Performance Best Practices and Benchmarking Guidelines

VMware® Infrastructure 3 version 3.5 with ESX 3.5, ESXi 3.5, and VirtualCenter 2.5

Performance Best Practices and Benchmarking Guidelines
Revision: 20081106
Item: EN-000005-01

You can find the most up-to-date technical documentation on our Web site at:

http://www.vmware.com/support/

The VMware Web site also provides the latest product updates.

If you have comments about this documentation, submit your feedback to:

docfeedback@vmware.com

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Contents

Tables

About This Book

This book, *Performance Best Practices and Benchmarking Guidelines for VMware® Infrastructure 3 version 3.5*, provides a list of performance tips that cover the most performance-critical areas of VMware® Infrastructure 3 version 3.5. It is not intended as a comprehensive guide for planning and configuring your deployments.

Chapter 1, "VMware Infrastructure Performance," on page 11, provides guidance on maximizing performance of VMware Infrastructure 3 version 3.5.

Chapter 2, "ESX Benchmarking," on page 33, provides guidance on preparing for and conducting benchmark tests on VMware ESX systems.

Intended Audience

This manual is intended for system administrators who have already deployed VMware Infrastructure 3 version 3.5 and are looking to maximize its performance. The book assumes the reader is already familiar with VMware Infrastructure concepts and terminology.

Document Feedback

VMware welcomes your suggestions for improving our documentation. If you have comments, send your feedback to:

docfeedback@vmware.com

VMware Infrastructure Documentation

The VMware Infrastructure documentation consists of the combined VirtualCenter and VMware ESX documentation set.

You can access the most current versions of this manual and other books by going to:

http://www.vmware.com/support/pubs

Conventions

Table 1 illustrates the typographic conventions used in this manual.

Table 1. Conventions Used in This Manual

Style	Elements
Blue (online only)	Links, cross-references, and email addresses
Black boldface	User interface elements such as button names and menu items
`Monospace`	Commands, filenames, directories, and paths
`Monospace bold`	User input

Table 1. Conventions Used in This Manual (Continued)

Style	Elements
Italic	Document titles, glossary terms, and occasional emphasis
<Name>	Variable and parameter names

Technical Support and Education Resources

The following sections describe the technical support resources available to you.

Self-Service Support

Use the VMware Technology Network (VMTN) for self-help tools and technical information:

- Product information – http://www.vmware.com/products/
- Technology information – http://www.vmware.com/vcommunity/technology
- Documentation – http://www.vmware.com/support/pubs
- VMTN Knowledge Base – http://kb.vmware.com
- Discussion forums – http://www.vmware.com/community
- User groups – http://www.vmware.com/vcommunity/usergroups.html

For more information about the VMware Technology Network, go to http://www.vmtn.net.

Online and Telephone Support

Use online support to submit technical support requests, view your product and contract information, and register your products. Go to http://www.vmware.com/support.

Customers with appropriate support contracts should use telephone support for the fastest response on priority 1 issues. Go to http://www.vmware.com/support/phone_support.html.

Support Offerings

Find out how VMware support offerings can help meet your business needs. Go to http://www.vmware.com/support/services.

VMware Education Services

VMware courses offer extensive hands-on labs, case study examples, and course materials designed to be used as on-the-job reference tools. For more information about VMware Education Services, go to http://mylearn1.vmware.com/mgrreg/index.cfm.

Related Publications

For technical papers providing more detail on topics discussed in this book refer to http://www.vmware.com/vmtn/resources.

Issue	Publication
Mismatched HALs	KB article 1077
IRQ sharing	KB article 1290
Idle loop	KB article 1730
CPU utilization	KB article 2032
Network throughput between virtual machines	KB article 1428

Issue	Publication
Queue depths on QLogic cards	KB article 1267
Guest storage drivers	KB article 9645697
Disk outstanding commands parameter	KB article 1268
Linux guest timing	KB article 1420
VMotion CPU Compatibility Requirements for Intel Processors	KB article 1991
VMotion CPU Compatibility Requirements for AMD Processors	KB article 1992
Migrations with VMotion Prevented Due to CPU Mismatch—How to Override Masks	KB article 1993
Timing in virtual machines	Timekeeping in VMware Virtual Machines
VMFS partitions	Recommendations for Aligning VMFS Partitions
esxtop	Using esxtop to Troubleshoot Performance Problems
SAN best practices	*SAN Configuration Guide*
NFS best practices	*Server Configuration Guide*
iSCSI best practices	*Server Configuration Guide*
Memory allocation	*Resource Management Guide*
Swap space	*Resource Management Guide*
Supported maximums	*VMware Infrastructure 3 Release Notes* *VMware Infrastructure 3 Configuration Maximums*
Hardware requirements	*ESX Server 3.5 and VirtualCenter 2.5 Installation Guide*

VMware Infrastructure Performance

1

The chapter provides a list of performance tips covering the most performance-critical areas of VMware Infrastructure 3 version 3.5.

The chapter includes the following sections:

NOTE For planning purposes we recommend reading this entire chapter before beginning a deployment. Material in the Virtual Infrastructure Management section, for example, may influence your hardware choices.

NOTE Most of the recommendations in this chapter apply to both ESX and ESXi. The few exceptions are noted in the text.

Hardware

This section provides guidance on selecting and configuring hardware for use with ESX.

Validate Your Hardware

- Test system memory for 72 hours, checking for hardware errors.
- Check installed items against the hardware compatibility list for the VMware ESX version you will be running.
- Make sure that your hardware meets the minimum configuration supported by the VMware ESX version you will be running.

Hardware CPU Considerations

- When purchasing hardware, it is a good idea to consider CPU compatibility for VMotion. See "VirtualCenter Best Practices" on page 29.

Rapid Virtualization Indexing (RVI) CPUs

Rapid virtualization indexing (RVI)—also called nested page tables (NPT) or hardware virtual memory management unit (hardware virtual MMU)—is a hardware feature included in some recent AMD processors that addresses the overheads due to memory management unit virtualization. This feature is supported beginning with ESX 3.5 Update 1.

Without RVI, the guest operating system maintains guest virtual memory to guest physical memory address mappings, while ESX maintains "shadow page tables" that directly map guest virtual memory to host physical memory addresses. These shadow page tables are maintained for use by the processor and are kept consistent with the guest physical memory to host physical memory address mappings. This allows ordinary memory references to execute without additional overhead, since the hardware translation lookaside buffer (TLB) will cache direct guest virtual memory to host physical memory address translations read from the shadow page table. However, extra work is required to maintain the shadow page table structures.

RVI allows an additional level of page tables that map guest physical memory to host physical memory addresses, eliminating the need for ESX to intervene to virtualize the MMU in software.

For information about configuring the way ESX uses RVI, see "Configuring ESX for Rapid Virtualization Indexing (RVI)" on page 18.

Hardware Storage Considerations

Back-end storage configuration can greatly affect performance. Refer to the following sources for more information:

- For SAN best practices: "Using ESX Server with SAN: Concepts" in the *SAN Configuration Guide*.
- For NFS best practices: "Advanced Networking" in the *Server Configuration Guide*.
- For iSCSI best practices: "Configuring Storage" in the *Server Configuration Guide*.
- For a comparison of Fibre Channel, iSCSI, and NFS: *Comparison of Storage Protocol Performance*.

Storage performance issues are most often the result of configuration issues with underlying storage devices and are not specific to ESX.

Storage performance is a vast topic that depends on workload, hardware, vendor, RAID level, cache size, stripe size, and so on. Consult the appropriate documentation from VMware as well as the storage vendor.

Many workloads are very sensitive to the latency of I/O operations. It is therefore important to have storage devices configured correctly. The remainder of this section lists practices and configurations recommended by VMware for optimal storage performance.

- For iSCSI and NFS, make sure that your network topology does not contain Ethernet bottlenecks, where multiple links are routed through fewer links, potentially resulting in oversubscription and dropped network packets. Any time a number of links transmitting near capacity are switched to a smaller number of links, such oversubscription is a possibility.

  Recovering from dropped network packets results in large performance degradation. In addition to time spent determining that data was dropped, the retransmission uses network bandwidth that could otherwise be used for new transactions.

- Applications or systems that write large amounts of data to storage, such as data acquisition or transaction logging systems, should not share Ethernet links to a storage device. These types of applications perform best with multiple connections to storage devices.

- Performance design for a storage network must take into account the physical constraints of the network, not logical allocations. Using VLANs or VPNs does not provide a suitable solution to the problem of link oversubscription in shared configurations. VLANs and other virtual partitioning of a network provide a way of logically configuring a network, but don't change the physical capabilities of links and trunks between switches.

- For NFS and iSCSI, if the network switch deployed for the data path supports VLAN, it is beneficial to create a VLAN just for the ESX host's vmknic and the NFS/iSCSI server. This minimizes network interference from other packet sources.

- If you have heavy disk I/O loads, you may need to assign separate storage processors to separate systems to handle the amount of traffic bound for storage.

- To optimize storage array performance, spread I/O loads over the available paths to the storage (across multiple host bus adapters (HBAs) and storage processors (SPs)).

- Configure maximum queue depth for HBA cards. For additional information see KB article 1267, listed in "Related Publications" on page 8.

- Be aware that with software-initiated iSCSI and NAS the network protocol processing takes place on the host system, and thus these can require more CPU resources than other storage options.

Hardware Networking Considerations

- Before undertaking any network optimization effort, you should understand the physical aspects of the network. The following are just a few aspects of the physical layout that merit close consideration:
  - Consider using server-class network interface cards (NICs) for the best performance.
  - Make sure the NICs are configured to use auto-negotiation to set speed and duplex settings, and are configured for full-duplex mode.
  - Make sure the network infrastructure between the source and destination NICs doesn't introduce bottlenecks. For example, if both NICs are 1 Gigabit, make sure all cables and switches are capable of the same speed and that the switches are not configured to a lower speed.
  - Make sure data packets are sized such that fragmentation and reassembled along the transmission path is minimized.
- For the best networking performance, VMware recommends the use of network adapters that support the following hardware features:
  - Checksum offload
  - TCP segmentation offload (TSO)
  - Ability to handle high-memory DMA (that is, 64-bit DMA addresses)
  - Ability to handle multiple Scatter Gather elements per Tx frame
  - Jumbo frames (JF)
- On some 10 Gigabit Ethernet hardware network adapters ESX 3.5 supports NetQueue, a technology that significantly improves performance of 10 Gigabit Ethernet network adapters in virtualized environments.
- In addition to the PCI and PCI-X bus architectures, we now have the PCI Express (PCIe) architecture. Ideally single-port 10 Gigabit Ethernet network adapters should use PCIe x8 (or higher) or PCI-X 266 and dual-port 10 Gigabit Ethernet network adapters should use PCIe x16 (or higher). There should preferably be no "bridge chip" (e.g., PCI-X to PCIe or PCIe to PCI-X) in the path to the actual Ethernet device (including any embedded bridge chip on the device itself), as these chips can reduce performance.
- Multiple physical network adapters between a single virtual switch (vSwitch) and the physical network constitute a NIC team. NIC teams can provide passive failover in the event of hardware failure or network outage and, in some configurations, can increase performance by distributing the traffic across those physical network adapters. For more information, see VMware Virtual Networking Concepts, listed in "Related Publications" on page 8.
- Because in ESX (though not ESXi) cloning of virtual machines uses the service console's network interface (rather than the VMkernel's interface), if any cloning will be done the service console should be connected to a Gigabit-rated NIC (which should in turn be connected to Gigabit-rated infrastructure).

