INFORMATION GUIDE

The Role of Memory in VMware ESX Server 3

Contents

Introduction **1**

Memory Management in ESX Server 3 ..... **1**

ESX Server Architecture 1

Virtual Machine Memory 1

Memory Overcommitment 2

Memory Sharing 2

ESX Server Memory 2

Memory Balloon Driver 3

Swapping 4

Memory Usage **4**

Workloads 5

CPU Technology Trends 5

Memory Technology Trends 6

DDR2 FB-DIMM Memory Technology 7

DDR2 Registered DIMM Memory Technology 8

Conclusion **8**

About Kingston Technology Company, Inc. 8

Introduction

The memory management capabilities of VMware ESX Server 3.0 provide a unique and sophisticated way to maximize the usage of physical memory within a single box. 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. Advances in virtualization, CPU, and memory technology make the addition of memory one of the most effective investments for maximizing the utilization of an ESX Server host.

Memory Management in ESX Server 3

ESX Server Architecture

The VMkernel is a high-performance operating system developed by VMware that runs directly on the ESX Server host. VMkernel controls and manages most of the physical resources on the hardware, including:

- Memory
- Physical processors
- Storage controllers
- Networking
- Keyboard, video, and mouse

The VMkernel includes schedulers for CPU, memory, and disk access, and has full-fledged storage and network stacks.

The virtual machine monitor is the component actually responsible for virtualizing the CPUs. When a virtual machine is turned on, control transfers to the virtual machine monitor, which begins executing instructions from the virtual machine. The transfer of control to the virtual machine monitor involves setting the system state so that the virtual machine monitor runs directly on the hardware.

The VMkernel manages all machine memory except for the memory that is allocated to the service console. In prior versions of ESX Server, the service console required a specific amount of memory overhead for each virtual machine running on a host. ESX Server 3.0 employs a new architecture in which the service console is no longer burdened with memory requirements for each virtual machine running on a host. Individual virtual machine process threads are handled directly by the VMkernel, thereby eliminating the need to allocate service console memory for each virtual machine. It may be necessary to allocate more memory to the service console, however, if there are agents running there — for example, for monitoring or backup.

The VMkernel dedicates part of its managed machine memory for its own use, while the rest is available for use by virtual machines. Virtual machines use machine memory for two purposes: each virtual machine requires its own memory, and the virtual machine monitor requires some memory for its code and data.

Virtual Machine Memory

Each virtual machine consumes memory based on its configured size, plus a small amount of additional overhead memory for virtualization. Overhead memory includes space reserved for the

virtual machine frame buffer and various virtualization data structures. The configured size is the dynamic memory allocation for a virtual machine, and is based on three factors.

The *reservation* is a guaranteed lower bound on the amount of memory that the host reserves for the virtual machine. The *limit* is the upper limit on memory the host makes available to virtual machines. Finally, *shares* specify the priority for a virtual machine relative to other virtual machines on the server if more than the reservation amount is available. Later sections of this paper discuss these three factors in more detail.

Memory Overcommitment

For each running virtual machine, the system reserves physical memory for both the virtual machine's reservation (if any) and for its virtualization overhead. Because of the memory management techniques the ESX Server host employs, however, your virtual machines can use more memory than the physical machine (the host) has available. For example, you can have a host with 2GB memory and run four virtual machines with 1GB memory each. In that case, the memory is *overcommitted*. Overcommitment is an especially effective technique for maximizing memory use because, typically, some virtual machines are lightly loaded while others are more heavily loaded, and relative activity levels vary over time.

To improve memory utilization, the ESX Server host automatically transfers memory from idle virtual machines to virtual machines that need more memory. This memory is cleared prior to the reallocation, thus enforcing the isolation between virtual machines. The reservation and shares parameters are used to preferentially allocate memory to important virtual machines. This memory remains available to other virtual machines if it's not in use.

Memory Sharing

Many workloads present opportunities for sharing memory across virtual machines. For example, several virtual machines may be running instances of the same guest operating system, have the same applications or components loaded, or contain common data. ESX Server systems use a proprietary page-sharing technique to securely eliminate redundant copies of memory pages.

When ESX Server detects an extended period of idleness in the system, the VMkernel begins to compare physical memory pages using a hashing algorithm. After encountering two memory pages that appear to have the same contents, a binary compare is executed to ensure similar content. ESX Server then frees up one of the memory pages by updating the memory mappings for both virtual machines to point to the same physical memory address. Should a virtual machine attempt to write to or modify a shared memory page, ESX Server first copies the shared memory page, so that a distinct instance is created for each virtual machine. The virtual machine requesting the write operation to the memory page is then able to its contents without affecting other virtual machines sharing this same page.

