Performance Best Practices for VMware vSphere™ 5.0

VMware ESXi™ 5.0

vCenter™ Server 5.0

EN-000005-04

vmware®

You can find the most up-to-date technical documentation on the VMware 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

Revision: 20110822

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

Contents

Tables

About This Book

This book, *Performance Best Practices for VMware vSphere™ 5.0*, provides performance tips that cover the most performance-critical areas of VMware vSphere 5.0. It is not intended as a comprehensive guide for planning and configuring your deployments.

Chapter 1, "Hardware for Use with VMware vSphere," on page 9, provides guidance on selecting hardware for use with vSphere.

Chapter 2, "ESXi and Virtual Machines," on page 17, provides guidance regarding VMware ESXi™ software and the virtual machines that run in it.

Chapter 3, "Guest Operating Systems," on page 37, provides guidance regarding the guest operating systems running in vSphere virtual machines.

Chapter 4, "Virtual Infrastructure Management," on page 45, provides guidance regarding infrastructure management best practices.

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

Intended Audience

This book is intended for system administrators who are planning a VMware vSphere 5.0 deployment and want to maximize its performance. The book assumes the reader is already familiar with VMware vSphere 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 vSphere Documentation

The VMware vSphere documentation consists of the combined VMware vCenter™ and VMware ESXi™ documentation set.

You can access the most current versions of the vSphere documentation 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
Italic	Document titles, glossary terms, and occasional emphasis
<Name>	Variable and parameter names

Hardware for Use with VMware vSphere 1

This chapter provides guidance on selecting and configuring hardware for use with VMware vSphere.

Validate Your Hardware

Before deploying a system we recommend the following:

- Verify that all hardware in the system is on the hardware compatibility list for the specific version of VMware software you will be running.
- Make sure that your hardware meets the minimum configuration supported by the VMware software you will be running.
- Test system memory for 72 hours, checking for hardware errors.

Hardware CPU Considerations

This section provides guidance regarding CPUs for use with vSphere 5.0.

General CPU Considerations

- When selecting hardware, it is a good idea to consider CPU compatibility for VMware vMotion™ (which in turn affects DRS) and VMware Fault Tolerance. See "VMware vMotion and Storage vMotion" on page 51, "VMware Distributed Resource Scheduler (DRS)" on page 52, and "VMware Fault Tolerance" on page 59.

Hardware-Assisted Virtualization

Most recent processors from both Intel and AMD include hardware features to assist virtualization. These features were released in two generations:

- the first generation introduced CPU virtualization;
- the second generation added memory management unit (MMU) virtualization;

For the best performance, make sure your system uses processors with second-generation hardware-assist features.

NOTE For more information about virtualization techniques, see http://www.vmware.com/files/pdf/software_hardware_tech_x86_virt.pdf.

Hardware-Assisted CPU Virtualization (VT-x and AMD-V)

The first generation of hardware virtualization assistance, VT-x from Intel and AMD-V from AMD, became available in 2006. These technologies automatically trap sensitive events and instructions, eliminating the overhead required to do so in software. This allows the use of a hardware virtualization (HV) virtual machine monitor (VMM) as opposed to a binary translation (BT) VMM. While HV outperforms BT for the vast majority of workloads, there are a few workloads where the reverse is true.

For information about configuring the way ESXi uses hardware-assisted CPU virtualization, see "Configuring ESXi for Hardware-Assisted Virtualization" on page 22.

NOTE ESXi support for AMD-V requires AMD "Barcelona" processors or later.

Hardware-Assisted MMU Virtualization (Intel EPT and AMD RVI)

More recent processors also include second-generation hardware virtualization assistance that addresses the overheads due to memory management unit (MMU) virtualization by providing hardware support to virtualize the MMU. ESXi supports this feature both in AMD processors, where it is called rapid virtualization indexing (RVI) or nested page tables (NPT), and in Intel processors, where it is called extended page tables (EPT).

Without hardware-assisted MMU virtualization, the guest operating system maintains guest virtual memory to guest physical memory address mappings in guest page tables, while ESXi 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 page tables. 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 tables. However, extra work is required to maintain the shadow page tables.

Hardware-assisted MMU virtualization allows an additional level of page tables that map guest physical memory to host physical memory addresses, eliminating the need for ESXi to maintain shadow page tables. This reduces memory consumption and speeds up workloads that cause guest operating systems to frequently modify page tables. While hardware-assisted MMU virtualization improves the performance of the vast majority of workloads, it does increase the time required to service a TLB miss, thus potentially reducing the performance of workloads that stress the TLB.

For information about configuring the way ESXi uses hardware-assisted MMU virtualization, see "Configuring ESXi for Hardware-Assisted Virtualization" on page 22.

Hardware-Assisted I/O MMU Virtualization (VT-d and AMD-Vi)

An even newer processor feature is an I/O memory management unit that remaps I/O DMA transfers and device interrupts. This can allow virtual machines to have direct access to hardware I/O devices, such as network cards, storage controllers (HBAs) and GPUs. In AMD processors this feature is called AMD I/O Virtualization (AMD-Vi or IOMMU) and in Intel processors the feature is called Intel Virtualization Technology for Directed I/O (VT-d).

For information about using hardware-assisted I/O MMU virtualization, see "DirectPath I/O" on page 34.

Hardware Storage Considerations

Back-end storage configuration can greatly affect performance. For more information on storage configuration, refer to the *vSphere Storage* document for VMware vSphere 5.0.

Lower than expected storage performance is most often the result of configuration issues with underlying storage devices rather than anything specific to ESXi.

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.

- VMware Storage vMotion performance is heavily dependent on the available storage infrastructure bandwidth. We therefore recommend you consider the information in "VMware vMotion and Storage vMotion" on page 51 when planning a deployment.
- Consider choosing storage hardware that supports VMware vStorage APIs for Array Integration (VAAI). VAAI can improve storage scalability by offloading some operations to the storage hardware instead of performing them in ESXi.

  On SANs, VAAI offers the following features:

  - Hardware-accelerated cloning (sometimes called "full copy" or "copy offload") frees resources on the host and can speed up workloads that rely on cloning, such as Storage vMotion.
  - Block zeroing speeds up creation of eager-zeroed thick disks and can improve first-time write performance on lazy-zeroed thick disks and on thin disks.
  - Scalable lock management (sometimes called "atomic test and set," or ATS) can reduce locking-related overheads, speeding up thin-disk expansion as well as many other administrative and file system-intensive tasks. This helps improve the scalability of very large deployments by speeding up provisioning operations like boot storms, expansion of thin disks, snapshots, and other tasks.
  - Thin provision UNMAP allows ESXi to return no-longer-needed thin-provisioned disk space to the storage hardware for reuse.

  On NAS devices, VAAI offers the following features:

  - Hardware-accelerated cloning (sometimes called "full copy" or "copy offload") frees resources on the host and can speed up workloads that rely on cloning. (Note that Storage vMotion does not make use of this feature on NAS devices.)
  - Space reservation allows ESXi to fully preallocate space for a virtual disk at the time the virtual disk is created. Thus, in addition to the thin provisioning and eager-zeroed thick provisioning options that non-VAAI NAS devices support, VAAI NAS devices also support lazy-zeroed thick provisioning.

  Though the degree of improvement is dependent on the storage hardware, VAAI can reduce storage latency for several types of storage operations, can reduce the ESXi host CPU utilization for storage operations, and can reduce storage network traffic.

  For information about configuring the way ESXi uses VAAI, see "ESXi Storage Considerations" on page 30.
- 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.
- If you have heavy disk I/O loads, you might need to assign separate storage processors (SPs) 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 (that is, across multiple host bus adapters (HBAs) and storage processors).
- Make sure that end-to-end Fibre Channel speeds are consistent to help avoid performance problems. For more information, see KB article 1006602.
- Configure maximum queue depth for Fibre Channel HBA cards. For additional information see VMware KB article 1267.
- 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 with other applications or systems. These types of applications perform best with dedicated connections to storage devices.
- 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 these 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.
- For iSCSI and NFS, if the network switch deployed for the data path supports VLAN, it might be beneficial to create a VLAN just for the ESXi host's vmknic and the iSCSI/NFS server. This minimizes network interference from other packet sources.
- Be aware that with software-initiated iSCSI and NFS the network protocol processing takes place on the host system, and thus these might require more CPU resources than other storage options.
- Local storage performance might be improved with write-back cache. If your local storage has write-back cache installed, make sure it's enabled and contains a functional battery module. For more information, see KB article 1006602.

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 network infrastructure between the source and destination NICs doesn't introduce bottlenecks. For example, if both NICs are 10 Gigabit, make sure all cables and switches are capable of the same speed and that the switches are not configured to a lower speed.
- For the best networking performance, we recommend 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)
  - Large receive offload (LRO)
- On some 10 Gigabit Ethernet hardware network adapters, ESXi 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.

Hardware BIOS Settings

The default hardware BIOS settings on servers might not always be the best choice for optimal performance. This section lists some of the BIOS settings you might want to check, particularly when first configuring a new server.

NOTE Because of the large number of different server models and configurations, the BIOS options discussed below might not be comprehensive for your server.

General BIOS Settings

- Make sure you are running the latest version of the BIOS available for your system.

  NOTE After updating the BIOS you should revisit your BIOS settings in case new BIOS options become available or the settings of old options have changed.

- Make sure the BIOS is set to enable all populated processor sockets and to enable all cores in each socket.
- Enable "Turbo Boost" in the BIOS if your processors support it.
- Make sure hyper-threading is enabled in the BIOS for processors that support it.
- Some NUMA-capable systems provide an option in the BIOS to disable NUMA by enabling node interleaving. In most cases you will get the best performance by disabling node interleaving (in other words, leaving NUMA enabled).
- Make sure any hardware-assisted virtualization features (VT-x, AMD-V, EPT, RVI, and so on) are enabled in the BIOS.

  NOTE After changes are made to these hardware-assisted virtualization features, some systems might need a complete power down before the changes take effect. See http://communities.vmware.com/docs/DOC-8978 for details.

- Disable from within the BIOS any devices you won't be using. This might include, for example, unneeded serial, USB, or network ports. See "ESXi General Considerations" on page 17 for further details.
- Cache prefetching mechanisms (sometimes called DPL Prefetch, Hardware Prefetcher, L2 Streaming Prefetch, or Adjacent Cache Line Prefetch) usually help performance, especially when memory access patterns are regular. When running applications that access memory randomly, however, disabling these mechanisms might result in improved performance.
- If the BIOS allows the memory scrubbing rate to be configured, we recommend leaving it at the manufacturer's default setting.

Power Management BIOS Settings

VMware ESXi includes a full range of host power management capabilities in the software that can save power when a host is not fully utilized (see "Host Power Management in ESXi" on page 23). We recommend that you configure your BIOS settings to allow ESXi the most flexibility in using (or not using) the power management features offered by your hardware, then make your power-management choices within ESXi.

- In order to allow ESXi to control CPU power-saving features, set power management in the BIOS to "OS Controlled Mode" or equivalent. Even if you don't intend to use these power-saving features, ESXi provides a convenient way to manage them.

  NOTE Some systems have Processor Clocking Control (PCC) technology, which allows ESXi to manage power on the host system even if its BIOS settings do not specify "OS Controlled mode." With this technology, ESXi does not manage P-states directly, but instead cooperates with the BIOS to determine the processor clock rate. On HP systems that support this technology, it's called Cooperative Power Management in the BIOS settings and is enabled by default.

  This feature is fully supported by ESXi and we therefore recommend enabling it (or leaving it enabled) in the BIOS.

- Availability of the C1E halt state typically provides a reduction in power consumption with little or no impact on performance. When "Turbo Boost" is enabled, the availability of C1E can sometimes even increase the performance of certain single-threaded workloads. We therefore recommend that you enable C1E in BIOS.

  However, for a very few workloads that are highly sensitive to I/O latency, especially those with low CPU utilization, C1E can reduce performance. In these cases, you might obtain better performance by disabling C1E in BIOS, if that option is available.

- C-states deeper than C1/C1E (i.e., C3, C6) allow further power savings, though with an increased chance of performance impacts. We recommend, however, that you enable all C-states in BIOS, then use ESXi host power management to control their use.

ESXi and Virtual Machines 2

This chapter provides guidance regarding ESXi software itself and the virtual machines that run in it.

ESXi General Considerations

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

- Plan your deployment by allocating enough resources for all the virtual machines you will run, as well as those needed by ESXi itself.
- Allocate to each virtual machine only as much virtual hardware as that virtual machine requires. 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 any physical hardware devices that you will not be using. These might include devices such as:
  - COM ports
  - LPT ports
  - USB controllers
  - Floppy drives
  - Optical drives (that is, CD or DVD drives)
  - Network interfaces
  - Storage controllers

  Disabling hardware devices (typically done in BIOS) can free interrupt resources. Additionally, some devices, such as USB controllers, operate on a polling scheme that consumes extra CPU resources. Lastly, some PCI devices reserve blocks of memory, making that memory unavailable to ESXi.
- Unused or unnecessary virtual hardware devices can impact performance and should be disabled.