ESX and Virtual Machines

This section provides guidance regarding ESX software itself and the virtual machines that run in it.

ESX General Considerations

This subsection provides guidance regarding a number of general performance considerations in ESX.

- Plan the deployment. Allocate enough resources, especially (in the case of ESX, but not ESXi) for the service console.

  Table 1-1 lists symptoms of insufficient resources for the service console.

  Table 1-1. Symptoms of Insufficient Resources for the ESX Service Console

Insufficient Resource Type	Symptoms
Memory	Poor VMware Infrastructure Client (VI Client) response
Disk Space	Inability to write diagnostic messages to log; inability to log into the service console

- Allocate only as much virtual hardware as required for each virtual machine. Provisioning a virtual machine with more resources than it requires can, in some cases, reduce the performance of that virtual machine as well as other virtual machines sharing the same host.
- Disconnect or disable unused or unnecessary physical hardware devices, such as:
  - COM ports
  - LPT ports
  - USB controllers
  - Floppy drives
  - Optical drives (that is, CD or DVD drives)

  Disabling hardware devices (typically done in BIOS) can free IRQ resources and can reduce the likelihood of encountering the interrupt-sharing issue described below. Additionally, some devices, such as USB controllers, operate on a polling scheme that consumes extra CPU resources.
- Avoid configurations in which multiple physical hardware devices sharing the same interrupt are managed by different entities (that is, one or more devices managed by the service console, and one or more devices managed by the VMkernel). These configurations require extra context switches, potentially having a significant impact on performance. (Because it does not include a service console, this point doesn't apply to ESXi.)

  You can find instructions to determine if this issue applies to your configuration, as well as how to reduce the performance impact if it does, in KB article 1290, listed in "Related Publications" on page 8.
- Unused or unnecessary virtual hardware devices can impact performance and should be disabled.

  For example, windows guests poll optical drives (that is, CD or DVD drives) quite frequently. When virtual machines are configured to use a physical drive, and multiple guests simultaneously try to access that drive, performance could suffer. This can be reduced by configuring the virtual machines to use ISO images instead of physical drives, and can be avoided entirely by disabling optical drives in virtual machines when the devices are not needed.

ESX CPU Considerations

This subsection provides guidance regarding CPU considerations in ESX.

CPU virtualization adds varying amounts of overhead depending on the percentage of the virtual machine's workload that can be executed on the physical processor as is and the cost of virtualizing the remaining workload.

Even for those applications in which virtualization does add overhead, CPU virtualization will reduce their performance only if they are CPU-bound— that is, if most of their time is spent executing instructions rather than waiting for external events such as user interaction, device input, or data retrieval. Otherwise such applications might still deliver performance comparable to native because there are CPU cycles available to absorb the virtualization overhead.

The rest of this subsection lists practices and configurations recommended by VMware for optimal CPU performance.

- When configuring virtual machines, the total CPU resources needed by the virtual machines running on the system should not exceed the CPU capacity of the host. If the host CPU capacity is overloaded, the performance of individual virtual machines may degrade.
- It is a good idea to periodically monitor the CPU usage of the host. This can be done through the VI Client or by using `esxtop` or `resextop`. Below we describe how to interpret `esxtop` data:
  - If the load average on the first line of the `esxtop` CPU Panel is equal to the number of physical processors in the system, this indicates that the system is overloaded.
  - The usage percentage for the physical CPUs on the PCPU line can be another indication of a possibly overloaded condition. In general, 80% usage is a reasonable ceiling and 90% should be a warning that the CPUs are approaching an overloaded condition. However organizations will have varying standards regarding the desired load percentage.

  For information about using `esxtop` see Appendix A of the VMware *Resource Management Guide*, listed in "Related Publications" on page 8.
- Use as few virtual CPUs (vCPUs) as possible. For example, do not use virtual SMP if your application is single-threaded and will not benefit from the additional vCPUs.

  Having virtual machines configured with vCPUs that are not used still imposes some small resource requirements on ESX. Unused vCPUs still consume timer interrupts and in some cases execute the idle loops of the guest operating system. This translates to real CPU consumption from the point of view of ESX. For additional information see KB articles 1077 and 1730, listed in "Related Publications" on page 8.
- Configure single-processor virtual machines with a UP HAL (hardware abstraction layer) or kernel. Multi-processor virtual machines must be configured with an SMP HAL/kernel.

  Most guest operating systems can be configured to use either a UP HAL/kernel or an SMP HAL/kernel. The UP operating system versions are for single-processor systems. If used on a multiprocessor system, a UP operating system version will recognize and use only one of the processors. The SMP versions, while required in order to fully utilize multiprocessor systems, may also be used on single-processor systems. Due to their extra synchronization code, however, SMP operating system versions used on single-processor systems are slightly slower than UP operating system versions used on the same systems. For additional information see KB article 1730, listed in "Related Publications" on page 8.

  NOTE When changing an existing virtual machine running Windows from multiprocessor to single-processor the HAL usually remains SMP.
- Certain guest operating systems do not actually halt the vCPU when they idle. Since the vCPU is not halted, ESX will continue to run and schedule this vCPU as if it were still doing useful work. This execution of the idle loop results in unnecessary CPU utilization.

To prevent this unneeded CPU utilization, ESX detects that a guest vCPU is idle and deschedules it. The detection heuristic is controlled by the idle loop spin parameter and can sometimes affect performance. For more details on optimally configuring this parameter refer to KB article 1730, listed in "Related Publications" on page 8.

Hyper-Threading

- Hyper-threading technology allows a single physical processor core to behave like two logical processors, essentially allowing two independent threads to run simultaneously. Unlike having twice as many processor cores — that can roughly double performance — hyper-threading can slightly increase system performance by keeping the processor pipeline busier.

  If the hardware and BIOS support hyper-threading, ESX automatically makes use of it. To enable hyper-threading:

  a Ensure that your system supports hyper-threading technology. It is not enough that the processors support hyper-threading — the BIOS must support it as well. Consult your system documentation to see if the BIOS includes support for hyper-threading.

  b Enable hyper-threading in the system BIOS. Some manufacturers label this option **Logical Processor** while others label it **Enable Hyper-threading**.

- An ESX system enabled for hyper-threading should behave almost exactly like a standard system. Logical processors on the same core have adjacent CPU numbers, so that CPUs 0 and 1 are on the first core, CPUs 2 and 3 are on the second core, and so on.

  ESX systems manage processor time intelligently to guarantee that load is spread smoothly across all physical cores in the system. If there is no work for a logical processor it is put into a special halted state that frees its execution resources and allows the virtual machine running on the other logical processor on the same core to use the full execution resources of the core.

- Be careful when using CPU affinity on systems with hyper-threading. Because the two logical processors share most of the processor resources, pinning vCPUs, whether from different virtual machines or from one SMP virtual machine, to both logical processors on one core (CPUs 0 and 1, for example) could cause poor performance.

- ESX 3.5 provides configuration parameters for controlling the scheduling of virtual machines on hyper-threaded systems. When choosing hyper-threading sharing choices **Any** (which is the default) is almost always preferred over **None** or **Internal**.

  The **None** option indicates that when a vCPU from this virtual machine is assigned to a logical processor, no other vCPU, whether from the same virtual machine or from a different virtual machine, should be assigned to the other logical processor that resides on the same core. That is, each vCPU from this virtual machine should always get a whole core to itself and the other logical CPU on that core should be placed in the halted state. This option is like disabling hyper-threading for that one virtual machine.

  The **Internal** option indicates that the vCPUs of the virtual machine can share cores with each other, but not with vCPUs from other virtual machines.

  For nearly all workloads, custom hyper-threading settings are not necessary. They can, however, help in cases of unusual workloads that interact badly with hyper-threading. For example, an application with cache-thrashing problems might slow down an application sharing its physical CPU core. You could configure the virtual machine running the problem application with the **None** or **Internal** hyper-threading option to isolate it from other virtual machines.

  The trade-off for configuring **None** or **Internal** should also be considered. With either of these settings, there can be cases where there is no core to which a descheduled virtual machine can be migrated, even though one or more logical cores are idle. As a result, it is possible that virtual machines with hyper-threading set to **None** or **Internal** can experience performance degradation, especially on systems with a limited number of CPU cores.

Non-Uniform Memory Access (NUMA)

- Both IBM (X-Architecture) and AMD (Opteron-based) non-uniform memory access (NUMA) systems are supported in ESX 3.5. On AMD Opteron-based systems, such as the HP ProLiant DL585 Server, BIOS settings for node interleaving determine whether the system behaves like a NUMA system or like a uniform memory accessing (UMA) system. For more information, refer to your server's documentation.

  If node interleaving is disabled, ESX detects the system as NUMA and applies NUMA optimizations. If node interleaving (also known as interleaved memory) is enabled, ESX does not detect the system as NUMA.

  The intelligent, adaptive NUMA scheduling and memory placement policies in ESX 3.5 can manage all virtual machines transparently, so that administrators do not need to deal with the complexity of balancing virtual machines between nodes by hand. However, manual override controls are available, and advanced administrators may prefer to control the memory placement (through the **Memory Affinity** option) and processor utilization (through the **Only Use Processors** option).

- By default, ESX NUMA scheduling and related optimizations are enabled only on systems with a total of at least four CPU cores and with at least two CPU core per NUMA node. On such systems, virtual machines can be separated into the following categories:

  - Virtual machines with a number of vCPUs equal to or less than the number of cores in each NUMA node will be managed by the NUMA scheduler and will have the best performance.
  - Virtual machines with more vCPUs than the number of cores in a NUMA node will *not* be managed by the NUMA scheduler. They will still run correctly, but they will not benefit from the ESX NUMA optimizations.

NOTE More information about using NUMA systems with ESX can be found in the "Advanced Attributes and What They Do" and "Using NUMA Systems with ESX Server" sections of the VMware *Resource Management Guide*.