With memory sharing, a workload consisting of multiple virtual machines often consumes less memory than it would when running on physical machines. As a result, the system can efficiently support higher levels of overcommitment. The amount of memory saved by memory sharing depends on workload characteristics. A workload of many nearly identical virtual machines may free up more than 30 percent of memory, while a more diverse workload may result in savings of less than 5 percent of memory.

ESX Server Memory

An ESX Server host allocates the memory specified by the limit parameter to each virtual machine unless memory is overcommitted. An ESX Server host never allocates more memory to a virtual machine than its specified physical memory size. For example, a 1GB virtual machine might have

the default limit (unlimited) or a user-specified limit (for example 2GB). In both cases, the ESX Server host never allocates more than 1GB, the physical memory size that was specified for it.

When memory is overcommitted, each virtual machine is allocated an amount of memory somewhere between what is specified by reservation and what is specified by limit. An ESX Server host determines allocations for each virtual machine based on the number of shares allocated to it. Memory shares entitle a virtual machine to a fraction of available physical memory. ESX Server hosts use a modified proportional-share memory allocation policy.

Shares specify the *relative priority* or importance of a virtual machine. If a virtual machine has twice as many shares of a resource as another virtual machine, it is entitled to consume twice as much of that resource. Shares are typically specified as high, normal, or low, with a 4:2:1 ratio. Virtual machines on the same server share resources according to their relative share values, bounded by the reservation and limit. When you assign shares to a virtual machine, you always specify the relative priority for that virtual machine. This is like handing out pieces of a pie. Assume you get one piece of pie and your sister gets two pieces of pie. How much pie each of you actually gets depends on the size of the pie and on the total number of pieces of the pie.

The amount of memory granted to a virtual machine above its reservation usually varies with the current memory load. ESX Server hosts estimate the working set for a virtual machine by monitoring memory activity over successive periods of virtual machine execution time. Estimates are smoothed over several time periods using techniques that respond rapidly to increases in working set size and more slowly to decreases in working set size.

If a virtual machine is not actively using its currently allocated memory, ESX Server charges a *memory tax* — more for idle memory than for memory that is in use. That is, the idle memory counts more towards the share allocation than memory in use. The default tax rate is 75 percent, that is, an idle page of memory costs as much as four active pages. This rate can be changed by modifying a parameter setting.

Memory tax helps prevent virtual machines from hoarding idle memory. The approach in ESX Server ensures that a virtual machine from which idle memory has been reclaimed can ramp up quickly to its full share-based allocation once it starts using its memory more actively.

ESX Server employs two distinct techniques for dynamically expanding or contracting the amount of memory allocated to virtual machines:

- A memory balloon driver (vmmemctl), loaded into the guest operating system running in a virtual machine, part of the VMware Tools package
- Paging from a virtual machine to a server swap file, without any involvement by the guest operating system

Memory Balloon Driver

The balloon driver, also known as the *vmmemctl* driver, collaborates with the server to reclaim pages that are considered least valuable by the guest operating system. It essentially acts like a native program in the operating system that requires more and more memory. The driver uses a proprietary ballooning technique that provides predictable performance that closely matches the behavior of a native system under similar memory constraints. This technique effectively increases or decreases memory pressure on the guest operating system, causing the guest to invoke its own native memory management algorithms. When memory is tight, the guest operating system decides which particular pages to reclaim and, if necessary, swaps them to its own virtual disk.

You need to be sure your guest operating systems have sufficient swap space. This swap space must be greater than or equal to the difference between the virtual machine's configured memory size and its reservation.

