted to be in physical RAM at any one time. The defined working set of a process is determined by several factors, including process memory demands and available system memory, and may fluctuate based on needs. Operating system components, reserved memory and file system cache also consume memory.

As memory demands rise on a machine, be it physical or virtual, there is an increased likelihood that information may need to be swapped to a paging file on disk if sufficient free memory space is not available. The Windows pagefile is

leveraged as a way to extend the amount of available physical RAM. Unfortunately, the use of this expanded memory space comes at the expense of performance. Although some pagefile usage is almost always present in Windows-based systems, excessive paging will cause immediate and noticeable performance degradation. In addition, as available memory levels drop, file system and network performance also suffer as the system cache is reduced to make room for process memory. Finally, paging also comes with the added effect of increased CPU overhead to manage the Virtual Memory Manager's processes and paging activities.

Paging activity takes an additional performance hit in the virtualization layer of ESX as the guest operating system's paging activity must traverse the VMkernel to gain access to the physical disk.

Memory behavior from the perspective of a guest OS within a virtual machine is no different from that of a physical machine, so strategies that help with the detection and elimination of memory bottlenecks in physical systems also apply to virtual machines. In fact, it is by optimizing the memory usage of virtual machines and creating complimentary workload profiles within ESX Server clusters that the greatest consolidation factors are achieved.

## Using Resource Analysis Tools to Determine Memory Bottlenecks

---

Utilities available within Windows and Linux Operating Systems (such as *Windows Perfmon* and *Linux top*) are useful for gathering both real-time and historical performance and resource usage data. These utilities are more appropriate for real-time or limited historical trend analysis of resource utilization than for long-term or in-depth examination as there are several third-party utilities available to more effectively handle those tasks.

Two such utilities are particularly useful for capturing resource utilization statistics, through which complex "what-if" scenarios for consolidation planning can be created. The first is VMware's Capacity Planner, which is available only by engaging either with VMware Professional Services or a VMware Accredited Consulting (VAC) partner<sup>1</sup>. Another popular third-party tool is PlateSpin's PowerRecon, available for purchase directly from PlateSpin or through one of its strategic partners<sup>2</sup>. Both products use an agent-less technology to periodically gather resource utilization data. PowerRecon leverages a locally installed Microsoft SQL Server database to store the information, while VMware Capacity Planner uploads collected data into a centralized and secure information data warehouse. Both products provide sophisticated planning and scenario-building capabilities.

Both tools are very valuable for determining actual memory usage and utilization trends and can be leveraged to both identify memory (and other resource usage) bottlenecks, as well as creating consolidation scenarios to determine the optimal physical memory footprint for ESX Server hosts.

There are key performance counters that can be leveraged to determine the memory demands of a physical or virtual Windows-based machine. These counters can be observed and analyzed in myriad ways, including Windows Performance Monitor and third-party products such as VMware Capacity Planner and PlateSpin PowerRecon (described previously). It is important to emphasize that a guest operating system within a virtual machine can only access memory that has been provisioned to it, regardless of the actual amount of physical memory in the ESX

---

<sup>1</sup> For more information visit [www.vmware.com](http://www.vmware.com)

<sup>2</sup> For more information visit [www.platespin.com](http://www.platespin.com)

Server host. The following key performance counters relate to discovering memory bottlenecks in Windows operating systems:

- **Memory: Available bytes** – This counter tracks the amount of memory that is available to allocate to processes. Microsoft recommends maintaining 10% of total physical RAM as unallocated.
- **Memory: Page Faults/sec** – This counter contains both hard and soft page faults. A soft page fault is when a requested page is not contained in a process's working set but is found elsewhere in memory. A hard page fault occurs when the page in question must be retrieved from disk. Hard page faults indicate pagefile activity. If requested pages are consistently not found, it is an indication that the process' working set is too small, most likely due to insufficient available memory.
- **Memory: Pages Input/sec** – This indicates the number of pages being retrieved from disk to satisfy page faults. This counter should be compared to Page Faults/sec to determine how many faults are satisfied from disk vs. from elsewhere in RAM.
- **Memory: Pages Output/sec** – This indicates the number of pages being written to disk. When a process changes the contents of a page, the update must be written to disk, but are typically held in a list in memory and written to disk in batches for efficiency. When Pages Output/sec is high, it indicates that most page faults are data pages and memory has become scarce.
- **Memory: Pages/sec** – This counter is simply the sum of Pages Input/sec and Pages Output/sec.
- **Memory: Page Reads/sec** – This counter indicates how often the pagefile must be read to satisfy a page fault. A higher Pages/sec reading indicates that more than one page is being read at a time. According to Microsoft, a value of 5 or higher is a direct indication of a memory shortage; values in excess of 10 indicate severe paging activity.
- **Memory: Pages Writes/sec** – This counter indicates how often the pagefile must be written to and is also a direct indication of a memory shortage. A higher Pages/sec reading indicates that more than one page is being written at a time.

Of the counters listed above, non-zero values in **Page Faults/sec**, **Pages Input/sec** and **Page Reads/sec** are a clear indication of excessive pagefile activity and memory constraints.

The process of identifying and resolving memory bottlenecks is often time consuming and requires thorough analysis. The results are often dramatic and satisfying.

## Memory Management in ESX Server 3

Memory management in ESX Server 3 falls into two categories; memory that is managed by the VMkernel, and memory that is configured administratively. The VMkernel maintains three separate robust memory management mechanisms to govern memory allocation and usage, which include Transparent Page Sharing, Memory Ballooning, and a per-virtual machine swap file. Administrator-managed features include memory reservations, memory limits, and share allocations.

---

## VMkernel Memory Management Techniques

---

The VMkernel maintains three distinct mechanisms for governing memory management in ESX server:

- Transparent Page Sharing
- Memory Ballooning
- Swap File Usage

Please note that this section is not intended to be a thorough explanation of these features; for more information, please refer to the VMware white paper entitled “The Role of Memory in ESX Server 3”:

### Transparent Page Sharing

Transparent Page Sharing is a process executed frequently by the VMkernel that scans physical memory to identify redundant virtual machine memory pages. When a match is found, the physical page is marked read-only. The VMkernel removes the duplicate page from physical RAM and the page table is adjusted to redirect the virtual machine’s virtual page back to the read-only page in physical RAM. This mechanism results in increasing memory that is shared among running virtual machines over time. The savings from Transparent Page Sharing is attributed to a combination of duplicate virtual machine memory contents and zero-byte memory pages (memory granted to a virtual machine which is not currently being used by the guest operating system).

### Memory Ballooning

Memory ballooning is handled through a driver (*vmmemctl.sys*) that is installed as part of the VMware Tools. This driver is loaded in the guest OS to interact with the VMkernel and is leveraged to reclaim memory pages when ESX memory resources are in demand and available physical pages cannot meet requirements. When memory demands rise on the ESX host, the VMkernel will instruct the balloon driver to “inflate” and consume memory in the running guest OS, forcing the guest operating system to leverage its own native memory management techniques to handle changing conditions. Free pages are typically released first, but the guest OS may decide to page some memory out to its pagefile on the virtual disk. The reclaimed memory is then used by ESX to satisfy memory demands of other running workloads, but will be relinquished back to the guest OS when memory demands decrease by “deflating” the balloon driver. Balloon driver activity can be viewed either through VirtualCenter performance monitoring graphs or ESXTOP on the local host.