Configuring ESX for Rapid Virtualization Indexing (RVI)

For a description of RVI, see "Rapid Virtualization Indexing (RVI) CPUs" on page 12.

RVI is supported beginning with ESX 3.5 Update 1. By default, on AMD processors that support it ESX Update 1 uses RVI for virtual machines running 64-bit guest operating systems and does not use RVI for virtual machines running 32-bit guest operating systems. These defaults should work well for the majority of guest operating systems and workloads. If desired, however, this can be changed using the VI Client by selecting the virtual machine to be configured, clicking **Edit virtual machine settings**, choosing the **Options** tab, selecting **Virtualized MMU**, and selecting the desired radio button. **Force use of these features where available** enables RVI, **Forbid use of these features** disables RVI, and **Allow the host to determine automatically** results in the default behavior described above.

When running 64-bit operating systems, the use of RVI will result in performance for most workloads that is equal to or better than without RVI. Workloads that show the most benefit from RVI include terminal services and multi-process applications. The benefit of RVI also increases as the number of virtual processors is increased.

Although RVI is disabled by default for virtual machines running 32-bit guest operating systems, enabling it for certain 32-bit operating systems may achieve performance benefits similar to those achieved for 64-bit operating systems. These 32-bit operating systems include Windows 2003 SP2, Windows Vista, and Linux.

When RVI is enabled for a virtual machine we recommend you also—when possible—configure that virtual machine's guest operating system and applications to make use of large memory pages.

ESX Memory Considerations

This subsection provides guidance regarding memory considerations in ESX.

Memory Overhead

Virtualization causes an increase in the amount of physical memory required due to the extra memory needed by ESX for its own code and for data structures. This additional memory requirement can be separated into two components:

1 A system-wide memory space overhead for the service console (typically 272MB, but not present in ESXi) and for the VMkernel (typically about 100MB).

2 An additional memory space overhead for each virtual machine.

The per-virtual-machine memory space overhead includes space reserved for the virtual machine frame buffer and various virtualization data structures. These amounts depend on the number of vCPUs, the configured memory for the guest operating system, and whether the guest operating system is 32-bit or 64-bit.

ESX provides optimizations such as memory sharing (see "Memory Reclamation Techniques" on page 19) to reduce the amount of physical memory used on the underlying server. In some cases these optimizations can save more memory than is taken up by the overhead.

The *Resource Management Guide for ESX Server 3.5, ESX Server 3i version 3.5, and VirtualCenter 2.5* includes a table with detailed memory space overhead numbers.

Memory Sizing

- Carefully select the amount of virtual memory you allocate to your virtual machines. You should allocate enough memory to hold the working set of applications you will run in the virtual machine, thus minimizing swapping, but avoid over-allocating memory. Allocating more memory than needed unnecessarily increases the virtual machine memory overhead, thus using up memory that could be used to support more virtual machines.
- When possible, configure 32-bit Linux virtual machines with no more than 896MB of memory. 32-bit Linux kernels use different techniques to map memory on systems with more than 896MB. These techniques impose additional overhead on the virtual machine monitor and can result in slightly reduced performance.
- Avoid frequent memory reclamation. Make sure the host has more physical memory than the total amount of memory that will be used by ESX plus the sum of the working set sizes that will be used by all the virtual machines running at any one time.

  NOTE ESX does, however, allow some memory overcommitment without impacting performance by using the techniques described in "Memory Reclamation Techniques."

Memory Reclamation Techniques

- ESX uses three memory management mechanisms — page sharing, ballooning, and swapping — to dynamically reduce the amount of memory allocated to each virtual machine.
  - **Page Sharing:** ESX uses a proprietary technique to transparently and securely share memory pages between virtual machines, thus eliminating redundant copies of memory pages. By default, page sharing is used regardless of the memory demands on the host system.
  - **Ballooning:** When page sharing does not reclaim enough memory, ESX uses ballooning. ESX works with a VMware-supplied vmmemctl module loaded into the guest operating system to reclaim pages that are considered least valuable by the guest operating system. The vmmemctl driver is installed as part of VMware Tools. Ballooning provides performance closely matching that of a native system under similar memory constraints. To use ballooning, the guest operating system must be configured with sufficient swap space.
  - **Swapping:** When both page sharing and ballooning fail to reclaim sufficient memory from an overcommitted system, host-level swapping is used to forcibly reclaim memory from a virtual machine. Because this will swap out active pages, it can cause virtual machine performance to degrade significantly.

Memory Swapping

As described in "Memory Reclamation Techniques," when other memory reclamation techniques are not sufficient, ESX uses memory swapping. This swapping occurs at the host level, and is distinct from the swapping that can occur within the virtual machine under the control of the guest operating system.

This subsection describes ways to avoid or reduce host-level swapping, and presents techniques to reduce its impact on performance when it is unavoidable.

- Due to page sharing and other techniques (described below) ESX can handle a limited amount of memory overcommitment without swapping. However, if the overcommitment is large enough to cause ESX to swap, performance in the virtual machines is significantly reduced.
- If you choose to overcommit memory with ESX, be sure you have sufficient swap space on your ESX system. At the time a virtual machine is first powered on, ESX creates a swap file for that virtual machine that is equal in size to the difference between the virtual machine's configured memory size and its memory reservation. The available swap space must therefore be at least this large.

  NOTE The memory reservation is a guaranteed lower bound on the amount of physical memory ESX reserves for the virtual machine. It can be configured through the VI Client in the settings window for each virtual machine (select **Edit virtual machine settings**, choose the **Resources** tab, select **Memory**, then set the desired reservation).

- Swapping can be avoided in a specific virtual machine by using the VI Client to reserve memory for that virtual machine at least equal in size to the machine's active working set. Be aware, however, that configuring resource reservations will reduce the number of virtual machines that can be run on a system. This is because ESX will keep available enough host memory to fulfill all reservations and won't power-on a virtual machine if doing so would reduce the available memory to less than the amount reserved.
- If swapping cannot be avoided, placing the virtual machine's swap file on a high speed/high bandwidth storage system results in the smallest performance impact. The swap file location can be set with the **sched.swap.dir** option in the VI Client (select **Edit virtual machine settings**, choose the **Options** tab, select **Advanced**, and click **Configuration Parameters**). If this option is not set, the swap file will be created in the virtual machine's working directory: either the directory specified by `workingDir` in the virtual machine's `.vmx` file, or, if this variable is not set, in the directory where the `.vmx` file is located. The latter is the default behavior.

Large Memory Pages for Hypervisor and Guest Operating System

In addition to the usual 4KB memory pages, ESX 3.5 also makes 2MB memory pages available (commonly referred to as "large pages"). By default ESX 3.5 assigns these 2MB machine memory pages to guest operating systems that request them, giving the guest operating system the full advantage of using large pages. The use of large pages results in reduced memory management overhead and can therefore increase hypervisor performance.

If an operating system or application can benefit from large pages on a native system, that operating system or application can potentially achieve a similar performance improvement on a virtual machine backed with 2MB machine memory pages. Consult the documentation for your operating system and application to determine how to configure them each to use large memory pages.

More information about large page support can be found in the performance study entitled *Large Page Performance* (available at http://www.vmware.com/resources/techresources/1039).

Rapid Virtualization Indexing (RVI)

Rapid Virtualization Indexing (RVI) is a technique that virtualizes the CPU's memory management unit (MMU). For a description of RVI, see "Rapid Virtualization Indexing (RVI) CPUs" on page 12; for information about configuring the way ESX uses RVI, see "Configuring ESX for Rapid Virtualization Indexing (RVI)" on page 18.

ESX Storage Considerations

This subsection provides guidance regarding storage considerations in ESX.

- ESX supports raw device mapping (RDM), which allows management and access of raw SCSI disks or LUNs as VMFS files. An RDM is a special file on a VMFS volume that acts as a proxy for a raw device. The RDM file contains meta data used to manage and redirect disk accesses to the physical device. Ordinary VMFS is recommended for most virtual disk storage, but raw disks may be desirable in some cases.
- ESX supports three virtual disk modes: Independent persistent, Independent nonpersistent, and Snapshot. These modes have the following characteristics:
  - **Independent persistent** – Changes are immediately written to the disk, so this mode provides the best performance.
  - **Independent nonpersistent** – Changes to the disk are discarded when you power off or revert to a snapshot. In this mode disk writes are appended to a redo log. When a virtual machine reads from disk, ESX first checks the redo log (by looking at a directory of disk blocks contained in the redo log) and, if the relevant blocks are listed, reads that information. Otherwise, the read goes to the base disk for the virtual machine. These redo logs, which track the changes in a virtual machine's file system and allow you to commit changes or revert to a prior point in time, can incur a performance penalty.
  - **Snapshot** – A snapshot captures the entire state of the virtual machine. This includes the memory and disk states as well as the virtual machine settings. When you revert to a snapshot, you return all these items to their previous states. Like the independent nonpersistent disks described above, snapshots use redo logs and can incur a performance penalty.
- ESX supports multiple disk types:
  - **Thick** – Thick disks, which have all their space allocated at creation time, are further divided into two types: eager zeroed and lazy zeroed.
    - **Eager-zeroed** – An eager-zeroed thick disk has all space allocated and zeroed out at the time of creation. This extends the time it takes to create the disk, but results in the best performance, even on first write to each block.
    - **Lazy-zeroed** – A lazy-zeroed thick disk has all space allocated at the time of creation, but each block is only zeroed on first write. This results in a shorter creation time, but reduced performance the first time a block is written to. Subsequent writes, however, have the same performance as an eager-zeroed thick disk.
  - **Thin** – Space required for a thin-provisioned virtual disk is allocated and zeroed upon demand, as opposed to upon creation. There is a higher I/O penalty during the first write to an unwritten file block, but the same performance as an eager-zeroed thick disk on subsequent writes.

  Virtual machine disks created through the VI Client (whether connected directly to the ESX host or through VirtualCenter) are lazy-zeroed thick disks, thus incurring the first-write performance penalty described above. The other disk types can be created from the console command line using `vmkfstools`. For more details refer to the `vmkfstools` man page.
- The alignment of your file system partitions can impact performance. VMware makes the following recommendations for VMFS partitions:
  - Like other disk-based file systems, VMFS suffers a penalty when the partition is unaligned. Using the VI Client to create VMFS partitions avoids this problem since it automatically aligns the partitions along the 64KB boundary.
  - To manually align your VMFS partitions, check your storage vendor's recommendations for the partition starting block. If your storage vendor makes no specific recommendation, use a starting block that is a multiple of 8KB.

- Before performing an alignment procedure, carefully evaluate the performance impact of the unaligned VMFS partition on your particular workload. For example, if the workload is light, if the system already meets service level agreements, or if the workload contains many random writes, consider delaying the partition alignment procedure until a scheduled outage, or not performing it at all.