  For example, Windows guest operating systems poll optical drives (that is, CD or DVD drives) quite frequently. When virtual machines are configured to use a physical drive, and multiple guest operating systems 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.
- ESXi 5.0 introduces virtual hardware version 8. By creating virtual machines using this hardware version, or upgrading existing virtual machines to this version, a number of additional capabilities become available. Some of these, such as support for virtual machines with up to 1TB of RAM and up to 32 vCPUs, support for virtual NUMA, and support for 3D graphics, can improve performance for some workloads.

This hardware version is not compatible with versions of ESXi prior to 5.0, however, and thus if a cluster of ESXi hosts will contain some hosts running pre-5.0 versions of ESXi, the virtual machines running on hardware version 8 will be constrained to run only on the ESXi 5.0 hosts. This could limit vMotion choices for Distributed Resource Scheduling (DRS) or Distributed Power Management (DPM).

ESXi CPU Considerations

This subsection provides guidance regarding CPU considerations in VMware ESXi.

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 remainder of the workload:

- For many workloads, CPU virtualization adds only a very small amount of overhead, resulting in performance essentially comparable to native.
- Many workloads to which CPU virtualization does add overhead are not CPU-bound—that is, most of their time is spent waiting for external events such as user interaction, device input, or data retrieval, rather than executing instructions. Because otherwise-unused CPU cycles are available to absorb the virtualization overhead, these workloads will typically have throughput similar to native, but potentially with a slight increase in latency.
- For a small percentage of workloads, for which CPU virtualization adds overhead and which are CPU-bound, there might be a noticeable degradation in both throughput and latency.

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

- In most environments ESXi allows significant levels of CPU overcommitment (that is, running more vCPUs on a host than the total number of physical processor cores in that host) without impacting virtual machine performance.

  If an ESXi host becomes CPU saturated (that is, the virtual machines and other loads on the host demand all the CPU resources the host has), latency-sensitive workloads might not perform well. In this case you might want to reduce the CPU load, for example by powering off some virtual machines or migrating them to a different host (or allowing DRS to migrate them automatically).

- It is a good idea to periodically monitor the CPU usage of the host. This can be done through the vSphere Client or by using `esxtop` or `resxtop`. Below we describe how to interpret `esxtop` data:

  - If the load average on the first line of the `esxtop` CPU panel is equal to or greater than **1**, 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` or `resxtop` see Appendix A of the VMware *Resource Management Guide*.

- Configuring a virtual machine with more virtual CPUs (vCPUs) than its workload can use might cause slightly increased resource usage, potentially impacting performance on very heavily loaded systems. Common examples of this include a single-threaded workload running in a multiple-vCPU virtual machine or a multi-threaded workload in a virtual machine with more vCPUs than the workload can effectively use.

  Even if the guest operating system doesn't use some of its vCPUs, configuring virtual machines with those vCPUs still imposes some small resource requirements on ESXi that translate to real CPU consumption on the host. For example:

  - Unused vCPUs still consume timer interrupts in some guest operating systems. (Though this is not true with "tickless timer" kernels, described in "Guest Operating System CPU Considerations" on page 39.)
  - Maintaining a consistent memory view among multiple vCPUs can consume additional resources, both in the guest operating system and in ESXi. (Though hardware-assisted MMU virtualization significantly reduces this cost.)

- Most guest operating systems execute an idle loop during periods of inactivity. Within this loop, most of these guest operating systems halt by executing the HLT or MWAIT instructions. Some older guest operating systems (including Windows 2000 (with certain HALs), Solaris 8 and 9, and MS-DOS), however, use busy-waiting within their idle loops. This results in the consumption of resources that might otherwise be available for other uses (other virtual machines, the VMkernel, and so on).

  ESXi automatically detects these loops and de-schedules the idle vCPU. Though this reduces the CPU overhead, it can also reduce the performance of some I/O-heavy workloads. For additional information see VMware KB articles 1077 and 2231.

- The guest operating system's scheduler might migrate a single-threaded workload amongst multiple vCPUs, thereby losing cache locality.

These resource requirements translate to real CPU consumption on the host.

UP vs. SMP HALs/Kernels

There are two types of hardware abstraction layers (HALs) and kernels: UP and SMP. UP historically stood for "uniprocessor," but should now be read as "single-core." SMP historically stood for "symmetric multi-processor," but should now be read as multi-core.

- Although some recent operating systems (including Windows Vista, Windows Server 2008, and Windows 7) use the same HAL or kernel for both UP and SMP installations, many operating systems can be configured to use either a UP HAL/kernel or an SMP HAL/kernel. To obtain the best performance on a single-vCPU virtual machine running an operating system that offers both UP and SMP HALs/kernels, configure the operating system with a UP HAL or kernel.

  The UP operating system versions are for single-core machines. If used on a multi-core machine, a UP operating system version will recognize and use only one of the cores. The SMP versions, while required in order to fully utilize multi-core machines, can also be used on single-core machines. Due to their extra synchronization code, however, SMP operating system versions used on single-core machines are slightly slower than UP operating system versions used on the same machines.

  NOTE When changing an existing virtual machine running Windows from multi-core to single-core the HAL usually remains SMP. For best performance, the HAL should be manually changed back to UP.

Hyper-Threading

- Hyper-threading technology (sometimes also called simultaneous multithreading, or SMT) 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 provide anywhere from a slight to a significant increase in system performance by keeping the processor pipeline busier.

  If the hardware and BIOS support hyper-threading, ESXi automatically makes use of it. For the best performance we recommend that you enable hyper-threading, which can be accomplished as follows:

  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**.

- When ESXi is running on a system with hyper-threading enabled, it assigns adjacent CPU numbers to logical processors on the same core. Thus CPUs 0 and 1 are on the first core, CPUs 2 and 3 are on the second core, and so on.

ESXi 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 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 a single SMP virtual machine, to both logical processors on one core (CPUs 0 and 1, for example) could cause poor performance.
- ESXi provides configuration parameters for controlling the scheduling of virtual machines on hyper-threaded systems (**Edit virtual machine settings** > **Resources** tab > **Advanced CPU**). When choosing hyper-threaded core sharing choices, the **Any** option (which is the default) is almost always preferred over **None**.

  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.

  For nearly all workloads, custom hyper-threading settings are not necessary. In cases of unusual workloads that interact badly with hyper-threading, however, choosing the **None** hyper-threading option might help performance. For example, even though the ESXi scheduler tries to dynamically run higher-priority virtual machines on a whole core for longer durations, you can further isolate a high-priority virtual machine from interference by other virtual machines by setting its hyper-threading sharing property to **None**.

  The trade-off for configuring **None** should also be considered. With this setting, 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** can experience performance degradation, especially on systems with a limited number of CPU cores.

Non-Uniform Memory Access (NUMA)

This section describes how to obtain the best performance when running ESXi on NUMA hardware.

NOTE A different feature, Virtual NUMA (vNUMA), allowing the creation of NUMA virtual machines, is described in "Guest Operating System CPU Considerations" on page 39.

VMware vSphere supports AMD (Opteron, Barcelona, etc.), Intel (Nehalem, Westmere, etc.), and IBM (X-Architecture) non-uniform memory access (NUMA) systems.

NOTE On some systems BIOS settings for node interleaving (also known as interleaved memory) determine whether the system behaves like a NUMA system or like a uniform memory accessing (UMA) system. If node interleaving is disabled, ESXi detects the system as NUMA and applies NUMA optimizations. If node interleaving is enabled, ESXi does not detect the system as NUMA. For more information, refer to your server's documentation.

The intelligent, adaptive NUMA scheduling and memory placement policies in ESXi can manage all virtual machines transparently, so that administrators don't need to deal with the complexity of balancing virtual machines between nodes by hand. Manual override controls are available, however, and advanced administrators might prefer to control the memory placement (through the **Memory Affinity** option) and processor utilization (through the **Only Use Processors** option).

By default, ESXi 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 cores per NUMA node.

On such systems, virtual machines can be separated into the following two categories:

- Virtual machines with a number of vCPUs equal to or less than the number of cores in each physical NUMA node. These virtual machines will be assigned to cores all within a single NUMA node and will be preferentially allocated memory local to that NUMA node. This means that, subject to memory availability, all their memory accesses will be local to that NUMA node, resulting in the lowest memory access latencies.
- Virtual machines with more vCPUs than the number of cores in each physical NUMA node (called "wide virtual machines"). These virtual machines will be assigned to two (or more) NUMA nodes and will be preferentially allocated memory local to those NUMA nodes. Because vCPUs in these wide virtual machines might sometimes need to access memory outside their own NUMA node, they might experience higher average memory access latencies than virtual machines that fit entirely within a NUMA node.

  NOTE This potential increase in average memory access latencies can be mitigated by appropriately configuring Virtual NUMA (described in "Virtual NUMA (vNUMA)" on page 39), thus allowing the guest operating system to take on part of the memory-locality management task.

Because of this difference, there can be a slight performance advantage in some environments to virtual machines configured with no more vCPUs than the number of cores in each physical NUMA node.

Conversely, some memory bandwidth bottlenecked workloads can benefit from the increased aggregate memory bandwidth available when a virtual machine that would fit within one NUMA node is nevertheless split across multiple NUMA nodes. This split can be accomplished by using the `numa.vcpu.maxPerMachineNode` option.

On hyper-threaded systems, virtual machines with a number of vCPUs greater than the number of cores in a NUMA node but lower than the number of logical processors in each physical NUMA node might benefit from using logical processors with local memory instead of full cores with remote memory. This behavior can be configured for a specific virtual machine with the `numa.vcpu.preferHT` flag.

More information about using NUMA systems with ESXi can be found in *vSphere Resource Management*.

Configuring ESXi for Hardware-Assisted Virtualization

For a description of hardware-assisted virtualization, see "Hardware-Assisted Virtualization" on page 9.

On processors that support hardware-assisted CPU virtualization but not hardware-assisted MMU virtualization, ESXi by default chooses—based on the processor model and the guest operating system—either the binary translation (BT) with software MMU (swMMU) virtual machine monitor (VMM) mode or the hardware virtualization (HV) with swMMU VMM mode.

On processors that support hardware-assisted MMU virtualization, ESXi by default chooses—based on guest operating system—among HV with hardware MMU (hwMMU) VMM mode, HV with swMMU VMM mode, and BT with swMMU VMM mode.

These defaults provide the best performance in the majority of cases. If desired, however, this behavior can be changed, as described below.

NOTE When hardware-assisted MMU virtualization is enabled for a virtual machine we strongly recommend you also—when possible—configure that virtual machine's guest operating system and applications to make use of large memory pages.

When running on a system with hardware-assisted MMU virtualization enabled, ESXi will attempt to use large pages to back the guest's memory pages even if the guest operating system and applications do not make use of large memory pages. For more information about large pages, see "Large Memory Pages for Hypervisor and Guest Operating System" on page 28.

The default behavior of ESXi regarding hardware-assisted virtualization can be changed using the vSphere Client. To do so:

1 Select the virtual machine to be configured.

2 Click **Edit virtual machine settings**, choose the **Options** tab, and select **CPU/MMU Virtualization**.

3 Select the desired radio button:

- **Automatic** allows ESXi to determine the best choice. This is the default; http://communities.vmware.com/docs/DOC-9882 provides a detailed list of which VMM is chosen for each combination of CPU and guest operating system.
- **Use software for instruction set and MMU virtualization** disables both hardware-assisted CPU virtualization (VT-x/AMD-V) and hardware-assisted MMU virtualization (EPT/RVI).
- **Use Intel® VT-x/AMD-V™ for instruction set virtualization and software for MMU virtualization** enables hardware-assisted CPU virtualization (VT-x/AMD-V) but disables hardware-assisted MMU virtualization (EPT/RVI).
- **Use Intel® VT-x/AMD-V™ for instruction set virtualization and Intel® EPT/AMD RVI for MMU virtualization** enables both hardware-assisted CPU virtualization (VT-x/AMD-V) and hardware-assisted MMU virtualization (EPT/RVI).

NOTE Some combinations of CPU, guest operating system, and other variables (i.e, turning on Fault Tolerance) limit these options. If the setting you select is not available for your particular combination, the setting will be ignored and **Automatic** will be used.

Host Power Management in ESXi

Host power management in ESXi 5.0 is designed to reduce the power consumption of ESXi hosts while they are running.

NOTE A very different power-saving technique, Distributed Power Management, attempts to shut down ESXi hosts when they are not needed. This is described in "VMware Distributed Power Management (DPM)" on page 56.

Power Policy Options in ESXi

ESXi 5.0 offers the following power policy options:

- **High performance**
  This power policy maximizes performance, using no power management features.
- **Balanced**
  This power policy (the default in ESXi 5.0) is designed to reduce host power consumption while having little or no impact on performance.
- **Low power**
  This power policy is designed to more aggressively reduce host power consumption at the risk of reduced performance.
- **Custom**
  This power policy starts out the same as **Balanced**, but allows for the modification of individual parameters.

For details on selecting a power policy, search for "Select a CPU Power Management Policy" in the *vSphere Resource Management* guide.