### Swap File Usage

The third mechanism leveraged is a unique swap file created for each virtual machine. By default, this swap file is maintained in the same location used to store all files for the relevant virtual machine and is equal in size to the amount of memory configured in the .vmx file, minus any memory reservation configured specifically for the virtual machine. By default the swap file is created at the time the virtual machine is powered on and automatically deleted when powered off; the required disk space must be available on the relevant datastore to successfully power on the VM. As ESX Server host memory demands rise and physical resources become scarce and when the balloon drive is temporarily unavailable to reclaim memory quickly enough, the VMkernel will move non-active memory pages out of physical RAM to the per-VM swap file. Because of the severe overhead involved in managing swapped pages,

swapping is typically reserved as a last-case mechanism for recovering memory, usually only leveraged once all other memory conservation avenues have been exhausted.

Swap file usage can be viewed in both ESXTOP within the Service Console or in VirtualCenter's performance monitoring graphs.

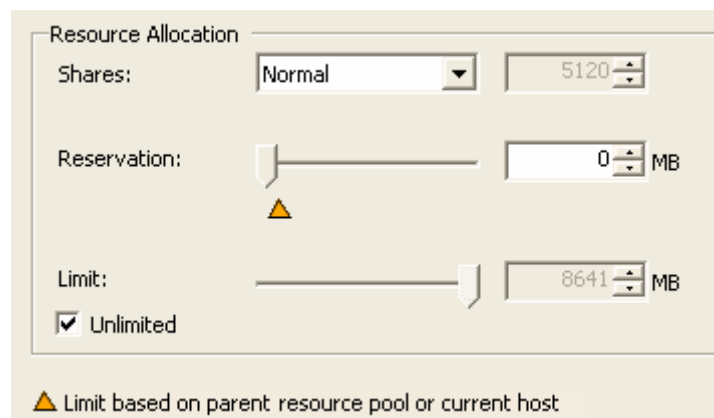
## Memory and VMware Infrastructure Performance

Memory is a key performance enabler for x86 servers and workloads, and the right mix of provisioning and tuning of memory resources in ESX Servers, as well as the hosted virtual machines, can unlock the greater potential of VMware Infrastructure 3. Therefore, it is imperative to understand the relationship between the memory usage of virtual machines and the memory management features of ESX Server.

### Memory Reservations, Limits, and Shares

Memory Reservations are an ESX Server memory management technique used to control how memory is allocated from an either an explicit or implicit resource pool by the VMkernel to a virtual machine or group of virtual machines.

Reservations consist of two settings – a reservation, or guaranteed amount of physical memory that will always be available to the virtual machine(s), and a limit, which is the absolute maximum physical memory a virtual machine or group of virtual machines can consume on the host (see Figure 1 - Memory Reservation Options). Each can be configured independently of the other, so some configurations may only employ a reservation with no limit while others will only set a limit with no reserved amount. In addition, both settings can coexist on the same object to control both the guaranteed minimum amount of RAM and the overall allocation limit.



**Figure 1 - Memory Reservation Options**

Memory reservations and limits are useful memory management features. Placing a memory limit on a resource pool allows individual virtual machines within the pool to be liberally allocated RAM while maintaining an absolute cap on memory that is consumed by the group. This strategy is useful in cases where VM memory consumption varies individually but it must be controlled at the group or departmental level. Memory reservations can ensure that a



certain allocated amount of physical memory is always available to a virtual machine or resource pool, even when virtual machines migrate from one physical host to another. By establishing a reservation, the physical memory necessary to ensure expected performance is always available.

The *Shares* allocation is used to identify preferential treatment when memory resources are under constraint on an ESX host. Resource *shares* are based on a proportional allocation system, where a virtual machine's "due share" is a ratio based on its shares compared to the total shares allocated to all objects in the respective group. When two virtual machines have the same amount of memory provisioned to them and are in the same resource pool, the virtual machine with the greater number of shares will enjoy preferential access to physical memory on a proportional basis equal to the differences in the share allocations.