For additional information see *Recommendations for Aligning VMFS Partitions*, listed in "Related Publications" on page 8.

- Multiple heavily-used virtual machines concurrently accessing the same VMFS volume, or multiple VMFS volumes backed by the same LUNs, can result in decreased storage performance. Appropriately-configured storage architectures can avoid this issue. For information about storage configuration and performance, see *Scalable Storage Performance* (available at http://www.vmware.com/resources/techresources/1059).
- Meta-data-intensive operations can impact virtual machine I/O performance. These operations include:
  - Administrative tasks such as backups, provisioning virtual disks, cloning virtual machines, or manipulating file permissions.
  - Scripts or cron jobs that open and close files. These should be written to minimize open and close operations (open, do all that needs to be done, then close, rather than repeatedly opening and closing).
  - Dynamically growing `.vmdk` files for thin-provisioned virtual disks. When possible, use thick disks for better performance.

  You can reduce the effect of these meta-data-intensive operations by scheduling them at times when you don't expect high virtual machine I/O load.

  More information on this topic can be found in *Scalable Storage Performance* (available at http://www.vmware.com/resources/techresources/1059).
- I/O latency statistics can be monitored using `esxtop` (or `resxtop`), which reports device latency, time spent in the kernel, and latency seen by the guest.
- Make sure I/O is not queueing up in the VMkernel by checking the number of queued commands reported by `esxtop` (or `resxtop`).

  Since queued commands are an instantaneous statistic, you will need to monitor the statistic over a period of time to see if you are hitting the queue limit. To determine queued commands, look for the **QUED** counter in the `esxtop` (or `resxtop`) storage resource screen. If queueing occurs, try adjusting queue depths. For further information see KB article 1267, listed in "Related Publications" on page 8.
- The driver queue depth can be set for some VMkernel drivers. For example, while the default queue depth of the QLogic driver is 16, specifying a larger queue depth may yield higher performance. You can also adjust the maximum number of outstanding disk requests per virtual machine in the VMkernel through the VI Client. Setting this parameter can help equalize disk bandwidth across virtual machines. For additional information see KB article 1268, listed in "Related Publications" on page 8.
- By default, Active/Passive storage arrays use **Most Recently Used** path policy. Do not use **Fixed** path policy for Active/Passive storage arrays to avoid LUN thrashing. For more information, see the *VMware SAN Configuration Guide*, listed in "Related Publications" on page 8.
- By default, Active/Active storage arrays use **Fixed** path policy. When using this policy you can maximize the utilization of your bandwidth to the storage array by designating preferred paths to each LUN through different storage controllers. For more information, see the *VMware SAN Configuration Guide*, listed in "Related Publications" on page 8.
- Do not allow your service console's root file system to become full. (Because it does not include a service console, this point doesn't apply to ESXi.)
- Disk I/O bandwidth can be unequally allocated to virtual machines by using the VI Client (select **Edit virtual machine settings**, choose the **Resources** tab, select **Disk**, then change the **Shares** field).

ESX Networking Considerations

This subsection provides guidance regarding networking considerations in ESX.

- In a native environment, CPU utilization plays a significant role in network throughput. To process higher levels of throughput, more CPU resources are needed. The effect of CPU resource availability on the network throughput of virtualized applications is even more significant. Because insufficient CPU resources will reduce maximum throughput, it is important to monitor the CPU utilization of high-throughput workloads.
- Use separate virtual switches each connected to its own physical network adapter to avoid contention between the ESX service console, the VMkernel, and virtual machines, especially virtual machines running heavy networking workloads.
- To establish a network connection between two virtual machines that reside on the same ESX system, connect both virtual machines to the same virtual switch. If the virtual machines are connected to different virtual switches, traffic will go through wire and incur unnecessary CPU and network overhead.
- The VMkernel network device driver defaults to speed and duplex settings of auto-negotiate. These settings will work correctly with network switches that are also set to auto-negotiate. If your switch is configured for a specific speed and duplex setting, however, you must force the network driver to use the same speed and duplex setting (for Gigabit links, network settings should be set to auto-negotiate and not forced).

Guest Operating Systems

This section provides guidance regarding the guest operating systems running in virtual machines.

Guest Operating System General Considerations

- Use guest operating systems that are supported by ESX. See the *Guest Operating System Installation Guide* for a list.

  NOTE VMware Tools might not be available for unsupported guest operating systems.

- Install the latest version of VMware Tools in the guest operating system. Make sure to update VMware Tools after each ESX upgrade.

  Installing VMware Tools in Windows guests updates the BusLogic SCSI driver included with the guest operating system to the VMware-supplied driver. The VMware driver has optimizations that guest-supplied Windows drivers do not.

  VMware Tools also includes the balloon driver used for memory reclamation in ESX. Ballooning (described in "Memory Reclamation Techniques" on page 19) will not work if VMware Tools is not installed.

- Disable screen savers and Window animations in virtual machines. On Linux, if using an X server is not required, disable it. Screen savers, animations, and X servers all consume extra physical CPU resources, potentially affecting consolidation ratios and the performance of other virtual machines.

- Schedule backups and anti-virus programs in virtual machines to run at off-peak hours, and avoid scheduling them to run simultaneously in multiple virtual machines on the same ESX host. In general, it is a good idea to evenly distribute CPU usage, not just across CPUs but also across time. For workloads such as backups and antivirus where the load is predictable, this is easily achieved by scheduling the jobs appropriately.

- Always use VMware Tools, the VI Client, `esxtop`, or `resxtop` to measure resource utilization. CPU and memory usage reported within the guest can be different from what ESX reports. For additional information see KB article 2032, listed in "Related Publications" on page 8.

- For the most accurate timekeeping, consider configuring your guest operating system to use NTP, Windows Time Service, or another timekeeping utility suitable for your operating system. The time-synchronization option in VMware Tools can be used instead, but it is not designed for the same level of accuracy as these other tools and does not adjust the guest time when it is ahead of the host time. We recommend, however, that within any particular virtual machine you use either the VMware Tools time-synchronization option, or another timekeeping utility, but not both.

Running Paravirtualized Operating Systems

ESX 3.5 includes support for virtual machine interface (VMI), used for communication between the guest operating system and the hypervisor, thus improving performance and efficiency. Enabling this support will improve the performance of virtual machines running operating systems with VMI by reducing their CPU utilization and memory space overhead (the later being especially true for SMP virtual machines). Even when only some of the virtual machines on an ESX system use VMI, the performance of all virtual machines on that server might benefit due to the hardware resources freed up for ESX to allocate elsewhere.

ESX VMI support can be enabled for a virtual machine through the VI Client by selecting **Edit virtual machine settings**, choosing the **Options** tab, selecting **Paravirtualization**, then marking the box next to **Support VMI Paravirtualization**. Enabling VMI for a virtual machine reduces the number of available PCI slots in the guest operating system running in that virtual machine by one. There is no performance benefit to enabling VMI for a virtual machine running a non-VMI operating system.

Kernel support for VMI is included in some recent Linux distributions (Ubuntu 7.04 and later and SLES 10 SP2, for example), and can be compiled into other Linux distributions, typically by compiling the kernel with `CONFIG_PARAVIRT` and `CONFIG_VMI`. No Microsoft Windows operating systems support VMI. Check the VMware *Guest Operating System Installation Guide* to see which VMI operating systems are supported in ESX. More information about VMI can be found in *Performance of VMware VMI* (http://www.vmware.com/resources/techresources/1038) and the Paravirtualization API Version 2.5 (http://www.vmware.com/pdf/vmi_specs.pdf).

Guest Operating System CPU Considerations

- In SMP guests the guest operating system can migrate processes from one vCPU to another. This migration can incur a small CPU overhead. If the migration is very frequent it might be helpful to pin guest threads or processes to specific vCPUs. (Note that this is another reason not to configure virtual machines with more vCPUs than they need.)
- Many operating systems keep time by counting timer interrupts. The timer interrupt rates vary between different operating systems and versions. For example:
  - Unpatched 2.4 and earlier Linux kernels request timer interrupts at 100 Hz (100 interrupts per second).
  - Older 2.6 Linux kernels and some 2.4 Linux kernels request interrupts at 1000 Hz.
  - Newer 2.6 Linux kernels request interrupts at 250 Hz.
  - Microsoft Windows operating system timer interrupt rates are specific to the version of Microsoft Windows and the Windows HAL that is installed. Windows systems typically use timer interrupt rates of 66 Hz or 100 Hz.
  - Running applications that make use of the Microsoft Windows multimedia timer functionality can increase the timer interrupt rate. For example, some multimedia applications or Java applications increase the timer interrupt rate to 1000 Hz.

  The total number of timer interrupts delivered to the virtual machine depends on a number of factors:
  - Virtual machines running SMP HALs/kernels (even if they are running on a UP virtual machine) require more timer interrupts than those running UP HALs/kernels.
  - The more vCPUs a virtual machine has, the more interrupts it requires.

  Delivering many virtual timer interrupts negatively impacts guest performance and increases host CPU consumption. If you have a choice, use guest operating systems that require fewer timer interrupts. For example:
  - If you have a UP virtual machine use a UP HAL/kernel.
  - In RHEL 5.1 or later use the “divider=10” kernel boot parameter to reduce the timer interrupt rate to 100 Hz.

    NOTE A bug in the RHEL 5.1 x86_64 kernel causes problems with the divider option. For RHEL 5.1 use the patch that fixes the issue at https://bugzilla.redhat.com/show_bug.cgi?id=305011. This bug is also fixed in RHEL 5.2. For more information see http://rhn.redhat.com/errata/RHSA-2007-0993.html.

  - Use a VMI-enabled operating system and enable VMI for the virtual machine (see “Running Paravirtualized Operating Systems” on page 24).

  For background information on this topic refer to *Timekeeping in Virtual Machines*, listed in “Related Publications” on page 8.

Guest Operating System Storage Considerations

- The depth of the queue of outstanding commands in the guest operating system SCSI driver can significantly impact disk performance. A queue depth that is too small, for example, limits the disk bandwidth that can be pushed through the virtual machine. See the driver-specific documentation for more information on how to adjust these settings.
- If you choose to use the BusLogic virtual SCSI adapter, and are using a Windows guest operating system, you should use the custom BusLogic driver included in the VMware Tools package.
- Large I/O requests issued by applications in the guest can be split by the guest storage driver. Changing the guest registry settings to issue larger block sizes can eliminate this splitting, thus enhancing performance. For additional information see KB article 9645697, listed in "Related Publications" on page 8.
- Make sure that file system partitions within the guest are aligned.

Guest Operating System Networking Considerations

- The default virtual network adapter emulated inside a guest is either an AMD PCnet32 device (vlance) or an Intel e1000 device (e1000). However, a third driver type, vmxnet, provides better performance than these default devices and should be used for optimal performance within any guest operating system for which it is available.