For details on modifying individual power management parameters for the **Custom** policy, search for "Using CPU Power Management Policies" in the *vSphere Resource Management* guide.

Be sure, also, that your server's BIOS settings are configured correctly, as described in "Hardware BIOS Settings" on page 14.

When Some Power Policy Options Are Unavailable

In some cases, **Active Policy** will indicate **Not supported** and the **Properties...** option will not be clickable. This indicates that either:

1 The underlying hardware doesn't support power management.
In this case ESXi will function as if it were set to **High performance**.

or:

2 The underlying hardware *does* support power management but, because "OS Controlled Mode" or its equivalent is not enabled in the BIOS, ESXi does not have control of these options.
In this case, also, ESXi will function as if it were set to **High performance**, but the BIOS settings might override ESXi and cause the system to use some power management features.

In other cases, some of the power policy options will not be available. This indicates that either the underlying hardware doesn't support those options or the BIOS is not configured to allow them.

More information about BIOS settings can be found in "Hardware BIOS Settings" on page 14.

Choosing a Power Policy

While the default power policy in ESX/ESXi 4.1 was **High performance**, in ESXi 5.0 the default is now **Balanced**. This power policy will typically not impact the performance of CPU-intensive workloads. Rarely, however, the **Balanced** policy might slightly reduce the performance of latency-sensitive workloads. In these cases, selecting the **High performance** power policy will provide the full hardware performance. For more information on this, see "Running Network Latency Sensitive Applications" on page 35.

ESXi Memory Considerations

This subsection provides guidance regarding memory considerations in ESXi.

Memory Overhead

Virtualization causes an increase in the amount of physical memory required due to the extra memory needed by ESXi 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 VMkernel and various host agents (hostd, vpxa, etc.).

2 An additional memory space overhead for each virtual machine.

The per-virtual-machine memory space overhead can be further divided into the following categories:

- Memory reserved for the virtual machine executable (VMX) process.
  This is used for data structures needed to bootstrap and support the guest (i.e., thread stacks, text, and heap).
- Memory reserved for the virtual machine monitor (VMM).
  This is used for data structures required by the virtual hardware (i.e., TLB, memory mappings, and CPU state).
- Memory reserved for various virtual devices (i.e., mouse, keyboard, SVGA, USB, etc.)
- Memory reserved for other subsystems, such as the kernel, management agents, etc.

The amounts of memory reserved for these purposes depend on a variety of factors, including the number of vCPUs, the configured memory for the guest operating system, whether the guest operating system is 32-bit or 64-bit, and which features are enabled for the virtual machine. For more information about these overheads, see *vSphere Resource Management*.

While the VMM and virtual device memory needs are fully reserved at the time the virtual machine is powered on, a new feature in ESXi 5.0, called VMX swap, can reduce the VMX memory reservation from about 50MB or more per virtual machine to about 10MB per virtual machine, allowing the remainder to be swapped out when host memory is overcommitted. This represents a significant reduction in the overhead memory reserved for each virtual machine.

The creation of a VMX swap file for each virtual machine (and thus the reduction in host memory reservation for that virtual machine) is automatic. By default, this file is 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) but a different location can be set with `sched.swap.vmxSwapDir`.

The amount of disk space required varies, but even for a large virtual machine is typically less than 100MB (typically less than 300MB if the virtual machine is configured for 3D graphics). VMX swap file creation can be disabled by setting `sched.swap.vmxSwapEnabled` to `FALSE`.

NOTE The VMX swap file is entirely unrelated to swap to host cache or regular host-level swapping, both of which are described in "Memory Overcommit Techniques" on page 26.

In addition, ESXi also provides optimizations, such as page sharing (see "Memory Overcommit Techniques" on page 26), 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.

Memory Sizing

Carefully select the amount of 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 thrashing.

- You should also avoid over-allocating memory. Allocating more memory than needed unnecessarily increases the virtual machine memory overhead, thus consuming memory that could be used to support more virtual machines.

Memory Overcommit Techniques

- ESXi uses five memory management mechanisms—page sharing, ballooning, memory compression, swap to host cache, and regular swapping—to dynamically reduce the amount of machine physical memory required for each virtual machine.
  - **Page Sharing:** ESXi uses a proprietary technique to transparently and securely share memory pages between virtual machines, thus eliminating redundant copies of memory pages. In most cases, page sharing is used by default regardless of the memory demands on the host system. (The exception is when using large pages, as discussed in "Large Memory Pages for Hypervisor and Guest Operating System" on page 28.)
  - **Ballooning:** If the virtual machine's memory usage approaches its memory target, ESXi will use ballooning to reduce that virtual machine's memory demands. Using a VMware-supplied vmmemctl module installed in the guest operating system as part of VMware Tools suite, ESXi can cause the guest operating system to relinquish the memory pages it considers least valuable. 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.
  - **Memory Compression**: If the virtual machine's memory usage approaches the level at which host-level swapping will be required, ESXi will use memory compression to reduce the number of memory pages it will need to swap out. Because the decompression latency is much smaller than the swap-in latency, compressing memory pages has significantly less impact on performance than swapping out those pages.
  - **Swap to Host Cache:** If memory compression doesn't keep the virtual machine's memory usage low enough, ESXi will next forcibly reclaim memory using host-level swapping to a host cache (if one has been configured). Swap to host cache is a new feature in ESXi 5.0 that allows users to configure a special swap cache on SSD storage. In most cases this host cache (being on SSD) will be much faster than the regular swap files (typically on hard disk storage), significantly reducing access latency. Thus, although some of the pages ESXi swaps out might be active, swap to host cache has a far lower performance impact than regular host-level swapping.
  - **Regular Swapping:** If the host cache becomes full, or if a host cache has not been configured, ESXi will next reclaim memory from the virtual machine by swapping out pages to a regular swap file. Like swap to host cache, some of the pages ESXi swaps out might be active. Unlike swap to host cache, however, this mechanism can cause virtual machine performance to degrade significantly due to its high access latency.

  For further information about memory management, see *Understanding Memory Resource Management in VMware vSphere 5.0*.
- While ESXi uses page sharing, ballooning, memory compression, and swap to host cache to allow significant memory overcommitment, usually with little or no impact on performance, you should avoid overcommitting memory to the point that active memory pages are swapped out with regular host-level swapping.

If you suspect that memory overcommitment is beginning to affect the performance of a virtual machine you can:

a In the vSphere Client, select the virtual machine in question, select the **Performance** tab, then look at the value of **Memory Balloon (Average)**.
An absence of ballooning suggests that ESXi is not under heavy memory pressure and thus memory overcommitment is not affecting the performance of that virtual machine.

NOTE This indicator is only meaningful if the balloon driver is installed in the virtual machine and is not prevented from working.

NOTE Some ballooning is quite normal and not indicative of a problem.

b In the vSphere Client, select the virtual machine in question, select the **Performance** tab, then compare the values of **Consumed Memory** and **Active Memory**. If consumed is higher than active, this suggests that the guest is currently getting all the memory it requires for best performance.

c In the vSphere Client, select the virtual machine in question, select the **Performance** tab, then look at the values of **Swap-In** and **Decompress**.
Swapping in and decompressing at the host level indicate more significant memory pressure.

d Check for guest operating system swap activity within that virtual machine.
This can indicate that ballooning might be starting to impact performance, though swap activity can also be related to other issues entirely within the guest (or can be an indication that the guest memory size is simply too small).

Memory Swapping Optimizations

As described in "Memory Overcommit Techniques," above, ESXi supports a bounded amount of memory overcommitment without host-level swapping. If the overcommitment is large enough that the other memory reclamation techniques are not sufficient, however, ESXi uses host-level memory swapping, with a potentially significant effect on virtual machine performance. (Note that this swapping 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.

- Because ESXi uses page sharing, ballooning, and memory compression to reduce the need for host-level memory swapping, don't disable these techniques.
- If you choose to overcommit memory with ESXi, be sure you have sufficient swap space on your ESXi system. At the time a virtual machine is first powered on, ESXi creates a swap file for that virtual machine equal in size to the difference between the virtual machine's configured memory size and its memory reservation. The available disk space must therefore be at least this large (plus the space required for VMX swap, as described in "Memory Overhead" on page 25).
- You can optionally configure a special host cache on an SSD (if one is installed) to be used for the new swap to host cache feature. This swap cache will be shared by all the virtual machines running on the host, and host-level swapping of their most active pages will benefit from the low latency of SSD. This allows a relatively small amount of SSD storage to have a potentially significant performance impact

  NOTE Using swap to host cache and putting the regular swap file in SSD (as described below) are two different approaches for improving host swapping performance. Swap to host cache makes the best use of potentially limited SSD space while also being optimized for the large block sizes at which some SSDs work best.

- Even if an overcommitted host uses swap to host cache, it still needs to create regular swap files. Swap to host cache, however, makes much less important the speed of the storage on which the regular swap files are placed.

If a host does *not* use swap to host cache, and memory is overcommitted to the point of thrashing, placing the virtual machine swap files on low latency, high bandwidth storage systems will result in the smallest performance impact from swapping.

- The best choice will usually be local SSD.

  NOTE Placing the regular swap file in SSD and using swap to host cache in SSD (as described above) are two different approaches to improving host swapping performance. Because it is unusual to have enough SSD space for a host's entire swap file needs, we recommend using local SSD for swap to host cache.

- If the host doesn't have local SSD, the second choice would be remote SSD. This would still provide the low-latencies of SSD, though with the added latency of remote access.
- If you can't use SSD storage, place the regular swap file on the fastest available storage. This might be a Fibre Channel SAN array or a fast local disk.
- Placing swap files on local storage (whether SSD or hard drive) could potentially reduce vMotion performance. This is because if a virtual machine has memory pages in a local swap file, they must be swapped in to memory before a vMotion operation on that virtual machine can proceed.
- Regardless of the storage type or location used for the regular swap file, for the best performance, and to avoid the possibility of running out of space, swap files should not be placed on thin-provisioned storage.

The regular swap file location for a specific virtual machine can be set in the vSphere Client (select **Edit virtual machine settings**, choose the **Options** tab, and under **Advanced** select **Swapfile location**). 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.

- Host-level memory swapping can be avoided for a specific virtual machine by using the vSphere 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 ESXi 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 reserved amount.

  NOTE The memory reservation is a guaranteed lower bound on the amount of physical memory ESXi reserves for the virtual machine. It can be configured through the vSphere 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).

Large Memory Pages for Hypervisor and Guest Operating System

In addition to the usual 4KB memory pages, ESXi also provides 2MB memory pages (commonly referred to as "large pages"). By default ESXi 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 each of them to use large memory pages.

Use of large pages can also change page sharing behavior. While ESXi ordinarily uses page sharing regardless of memory demands, it does not share large pages. Therefore with large pages, page sharing might not occur until memory overcommitment is high enough to require the large pages to be broken into small pages. For further information see VMware KB articles 1021095 and 1021896.

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).

Hardware-Assisted MMU Virtualization

Hardware-assisted MMU virtualization is a technique that virtualizes the CPU's memory management unit (MMU). For a description of hardware-assisted MMU virtualization, see "Hardware-Assisted MMU Virtualization (Intel EPT and AMD RVI)" on page 10; for information about configuring the way ESXi uses hardware-assisted MMU virtualization, see "Configuring ESXi for Hardware-Assisted Virtualization" on page 22.

ESXi Storage Considerations

This subsection provides guidance regarding storage considerations in ESXi.

VMware vStorage APIs for Array Integration (VAAI)

- For the best storage performance, consider using VAAI-capable storage hardware. The performance gains from VAAI (described in "Hardware Storage Considerations" on page 11) can be especially noticeable in VDI environments (where VAAI can improve boot-storm and desktop workload performance), large data centers (where VAAI can improve the performance of mass virtual machine provisioning and of thin-provisioned virtual disks), and in other large-scale deployments.

  If your storage hardware supports VAAI, ESXi will automatically recognize and use these capabilities. To confirm that your hardware does support VAAI and that it is being used, follow the instructions in VMware KB article 1021976. If you determine that VAAI is not being used, contact your storage hardware vendor to see if a firmware upgrade is required for VAAI support.

LUN Access Methods, Virtual Disk Modes, and Virtual Disk Types

- ESXi 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 metadata used to manage and redirect disk accesses to the physical device. Ordinary VMFS is recommended for most virtual disk storage, but raw disks might be desirable in some cases.

  You can use RDMs in virtual compatibility mode or physical compatibility mode:

  - Virtual mode specifies full virtualization of the mapped device, allowing the guest operating system to treat the RDM like any other virtual disk file in a VMFS volume.
  - Physical mode specifies minimal SCSI virtualization of the mapped device, allowing the greatest flexibility for SAN management software or other SCSI target-based software running in the virtual machine.

  For more information about RDM, see *vSphere Storage*.

- ESXi supports three virtual disk modes: Independent persistent, Independent nonpersistent, and Snapshot.

  NOTE An independent disk does not participate in virtual machine snapshots. That is, the disk state will be independent of the snapshot state and creating, consolidating, or reverting to snapshots will have no effect on the disk.