## **Effect of DRS on Transparent Page Sharing**

---

When a virtual machine is relocated to another physical ESX Server host through VMotion, either administratively or as a result of DRS optimization activity, the virtual machine's memory contents are duplicated on the new host and any existing optimizations realized through Transparent Page Sharing on the original host are lost. Transparent Page Sharing activity on the new host will begin identifying optimization opportunities immediately, though this process requires time to produce the greatest amount of optimization.

DRS will take into consideration the amount of free memory in potential target ESX Server hosts with respect to the virtual machine's memory utilization and demands prior to performing or recommending a migration. In a DRS cluster, physical memory headroom in the ESX hosts will allow more freedom for virtual machine migrations, and for DRS activity to occur while maintaining optimal performance.

## **Memory Headroom and VMware HA**

---

VMware HA provides easy-to-use and cost-effective high availability for any applications running in virtual machines. In the event of server failure, affected virtual machines are automatically restarted on other ESX hosts in the VMware HA cluster that have spare capacity. When planning to leverage VMware HA in scenarios where availability and virtual machine performance are weighed equally, a certain amount of memory headroom is required on each ESX Server in the HA cluster to accommodate powering on virtual machines in the event of an HA failover occurrence, though a V13 administrator has the ability to configure which virtual machines should start automatically during an outage. Although VMware HA will preemptively display a warning message indicating that a cluster is 'inconsistent' when resources have been over-committed, meaning there are not enough available resources to start virtual machines in the event of an ESX Server failure, memory requirements are not considered in this consistency check unless memory reservations are established. If VMware HA is of the utmost importance to the V13 platform implementation, ample memory on all servers is a critical component.

Memory Over-Commitment

Because of the advanced memory management features in the VMkernel, VMware ESX Server has the capability of over-committing RAM. In other words, the sum total of memory assigned to virtual machines can exceed what is actually available in the ESX Server host. For instance, if an ESX host has 32 GB of RAM installed, it is possible to create 40 virtual machines with 1-GB of virtual memory allocated to each of them.

This strategy is most effective when there are many duplicate or idle pages in physical memory, such as when many virtual machines of the same type guest operating system are hosted on the same ESX Server. This strategy can backfire when there is a large quantity of unique pages in physical memory, resulting in contention for physical memory space. As such, the following is true for memory over-commitment in ESX Server:

- Over-committing RAM is most effective when combining workloads that leverage the same resources at different times within an ESX Server host. This method allows for optimal usage of otherwise idle resources.
- Over-committing RAM increases the likelihood of memory constraint conditions when memory demands on the ESX Server host suddenly rise, such as DRS or administrative VMotion activity.
- Over-committing RAM in VMware HA clusters is most effective when there is sufficient memory headroom on the remaining ESX Server hosts in the cluster to accommodate restarting the virtual machines from the failed ESX Server host. Restart priority can provide time needed by the ESX Servers to rebalance resource demands, and a careful strategy will payoff in a crisis.

A common practice among V13 administrators is to over-allocate RAM to virtual machines with the expectation that transparent page sharing will recover the unused pages (zero memory pages), thereby resulting in efficient over-commitment of ESX Server host physical memory. However, RAM that is provisioned to a virtual machine is considered available for use by that virtual machine and may produce unexpected results if memory demands within the guest operating system suddenly change. Planning for memory over-commitment then requires knowledge about the memory demands of the x86 workloads and managing these demands in a complimentary way among other virtual machines within the boundary of an ESX Server host. This can be accomplished very easily with DRS and the conscientious use of affinity and anti-affinity rules to group complementary virtual machines on the same host and to separate virtual machines with mismatched workload demands.

Overall, ESX Server host RAM should be provisioned to accommodate the anticipated collective workload, taking into account projected memory savings through Transparent Page Sharing, HA failover capacity needs, future capacity growth and administratively defined reservations and limits.