  The vmxnet driver implements an idealized network interface that passes network traffic from the virtual machine to the physical cards with minimal overhead. ESX supports vmxnet in most 32-bit guests and many 64-bit guests. To use the vmxnet network adapter, install the vmxnet driver (available in VMware Tools) on your virtual machines.

  In ESX, "NIC morphing" (also called "flexible NIC") automatically converts each vlance network device to a vmxnet device for most guests in which vmxnet is supported. This only occurs if VMware Tools is installed within the guest.

  Note that the network speeds reported by the guest network driver on the virtual machine do not necessarily reflect the actual speed of the underlying physical network interface card. For example, the vlance guest driver on a virtual machine reports a speed of 10Mbps, even if the physical card on the server is 100Mbps or 1Gbps, because the AMD PCnet cards that ESX emulates are 10Mbps. However, ESX is not limited to 10Mbps and transfers network packets as fast as the resources on the physical server machine allows.
- In ESX 3.5, there are two versions of the vmxnet virtual network adapter, *normal vmxnet* and *enhanced vmxnet*.
  - The normal vmxnet device does not support jumbo Ethernet frames or TSO.
  - A virtual machine with an enhanced vmxnet device cannot VMotion to an ESX 3.0.x host.
  - The vmxnet driver from VMware Tools included with ESX 3.5 supports both normal vmxnet and enhanced vmxnet devices. (Note that the vmxnet driver from VMware Tools included with ESX 3.0.x does not support enhanced vmxnet devices.)
  - NIC morphing installs the normal vmxnet device.
  - To use the enhanced vmxnet device you must explicitly select **Enhanced vmxnet** within the VI Client hardware configuration page.
- In ESX 3.5, the enhanced vmxnet device supports jumbo frames for better performance. (Note that the e1000 and vlance devices do not support jumbo frames.) To enable jumbo frames, set the MTU size to 9000 both in the guest and in the virtual switch configuration. The physical NICs at both ends and all the intermediate hops/routers/switches must also support jumbo frames.
- In ESX 3.5, TSO is enabled by default in the VMkernel, but is supported in guests only when they are using the enhanced vmxnet device or the e1000 device. TSO can improve performance even if the underlying hardware does not support TSO.

- In some cases, low throughput between virtual machines on the same ESX host may be caused by buffer overflows in the guest network driver. Buffer overflows require the sender to retransmit data, thereby limiting bandwidth.

  Possible workarounds are to increase the number of receive buffers, reduce the number of transmit buffers, or both. These workarounds may increase workload on the physical CPUs. The default number of receive and transmit buffers for vmxnet is 100 each. The maximum possible for vmxnet is 128. You can alter these settings by changing the buffer size defaults in the `.vmx` (configuration) files for the affected virtual machines. For additional information see KB article 1428, listed in "Related Publications" on page 8.

Virtual Infrastructure Management

This section provides guidance regarding resource management best practices. Most of the suggestions included in this section can be implemented using the VI Client connected to a VMware VirtualCenter server. Some can also be implemented using the VI Client connected to an individual ESX host.

General Resource Management Best Practices

ESX 3.5 provides several mechanisms to configure and adjust the allocation of CPU and memory resources for virtual machines running within it. Resource management configurations can have a significant impact on virtual machine performance.

This section lists resource management practices and configurations recommended by VMware for optimal performance.

- If you expect frequent changes to the total available resources, use **Shares**, not **Reservation**, to allocate resources fairly across virtual machines. If you use **Shares** and you upgrade the hardware, each virtual machine stays at the same relative priority (keeps the same number of shares) even though each share represents a larger amount of memory or CPU.
- Use **Reservation** to specify the minimum acceptable amount of CPU or memory, not the amount you would like to have available. After all resource reservations have been met, ESX allocates the remaining resources based on the number of shares and the resource limits configured for your virtual machine.

  As indicated above, reservations can be used to specify the minimum CPU and memory reserved for each virtual machine. In contrast to shares, the amount of concrete resources represented by a reservation does not change when you change the environment, for example, by adding or removing virtual machines. Don't set **Reservation** too high. A reservation that's too high can limit the number of virtual machines you can power on in a resource pool, cluster, or host.

  When specifying the reservations for virtual machines, always leave some headroom for memory virtualization overhead and migration overhead. In a DRS-enabled cluster, reservations that fully commit the capacity of the cluster or of individual hosts in the cluster can prevent DRS from migrating virtual machines between hosts. As you move closer to fully reserving all capacity in the system, it also becomes increasingly difficult to make changes to reservations and to the resource pool hierarchy without violating admission control.
- Use resource pools for delegated resource management. To fully isolate a resource pool, make the resource pool type **Fixed** and use **Reservation** and **Limit**.
- Group virtual machines for a multi-tier service into a resource pool. This allows resources to be assigned for the service as a whole.

VirtualCenter Best Practices

This section lists VirtualCenter practices and configurations recommended by VMware for optimal performance.

- Large numbers of managed hosts, managed virtual machines, and connected VMware Infrastructure Clients can affect the performance of a VirtualCenter server. Exceeding the supported maximums, though it might work, is even more likely to impact VirtualCenter performance. For more information, see *VMware Infrastructure 3 Release Notes* and *VMware Infrastructure 3 Configuration Maximums*, both listed in "Related Publications" on page 8.
- Make sure you are running VirtualCenter on hardware with sufficient CPU, memory, and storage resources. For additional information see *ESX Server 3.5 and VirtualCenter 2.5 Installation Guide*, listed in "Related Publications" on page 8.
- For the best performance, avoid attaching to a VirtualCenter server more VMware Infrastructure Clients than you need. Because the VirtualCenter server must keep all client sessions current with inventory changes, the number of VMware Infrastructure Client sessions attached to the VirtualCenter server can affect the server's CPU usage and user interface speed.
- For the best performance, avoid overly-aggressive VirtualCenter alarm settings. Each time an alarm condition is met the VirtualCenter server must take appropriate action. If this happens very often, the added load could affect system performance.
- VMware Update Manager can be run on the same system as VirtualCenter, and can use the same database. For maximum performance, however, especially on heavily-loaded VirtualCenter systems, you should consider running Update Manager on its own system, and providing it with a dedicated database.
- Similarly, VMware Converter can be run on the same system as VirtualCenter, but doing so might impact performance, especially on heavily-loaded VirtualCenter systems.

Database Considerations

VirtualCenter relies heavily on a database to store configuration information and statistics data. Due to the importance of this database to the reliability and performance of your VirtualCenter server, VMware recommends the following database practices:

- Separate the transaction logs from the datastore.
- Periodically reindex the database to maximize its performance.
- Configure the database log setting to **Normal** unless you have a specific reason to set it to **High**.
- Configure the VirtualCenter statistics level to a setting appropriate for your uses. This setting can range from 1 to 4, but a setting of 2 is recommended for most situations. Higher settings can slow the VirtualCenter system.

Distributed Resource Scheduler (DRS) Best Practices

Clustering configurations can have a significant impact on performance. This section lists Distributed Resource Scheduler (DRS) practices and configurations recommended by VMware for optimal performance.

- When deciding which hosts to group into DRS clusters, try to choose hosts that are as homogeneous as possible in terms of CPU and memory. This ensures higher performance predictability and stability. VMotion is not supported across hosts with incompatible CPU's. Hence with 'incompatible CPU' heterogeneous systems, the opportunities DRS has to improve the load balance across the cluster are limited.

  To ensure CPU compatibility make sure systems are configured with the same CPU vendor, with similar CPU families, and with matching SSE instruction-set capability. For more information on this topic see KB articles 1991, 1992, and 1993, listed in "Related Publications" on page 8.

  When heterogeneous systems have compatible CPUs, but have different CPU frequencies and/or amounts of memory, DRS generally prefers to locate virtual machines on the systems with more memory and higher CPU frequencies (all other things being equal) since those systems have more capacity to accommodate peak load.

- The more VMotion compatible ESX hosts DRS has available, the more choices it has to better balance the DRS cluster. Besides CPU incompatibility, there are other misconfigurations that can block VMotion between two or more hosts. For example, if the hosts' VMotion network adapters are not connected by a Gigabit Ethernet link then the VMotion might not occur between the hosts. Other configuration settings to check for are misconfiguration of the VMotion gateway, incompatible security policies between the source and destination host VMotion network adapter, and virtual machine network availability on the destination host. Refer to the *Resource Management Guide* and the *Server Configuration Guide* for further details.

- Don't configure more hosts in a DRS cluster than the maximum allowed in the *VMware Infrastructure 3 Configuration Maximums*, listed in "Related Publications" on page 8.

- Virtual machines with smaller memory sizes and/or fewer vCPUs provide more opportunities for DRS to migrate them in order to improve balance across the cluster. Virtual machines with larger memory size and/or more vCPUs add more constraints in migrating the virtual machines. This is one more reason to configure virtual machines with only as many vCPUs and only as much virtual memory as they need.

- Have virtual machines in DRS automatic mode when possible, as they are considered for cluster load balance migrations across the ESX hosts before the virtual machines that are not in automatic mode.

- All powered-on virtual machines consume CPU resources. Thus even idle virtual machines, though their CPU utilization is usually small, can affect DRS decisions. For this and other reasons, a marginal performance increase might be obtained by shutting down (or suspending) virtual machines that are not being used.

- The migration threshold should be set to more aggressive levels when the following conditions are satisfied:

  - If the hosts in the cluster are relatively homogeneous
  - If the virtual machines' resource utilization does not vary too much over time and you have relatively few constraints on where a virtual machine can be placed

  The migration threshold should be set to more conservative levels in the converse situations.

- The frequency with which the DRS algorithm is invoked for balancing can be controlled through the vpxd configuration file, `vpxd.cfg`, with the following option:

```
<config>
   <drm>
      <pollPeriodSec>
         300
      </pollPeriodSec>
   </drm>
</config>
```

The default frequency is 300 seconds, but it can be set to anything between 60 seconds and 3600 seconds. Users are discouraged from changing the default value. This is recommended only in specific cases where the user would like to invoke the algorithm less frequently at the cost of potential loss in throughput.

- DRS affinity rules can keep virtual machines on the same ESX host or make sure they are always on different hosts. In most cases, the default affinity settings provide the best results. In rare cases, however, specifying affinity rules can help improve performance. To change affinity settings from the VI Client select the cluster, choose the **Summary** tab, click **Edit Settings**, choose **Rules**, click **Add**, create a name for the new rule, choose one of the two settings, click **Add**, select the virtual machines to which this rule should apply, click **OK**, then click **OK** again.