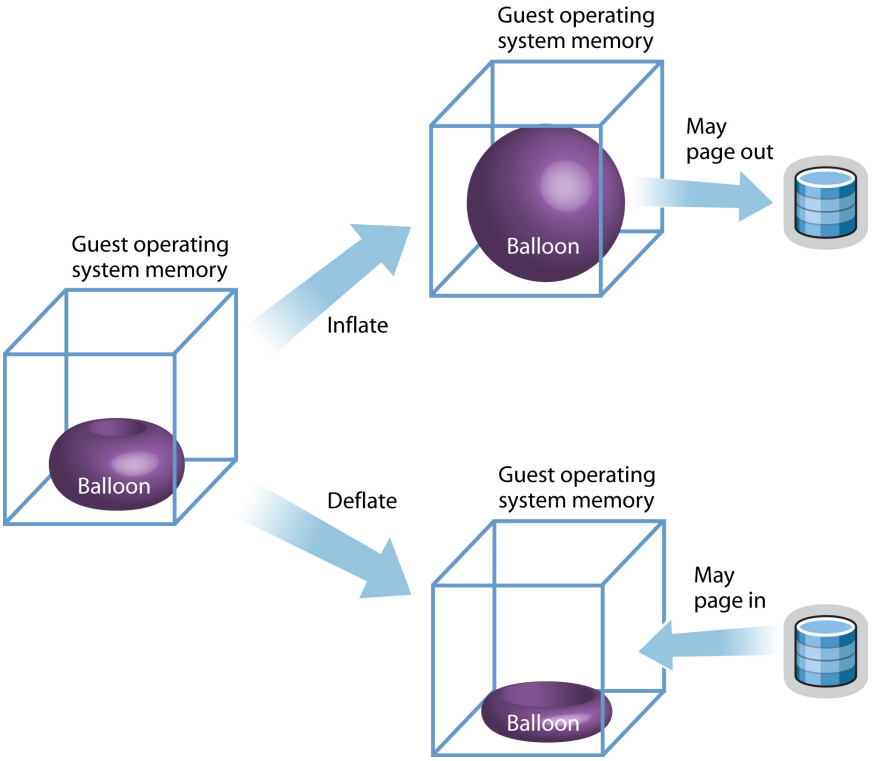

Figure 1: Memory balloon driver in action

Swapping

When you power on a virtual machine, a corresponding swap file is created and placed in the same location as the virtual machine configuration file (`.vmx` file). The virtual machine can power on only when the swap file is available. ESX Server hosts use swapping to forcibly reclaim memory from a virtual machine when no balloon driver is available. The balloon driver may be unavailable either because VMware Tools was never installed or because the driver has been disabled or is not running (for example, while the guest operating system is booting).

For optimum performance, ESX Server uses the balloon approach whenever possible. However, swapping is used when the driver is temporarily unable to reclaim memory quickly enough to satisfy current system demands. Since the memory is being swapped out to disk, there is a rather significant performance penalty when this technique is used. Swapping is a reliable mechanism of last resort that a host uses only when necessary to reclaim memory. Standard-demand paging techniques swap pages back in when the virtual machine needs them.

Memory Usage

Although this paper focuses on memory, ESX Server must also manage other resources used by virtual machines. When considering resource utilization, the four main components are memory, CPU, disk I/O and network I/O. The capacity of these resources is fixed during the live operation of

an ESX Server system, and the task of management represents the balancing of the usage of each of these resources across all virtual machines on the host.

Workloads

Experience with production workloads has shown that on modern, x86-based hardware memory is often the most used resource. The following table shows the average utilization rates of the four main computing resources for a typical system.

Resource	Utilization
CPU	6%
Memory	40%
Network I/O	<5%
Disk I/O	<5%

Of course, utilization highly depends on the particular workload being run. For example, a database can be expected to show much more disk activity. In addition, although the average utilization might be very small, utilization rates experience peaks that occur as a result of either normal usage cycles during the business day or unexpected demands. Nevertheless, it is often the case that even these variations for the other three resources remain within relatively low boundaries, while memory usage remains large.

The utilization of a workload directly translates into the utilization of a virtual machine running that workload. Since the utilization of the four resources on one virtual machine determines how much of those resources is available for other virtual machines on the same host, a doubling of the amount of physical memory on an ESX Server host enables twice as many virtual machines to be put on that host.

VMware provides various tools to determine resource utilization on virtual machines, as well as observer other performance indicators such as memory ballooning and swapping. By using these tools while a virtual machine runs a production (not test) workload, you can determine the actual utilization rates and decide if additional memory will enable more virtual machines to be co-resident on the same host.

CPU Technology Trends

There are a number of trends in IT that point to an even greater role of memory management in server virtualization. First and foremost, the notion of a CPU is rapidly changing. As fabrication technology enables a greater number of transistors to be put onto smaller and smaller areas, chip makers have begun to focus less on raw clock cycles and more on putting more functionality onto the same CPU. The biggest trend in this area is that of multicore CPUs. Companies such as Intel and AMD (in the x86 world), and Sun and Fujitsu (in the SPARC world) are creating CPUs in which multiple chip cores act as if they were completely independent CPUs, sharing only auxiliary functions such as memory transport. This means that, in the same physical space as an ordinary two-way server, you can today have up to eight CPU cores. In a nonvirtualized environment, the operating system is given the task of managing what is effectively an eight-way system. With ESX Server, the virtualization layer is given eight physical CPUs to construct the total pool of virtual CPUs for all the virtual machines on the host.