  These modes have the following characteristics:

  - **Independent persistent** – In this mode changes are persistently written to the disk, providing the best performance.
  - **Independent nonpersistent** – In this mode disk writes are appended to a redo log. The redo log is erased when you power off the virtual machine or revert to a snapshot, causing any changes made to the disk to be discarded. When a virtual machine reads from an independent nonpersistent mode disk, ESXi 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. Because of 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, performance might not be as high as independent persistent mode disks.
  - **Snapshot** – In this mode disk writes are appended to a redo log that persists between power cycles. Thus, like the independent nonpersistent mode disks described above, snapshot mode disk performance might not be as high as independent persistent mode disks.

- ESXi supports multiple virtual disk types:

- **Thick** – Thick virtual 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 increases the time it takes to create the disk, but results in the best performance, even on the first write to each block.

    NOTE The use of VAAI-capable SAN storage (described in "Hardware Storage Considerations" on page 11) can speed up eager-zeroed thick disk creation by offloading zeroing operations to the storage array.

  - **Lazy-zeroed** – A lazy-zeroed thick disk has all space allocated at the time of creation, but each block is zeroed only 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 on eager-zeroed thick disks.

    NOTE The use of VAAI-capable SAN or NAS storage can improve lazy-zeroed thick disk first-time-write performance by offloading zeroing operations to the storage array.

- **Thin** – Space required for a thin-provisioned virtual disk is allocated and zeroed upon first write, as opposed to upon creation. There is a higher I/O cost (similar to that of lazy-zeroed thick disks) during the first write to an unwritten file block, but on subsequent writes thin-provisioned disks have the same performance as eager-zeroed thick disks.

  NOTE The use of VAAI-capable SAN storage can improve thin-provisioned disk first-time-write performance by improving file locking capability and offloading zeroing operations to the storage array.

All three types of virtual disks can be created using the vSphere Client (**Edit virtual machine settings** > **Hardware** tab > **Add...** > **Hard Disk**). At the **Create a Disk** window:

- Selecting **Flat Disk** creates a lazy-zeroed thick disk.
- Selecting **Thick Provision** creates an eager-zeroed thick disk.
- Selecting **Thin Provision** creates a thin disk.

Virtual disks can also be created from the vSphere Command-Line Interface (vSphere CLI) using `vmkfstools`. For details refer to *vSphere Command-Line Interface Reference* and the `vmkfstools` man page.

NOTE Virtual disks created on NFS volumes can be either thin-provisioned or eager-zeroed thick unless the NAS device supports VAAI, which can add support for lazy-zeroed thick provisioning.

Partition Alignment

The alignment of file system partitions can impact performance. VMware makes the following recommendations for VMFS partitions:

- Like other disk-based filesystems, VMFS filesystems suffer a performance penalty when the partition is unaligned. Using the vSphere Client to create VMFS partitions avoids this problem since, beginning with ESXi 5.0, it automatically aligns VMFS3 or VMFS5 partitions along the 1MB boundary.

  NOTE If a VMFS3 partition was created using an earlier version of ESX/ESXi that aligned along the 64KB boundary, and that filesystem is then upgraded to VMFS5, it will retain its 64KB alignment. 1MB alignment can be obtained by deleting the partition and recreating it using the vSphere Client and an ESXi 5.0 host.

- 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, carefully evaluate the performance impact of the unaligned VMFS partition on your particular workload. The degree of improvement from alignment is highly dependent on workloads and array types. You might want to refer to the alignment recommendations from your array vendor for further information.

SAN Multipathing

- By default, ESXi uses the **Most Recently Used** (MRU) path policy for devices on Active/Passive storage arrays. Do not use **Fixed** path policy for Active/Passive storage arrays to avoid LUN path thrashing. For more information, see the *VMware SAN Configuration Guide*.

  NOTE With some Active/Passive storage arrays that support ALUA (described below) ESXi can use **Fixed** path policy without risk of LUN path thrashing.

- By default, ESXi uses the **Fixed** path policy for devices on Active/Active storage arrays. 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*.
- In addition to the Fixed and MRU path policies, ESXi can also use the **Round Robin** path policy, which can improve storage performance in some environments. Round Robin policy provides load balancing by cycling I/O requests through all Active paths, sending a fixed (but configurable) number of I/O requests through each one in turn.
- If your storage array supports ALUA (Asymmetric Logical Unit Access), enabling this feature on the array can improve storage performance in some environments. ALUA, which is automatically detected by ESXi, allows the array itself to designate paths as "Active Optimized." When ALUA is combined with the Round Robin path policy, ESXi cycles I/O requests through these Active Optimized paths.

Storage I/O Resource Allocation

VMware vSphere provides mechanisms to dynamically allocate storage I/O resources, allowing critical workloads to maintain their performance even during peak load periods when there is contention for I/O resources. This allocation can be performed at the level of the individual host or for an entire datastore. Both methods are described below.

- The storage I/O resources available to an ESXi host can be proportionally allocated to the virtual machines running on that host by using the vSphere Client to set disk shares for the virtual machines (select **Edit virtual machine settings**, choose the **Resources** tab, select **Disk**, then change the **Shares** field).
- The maximum storage I/O resources available to each virtual machine can be set using limits. These limits, set in I/O operations per second (IOPS), can be used to provide strict isolation and control on certain workloads. By default, these are set to **unlimited**. When set to any other value, ESXi enforces the limits even if the underlying datastores are not fully utilized.
- An entire datastore's I/O resources can be proportionally allocated to the virtual machines accessing that datastore using Storage I/O Control (SIOC). When enabled, SIOC evaluates the disk share values set for all virtual machines accessing a datastore and allocates that datastore's resources accordingly. SIOC can be enabled using the vSphere Client (select a datastore, choose the **Configuration** tab, click **Properties...** (at the far right), then under **Storage I/O Control** add a checkmark to the **Enabled** box).

With SIOC disabled (the default), all hosts accessing a datastore get an equal portion of that datastore's resources. Any shares values determine only how each host's portion is divided amongst its virtual machines.

With SIOC enabled, the disk shares are evaluated globally and the portion of the datastore's resources each host receives depends on the sum of the shares of the virtual machines running on that host relative to the sum of the shares of all the virtual machines accessing that datastore.

General ESXi Storage Recommendations

- 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 operating system.
- Make sure that the average latency for storage devices is not too high. This latency can be seen in `esxtop` (or `resxtop`) by looking at the **GAVG/cmd** metric. A reasonable upper value for this metric depends on your storage subsystem. If you use SIOC, you can use your SIOC setting as a guide — your **GAVG/cmd** value should be well below your SIOC setting. The default SIOC setting is 30 ms, but if you have very fast storage (SSDs, for example) you might have reduced that value. For further information on average latency see VMware KB article 1008205.
- You can adjust the maximum number of outstanding disk requests per VMFS volume, which can help equalize the bandwidth across virtual machines using that volume. For further information see VMware KB article 1268.
- If you will not be using Storage I/O Control and often observe QFULL/BUSY errors, enabling and configuring queue depth throttling might improve storage performance. This feature can significantly reduce the number of commands returned from the array with a QFULL/BUSY error. If any system accessing a particular LUN or storage array port has queue depth throttling enabled, all systems (both ESX hosts and other systems) accessing that LUN or storage array port should use an adaptive queue depth algorithm. Queue depth throttling is not compatible with Storage DRS. For more information about both QFULL/BUSY errors and this feature see KB article 1008113.

Running Storage Latency Sensitive Applications

By default the ESXi storage stack is configured to drive high storage throughout at low CPU cost. While this default configuration provides better scalability and higher consolidation ratios, it comes at the cost of potentially higher storage latency. Applications that are highly sensitive to storage latency might therefore benefit from the following:

- Adjust the host power management settings:

  Some of the power management features in newer server hardware can increase storage latency. Disable them as follows:

  - Set the ESXi host power policy to **Maximum performance** (as described in "Host Power Management in ESXi" on page 23; this is the preferred method) or disable power management in the BIOS (as described in "Power Management BIOS Settings" on page 14).
  - Disable C1E and other C-states in BIOS (as described in "Power Management BIOS Settings" on page 14).
  - Enable Turbo Boost in BIOS (as described in "General BIOS Settings" on page 14).

ESXi Networking Considerations

This subsection provides guidance regarding networking considerations in ESXi.

General ESXi Networking Considerations

- 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 limit 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 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 ESXi 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.

Network I/O Control (NetIOC)

Network I/O Control (NetIOC) allows the allocation of network bandwidth to network resource pools. You can either select from among seven predefined resource pools (Fault Tolerance traffic, iSCSI traffic, vMotion traffic, management traffic, vSphere Replication (VR) traffic, NFS traffic, and virtual machine traffic) or you can create user-defined resource pools. Each resource pool is associated with a portgroup and, optionally, assigned a specific 802.1p priority level.

Network bandwidth can be allocated to resource pools using either shares or limits:

- Shares can be used to allocate to a resource pool a proportion of a network link's bandwidth equivalent to the ratio of its shares to the total shares. If a resource pool doesn't use its full allocation, the unused bandwidth is available for use by other resource pools.
- Limits can be used to set a resource pool's maximum bandwidth utilization (in Mbps) from a host through a specific virtual distributed switch (vDS). These limits are enforced even if a vDS is not saturated, potentially limiting a resource pool's bandwidth while simultaneously leaving some bandwidth unused. On the other hand, if a resource pool's bandwidth utilization is less than its limit, the unused bandwidth *is* available to other resource pools.

NetIOC can guarantee bandwidth for specific needs and can prevent any one resource pool from impacting the others.

For further information about NetIOC, see *VMware Network I/O Control: Architecture, Performance and Best Practices*.

DirectPath I/O

vSphere DirectPath I/O leverages Intel VT-d and AMD-Vi hardware support (described in "Hardware-Assisted I/O MMU Virtualization (VT-d and AMD-Vi)" on page 10) to allow guest operating systems to directly access hardware devices. In the case of networking, DirectPath I/O allows the virtual machine to access a physical NIC directly rather than using an emulated device (E1000) or a para-virtualized device (VMXNET, VMXNET3). While DirectPath I/O provides limited increases in throughput, it reduces CPU cost for networking-intensive workloads.

DirectPath I/O is not compatible with certain core virtualization features, however. This list varies with the hardware on which ESXi is running:

- New for vSphere 5.0, when ESXi is running on certain configurations of the Cisco Unified Computing System (UCS) platform, DirectPath I/O for networking is compatible with vMotion, physical NIC sharing, snapshots, and suspend/resume. It is not compatible with Fault Tolerance, NetIOC, memory overcommit, VMCI, or VMSafe.

- For server hardware other than the Cisco UCS platform, DirectPath I/O is not compatible with vMotion, physical NIC sharing, snapshots, suspend/resume, Fault Tolerance, NetIOC, memory overcommit, or VMSafe.

Typical virtual machines and their workloads don't require the use of DirectPath I/O. For workloads that are very networking intensive and don't need the core virtualization features mentioned above, however, DirectPath I/O might be useful to reduce CPU usage.

SplitRx Mode

SplitRx mode, a new feature in ESXi 5.0, uses multiple physical CPUs to process network packets received in a single network queue. This feature can significantly improve network performance for certain workloads. These workloads include:

- Multiple virtual machines on one ESXi host all receiving multicast traffic from the same source. (SplitRx mode will typically improve throughput and CPU efficiency for these workloads.)
- Traffic via the vNetwork Appliance (DVFilter) API between two virtual machines on the same ESXi host. (SplitRx mode will typically improve throughput and maximum packet rates for these workloads.)

This feature, which is supported only for VMXNET3 virtual network adapters, is individually configured for each virtual NIC using the `ethernet`*X*`.emuRxMode` variable in each virtual machine's `.vmx` file (where *X* is replaced with the network adapter's ID).

The possible values for this variable are:

- `ethernet`*X*`.emuRxMode = "0"`
  This value disables splitRx mode for `ethernet`*X*.
- `ethernet`*X*`.emuRxMode = "1"`
  This value enables splitRx mode for `ethernet`*X*.

To change this variable through the vSphere Client:

1 Select the virtual machine you wish to change, then click **Edit virtual machine settings**.

2 Under the **Options** tab, select **General**, then click **Configuration Parameters**.

3 Look for **ethernet*X*.emuRxMode** (where ***X*** is the number of the desired NIC). If the variable isn't present, click **Add Row** and enter it as a new variable.

4 Click on the value to be changed and configure it as you wish.

The change will not take effect until the virtual machine has been restarted.

Running Network Latency Sensitive Applications

By default the ESXi network stack is configured to drive high network throughout at low CPU cost. While this default configuration provides better scalability and higher consolidation ratios, it comes at the cost of potentially higher network latency. Applications that are highly sensitive to network latency might therefore benefit from the following:

- Use VMXNET3 virtual network adapters (see "Guest Operating System Networking Considerations" on page 42).
- Adjust the host power management settings:

  Some of the power management features in newer server hardware can increase network latency. Disable them as follows:

  - Set the ESXi host power policy to **Maximum performance** (as described in "Host Power Management in ESXi" on page 23; this is the preferred method) or disable power management in the BIOS (as described in "Power Management BIOS Settings" on page 14).
  - Disable C1E and other C-states in BIOS (as described in "Power Management BIOS Settings" on page 14).