### **Planning and Evaluating ESX Server Memory Requirements**

The VMkernel virtualization layer itself consumes very little memory. The most important consideration for the optimal amount of RAM to provision in an ESX Server host is the memory requirements of the virtual machines to be hosted by the ESX Server. Memory demands are driven at the virtual machine level, and so any attempt to tune or optimize ESX Server memory before addressing virtual machine memory bottlenecks may achieve positive results only temporarily.

Identifying and addressing memory bottlenecks at the virtual machine level is the first task of a two-part optimization process. Detecting and resolving ESX Server memory bottlenecks is the second part of the process.

---

## VMkernel Memory Counters Explained

---

This section describes the most useful (but not all) of the available memory performance counters in VirtualCenter.

- **Memory Usage %** – This metric identifies the memory used on the host as a percentage, based on Memory Consumed divided by the total memory in the ESX host.
- **Memory Granted** – This is the amount of memory that the VMkernel has allocated to all virtual machines running on the server. It's based on the virtual machine's VMkernel working set (amount of active RAM), virtualization overhead and expected needs. The VMkernel will always attempt to grant the full amount of RAM provisioned to a virtual machine when there is sufficient available physical RAM in the host.
- **Memory Consumed** – This is the amount of memory consumed to meet virtual machine demands and includes any memory savings achieved by Transparent Page Sharing, as well as memory in use by the Service Console and the VMkernel itself.
- **Memory Shared** – The total net savings of ESX Server host physical memory as a result of Transparent Page Sharing, including any zero-byte pages
- **Memory Shared Common** – The amount of RAM consumed by unique shared memory pages, identified as a result of Transparent Page Sharing. These pages are the single instance read-only copy, to which the VMkernel refers redundant memory pages.
- **Memory Balloon** – This counter reflects the total amount of memory the VMkernel is effectively “borrowing” from virtual machines via the VMTools balloon driver running inside the guest OS. Balloon driver activity indicates over-commitment of ESX host memory. By default, the VMkernel may balloon up to 65% of a virtual machine's configure RAM.
- **Memory Swap Used** – This counter reflects the total amount of VMkernel swap usage on the host. In almost any scenario, this counter should be at or close to zero as VMkernel memory swapping is used as a last resort. Significant or consistent memory swapping indicates that ESX host memory is severely over-committed and that performance degradation is imminent or actively occurring.
- **Memory Zero** – This counter is helpful in identifying virtual machines that may have been configured with more RAM than necessary or are not presently using provisioned memory.

Memory Counters can be observed via the Virtual Infrastructure Client by first selecting a container (ESX Server host or cluster), choosing the *Performance* tab, then clicking *Change Chart Options*, and then selecting *Memory* from the Chart Options.