This increase in the available CPU resource on a host means that, for certain workloads, physical memory becomes an even more critical resource. In the case of a CPU-bound workload, where once you could have only two virtual machines, you can now have eight. However, these additional six virtual machines all come with the consequent need for additional memory to support the six additional operating systems and application infrastructure.

The availability of more cores on a single server also offers the opportunity to have larger workloads running in a virtual machine. Applications such as databases that can make effective use of two or more cores can now be placed on the same physical host in separate virtual machines. But, again, these applications typically are quite demanding of memory. With ESX Server 3, the virtual memory limit per virtual machine has been increased to 16GB, thus allowing these memory-hungry applications to run on virtual machines.

All of these trends lead to the conclusion that additional memory is one of the most important investments that can be made to get more out of a virtualized infrastructure.

Memory Technology Trends

Along with the fast deployment of servers based on multicore CPUs, memory technologies have evolved to deliver even higher performance. While CPU performance is often the top criterion for server performance, memory is a key enabler of application performance. Robust virtualization platforms require fast and reliable memory in capacities high enough to optimize the platforms' performance.

Memory acts as a cache for hard disk storage — memory accesses can take just a couple of nanoseconds (1ns = 1 billionth of a second); in comparison, hard drive access times are in milliseconds (1ms = 1 one-thousandth of a second). When a CPU cannot find data needed by an application in its own built-in cache or the server's main memory, the server must retrieve the data from the much slower hard drives, thereby forcing performance-lowering wait states on the processor. There is a demonstrable and direct correlation between transaction processing server performance and memory capacity and speed.

Most consolidation projects utilize two-way servers as the preferred platform for consolidating and virtualizing servers. In addition to supporting multicore CPUs, new Intel and AMD CPU-based platforms combine higher computing power with support for faster DDR2 memory and higher memory capacities. DDR2 memory technology, used on Intel servers starting in 2005 and on AMD servers in 2006, has increased memory bandwidth over the older DDR memory technology, as shown in figure 2.

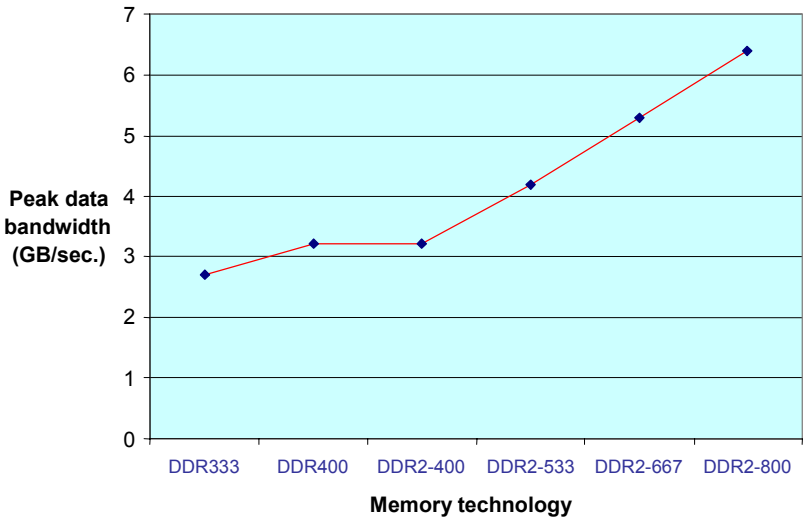

Figure 2: Memory module performance

As of the end of 2005, servers with DDR333/400 or DDR2 memory were hitting a memory technology limit (memory loading or ranks limitation with registered DIMMs) that limits memory capacity growth.

Starting in 2006, both Intel and AMD CPU-based server platforms support newer DDR2 memory. Intel is shifting to next-generation fully buffered DIMM (FB-DIMM) memory x86 platforms. AMD is supporting enhanced-speed DDR2 server memory based upon the older registered DIMM memory technology for its processors. Both of these are robust, memory-rich platforms that will accelerate virtualization and consolidation efforts by giving virtual machines faster access to greater amounts of memory.