- Enable Turbo Boost in BIOS (as described in "General BIOS Settings" on page 14).

- Disable VMXNET3 virtual interrupt coalescing for the desired NIC.

  In some cases this can improve performance for latency-sensitive applications. In other cases—most notably applications with high numbers of outstanding network requests—it can reduce performance.

  To do this through the vSphere Client:

  a Select the virtual machine you wish to change, then click **Edit virtual machine settings**.

  b Under the **Options** tab, select **General**, then click **Configuration Parameters**.

  c Look for **ethernet*X*.coalescingScheme** (where **X** is the number of the desired NIC). If the variable isn't present, click **Add Row** and enter it as a new variable.

  d Click on the value to be changed and set it to **`disabled`**.

  The change will not take effect until the virtual machine has been restarted.

Guest Operating Systems 3

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

Guest Operating System General Considerations

- Use guest operating systems that are supported by ESXi. 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 ESXi 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 ESXi. Ballooning (described in "Memory Overcommit Techniques" on page 26) 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 virus scanning programs in virtual machines to run at off-peak hours. Avoid scheduling them to run simultaneously in multiple virtual machines on the same ESXi 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 virus scanning, where the load is predictable, this is easily achieved by scheduling the jobs appropriately.

- For the most accurate timekeeping, consider configuring your guest operating system to use NTP, Windows Time Service, the VMware Tools time-synchronization option, or another timekeeping utility suitable for your operating system.

  NOTE As of the version included in ESXi 5.0, the VMware Tools time-synchronization option is a suitable choice. Versions prior to ESXi 5.0 were not designed for the same level of accuracy and do 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.

  For additional information about best practices for timekeeping within virtual machines, see VMware KB articles 1318 and 1006427.

Measuring Performance in Virtual Machines

Be careful when measuring performance from within virtual machines.

- Timing numbers measured from within virtual machines can be inaccurate, especially when the processor is overcommitted.

  NOTE One possible approach to this issue is to use a guest operating system that has good timekeeping behavior when run in a virtual machine, such as a guest that uses the NO_HZ kernel configuration option (sometimes called "tickless timer"). More information about this topic can be found in *Timekeeping in VMware Virtual Machines* (http://www.vmware.com/files/pdf/Timekeeping-In-VirtualMachines.pdf).

- Measuring performance from within virtual machines can fail to take into account resources used by ESXi for tasks it offloads from the guest operating system, as well as resources consumed by virtualization overhead.

Measuring resource utilization using tools at the ESXi level, such as the vSphere Client, VMware Tools, `esxtop`, or `resxtop`, can avoid these problems.

Guest Operating System CPU Considerations

- In SMP virtual machines 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 typically request timer interrupts at 100 Hz (that is, 100 interrupts per second), though this can vary with version and distribution.
  - 2.6 Linux kernels have used a variety of timer interrupt rates, including 100 Hz, 250 Hz, and 1000 Hz, again varying with version and distribution.
  - The most recent 2.6 Linux kernels introduce the NO_HZ kernel configuration option (sometimes called "tickless timer") that uses a variable timer interrupt rate.
  - 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 a base timer interrupt rate of 64 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 approximately 1000 Hz.

  In addition to the timer interrupt rate, the total number of timer interrupts delivered to a virtual machine also depends on a number of other 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 virtual machine 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 some Linux versions, such as RHEL 5.1 and later, the "divider=10" kernel boot parameter reduces the timer interrupt rate to one tenth its default rate. See VMware KB article 1006427 for further information.

    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.

  - Kernels with tickless-timer support (NO_HZ kernels) do not schedule periodic timers to maintain system time. As a result, these kernels reduce the overall average rate of virtual timer interrupts, thus improving system performance and scalability on hosts running large numbers of virtual machines.

  For background information on the topic of timer interrupts, refer to *Timekeeping in Virtual Machines*.

Virtual NUMA (vNUMA)

Virtual NUMA (vNUMA), a new feature in ESXi 5.0, exposes NUMA topology to the guest operating system, allowing NUMA-aware guest operating systems and applications to make the most efficient use of the underlying hardware's NUMA architecture.

Virtual NUMA, which requires virtual hardware version 8, can provide significant performance benefits, though the benefits depend heavily on the level of NUMA optimization in the guest operating system and applications.

- You can obtain the maximum performance benefits from vNUMA if your clusters are composed entirely of hosts with matching NUMA architecture.

  This is because the very first time a vNUMA-enabled virtual machine is powered on, its vNUMA topology is set based in part on the NUMA topology of the underlying physical host on which it is running. Once a virtual machine's vNUMA topology is initialized it doesn't change unless the number of vCPUs in that virtual machine is changed. This means that if a vNUMA virtual machine is moved to a host with a different NUMA topology, the virtual machine's vNUMA topology might no longer be optimal for the underlying physical NUMA topology, potentially resulting in reduced performance.

- Size your virtual machines so they align with physical NUMA boundaries. For example, if you have a host system with six cores per NUMA node, size your virtual machines with a multiple of six vCPUs (i.e., 6 vCPUs, 12 vCPUs, 18 vCPUs, 24 vCPUs, and so on).

  NOTE Some multi-core processors have NUMA node sizes that are different than the number of cores per socket. For example, some 12-core processors have two six-core NUMA nodes per processor.

- When creating a virtual machine you have the option to specify the number of virtual sockets and the number of cores per virtual socket. If the number of cores per virtual socket on a vNUMA-enabled virtual machine is set to any value other than the default of 1, and that value doesn't align with the underlying physical host topology, performance might be slightly reduced.

  Therefore if a virtual machine is to be configured with a non-default number of cores per virtual socket, for best performance that number should be an integer multiple or integer divisor of the physical NUMA node size.

- By default, vNUMA is enabled only for virtual machines with more than eight vCPUs. This feature can be enabled for smaller virtual machines, however, by adding to the `.vmx` file the line:
  `numa.vcpu.maxPerVirtualNode = X`
  (where *X* is the number of vCPUs per vNUMA node).

  NOTE This change can be made through the vSphere Client:

  1. Select the virtual machine you wish to change, then click **Edit virtual machine settings**.
  2. Under the **Options** tab, select **General**, then click **Configuration Parameters**.
  3. Look for **numa.vcpu.maxPerVirtualNode**. If it's not present, click **Add Row** and enter the new variable.
  4. Click on the value to be changed and configure it as you wish.

Guest Operating System Storage Considerations

- The default virtual storage adapter in ESXi 5.0 is either BusLogic Parallel, LSI Logic Parallel, or LSI Logic SAS, depending on the guest operating system and the virtual hardware version. However, ESXi also includes a paravirtualized SCSI storage adapter, PVSCSI (also called VMware Paravirtual). The PVSCSI adapter offers a significant reduction in CPU utilization as well as potentially increased throughput compared to the default virtual storage adapters, and is thus the best choice for environments with very I/O-intensive guest applications.

  NOTE In order to use PVSCSI, your virtual machine must be using virtual hardware version 7 or later, as described under "ESXi General Considerations" on page 17.

  NOTE PVSCSI adapters are supported for boot drives in only some operating systems. For additional information see VMware KB article 1010398.

- If you choose to use the BusLogic Parallel virtual SCSI adapter, and are using a Windows guest operating system, you should use the custom BusLogic driver included in the VMware Tools package.
- 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.
- In some cases large I/O requests issued by applications in a virtual machine can be split by the guest storage driver. Changing the guest operating system's registry settings to issue larger block sizes can eliminate this splitting, thus enhancing performance. For additional information see VMware KB article 9645697.
- Make sure the disk partitions within the guest are aligned. For further information you might want to refer to the literature from the operating system vendor regarding appropriate tools to use as well as recommendations from the array vendor.

Guest Operating System Networking Considerations

The default virtual network adapter emulated in a virtual machine is either an AMD PCnet32 device (vlance) or an Intel E1000 device (E1000). VMware also offers the VMXNET family of paravirtualized network adapters, however, that provide better performance than these default adapters and should be used for optimal performance within any guest operating system for which they are available.

The paravirtualized network adapters in the VMXNET family implement an idealized network interface that passes network traffic between the virtual machine and the physical network interface cards with minimal overhead. Drivers for VMXNET-family adapters are available for most guest operating systems supported by ESXi.

The VMXNET family contains VMXNET, Enhanced VMXNET (available since ESX/ESXi 3.5), and VMXNET Generation 3 (VMXNET3; available since ESX/ESXi 4.0).

NOTE The network speeds reported by the guest network driver in the virtual machine do not necessarily reflect the actual speed of the underlying physical network interface card. For example, the vlance guest driver in 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 ESXi emulates are 10Mbps. However, ESXi is not limited to 10Mbps and transfers network packets as fast as the resources on the physical server machine allow

- For the best performance, use the VMXNET3 paravirtualized network adapter for operating systems in which it is supported. This requires that the virtual machine use virtual hardware version 7 or later, and that VMware Tools be installed in the guest operating system.

  NOTE A virtual machine with a VMXNET3 device cannot vMotion to a host running ESX/ESXi 3.5.x or earlier.

- For guest operating systems in which VMXNET3 is not supported, or if you don't wish to use virtual hardware version 7 or later (to maintain vMotion compatibility with older versions of ESX/ESXi, for example), the best performance can be obtained with the use of Enhanced VMXNET for operating systems in which it is supported. This requires that VMware Tools be installed in the guest operating system.

  NOTE A virtual machine with an Enhanced VMXNET device cannot vMotion to a host running ESX 3.0.x or earlier.

- For the operating systems in which Enhanced VMXNET is not supported, use the flexible device type. In ESXi, the "flexible NIC" automatically converts each vlance network device to a VMXNET device (a process also called "NIC Morphing") if the VMware Tools suite is installed in the guest operating system and the operating system supports VMXNET.
- The VMXNET3, Enhanced VMXNET, and E1000 devices support jumbo frames for better performance. (Note that the vlance device does not support jumbo frames.) To enable jumbo frames, set the MTU size to 9000 in both the guest network driver and the virtual switch configuration. The physical NICs at both ends and all the intermediate hops/routers/switches must also support jumbo frames.
- In ESXi, TCP Segmentation Offload (TSO) is enabled by default in the VMkernel, but is supported in virtual machines only when they are using the VMXNET3 device, 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 receive throughput in a virtual machine can be caused by insufficient receive buffers in the receiver network device. If the receive ring in the guest operating system's network driver overflows, packets will be dropped in the VMkernel, degrading network throughput. A possible workaround is to increase the number of receive buffers, though this might increase the host physical CPU workload.

For VMXNET, the default number of receive and transmit buffers is 100 each, with the maximum possible being 128. For Enhanced VMXNET, the default number of receive and transmit buffers are 150 and 256, respectively, with the maximum possible receive buffers being 512. 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 VMware KB article 1010071.

For VMXNET3 and E1000, the default number of receive and transmit buffers are controlled by the guest driver, with the maximum possible for both being 4096. For Linux, these values can be changed from within the guest by using `ethtool`. In Windows, the values can be changed from within the guest in the device properties window. For additional information see VMware KB article 1010071.

- Receive-side scaling (RSS) allows network packet receive processing to be scheduled in parallel on multiple CPUs. Without RSS, receive interrupts can be handled on only one CPU at a time. With RSS, received packets from a single NIC can be processed on multiple CPUs concurrently. This helps receive throughput in cases where a single CPU would otherwise be saturated with receive processing and become a bottleneck. To prevent out-of-order packet delivery, RSS schedules all of a flow's packets to the same CPU.

  The VMXNET3 device supports RSS for all Windows guest operating systems that support RSS natively (such as Windows Server 2008 and Windows 7) and also supports multiple transmit queues. By default, on an *n*-vCPU virtual machine RSS configures *n* receive queues (up to 4). (RSS doesn't affect the transmit queues, which default to 1 with or without RSS.) In order to obtain the maximum performance with your specific workloads and resource availability you can try out different values for both the number of transmit queues (up to a maximum of 8) and the number of receive queues (up to a maximum of 8). These values are set from the advanced driver configuration tab within the guest operating system.

Virtual Infrastructure Management 4

This chapter provides guidance regarding infrastructure management best practices. Most of the suggestions included in this section can be implemented using the vSphere Client connected to a VMware vCenter™ Server. Some can also be implemented using the vSphere Client connected to an individual ESXi host.

Further detail about many of these topics, as well as background information, can be found in the *VMware vCenter Server Performance and Best Practices* white paper.

General Resource Management

ESXi 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.

- Use resource settings (that is, **Reservation**, **Shares**, and **Limits**) only if needed in your environment.
- 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 subsequently 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, ESXi 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 approach 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.

VMware vCenter

This section lists VMware vCenter practices and configurations recommended for optimal performance. It also includes a few features that are controlled or accessed through vCenter.