---

## Tracking ESX Server Memory Usage

---

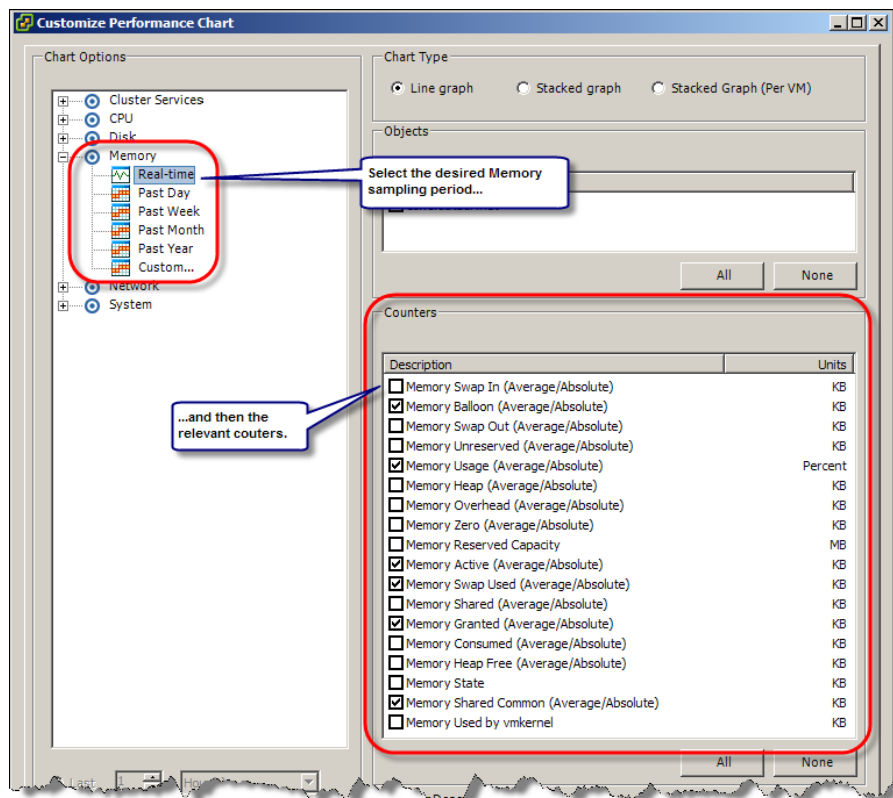
No matter which sizing strategy is employed when trying to determine the appropriate amount of physical memory to provision to an ESX Server host, it is critical to monitor ESX Server host memory utilization. The performance of a guest operating system within a virtual machine degrades dramatically if ESX is forced to swap out physical memory pages to disk; there is a system-wide cost of excessive virtual machine memory swapping that can compound the impact to multiple virtual machines quickly.

ESX Server host memory utilization is most simply monitored using VirtualCenter. Performance statistics can be observed by first selecting either a cluster or individual ESX Server host and clicking the *Performance* tab. Memory statistics can be selected then by clicking *Change Chart Options* (see Figure 2 - Modifying the VirtualCenter Performance Chart) and then selecting the relevant counters (Figure 3 - Customizing Performance Chart Options for Memory Counters).



**Figure 2 - Modifying the VirtualCenter Performance Chart**

For the purpose of tracking ESX Server memory usage, use the counters listed in the section entitled *VMkernel Memory Counters Explained*.



**Figure 3 - Customizing Performance Chart Options for Memory Counters**

When a cluster of ESX Servers has been deployed, it is important to incorporate some memory headroom to accommodate virtual machine migration using VMotion and DRS, since the VMkernel will require some time to evaluate the virtual machine's physical memory pages for transparent page sharing optimization, and during this time the virtual machine will occupy more physical memory. Indeed, there may not be any optimization if no matching

physical pages exist, but this is statistically unlikely. When using VMware HA, it is even more critical to ensure there is adequate resource headroom to accommodate the anticipated number of ESX Server host failures without sacrificing the availability or responsiveness of virtual machines outside of the ESX Server host outage.

## Memory RAID and VI3

The final component of an ESX Server memory provisioning strategy includes the use of Memory RAID. Newer models of servers supporting VMware ESX from IBM<sup>3</sup> and Dell<sup>4</sup> now also include memory redundancy features never available before. Much like disk striping and hot-spare technology, memory RAID allows for the failure of one or more DIMMs without the automatic failure of the entire system. These new features go beyond the preemptive warning of failure (such as ECC features have for years) and will actually keep the system operational despite the total failure of one or multiple DIMMs. This feat is accomplished slightly differently in each vendor's offering, but essentially data is striped in memory in a manner similar to disk parity striping. The use of Memory RAID technology does require an investment in physical memory which results in less RAM being available for active use, but the value of protecting this otherwise single point of failure is very high when considering that a single ESX Server can host forty or more virtual machines at any one time. The actual investment can be discounted dramatically by purchasing Kingston memory instead of the similarly reliable yet more costly OEM memory.

When physical memory fails, the operating system has no choice but to consider memory pages in the failed region as adulterated and panics, or terminates immediately, rather than risk corrupting data. Given the wide dynamic use of physical memory in ESX Server hosts, failed memory is more likely to result in an OS panic than typical x86 servers, which tend to have more unused memory headroom.