  Besides the default setting, the two affinity settings are:

  - **Keep Virtual Machines Together** can improve performance due to lower latencies of communication between machines.
  - **Separate Virtual Machines** maintains maximal availability of the virtual machines. For instance, if they are both web server front ends to the same application, you might want to make sure that they don't both go down at the same time. Also co-location of I/O intensive virtual machines could end up saturating the host I/O capacity, leading to performance degradation. DRS currently does not make virtual machine placement decisions based on their I/O resources usage.

Distributed Power Management (DPM) Best Practices

Distributed Power Management (DPM) conserves power by migrating virtual machines to fewer hosts when utilizations are low. DPM is most appropriate for clusters in which composite virtual machine demand varies greatly over time; for example, clusters in which overall demand is higher during the day and significantly lower at night. If demand is consistently high relative to overall cluster capacity DPM will have little opportunity to put hosts into standby mode to save power.

Most DRS best practices (described in "Distributed Resource Scheduler (DRS) Best Practices" on page 30) are relevant to DPM as well.

- The more hosts in the cluster, the more opportunity there is for power savings through DPM. However don't exceed the maximum allowed in the *VMware Infrastructure 3 Configuration Maximums*, listed in "Related Publications" on page 8.
  - DPM can only be enabled on hosts running ESX 3.5 or later.
  - Having VMotion-compatible hosts in the cluster is important, since DPM needs to be able to migrate running virtual machines onto fewer hosts to be able to put some hosts into standby to save power.
  - Similarly, having wake-on-LAN (WOL) capable hosts in the cluster is beneficial. The ability of DPM to place a host in standby and power it on again later depends on ESX host software and on the network interface having hardware support for wake-on-lan. The more hosts in the cluster support DPM standby/power-on operations the more choices DPM has for saving power.
- Enabling DPM with all hosts in the cluster being in automatic mode gives DPM the most flexibility in choosing hosts for power down/up. For hosts in manual DPM mode, DPM must query the user about power down/up of the host. For this reason, DPM prefers choosing hosts in automatic mode over those in manual mode. If desired, DPM can be disabled for specific hosts.
- DPM considers historical demand in determining how much capacity to keep powered on and keeps some excess capacity available for changes in demand. DRS will also power on additional hosts when needed for unexpected increases in the demand of existing virtual machines or to allow virtual machine admission.
- In a cluster with High Availability (HA) enabled, DRS/DPM maintains excess powered-on capacity to meet the High Availability settings. The cluster may therefore not allow additional virtual machines to be powered on and/or some hosts may not be powered down even though the cluster may appear to be sufficiently idle. These factors should be considered when configuring HA.

ESX Benchmarking 2

This section provides guidelines on how to benchmark virtual machines running in VMware ESX 3.5. It describes how to generate fair comparisons and avoid common benchmarking pitfalls in virtualized environments.

Before running benchmark tests you should also apply all relevant recommendations from Chapter 1, "VMware Infrastructure Performance," on page 11.

This chapter is divided into the following sections:

NOTE Make sure to check the relevant VMware product end-user license agreement (EULA) before publishing any benchmarking data regarding VMware products.

Benchmarking Design

This section provides tips and guidance on the design of benchmarking experiments.

General Methodology

VMware recommends the following general guidelines for benchmarking tests:

- Before planning the testing, clearly define the parameters being measured and a metric with which to measure them (such as operations per second, jobs per hour, or average response time).
- If you are running a publicly-available benchmark, make sure you follow all the guidelines associated with configuring, running, and reporting for that benchmark and apply these guidelines to all systems on which the benchmark is being run. If you are running a custom benchmark, make sure that it incorporates the well-understood principles of benchmarking: specific test purpose, a meaningful metric (such as time, operations per second, or bytes transferred), reproducibility, and so forth.
- When trying to saturate and benchmark any specific system component (such as CPU, storage, or network), ensure that no other resource on the system is constrained in either the native or the virtual machine. Be aware that virtualization has overhead in terms of CPU, memory, etc., and provision for this overhead for components not being tested. For example, before making a native to virtual machine network-bandwidth comparison, make sure the processor load is not limiting the bandwidth achieved in either case.
- During the execution of the workload, ensure that there is no activity other than the benchmarks under consideration (virus scanner, NFS, cron jobs, etc.).

- Run the tests enough times to ensure that the percentage difference between runs is minimal and that the numbers are reproducible.
- Any report of benchmark results should include enough detail about the experimental setup for the reader to reproduce the results. Make sure your report includes at least the following information:

  Host hardware configuration:

  - Manufacturer and model of all key hardware components
  - Firmware versions for all relevant components
  - Quantity, type, and speed of CPUs
  - Amount of memory
  - Storage: Array configuration, number and speed (RPM) of disks backing the LUN, RAID configuration
  - Quantity and type of network adapters
  - ESX version and build number
  - VMFS version
  - Any non-default or configurable hardware or BIOS settings (such as memory interleaving configuration, for example)
  - Any non-default ESX settings

  Virtual machine configuration:

  - Quantity of virtual CPUs
  - Amount of virtual memory
  - Quantity and type of virtual disk: virtual SCSI adapter type, size, and disk format
  - Virtual disk mode — that is, independent persistent, independent nonpersistent, or snapshot (for further information about these virtual disk modes see "ESX Storage Considerations" on page 21).
  - Method of creation for virtual disks (GUI or command line)
  - If the virtual disks were created using the command line, list which options were used — that is, thick eager zeroed, thick lazy zeroed, or thin (for further information about these virtual disk types see "ESX Storage Considerations" on page 21).
  - Quantity and type of virtual network adapters
  - Quantity and type of virtual storage adapters
  - Shares, reservations, limits, and affinities of CPU, memory and disk
  - Guest operating system and version (including service pack) installed
  - Application information (for any applications being benchmarked, or applications used while benchmarking)
  - Benchmark configuration
  - Any non-default system tunings (for example, `/etc/sysctl.conf` in Linux or registry settings in Windows)

Timing Considerations

- Timing numbers reported within virtual machines can be inaccurate, especially when the processor is overcommitted. If you report numbers from within the guest, this fact should be noted in your results.
- One method of timing workloads is to ping an external machine and capture timestamps when that machine receives the pings. To use `ping` to time workloads, run `tcpdump` (in Linux) or `WinDUMP` (in Windows) on an external system. (The examples below show `tcpdump` in a C shell. `WinDUMP` works similarly.)

  On the external system, run:

  ```
  [timeserver]# tcpdump ip proto \\icmp and host <vm_ip>
  ```

  In this example, *<vm_ip>* is the IP address of the virtual machine under test. From within the virtual machine, before starting the benchmark and after the benchmark finishes execution, run:

  ```
  [my_vm]$ ping -c 1 <external_system_ip>
  ```

  In this case, *<external_system_ip>* is the IP address of the external system running `tcpdump` or `WinDUMP`. You can then determine the duration of the experiment by simply subtracting one timestamp from the other.

  NOTE For more information about `tcpdump` and `WinDUMP`, see: http://www.tcpdump.org/ and http://www.winpcap.org/windump/.

- In some cases your workload might not be able to use the above ping method, perhaps because the timing granularity is so small that the network latency of the pings is significant. It is also possible for the guest operating system to read the hardware timestamp counter (TSC) value as a pseudo performance counter using a VMware feature. The `rdpmc` command can be used to measure time from within the virtual machine by:

  a Adding `monitor_control.pseudo_perfctr=1` to the virtual machine `.vmx` file.

  b Issuing `rdpmc 0x10000` in the virtual machine.

  For further information, see KB article 1420, listed in "Related Publications" on page 8.

- If the benchmark takes long enough, you can also use a stopwatch to measure time.

Benchmarking Tools

This section lists some benchmarking tools that may help you compare the performance of various systems.

CPU-Related Benchmarks

- Passmark (available at http://www.passmark.com/)
- SPEC CPU2000 (available at http://www.spec.org/cpu2000/)

Memory-Related Benchmarks

- Passmark (available at http://www.passmark.com/)

Disk-Related Benchmarks

- Passmark (available at http://www.passmark.com/)
- IOMeter (available at http://www.iometer.org/)

Networking-Related Benchmarks

- Netperf (available at http://www.netperf.org/netperf/NetperfPage.html)
- SPECweb2005 (available at http://www.spec.org/web2005/)

Application Benchmarks

- SPECjbb2005 (available at http://www.spec.org/jbb2005/)
- SPECjAppServer2004 (available at http://www.spec.org/jAppServer2004/)
- SPECweb2005 (available at http://www.spec.org/web2005/)

Multiple-Virtual Machine Consolidation Benchmarks

- VMmark (available at http://www.vmware.com/products/vmmark/)

Discouraged Benchmarks

We don't recommend use of the following benchmarks, as our experience has shown that they can produce inconsistent results in virtual machines:

- Sisoft Sandra
- LMbench
- Unixbench

Hardware

This section provides guidance on selecting and configuring hardware for use in benchmarking tests.

General

- For performance comparisons, it is best to run all tests on the same system. When running on the same system is not possible, at least use identical systems.
- Run all tests on hardware supported by the VMware software you are using.

CPU Considerations During Benchmarking

While benchmarking consider the following CPU-related benchmarking best practices:

- For a fair performance comparison between native and virtual machines, configure the same number of physical CPUs in the native system as you configured virtual CPUs in the virtual machine.

  This may require reducing the number of processors in your system when doing the native tests. To do this in Windows, use the `/numproc` switch in the `boot.ini` file. (Vista does not contain a `boot.ini` file; instead use the `Msconfig.exe` or `bcdedit.exe` binaries to configure the number of processors.) To do this in Linux, use the `maxcpus` variable in either the `grub.conf` or `lilo.conf` file, depending on your Linux version.

Memory Considerations During Benchmarking

While benchmarking consider the following memory-related benchmarking best practices:

- For a fair performance comparison between native and virtual machines, make sure the amount of physical memory configured for the native system is the same as the amount of memory allocated for the virtual machine.

  This may require reducing the amount of memory in your system when doing native tests. To do this in Windows, use the `/maxmem` switch in the `boot.ini` file. (Vista does not contain a `boot.ini` file, instead use the `Msconfig.exe` or `bcdedit.exe` binaries to configure the amount of system memory.) To do this in Linux, use the `mem` variable in either the `grub.conf` or `lilo.conf` file depending on your Linux version.
- For a fair performance comparison between native and virtualized systems, the virtualized system should not have memory overcommitment.
- For a fair performance comparison between two virtualized systems, both systems should have the same level of memory overcommitment.
- Make sure the physical memory is configured identically during native tests and virtualized tests. For example, make sure the memory interleaving or NUMA settings are the same.