- The performance of vCenter Server is dependent in large part on the number of managed entities (hosts and virtual machines) and the number of connected VMware vSphere Clients. Exceeding the maximums specified in *Configuration Maximums for VMware vSphere 5.0*, in addition to being unsupported, is thus likely to impact vCenter Server performance.
- Whether run on virtual machines or physical systems, make sure you provide vCenter Server and the vCenter Server database with sufficient CPU, memory, and storage resources for your deployment size. For additional information see *vSphere Installation and Setup* for vSphere 5.0.
- To minimize the latency of vCenter operations, keep to a minimum the number of network hops between the vCenter Server system and the vCenter Server database.
- Although VMware vCenter Update Manager can be run on the same system and use the same database as vCenter Server, for maximum performance, especially on heavily-loaded vCenter systems, consider running Update Manager on its own system and providing it with a dedicated database. For additional information see "VMware vCenter Update Manager" on page 61.
- Similarly, VMware vCenter Converter can be run on the same system as vCenter Server, but doing so might impact performance, especially on heavily-loaded vCenter systems.
- During installation of VMware vCenter you will be asked to choose a target inventory size. This choice will be used to set Java virtual machine (JVM) maximum heap memory sizes for various services. The default values (detailed in *vSphere Installation and Setup* for vSphere 5.0) should provide good performance while avoiding unnecessary memory commitment. If you will have inventory sizes well into the large range (as defined in *vSphere Installation and Setup*), however, you might obtain better performance by increasing one or more of these settings.

VMware vCenter Database Considerations

vCenter Server relies heavily on a database to store configuration information about inventory items and performance statistics data. It also stores data about alarms, events, and tasks. Due to the importance of this database to the reliability and performance of your vCenter Server, VMware recommends the database practices described in this section.

VMware vCenter Database Network and Storage Considerations

- To minimize the latency of operations between vCenter Server and the database, keep to a minimum the number of network hops between the vCenter Server system and the database system.
- The hardware on which the vCenter database is stored, and the arrangement of the files on that hardware, can have a significant effect on vCenter performance:
  - The vCenter database performs best when its files are placed on high-performance storage.
  - The database data files generate mostly random read I/O traffic, while the database transaction logs generate mostly sequential write I/O traffic. For this reason, and because their traffic is often significant and simultaneous, vCenter performs best when these two file types are placed on separate storage resources that share neither disks nor I/O bandwidth.

VMware vCenter Database Configuration and Maintenance

- Configure the vCenter statistics level to a setting appropriate for your uses. This setting can range from 1 to 4, but a setting of 1 is recommended for most situations. Higher settings can slow the vCenter Server system. You can also selectively disable statistics rollups for particular collection levels.
- To avoid frequent log file switches, ensure that your vCenter database logs are sized appropriately for your vCenter inventory. For example, with a large vCenter inventory running with an Oracle database, the size of each redo log should be at least 512MB.
- vCenter Server starts up with a database connection pool of 50 threads. This pool is then dynamically sized, growing adaptively as needed based on the vCenter Server workload, and does not require modification. However, if a heavy workload is expected on the vCenter Server, the size of this pool at startup can be increased, with the maximum being 128 threads. Note that this might result in increased memory consumption by vCenter Server and slower vCenter Server startup.

  To change the pool size, edit the `vpxd.cfg` file, adding:

  ```
  <vpxd>
   <odbc>
    <maxConnections>xxx</maxConnections>
   </odbc>
  </vpxd>
  ```

  (where *xxx* is the desired pool size.)
- Update statistics of the tables and indexes on a regular basis for better overall performance of the database.
- As part of the regular database maintenance activity, check the fragmentation of the index objects and recreate indexes if needed (i.e., if fragmentation is more than about 30%).

Recommendations for Specific Database Vendors

This subsection describes database-specific recommendations.

Microsoft SQL Server Database Recommendations

If you are using a Microsoft SQL Server database, the following points can improve vCenter Server performance:

- Setting the transaction logs to **Simple** recovery mode significantly reduces the database logs' disk space usage as well as their storage I/O load. If it isn't possible to set this to Simple, make sure to have a high-performance storage subsystem.

- To further improve database performance for large inventories, place tempDB on a different disk than either the database data files or the database transaction logs.
- We recommend a fill factor of about 70% for the four VPX_HIST_STAT tables (vpx_hist_stat1, vpx_hist_stat2, vpx_hist_stat3, and vpx_hist_stat4). If the fill factor is set too high, the server must take time splitting pages when they fill up. If the fill factor is set too low, the database will be larger than necessary due to the unused space on each page, thus increasing the number of pages that need to be read during normal operations.

Oracle Database Recommendations

If you are using an Oracle database, the following points can improve vCenter Server performance:

- When using Automatic Memory Management (AMM) in Oracle 11g, or Automatic Shared memory Management (ASMM) in Oracle 10g, allocate sufficient memory for the Oracle database.
- Set appropriate **PROCESSES** or **SESSIONS** initialization parameters. Oracle creates a new server process for every new connection that is made to it. The number of connections an application can make to the Oracle instance thus depends on how many processes Oracle can create. **PROCESSES** and **SESSIONS** together determine how many simultaneous connections Oracle can accept. In large vSphere environments (as defined in *vSphere Installation and Setup* for vSphere 5.0) we recommend setting **PROCESSES** to 800.
- If database operations are slow, after checking that the statistics are up to date and the indexes are not fragmented, you should move the indexes to separate tablespaces (i.e., place tables and primary key (PK) constraint index on one tablespace and the other indexes (i.e., BTree) on another tablespace).
- For large inventories (i.e., those that approach the limits for the number of hosts or virtual machines), increase the `db_writer_processes` parameter to `4`.

VMware vSphere Management

VMware vSphere environments are typically managed with the VMware vSphere Client, the VMware vSphere Web Client, or the VMware vSphere Web Services SDK.

Large numbers of connected vSphere Clients, vSphere Web Clients and vSphere Web Services SDK Clients can affect the performance of a vCenter Server. Exceeding the maximums specified in *Configuration Maximums for VMware vSphere 5.0* is not supported. Even if it seems to work, doing so is even more likely to affect vCenter Server performance.

vSphere Clients

The VMware vSphere Client is a Windows program used to configure ESXi hosts and operate virtual machines. You can download the vSphere Client from any ESXi host.

- For the best performance, disconnect vSphere Clients from the vCenter Server when they are no longer needed. Because the vCenter Server must keep all client sessions current with inventory changes, the number of vSphere Client sessions attached to the vCenter Server can affect the server's CPU usage and user interface speed.
- If running many instances of the vSphere Client on a single system, monitor the resource usage of that system.
- You can by default open a maximum of 25 pop-out console windows within a vSphere Client session. If you are running the vSphere Client on a powerful system, you can increase this limit in **Client Settings** > **General**.
- When selecting an entity (virtual machine, host, datastore, network, etc.) within the vSphere Client, using the inventory search feature is less resource intensive on the vCenter Server than navigating through the vCenter Client inventory panel.
- When viewing a map with a large number of entities (i.e., thousands of virtual machines in a datacenter), the map might be difficult to read and might be slow to appear. To avoid this, the parameter **Maximum requested topology entities** (in **Client Settings** > **Maps**) has a default value of `300`. You can view the topology map with a larger number of entities by increasing this value.

For more information on working with the vSphere Client, see *Customizing the vSphere Client*.

vSphere Web Clients

The VMware vSphere Web Client is a browser-based interface used to control ESXi hosts and operate virtual machines. It consists of the Java-based vSphere Web Client Server back end and one or more instances of an Adobe Flash-based front end running in a browser.

- It is possible to install the vSphere Web Client Server, the vCenter Server, and the vCenter Inventory Service all on the same system or split across two separate systems. We recommend the following:
  - For inventories of less than about 32 hosts and 3,000 virtual machines (that is, the maximum cluster size), install all three modules on the same system.
  - For larger inventories, install vCenter Server and vCenter Inventory Service on one system and install the vSphere Web Client Server on a second system.

  Like the vCenter modules, the vSphere Web Client Server can be run on a virtual machine or a physical system, as long as it is provided sufficient resources.

  If these modules are split across two systems, those systems should all be on the same network segment and should have low inter-system network latencies.
- Use the search function instead of navigating to managed objects. The vSphere Web Client is designed for the use of inventory search to find managed objects (clusters, hosts, virtual machines, datastores, and so on). Though a limited set of the most recently used managed objects are displayed in the object navigator pane, the inventory search function will typically provide better performance than navigating among these objects.

- Close the vSphere Web Client browser window occasionally (i.e., approximately daily). Limitations in the Adobe Flex Framework can cause actively-used client sessions to eventually consume more memory than needed. Closing and restarting the browser window in which the Web Client is running will avoid this problem.

vSphere Web Services SDK Clients

The VMware vSphere Web Services SDK can be an efficient way to manage the vSphere environment. To learn more about the VMware vSphere API and supported SDK libraries, refer to the vSphere API and SDK Documentation. For examples of good programming practices, see code samples from the VMware Communities sample code page (http://communities.vmware.com/community/vmtn/developer/codecentral).

VMware vMotion and Storage vMotion

This section provides performance best practices for vMotion™ and Storage vMotion.

VMware vMotion

- ESXi 5.0 introduces virtual hardware version 8. Because virtual machines running on hardware version 8 can't run on prior versions of ESX/ESXi, such virtual machines can be moved using VMware vMotion only to other ESXi 5.0 hosts. ESXi 5.0 is also compatible with virtual machines running on virtual hardware version 7 and earlier, however, and these machines can be moved using VMware vMotion to ESX/ESXi 4.x hosts.
- vMotion performance will increase as additional network bandwidth is made available to the vMotion network. Consider provisioning 10Gb vMotion network interfaces for maximum vMotion performance. Multiple vMotion vmknics, a new feature in ESXi 5.0, can provide a further increase in network bandwidth available to vMotion.

  All vMotion vmknics on a host should share a single vSwitch. Each vmknic's portgroup should be configured to leverage a different physical NIC as its active vmnic. In addition, all vMotion vmknics should be on the same vMotion network.
- While a vMotion operation is in progress, ESXi opportunistically reserves CPU resources on both the source and destination hosts in order to ensure the ability to fully utilize the network bandwidth. ESXi will attempt to use the full available network bandwidth regardless of the number of vMotion operations being performed. The amount of CPU reservation thus depends on the number of vMotion NICs and their speeds; 10% of a processor core for each 1Gb network interface, 100% of a processor core for each 10Gb network interface, and a minimum total reservation of 30% of a processor core. Therefore leaving some unreserved CPU capacity in a cluster can help ensure that vMotion tasks get the resources required in order to fully utilize available network bandwidth.
- vMotion performance could be reduced if host-level swap files are placed on local storage (whether SSD or hard drive). For more information on this, see "Memory Swapping Optimizations" on page 27.

VMware Storage vMotion

- VMware Storage vMotion performance depends strongly on the available storage infrastructure bandwidth between the ESXi host where the virtual machine is running and both the source and destination data stores.

  During a Storage vMotion operation the virtual disk to be moved is being read from the source data store and written to the destination data store. At the same time the virtual machine continues to read from and write to the source data store while also writing to the destination data store.

  This additional traffic takes place on storage that might also have other I/O loads (from other virtual machines on the same ESXi host or from other hosts) that can further reduce the available bandwidth.
- Storage vMotion will have the highest performance during times of low storage activity (when available storage bandwidth is highest) and when the workload in the virtual machine being moved is least active.
- During a Storage vMotion operation, the benefits of moving to a faster data store will be seen only when the migration has completed. However, the impact of moving to a slower data store will gradually be felt as the migration progresses.
- Storage vMotion will often have significantly better performance on VAAI-capable storage arrays (described in "Hardware Storage Considerations" on page 11).

VMware Distributed Resource Scheduler (DRS)

This section lists Distributed Resource Scheduler (DRS) practices and configurations recommended by VMware for optimal performance.

Cluster Configuration Settings

- 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 improves performance predictability and stability.

  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 loads.

- VMware 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 VMware KB articles 1991, 1992, and 1993.

  You can also use Enhanced vMotion Compatibility (EVC) to facilitate vMotion between different CPU generations. For more information on this topic see *VMware vMotion and CPU Compatibility* and VMware KB article 1003212.

- The more vMotion compatible ESXi 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 (or faster) Ethernet link then the vMotion might not occur between the hosts.

  Other configuration settings to check for are virtual hardware version compatibility, 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 *vSphere vCenter Server and Host Management* for further details.

- 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 sizes 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 balancing migrations across the ESXi hosts before the virtual machines that are not in automatic mode.

- Powered-on virtual machines consume memory resources—and typically consume some CPU resources—even when idle. Thus even idle virtual machines, though their 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.

- Resource pools help improve manageability and troubleshooting of performance problems. We recommend, however, that resource pools and virtual machines not be made siblings in a hierarchy. Instead, each level should contain only resource pools or only virtual machines. This is because by default resource pools are assigned share values that might not compare appropriately with those assigned to virtual machines, potentially resulting in unexpected performance.