DDR2 FB-DIMM Memory Technology

FB-DIMMs are the next-generation server memory technology from JEDEC, the industry's standards association. FB-DIMMs utilize a serial bus, similar to PCI Express. FB-DIMM technology allows for much faster memory to be supported (DDR2 currently, and DDR3 late in the decade with FB-DIMM2 technology) and scales up to memory capacities that will extend into hundreds of gigabytes in a few years. FB-DIMMs are called intelligent memory as they incorporate a special controller, called an advanced memory buffer, onto each FB-DIMM module. The advanced memory buffer manages the serial two-way connection and handles all the memory chip management functions.

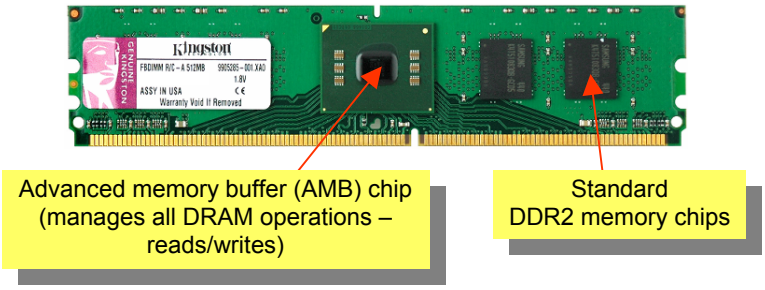

Figure 3: FB-DIMM memory

Intel servers support faster DDR2-533 and DDR2-667 FB-DIMMs and scale up to four memory channels. They support up to 16 FB-DIMM sockets. When memory modules are identically matched across four channels, these servers are able to transfer data in quad-channel mode (versus existing dual-channel memory mode). Quad-channel memory performance mode allows application data to be moved from and to memory in 256-bit chunks, compared to 128-bit for dual-channel memory mode. By contrast, four-way and higher Intel E8501 chipset-based servers will continue utilizing DDR2-400 Registered DIMM memory technology as the fastest memory supported until 2007, when FB-DIMMs are expected to be introduced

DDR2 Registered DIMM Memory Technology

AMD CPUs are architected differently from Intel CPUs, and AMD servers are able to support DDR2-533 and faster speeds using the older industry-standard Registered DDR2 DIMM memory technology from JEDEC. The new servers based on two-way, four-way, and eight-way AMD Opteron CPUs utilize DDR2-533/667 registered DIMMs. The new Opteron CPUs support up to two channels of four memory modules each, for a total of up to eight memory modules per CPU (two Opteron CPUs can thus support up to 16 DDR2 memory sockets).

Conclusion

The combination of ESX Server 3.0 with the newer generation servers based on two-way, multicore CPUs delivers compelling platforms for enterprise virtualization and consolidation efforts. Multicore CPUs, high-bandwidth DDR2 memory, higher memory capacities, and other architectural improvements increase the return on investment for virtualization efforts and reduce life-cycle costs for IT managers by enabling greater use of existing servers.

For more information on virtualization with ESX Server 3, contact your VMware representative or visit www.vmware.com. To learn more about memory technology and products, contact your Kingston Technology representative or visit www.kingston.com.

About Kingston Technology Company, Inc.

Kingston Technology Company, Inc. is the world's largest independent manufacturer of memory products. Kingston designs, manufactures and distributes memory products for desktops, laptops, servers, printers, and flash memory products for PDAs, mobile phones, digital cameras, and MP3 players. Kingston has manufacturing facilities in California, Malaysia, Taiwan, and China and sales offices in the United Sates, United Kingdom, Europe, Russia, Australia, Taiwan, China, and Latin America. For more information, please call 800-337-8410 or visit www.kingston.com.

Revision: 20060926 Item: IN-001-PRD-001

VMware, Inc. 3145 Porter Drive Palo Alto CA 94304 USA Tel 650-475-5000 Fax 650-475-5001 www.vmware.com

Pages 1–6.5 Copyright ©2006 VMware, Inc. All rights reserved. Pages 6.5–8 Copyright ©2006 Kingston Technology Company, Inc. All rights reserved. Protected by one or more of U.S. Patent Nos. 6,397,242, 6,496,847, 6,704,925, 6,711,672, 6,725,289, 6,735,601, 6,785,886, 6,789,156, 6,795,966, 6,880,022 6,961,941, 6,961,806 and 6,944,699; patents pending. VMware, the VMware "boxes" logo and design, Virtual SMP and VMotion are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions. Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation. Linux is a registered trademark of Linus Torvalds. All other marks and names mentioned herein may be trademarks of their respective companies.

vmware®

Kingston® TECHNOLOGY

COMMITTED TO MEMORY®