Storage Considerations During Benchmarking

While benchmarking consider the following storage-related benchmarking best practices:

- If virtual disks are stored on a network storage device (such as SAN or NAS), make sure that any shared resources don't impact the test, or, if possible, use a dedicated device.
- If connecting to storage through a switch (Fibre Channel or Ethernet), make sure that either your machine is the only one on the switch or that your machine is the only one active on the zone (for a Fibre Channel switch) or on the VLAN (for an Ethernet switch). Other traffic on a storage device or switch can affect your results.
- For storage tests involving a single LUN, make sure you use the same LUN for all tests. If you are testing various versions of VMFS, recreate VMFS partitions on the same LUN.
- If it is not possible to use the same LUN/file system for the test, choose another LUN that satisfies the following conditions:

- The new LUN should be on the same array.
- The new LUN should have the same number and speed (RPM) of disks as the LUN to which it is being compared.
- The RAID type should be the same as the LUN to which it is being compared.

- For storage tests involving multiple LUNs, assign a set of LUNs for the experiment and make sure the same LUNs are used for all tests.
- Zone the LUNs used for the benchmarking experiments strictly to the benchmarking host.
- Before repeating benchmark tests any storage area to which data has been written should be returned to its original state, thus ensuring that each benchmark run starts with storage in the same state. For some storage devices that display aging effects writing the original data over the modified data might not be sufficient — for such devices returning to a previous state might involve using a snapshot feature (if supported by the array vendor) or recreating the LUNs or volumes if necessary. Refer to the documentation for your storage device to determine if this consideration applies.

Networking Considerations During Benchmarking

While benchmarking consider the following networking-related benchmarking best practices:

- Avoid cross-traffic noise over the network while conducting the tests. Either use direct cables between the systems or use a private network switch. Use dedicated network interface cards on both machines for the connection.
- Use network switches instead of hubs.
- Make sure that all the networking infrastructure is appropriately rated. For example, when connecting systems containing Gigabit network interface cards, make sure to use Gigabit switches and Gigabit-rated cables.
- If possible, use similar network interface cards on systems under test so that they function well with each other. Using similar cards also helps to ensure that send and receive have similar performance. Ideally, you should use similar client and server machines as well, with similar bus architecture and configuration. The difference between PCI and PCI-X, for example, can affect networking performance.
- Don't have more physical network interface cards than absolutely necessary. This avoids the unnecessary overhead associated with processing broadcast packets, protocol control packets, and so forth.

  If your system has multiple physical network interface cards, make sure you are using the intended network interface cards for the test. If you have more than one network interface card, it is easy to enable and use the wrong one. To avoid this confusion, disable the network interface cards you do not plan to use.
- Do not use the same NIC for the service console and for your virtual machine(s).

Other Devices

- Remove or disable all devices that are not part of your experiment. These might include audio devices, optical drives (i.e., CD or DVD), floppy drives, USB ports and devices, network interface cards, and so on.

ESX and Virtual Machines

This section provides guidance on configuring and using ESX when running benchmarking experiments.

General

- It is good practice to run benchmarking tests on virtual machine that were created specifically for your performance tests (that is, purpose-built virtual machines) rather than reusing existing virtual machines. This is because existing virtual machines could have unnecessary applications or services installed that could impact performance, and could be configured with an incorrect HAL/kernel.

  Creating a purpose-built virtual machine typically involves some or all of the following steps:

  a Create a fresh virtual machine.

  b Install an operating system.

  c Install any operating system upgrades or patches desired.

  d Install any applications or benchmarks that will be used.

  e Start and stop the machine to verify its correct functioning and to ensure that all hardware has been discovered and that all software installations have completed.

  f Check the log files for this virtual machine to make sure no errors or unexpected warnings are being reported.

  Use this clean virtual machine, or clones of it, for all tests.

- Avoid running virtual machines from snapshots.

Storage

- Make sure that the same version of VMFS is used for all storage tests and use the same block size when re-creating the file system. Use the default creation options if you are not sure.
- The differences in device drivers and their parameters can impact performance and make the results harder to interpret. The fairest comparison would be to use the same device type in both virtual and native environments (e.g. if using an emulated LSILogic device in the guest, use a physical LSILogic device for the native experiments). While it will sometimes be necessary to use dissimilar device types, be aware that doing so can, in certain circumstances, make the results more difficult to analyze. For example, we've seen cases in which dissimilar device types resulted in higher performance in the virtual environment than in the native environment.

Networking

- Most modern network interface cards can operate in multiple modes (such as 10, 100, or 1000Mbps; half duplex or full duplex). Make sure the network interface cards are in full duplex mode and are configured at their maximum possible bandwidth.
- Don't change any of the default network interface card driver settings unless there is a valid reason to do so. Use the OEM recommendations.
- To establish a network between two virtual machines that reside on the same ESX host, connect both virtual machines to the same virtual switch. If the virtual machines are connected to different virtual switches, traffic will go through wire and incur unnecessary CPU and network overhead.

Guest Operating Systems

This section provides guidance regarding configuration of the guest operating systems when running benchmarking experiments.

General

- Make sure you are running general-availability (GA) releases of all operating systems and benchmarking software (rather than beta or debug versions).
- If you have tuned the native system (registry, swap space, and so forth), a similar tuning procedure should be performed on the virtual machine before comparing native to virtual machine performance. Settings on the native system may not be applicable within the virtual machine and it may therefore be necessary to repeat the full tuning process on the virtual machine.
- For workloads that include virtual machine startup (such as automated tests) make sure that the virtual machine is not detecting new hardware. If new hardware is detected, the **New Hardware Wizard** (on Windows) or `kudzu` (on Linux) may start up. In this case, complete the new hardware configuration process, then shutdown and restart the operating system, making sure the new hardware detection process is completed.
- Disable any programs or services not needed for your tests. Depending on your setup and environment, these might include screen savers, virus checkers, and so on.
- Direct the output from the application or benchmark to a log file instead of to the display. This avoids the added overhead associated with the virtualized display.

CPU

UP Versus SMP HAL/Kernel

- If comparing a native system to a virtual machine, make sure the HAL/kernel types in both are kept the same: either both UP or both SMP. When running a single-threaded benchmark, use a single-processor virtual machine. When running a multi-threaded benchmark with an SMP HAL/kernel on an SMP virtual machine, make sure there is enough parallelism to keep all the virtual CPUs busy.
- When conducting SMP scaling experiments, consider configuring both the single-processor and the multi-processor virtual machines with SMP HALs/kernels. If you keep the HAL/kernel the same in both systems you change only one variable. If your benchmarks don't involve processor scaling, it may be best to stick with single-processor configurations.
- Create fresh virtual machines specifically for benchmarking tests rather than reusing machines created previously.

  Most operating systems automatically select an appropriate HAL/kernel when they are first installed in a virtual machine. When a virtual machine with a UP HAL/kernel is reconfigured to have two processors, the typical behavior of the guest operating system is to automatically switch to an SMP HAL/kernel. If that virtual machine is later reconfigured to have a single processor, however, it typically does not automatically switch to a UP HAL/kernel.

  NOTE Some newer versions of Windows, with appropriate BIOS and hardware support, may be able to seamlessly switch from an SMP to a UP HAL.

  For additional information, see KB article 1077, listed in "Related Publications" on page 8.
- Don't overcommit CPU resources while running benchmarks. For example:
  - Avoid running a four-processor virtual machine on a dual-processor host system, even if the dual-processor host has hyper-threading (that is, four logical CPUs).
  - Avoid running two or more SMP virtual machines on a dual-processor host system, even if the dual-processor host has hyper-threading.

- Since virtualization has its own CPU overhead, make sure that CPU resources are not overcommitted on the host system.
- If benchmarking server performance, use a remote client running on a physical machine and make sure that the client is not a bottleneck.

32-bit Versus 64-bit Software

- 64-bit versions of operating systems and application software are becoming increasingly common. Make sure you use similar types of operating system and application software (that is, 32-bit or 64-bit), both natively and within the guest (for example, compare a 64-bit operating system running a 32-bit application natively to a 64-bit operating system running a 32-bit application within the guest, not to a 64-bit operating system running a 64-bit application within the guest).

Storage

- For the best performance, defragment your virtual disks from within the guest operating system before running benchmark tests.

Networking

- Remove or disable any virtual networking devices that are not required for your tests.

Glossary

A

AMD Virtualization (AMD-V)
AMD's version of virtualization assist, included in some 64-bit AMD processors. See also Virtualization Assist.

B

Ballooning
A technique used in VMware ESX to reclaim the guest memory pages that are considered the least valuable by the guest operating system. This is accomplished using the `vmmemctl` driver, which is installed as part of the VMware Tools suite.

C

Checksum Offload
An option enabling a network adapter to calculate the TCP checksum, thus reducing CPU load.

Clone
A copy of a virtual machine. See also Full Clone and Linked Clone.

Console
See VMware Virtual Machine Console.

Core
A processing unit. Often used to refer to multiple processing units in one package (a so-called "multi-core CPU"). Also used by Intel to refer to a particular family of processors (with the "Core microarchitecture"). Note that the Intel "Core" brand did not include the Core microarchitecture. Instead, this microarchitecture began shipping with the "Core 2" brand.

D

Distributed Power Management (DPM)
A feature that uses DRS to unload servers, allowing them to be placed into standby, and thereby saving power. When the load increases, the servers can be automatically brought back online.

Distributed Resource Management (DRS)
A feature that monitors utilization across resource pools and uses VMotion to move running virtual machines to other servers.

E

e1000
One of the virtual network adapters available in a virtual machine running in ESX. The e1000 adapter emulates an Intel e1000 device. See also vlance and vmxnet.

EPT (Extended Page Tables)
Intel's implementation of nested page tables.

F

Fibre Channel
A networking technology used for storage. See also iSCSI, NAS, NFS, and SAN.

Full Clone
A copy of the original virtual machine that has no further dependence on the parent virtual machine. See also Linked Clone.

G

Growable Disk
A type of virtual disk in which only as much host disk space as is needed is initially set aside, and the disk grows as the virtual machine uses the space. Also called thin disk. See also Preallocated Disk.

Guest
A virtual machine running within VMware Workstation. See also Virtual Machine.

Guest Operating System
An operating system that runs inside a virtual machine. See also Host Operating System.

H

Hardware Abstraction Layer (HAL)
A layer between the physical hardware of a computer and the software that runs on that computer designed to hide differences in the underlying hardware, thus allowing software to run on a range of different architectures without being modified for each one. Windows uses different HALs depending, among other factors, on whether the underlying system has one CPU (Uniprocessor (UP) HAL) or multiple CPUs (Symmetric Multiprocessor (SMP) HAL). See also Kernel.

Hardware Virtualization Assist
See Virtualization Assist.