## Best Practice Recommendations

For efficient and effective provisioning of VMware ESX Server host memory capacity, use the following Best Practices as guidance:

- Identify and resolve memory bottlenecks in virtual machines before attempting to tune and optimize physical memory at the ESX Server host level. Improving the performance of memory-starved virtual machines might increase demand for physical memory in the ESX Server.
- Develop an understanding of the x86 workload characteristics. Leverage a Capacity Usage and Planning tool, such as Windows Perfmon, VMware Capacity Planner, or PlateSpin PowerRecon, to determine actual workload needs. Virtual machine memory should be configured based on actual needs plus a relative surplus for memory headroom. Many physical servers contain more memory than is necessary to service running applications and processes. When converting physical servers to virtual machines, evaluate the actual memory demands of the servers prior to conversion.

---

<sup>3</sup> One such IBM system is the System x3755, featuring Memory Online Spare and Chipkill memory

<sup>4</sup> One such system from Dell is the PowerEdge 6850 featuring Memory RAID

- Leverage memory Reservations, Limits and Shares effectively to control which virtual machines receive preferential treatment, should memory contention occur.
- Ensure that adequate disk space is available for per-VM swap files. ESX Server will create an individual swap file for every powered-on virtual machine that is equal to the amount of RAM provisioned to the VM minus any memory reservation.
- Alleviate VM swapping scenarios as quickly as possible, either by redistributing virtual machines workloads to other ESX Servers using VMotion, or by adding more physical memory to the ESX Server. Prolonged VM swapping can degrade performance of an ESX Server host.
- Monitor memory resource usage consistently and take a proactive approach to capacity planning, ensuring the environment performs at peak efficiency.
- For mission-critical ESX Server hosts, consider leveraging memory RAID to prevent ESX Server host outages due to faulty memory hardware.

## Summary

Physical memory is a critical component of all VMware Infrastructure 3 deployments, and appropriate sizing is necessary to ensure the success of solutions based on VI3, such as Server Consolidation. Provisioning the ideal amount of physical memory for ESX Servers becomes more important as systems with an ever-increasing number of CPU processing cores rise in popularity. Fully understand the ramifications of memory over-commitment in an ESX host and leverage ESX Server's memory management features to ensure a smooth-running and flexible platform. Understanding the meaning of memory-specific counters and knowing how to read them is a critical skill that frees VMware VirtualCenter Administrators from unexpected outages due to system crashes or under-performing hosts. Most importantly, gain a solid understanding of the memory demands of existing x86 workloads to be virtualized in the environment.

Finally, create and follow a list of Best Practices that modify behaviors that create positive habits and lead to stable systems, as opposed to the lead by exception rule common in environments with spotty performance. In terms of investment dollar per performance returns, physical RAM is the clear valuation leader. With all these considerations in mind, Kingston is your best source for memory. It offers legendary reliability, 100-percent testing and significant cost savings over OEM memory.

## About the Authors

---

**John Dodge**, VCP, MCSE, CCNA, is one of the managing partners of Foedus Group, LLC. Dodge's time is spent primarily designing and implementing virtual infrastructure platforms for Fortune 500 companies. He has more than twenty years of technical and management experience in IT infrastructure, and is also a highly regarded subject matter expert for Pharmaceutical Virtual Infrastructure qualification.

**Michael Burke**, VCP, MCP, is a Senior Virtual Infrastructure Engineer for Foedus, a leading Virtual Infrastructure Services company headquartered in the Northeast. Mike has more than six years of experience working closely with VMware products and over 12 years of experience with x86 systems in IT infrastructure. He has written several technical articles and white papers in addition to being a technical editor for the book "VMware ESX Server: Advanced Technical Design Guide."



---

17600 Newhope street  
Fountain Valley, CA 90278 MKMS-1016