- DRS affinity rules can keep two or more virtual machines on the same ESXi host ("VM/VM affinity") or make sure they are always on different hosts ("VM/VM anti-affinity"). DRS affinity rules can also be used to make sure a group of virtual machines runs only on (or has a preference for) a specific group of ESXi hosts ("VM/Host affinity") or never runs on (or has a preference against) a specific group of hosts ("VM/Host anti-affinity").

In most cases leaving the affinity settings unchanged will provide the best results. In rare cases, however, specifying affinity rules can help improve performance. To change affinity settings, select a cluster from within the vSphere Client, choose the **Summary** tab, click **Edit Settings**, choose **Rules**, click **Add**, enter a name for the new rule, choose a rule type, and proceed through the GUI as appropriate for the rule type you selected.

Besides the default setting, the affinity setting types are:

- **Keep Virtual Machines Together**
  This affinity type can improve performance due to lower latencies of communication between machines.
- **Separate Virtual Machines**
  This affinity type can maintain 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 (though see "Storage I/O Resource Allocation" on page 32 for one way to allocate storage resources).
- **Virtual Machines to Hosts** (including **Must run on...**, **Should run on...**, **Must not run on...**, and **Should not run on...**)
  These affinity types can be useful for clusters with software licensing restrictions or specific availability zone requirements.

- To allow DRS the maximum flexibility:
  - Place virtual machines on shared datastores accessible from all hosts in the cluster.
  - Make sure virtual machines are not connected to host devices that would prevent them from moving off of those hosts.
- The drmdump files produced by DRS can be very useful in diagnosing potential DRS performance issues during a support call. For particularly active clusters, or those with more than about 16 hosts, it can be helpful to keep more such files than can fit in the default maximum drmdump directory size of 20MB. This maximum can be increased using the `DumpSpace` option, which can be set using DRS **Advanced Options**.

Cluster Sizing and Resource Settings

- Exceeding the maximum number of hosts, virtual machines, or resource pools for each DRS cluster specified in *Configuration Maximums for VMware vSphere 5.0* is not supported. Even if it seems to work, doing so could adversely affect vCenter Server or DRS performance.
- Carefully select the resource settings (that is, reservations, shares, and limits) for your virtual machines.
  - Setting reservations too high can leave few unreserved resources in the cluster, thus limiting the options DRS has to balance load.
  - Setting limits too low could keep virtual machines from using extra resources available in the cluster to improve their performance.

  Use reservations to guarantee the minimum requirement a virtual machine needs, rather than what you might like it to get. Note that shares take effect only when there is resource contention. Note also that additional resources reserved for virtual machine memory overhead need to be accounted for when sizing resources in the cluster.

  If the overall cluster capacity might not meet the needs of all virtual machines during peak hours, you can assign relatively higher shares to virtual machines or resource pools hosting mission-critical applications to reduce the performance interference from less-critical virtual machines.
- If you will be using vMotion, it's a good practice to leave some unused CPU capacity in your cluster. As described in "VMware vMotion" on page 51, when a vMotion operation is started, ESXi reserves some CPU resources for that operation.

DRS Performance Tuning

- The migration threshold for fully automated DRS (cluster > **DRS** tab > **Edit...** > **vSphere DRS**) allows the administrator to control the aggressiveness of the DRS algorithm. In most cases, the default setting of the migration threshold should be used, representing a medium level of aggressiveness.

  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 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.

  NOTE If the most conservative threshold is chosen, DRS will only apply move recommendations that must be taken either to satisfy hard constraints, such as affinity or anti-affinity rules, or to evacuate virtual machines from a host entering maintenance or standby mode.

- In addition to the migration threshold, DRS offers a number of advanced options that allow further control over the algorithm's behavior. Table 4-1 lists some of these options and explains their use. We recommend that these DRS options be left at their default values unless there is a strong reason to change them; for example, a cluster with severe resource contention on some hosts and idle capacity on others.

  Table 4-1. DRS Advanced Options for Performance Tuning

Advanced option name	Description	Default Value	Most Aggressive Value
CostBenefit	Whether to take migration cost into account	1	0 (No cost-benefit analysis)
UseDownTime	Whether to use migration down time in cost analysis	1	0 (no consideration of down time)
IgnoreDownTimeLessThan	Threshold (in seconds) for ignoring migration down time in cost analysis	1	A large number (no consideration of down time)
MinImbalance	Used to compute target imbalance	50	0
MinGoodness	Minimum improvement in cluster imbalance required for each move	Adaptive	0 (All moves are considered)
MaxMovesPerHost	Maximum number of moves per host recommended per invocation	Adaptive	0 (No limit)

- The advanced options `UseDownTime` and `IgnoreDownTimeLessThan`, new in vSphere 5.0 and described in Table 4-1, provide specific control over how DRS evaluates for migration large or resource-heavy virtual machines that tend to have longer migration down times.
- There are a variety of reasons that a DRS cluster might be shown as **Load imbalanced**. These include:
  - Migrations being filtered out due to affinity/anti-affinity rules.
  - Migrations being filtered out due to VM-host incompatibilities.
  - The cost estimate for potential migrations (based on the costs of previous migrations) exceeding the expected benefits of the potential migrations.

  If achieving a "load balanced" cluster is critical for your specific environment, you can relax some rules, adjust the migration threshold, or use the advanced options in Table 4-1.

  On the other hand, if all the virtual machines are receiving 100% of their entitled resources, it might be acceptable to have a slightly imbalanced cluster.
- 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. We recommend against changing the default value, however, except in specific cases where the user would like to invoke the algorithm less frequently at the cost of a potential loss in application performance.

- A set of scripts with which advanced DRS users can conduct more proactive cluster load balancing is available on the *Scripts for Proactive DRS* VMware community page, at http://communities.vmware.com/docs/DOC-10231.

For more information on DRS, refer to *vSphere Resource Management*.

VMware Distributed Power Management (DPM)

VMware 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.

Because DPM uses DRS, most DRS best practices (described in "VMware Distributed Resource Scheduler (DRS)" on page 52) are relevant to DPM as well.

- DPM is complementary to host power management policies (described in "Host Power Management in ESXi" on page 23). Using DPM and host power management together can offer greater power savings than when either solution is used alone.
- DPM considers historical demand in determining how much capacity to keep powered on and keeps some excess capacity available for changes in demand. DPM will also power on additional hosts when needed for unexpected increases in the demand of existing virtual machines or to allow virtual machine admission.
- The aggressiveness of the DPM algorithm can be tuned by adjusting the **DPM Threshold** in the cluster settings menu. This parameter controls how far outside the target utilization range per-host resource utilization can be before DPM makes host power-on/power-off recommendations. The default setting for the threshold is 3 (medium aggressiveness).
- For datacenters that often have unexpected spikes in virtual machine resource demands, you can use the DPM advanced option `MinPoweredOnCpuCapacity` (default 1 MHz) or `MinPoweredOnMemCapacity` (default 1 MB) to ensure that a minimum amount of CPU or memory capacity is kept on in the cluster.
- Enabling DPM with all hosts in the cluster being in automatic mode gives DPM the most flexibility in choosing hosts for power down/power up. For hosts in manual DPM mode, vCenter requests user approval before changing a host's power state. For this reason, DPM prefers choosing hosts in automatic mode over those in manual mode. If desired, DPM can also be disabled for specific hosts.
- DPM can be disabled on individual hosts that are running mission-critical virtual machines, and the VM/Host affinity rules can be used to ensure that these virtual machines are not migrated away from these hosts.
- DPM can be enabled or disabled on a predetermined schedule using **Scheduled Tasks** in vCenter Server. When DPM is disabled, all hosts in a cluster will be powered on. This might be useful, for example, to reduce the delay in responding to load spikes expected at certain times of the day or to reduce the likelihood of some hosts being left in standby for extended periods.
- In a cluster with VMware High Availability (HA) enabled, DRS/DPM maintains excess powered-on capacity to meet the High Availability settings. The cluster might therefore not allow additional virtual machines to be powered on and/or some hosts might not be powered down even when the cluster appears to be sufficiently idle. These factors should be considered when configuring HA.
- If VMware HA is enabled in a cluster, DPM always keeps a minimum of two hosts powered on. This is true even if HA admission control is disabled or if no virtual machines are powered on.
- The VMware Community page *Scripts for "Proactive DPM"* (http://communities.vmware.com/docs/DOC-10230) provides a set of Perl scripts with which advanced DPM users can conduct more proactive power management.

For more information on DPM performance tuning, see *VMware Distributed Power Management Concepts and Use*.

VMware Storage Distributed Resource Scheduler (Storage DRS)

A new feature in vSphere 5.0, Storage Distributed Resource Scheduler (Storage DRS), provides I/O load balancing across datastores within a datastore cluster (a new vCenter object). This load balancing can avoid storage performance bottlenecks or address them if they occur.

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

- When deciding which datastores to group into a datastore cluster, try to choose datastores that are as homogeneous as possible in terms of host interface protocol (i.e., FCP, iSCSI, NFS), RAID level, and performance characteristics. We recommend not mixing SSD and hard disks in the same datastore cluster.
- Don't configure into a datastore cluster more datastores or virtual disks than the maximum allowed in *Configuration Maximums for VMware vSphere 5.0*.
- While a datastore cluster can have as few as two datastores, the more datastores a datastore cluster has, the more flexibility Storage DRS has to better balance that cluster's I/O load.
- As you add workloads you should monitor datastore I/O latency in the performance chart for the datastore cluster, particularly during peak hours. If most or all of the datastores in a datastore cluster consistently operate with latencies close to the congestion threshold used by Storage I/O Control (set to 30ms by default, but sometimes tuned to reflect the needs of a particular deployment), this might be an indication that there aren't enough spare I/O resources left in the datastore cluster. In this case, consider adding more datastores to the datastore cluster or reducing the load on that datastore cluster.

  NOTE Make sure, when adding more datastores to increase I/O resources in the datastore cluster, that your changes do actually add resources, rather than simply creating additional ways to access the same underlying physical disks.

- By default, Storage DRS affinity rules keep all of a virtual machine's virtual disks on the same datastore (using intra-VM affinity). However you can give Storage DRS more flexibility in I/O load balancing, potentially increasing performance, by overriding the default intra-VM affinity rule. This can be done for either a specific virtual machine (from the vSphere Client, select **Edit Settings** > **Virtual Machine Settings**, then deselect **Keep VMDKs together**) or for the entire datastore cluster (from the vSphere Client, select **Home** > **Inventory** > **Datastore and Datastore Clusters**, select a datastore cluster, select the **Storage DRS** tab, click **Edit**, select **Virtual Machine Settings**, then deselect **Keep VMDKs together**).
- Inter-VM anti-affinity rules can be used to keep the virtual disks from two or more different virtual machines from being placed on the same datastore, potentially improving performance in some situations. They can be used, for example, to separate the storage I/O of multiple workloads that tend to have simultaneous but intermittent peak loads, preventing those peak loads from combining to stress a single datastore.
- If a datastore cluster contains thin-provisioned LUNs, make sure those LUNs don't run low on backing disk space. If many thin-provisioned LUNs in a datastore cluster simultaneously run low on backing disk space (quite possible if they all share the same backing store), this could cause excessive Storage vMotion activity or limit the ability of Storage DRS to balance datastore usage.

VMware High Availability

VMware High Availability (HA) minimizes virtual machine downtime by monitoring hosts, virtual machines, or applications within virtual machines, then, in the event a failure is detected, restarting virtual machines on alternate hosts.

- When vSphere HA is enabled in a cluster, all active hosts (those not in standby mode, maintenance mode, or disconnected) participate in an election to choose the master host for the cluster; all other hosts become slaves. The master has a number of responsibilities, including monitoring the state of the hosts in the cluster, protecting the powered-on virtual machines, initiating failover, and reporting cluster health state to vCenter Server. The master is elected based on the properties of the hosts, with preference being given to the one connected to the greatest number of datastores. Serving in the role of master will have little or no effect on a host's performance.
- When the master host can't communicate with a slave host over the management network, the master uses datastore heartbeating to determine the state of that slave host. By default, vSphere HA uses two datastores for heartbeating, resulting in very low false failover rates. In order to reduce the chances of false failover even further—at the potential cost of a very slight performance impact—you can use the advanced option `das.heartbeatdsperhost` to change the number of datastores (up to a maximum of five).
- Enabling HA on a host reserves some host resources for HA agents, slightly reducing the available host capacity for powering on virtual machines.
- When HA is enabled, the vCenter Server reserves sufficient unused resources in the cluster to support the failover capacity specified by the chosen admission control policy. This can reduce the number of virtual machines the cluster can support.

For further details about HA, see *vSphere Availability*.

VMware Fault Tolerance

VMware Fault Tolerance (FT) provides continuous virtual machine availability in the event of a server failure.

Because FT uses HA, most HA best practices (described in "VMware High Availability" on page 58) are relevant to FT as well.

- For each virtual machine there are two FT-related actions that can be taken: turning on or off FT and enabling or disabling FT.