High Availability (HA)
VMware High Availability is a product that continuously monitors all physical servers in a resource pool and restarts virtual machines affected by server failure.

Host Bus Adapter (HBA)
A device that connects one or more peripheral units to a computer and manages data storage and I/O processing (often for Fibre Channel, IDE, or SCSI interfaces).An HBA can be physical (attached to a host) or virtual (part of a virtual machine).

Hyper-Threading
A processor architecture feature that allows a single processor to execute multiple independent threads simultaneously. Hyper-threading was added to Intel's Xeon and Pentium® 4 processors. Intel uses the term “package” to refer to the entire chip, and “logical processor” to refer to each hardware thread.

I

Independent Virtual Disk
Independent virtual disks are not included in snapshots. Independent virtual disks can in turn be either Persistent or Nonpersistent.

iSCSI
A protocol allowing SCSI commands to be transmitted over TCP/IP (typically using ordinary Ethernet cabling). An iSCSI client is called an initiator (can be software or hardware); an iSCSI server is called a target.

J

Jumbo frames
Ethernet frames with a payload of more than 1,500 bytes. Because there is a CPU cost per packet, larger packets can reduce the CPU cost of network traffic. Not all Gigabit Ethernet cards or switches support jumbo frames. Jumbo frames are supported by the enhanced vmxnet virtual network adapter (but not by the normal vmxnet adapter).

K

Kernel
The heart of an operating system. The kernel usually includes the functionality of a Hardware Abstraction Layer (HAL). Though applying to any operating system, the term is more often used in reference to Linux than to Windows.

L

Large Pages
A feature offered by most modern processors allowing the TLB (translation lookaside buffer) to index 2MB or 4MB pages in addition to the standard 4KB pages.

Linked Clone
A copy of the original virtual machine that must have access to the parent virtual machine's virtual disk(s). The linked clone stores changes to the virtual disk(s) in a set of files separate from the parent's virtual disk files. See also Full Clone.

LUN (Logical Unit Number)
A number identifying a single logical unit, can represent a single disk or a partition on an array. Used in many storage technologies, including SCSI, iSCSI, and Fibre Channel.

M

Management User Interface (MUI)
A web-based interface to previous versions of ESX Server. For ESX Server 3.0 and later the functionality of the MUI has largely been replaced by the Virtual Infrastructure Client (VI Client).

MMU (Memory Management Unit)
Part of a computer's hardware that acts as an interface between the core of the CPU and main memory. Typically part of the CPU package in modern processors.

N

NAS
See Network Attached Storage.

Native Execution
Execution of an application directly on a physical server, as contrasted with running the application in a virtual machine.

Native System
A computer running a single operating system, and in which the applications run directly in that operating system.

Network-Attached Storage (NAS)
A storage system connected to a computer network. NAS systems are file-based, and often use TCP/IP over Ethernet (although there are numerous other variations). See also Storage Area Network.

Network File System (NFS)
A specific network file system protocol supported by many storage devices and operating systems. Traditionally implemented over a standard LAN (as opposed to a dedicated storage network).

NIC
Historically meant "network interface card." With the recent availability of multi-port network cards, as well as the inclusion of network ports directly on system boards, the term NIC is now sometimes used to mean "network interface controller" (of which there may be more than one on a physical network card or system board).

NIC Morphing
The automatic conversion on some guest operating systems from the vlance virtual network adapter to the higher-performance vmxnet virtual network adapter.

NIC Team
The association of multiple NICs with a single virtual switch to form a team. Such teams can provide passive failover and share traffic loads between members of physical and virtual networks.

Non-Uniform Memory Access (NUMA)
A computer architecture in which memory located closer to a particular processor is accessed with less delay than memory located farther from that processor.

Nonpersistent Disk
All disk writes issued by software running inside a virtual machine with a nonpersistent virtual disk appear to be written to disk, but are in fact discarded after the session is powered down. As a result, a disk in nonpersistent mode is not modified by activity in the virtual machine. See also Persistent Disk.

NPT (Nested Page Tables)
A feature of some recent CPUs that performs virtualization of the MMU in hardware, rather than in the virtualization layer. Also called RVI by AMD and EPT by Intel.

P

Pacifica
A code name for AMD's version of virtualization assist, included in some 64-bit AMD processors. See AMD Virtualization.

Paravirtualization
Paravirtualization is a technique in which a modified guest operating system kernel communicates to the hypervisor its intent to perform privileged CPU and memory operations. See also VMI.

PCI (Peripheral Component Interconnect)
A computer bus specification. Now largely being superseded by PCIe.

PCI-X ()
A computer bus specification. Similar to PCI, but twice as wide and with a faster clock. Shares some compatibility with PCI devices (that is, PCI-X cards can sometimes be used in PCI slots and PCI cards can sometimes be used in PCI-X slots).

PCIe (PCI Express)
A computer bus specification. PCIe is available in a variety of different capacities (number of "lanes"): x1, x2, x4, x8, x16, and x32. Smaller cards will fit into larger slots, but not the reverse. PCIe is not slot-compatible with either PCI or PCI-X.

Persistent Disk
All disk writes issued by software running inside a virtual machine are immediately and permanently written to a persistent virtual disk. As a result, a disk in persistent mode behaves like a conventional disk drive on a physical computer. See also Nonpersistent Disk.

Physical CPU
A processor within a physical machine. See also Virtual CPU.

Preallocated Disk
A type of virtual disk in which all the host disk space for the virtual machine is allocated at the time the virtual disk is created. See also Growable Disk.

R

RAID (Redundant Array of Inexpensive Disks)
A technology using multiple hard disks to improve performance, capacity, or reliability.

Raw Device Mapping (RDM)
The use of a mapping file in a VMFS volume to point to a raw physical device.

RVI (Rapid Virtualization Indexing)
AMD's implementation of nested page tables.

S

SAN
See Storage Area Network.

Secure Virtual Machine (SVM)
Another name for AMD's version of virtualization assist, included in some 64-bit AMD processors. See AMD Virtualization.

Service Console
The service console boots the systems and runs support, management, and administration applications.

Shadow Page Tables
A set of page tables maintained by ESX that map the guest operating system's virtual memory pages to the underlying pages on the physical machine.

Snapshot
A snapshot preserves the virtual machine just as it was when you took that snapshot — including the state of the data on all the virtual machine's disks and whether the virtual machine was powered on, powered off, or suspended. VMware Workstation lets you take a snapshot of a virtual machine at any time and revert to that snapshot at any time.

Socket
A connector that accepts a CPU package. With multi-core CPU packages, this term is no longer synonymous with the number of cores.

Storage Area Network (SAN)
A storage system connected to a dedicated network designed for storage attachment. SAN systems are usually block-based, and typically use the SCSI command set over a Fibre Channel network (though other command sets and network types exist as well). See also Network-Attached Storage.

Symmetric Multiprocessor (SMP)
A multiprocessor architecture in which two or more processors are connected to a single pool of shared memory. See also Uniprocessor (UP).

T

Template
A virtual machine that cannot be deleted or added to a team. Setting a virtual machine as a template protects any linked clones or snapshots that depend on the template from being disabled inadvertently.

Thick Disk
A virtual disk in which all the space is allocated at the time of creation.

Thin Disk
A virtual disk in which space is allocated as it is used.

Thrashing
A situation that occurs when virtual or physical memory is not large enough to hold the full working set of a workload. This mismatch can cause frequent reading from and writing to a paging file, typically located on a hard drive, which can in turn severely impact performance.

TLB (Translation Lookaside Buffer)
A CPU cache used to hold page table entries.

TSO (TCP Segmentation Offload)
A feature of some NICs that offloads the packetization of data from the CPU to the NIC. TSO is supported by the enhanced vmxnet virtual network adapter (but not by the normal vmxnet adapter).

U

Uniprocessor (UP)
A single-processor architecture. See also Symmetric Multiprocessor (SMP).

V

Vanderpool
A code name for Intel's version of virtualization assist, included in some 64-bit Intel processors. See Virtualization Technology.

Virtual CPU (vCPU)
A processor within a virtual machine. ESX 3.5 currently supports up to four virtual CPUs per virtual machine.

Virtual Disk
A virtual disk is a file or set of files that appears as a physical disk drive to a guest operating system. These files can be on the host machine or on a remote file system. When you configure a virtual machine with a virtual disk, you can install a new operating system into the disk file without the need to repartition a physical disk or reboot the host.

VMware Infrastructure Client (VI Client)
A graphical user interface used to manage ESX hosts or Virtual Center servers.

Virtual Machine
A virtualized x86 PC environment in which a guest operating system and associated application software can run. Multiple virtual machines can operate on the same host system concurrently.

Virtual SMP
A VMware proprietary technology that supports multiple virtual CPUs in a single virtual machine.

Virtual Switch (vSwitch)
A software equivalent to a traditional network switch.

Virtualization Assist
A general term for technology included in some 64-bit processors from AMD and Intel that can allow 64-bit operating systems to be run in virtual machines (where supported by VMware Workstation). More information is available in VMware knowledge base article 1901. See also AMD Virtualization and Virtualization Technology.

Virtualization Overhead
The cost difference between running an application within a virtual machine and running the same application natively. Since running in a virtual machine requires an extra layer of software, there is by necessity an associated cost. This cost may be additional resource utilization or decreased performance.

Virtualization Technology (VT)
Intel's version of virtualization assist, included in some 64-bit Intel processors. See also Virtualization Assist.

vlance
One of the virtual network adapters available in a virtual machine running in ESX. The vlance adapter emulates an AMD PCnet32 device. Note that in some cases NIC morphing can automatically convert a vlance device into a vmxnet device. See also NIC Morphing, e1000, and vmxnet.

VMFS (Virtual Machine File System)
A high performance cluster file system.

VMI (Virtual Machine Interface)
A paravirtualization interface supported by ESX. See also Paravirtualization.

VMotion
A feature allowing running virtual machines to be moved from one physical server to another with no downtime.

VMware Tools
A suite of utilities and drivers that enhances the performance and functionality of your guest operating system. Key features of VMware Tools include some or all of the following, depending on your guest operating system: an SVGA driver, a mouse driver, the VMware Tools control panel, and support for such features as shared folders, shrinking virtual disks, time synchronization with the host, VMware Tools scripts, and connecting and disconnecting devices while the virtual machine is running.

vmxnet

One of the virtual network adapters available in a virtual machine running in ESX. The vmxnet adapter is a high-performance paravirtualized device with drivers (available in VMware Tools) for many guest operating systems. There are two versions of the vmxnet virtual hardware, normal vmxnet and enhanced vmxnet, though both use the same driver. See also NIC Morphing, e1000, and vlance.

W

Wake-on-LAN

A feature allowing a computer system to be powered on or brought out of suspend by sending a command over Ethernet.

Index

T

U

V

W