  "Turning on FT" prepares the virtual machine for FT by prompting for the removal of unsupported devices, disabling unsupported features, and setting the virtual machine's memory reservation to be equal to its memory size (thus avoiding ballooning or swapping).

  "Enabling FT" performs the actual creation of the secondary virtual machine by live-migrating the primary.

  NOTE Turning on FT for a powered-on virtual machine will also automatically "Enable FT" for that virtual machine.

  Each of these operations has performance implications.

  - Don't turn on FT for a virtual machine unless you will be using (i.e., Enabling) FT for that machine. Turning on FT automatically disables some features for the specific virtual machine that can help performance, such as hardware virtual MMU (if the processor supports it).
  - Enabling FT for a virtual machine uses additional resources (for example, the secondary virtual machine uses as much CPU and memory as the primary virtual machine). Therefore make sure you are prepared to devote the resources required before enabling FT.
- The live migration that takes place when FT is enabled can briefly saturate the vMotion network link and can also cause spikes in CPU utilization.
  - If the vMotion network link is also being used for other operations, such as FT logging (transmission of all the primary virtual machine's inputs (incoming network traffic, disk reads, etc.) to the secondary host), the performance of those other operations can be impacted. For this reason it is best to have separate and dedicated NICs (or use Network I/O Control, described in "Network I/O Control (NetIOC)" on page 34) for FT logging traffic and vMotion, especially when multiple FT virtual machines reside on the same host.
  - Because this potentially resource-intensive live migration takes place each time FT is enabled, we recommend that FT not be frequently enabled and disabled.
  - FT-enabled virtual machines must use eager-zeroed thick-provisioned virtual disks. Thus when FT is enabled for a virtual machine with thin-provisioned virtual disks or lazy-zeroed thick-provisioned virtual disks these disks need to be converted. This one-time conversion process uses fewer resources when the virtual machine is on storage hardware that supports VAAI (described in "Hardware Storage Considerations" on page 11).
- Because FT logging traffic is asymmetric (the majority of the traffic flows from primary to secondary), congestion on the logging NIC can be reduced by distributing primaries onto multiple hosts. For example on a cluster with two ESXi hosts and two virtual machines with FT enabled, placing one of the primary virtual machines on each of the hosts allows the network bandwidth to be utilized bidirectionally.
- FT virtual machines that receive large amounts of network traffic or perform lots of disk reads can create significant bandwidth on the NIC specified for the logging traffic. This is true of machines that routinely do these things as well as machines doing them only intermittently, such as during a backup operation. To avoid saturating the network link used for logging traffic limit the number of FT virtual machines on each host or limit disk read bandwidth and network receive bandwidth of those virtual machines.
- Make sure the FT logging traffic is carried by at least a Gigabit-rated NIC (which should in turn be connected to at least Gigabit-rated network infrastructure).

- Avoid placing more than four FT-enabled virtual machines on a single host. In addition to reducing the possibility of saturating the network link used for logging traffic, this also limits the number of simultaneous live-migrations needed to create new secondary virtual machines in the event of a host failure.
- If the secondary virtual machine lags too far behind the primary (which usually happens when the primary virtual machine is CPU bound and the secondary virtual machine is not getting enough CPU cycles), the hypervisor might slow the primary to allow the secondary to catch up. The following recommendations help avoid this situation:
  - Make sure the hosts on which the primary and secondary virtual machines run are relatively closely matched, with similar CPU make, model, and frequency.
  - Make sure that power management scheme settings (both in the BIOS and in ESXi) that cause CPU frequency scaling are consistent between the hosts on which the primary and secondary virtual machines run.
  - Enable CPU reservations for the primary virtual machine (which will be duplicated for the secondary virtual machine) to ensure that the secondary gets CPU cycles when it requires them.
- Though timer interrupt rates do not significantly affect FT performance, high timer interrupt rates create additional network traffic on the FT logging NICs. Therefore, if possible, reduce timer interrupt rates as described in "Guest Operating System CPU Considerations" on page 39.

VMware vCenter Update Manager

VMware vCenter Update Manager provides a patch management framework for VMware vSphere. It can be used to apply patches, updates, and upgrades to VMware ESX and ESXi hosts, VMware Tools and virtual hardware, and so on.

Further detail about many of these topics can be found in *VMware vCenter Update Manager 5.0 Performance and Best Practices*. The vCenter Update Manager sizing guide might also be useful: http://pubs.vmware.com/vsphere-50/topic/com.vmware.ICbase/PDF/vsphere-update-manager-50-sizing-estimator.xls

Update Manager Setup and Configuration

- When there are more than 300 virtual machines or more than 30 hosts, separate the Update Manager database from the vCenter Server database.
- When there are more than 1000 virtual machines or more than 100 hosts, separate the Update Manager server from the vCenter Server and the Update Manager database from the vCenter Server database.
- Allocate separate physical disks for the Update Manager patch store and the Update Manager database.
- To reduce network latency and packet drops, keep to a minimum the number of network hops between the Update Manager server system and the ESXi hosts.
- In order to cache frequently used patch files in memory, make sure the Update Manager server host has at least 2GB of RAM.

Update Manager General Recommendations

- For compliance view for all attached baselines, latency is increased linearly with the number of attached baselines. We therefore recommend the removal of unused baselines, especially when the inventory size is large.
- Upgrading VMware Tools is faster if the virtual machine is already powered on. Otherwise, Update Manager must power on the virtual machine before the VMware Tools upgrade, which could increase the overall latency.
- Upgrading virtual machine hardware is faster if the virtual machine is already powered off. Otherwise, Update Manager must power off the virtual machine before upgrading the virtual hardware, which could increase the overall latency.

  NOTE Because VMware Tools must be up to date before virtual hardware is upgraded, Update Manager might need to upgrade VMware Tools before upgrading virtual hardware. In such cases the process is faster if the virtual machine is already powered-on.

Update Manager Cluster Remediation

- Limiting the remediation concurrency level (i.e., the maximum number of hosts that can be simultaneously updated) to half the number of hosts in the cluster can reduce vMotion intensity, often resulting in better overall host remediation performance. (This option can be set using the cluster remediate wizard.)
- When all hosts in a cluster are ready to enter maintenance mode (that is, they have no virtual machines powered on), concurrent host remediation will typically be faster than sequential host remediation.
- Cluster remediation is most likely to succeed when the cluster is no more than 80% utilized. Thus for heavily-used clusters, cluster remediation is best performed during off-peak periods, when utilization drops below 80%. If this is not possible, it is best to suspend or power-off some virtual machines before the operation is begun.

Update Manager Bandwidth Throttling

- During remediation or staging operations, hosts download patches. On slow networks you can prevent network congestion by configuring hosts to use bandwidth throttling. By allocating comparatively more bandwidth to some hosts, those hosts can more quickly finish remediation or staging.
- To ensure that network bandwidth is allocated as expected, the sum of the bandwidth allocated to multiple hosts on a single network link should not exceed the bandwidth of that link. Otherwise, the hosts will attempt to utilize bandwidth up to their allocation, resulting in bandwidth utilization that might not be proportional to the configured allocations.
- Bandwidth throttling applies only to hosts that are downloading patches. If a host is not in the process of patch downloading, any bandwidth throttling configuration on that host will not affect the bandwidth available in the network link.

Glossary

A

ALUA (Asymmetric Logical Unit Access)
A feature included in some storage arrays that allows the array itself to designate paths as "Active Optimized."

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

AMD-Vi
AMD's version of hardware-assisted I/O MMU, included in some 64-bit Intel processors, also called AMD I/O Virtualization or IOMMU. See also I/O MMU.

B

Ballooning
A technique used in VMware ESXi 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.

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 Scheduler (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 ESXi. The E1000 adapter emulates an Intel E1000 device. See also vlance and VMXNET.

Enhanced VMXNET
One of the virtual network adapters available in a virtual machine running in ESXi. The Enhanced VMXNET adapter is a high-performance paravirtualized device with drivers (available in VMware Tools) for many guest operating systems. See also VMXNET, VMXNET3, E1000, vlance, and NIC Morphing.

EPT (Extended Page Tables)
Intel's implementation of hardware virtual MMU.

F

Fault Tolerance (FT)
A feature in vSphere 4.x that runs a secondary copy of a virtual machine on a secondary host and seamlessly switches to that secondary copy in the event of failure of the primary host.

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 Virtual MMU
A feature of some recent CPUs that performs virtualization of the memory management unit (MMU) in hardware, rather than in the virtualization layer. Also called RVI or NPT by AMD and EPT by Intel.

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).

Host Power Management
Host power management reduces the power consumption of ESXi hosts while they are running. See also Distributed Power Management.

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. Also called symmetric multithreading (SMT).

I

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

Intel VT-x
Intel's version of hardware-assisted CPU virtualization, included in some 64-bit Intel processors. See also Virtualization Assist.

Intel VT-d
Intel's version of hardware-assisted I/O MMU, included in some 64-bit Intel processors, also called Intel Virtualization Technology for Directed I/O. See also I/O MMU.

I/O MMU
A processor feature that remaps I/O DMA transfers and device interrupts. This can allow virtual machines to have direct access to hardware I/O devices, such as network cards. See also AMD Vi and Intel VT-d.

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 network 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 and the VMXNET3 virtual network adapters (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).

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.

LRO (Large Receive Offload)
A method of increasing network receive throughput included in some network adapters.

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

Memory Compression
One of a number of techniques used by ESXi to allow memory overcommitment.

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.

NetQueue
A technology that significantly improves performance of 10 Gigabit Ethernet network adapters in virtualized environments.

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).

Network I/O Control (NetIOC)
A vSphere feature that allows the allocation of network bandwidth to six network resource groups: vMotion, NFS, iSCSI, Fault Tolerance, virtual machine, and management.

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 might 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)
AMD's implementation of hardware virtual MMU. Also called RVI.

P

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

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

PCI-X (PCI Extended)
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.

PVSCSI
A virtual storage adapter that offers a significant reduction in CPU utilization as well as potentially increased throughput compared to the default virtual storage adapters.

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.

Receive-side scaling (RSS)
Allows network packet receive processing to be scheduled in parallel on multiple processors.

RVI (Rapid Virtualization Indexing)
AMD's implementation of hardware virtual MMU. Also called NPT.

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.

Shadow Page Tables
A set of page tables maintained by ESXi 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.

SplitRX Mode
A feature in ESXi that can significantly improve network performance for some workloads.

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.

Storage DRS
A vSphere feature that provides I/O load balancing across datastores within a datastore cluster. This load balancing can avoid storage performance bottlenecks or address them if they occur.

Storage I/O Control (SIOC)
A vSphere feature that allows an entire datastore's I/O resources to be proportionally allocated to the virtual machines accessing that datastore.

Storage vMotion
A feature allowing running virtual machines to be migrated from one datastore to another with no downtime.

Swap to host cache
A new feature in ESXi 5.0 that uses a relatively small amount of SSD storage to significantly reduce the performance impact of host-level memory swapping.

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

Symmetric Multithreading (SMT)
Another name for hyper-threading. See also Hyper-Threading.

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 E1000, Enhanced VMXNET, and VMXNET3 virtual network adapters (but not by the normal VMXNET adapter).

U

Uniprocessor (UP)
A single-processor architecture (single-core architecture, to use current terminology). 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.

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.

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 NUMA (vNUMA)
A feature in ESXi that exposes NUMA topology to the guest operating system, allowing NUMA-aware guest operating systems and applications to make the most efficient use of the underlying hardware's NUMA architecture.

Virtual SMP
A VMware proprietary technology that supports multiple virtual CPUs (vCPUs) 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 might 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 ESXi. 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.

DirectPath I/O
A vSphere feature that leverages Intel VT-d and AMD-Vi hardware support to allow guest operating systems to directly access hardware devices.

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

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

VMware Infrastructure Client (VI Client)
A graphical user interface used to manage ESX/ESXi hosts or vCenter servers. Renamed vSphere Client in vSphere 4.0.

VMware vCenter Update Manager
Provides a patch management framework for VMware vSphere. It can be used to apply patches, updates, and upgrades to VMware ESX and ESXi hosts, VMware Tools and virtual hardware, and so on.

VMware vStorage APIs for Array Integration (VAAI)
A set of APIs that can improve storage scalability by offloading to VAAI-capable storage hardware a number of operations instead of performing those operations in ESXi.

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.

VMX Swap
A feature allowing ESXI to swap to disk some of the memory it reserves for the virtual machine executable (VMX) process.

VMXNET
One of the virtual network adapters available in a virtual machine running in ESXi. The VMXNET adapter is a high-performance paravirtualized device with drivers (available in VMware Tools) for many guest operating systems. See also Enhanced VMXNET, VMXNET3, E1000, vlance, and NIC Morphing.

VMXNET3 (VMXNET Generation 3)
The latest in the VMXNET family of paravirtualized network drivers. Requires virtual hardware version 7 or later.

vSphere Client.
A graphical user interface used to manage ESX/ESXi hosts or vCenter servers. Previously called the VMware Infrastructure Client (VI Client).

vSphere Web Client.
A browser-based user interface used to manage ESX/ESXi hosts and vCenter servers.

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

X