vSphere Resource Management

Update 1
ESXi 5.0
vCenter Server 5.0

This document supports the version of each product listed and supports all subsequent versions until the document is replaced by a new edition. To check for more recent editions of this document, see http://www.vmware.com/support/pubs.

EN-000790-00

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

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

Contents

About vSphere Resource Management

vSphere Resource Management describes resource management for VMware® ESXi and vCenter® Server environments.

This documentation focuses on the following topics.

- Resource allocation and resource management concepts
- Virtual machine attributes and admission control
- Resource pools and how to manage them
- Clusters, vSphere® Distributed Resource Scheduler (DRS), vSphere Distributed Power Management (DPM), and how to work with them
- Datastore clusters, Storage DRS, Storage I/O Control, and how to work with them
- Advanced resource management options
- Performance considerations

Intended Audience

This information is for system administrators who want to understand how the system manages resources and how they can customize the default behavior. It's also essential for anyone who wants to understand and use resource pools, clusters, DRS, datastore clusters, Storage DRS, Storage I/O Control, or vSphere DPM.

This documentation assumes you have a working knowledge of VMware ESXi and of vCenter Server.

Getting Started with Resource Management 1

To understand resource management, you must be aware of its components, its goals, and how best to implement it in a cluster setting.

Resource allocation settings for a virtual machine (shares, reservation, and limit) are discussed, including how to set them and how to view them. Also, admission control, the process whereby resource allocation settings are validated against existing resources is explained.

Resource management is the allocation of resources from resource providers to resource consumers.

The need for resource management arises from the overcommitment of resources—that is, more demand than capacity and from the fact that demand and capacity vary over time. Resource management allows you to dynamically reallocate resources, so that you can more efficiently use available capacity.

This chapter includes the following topics:

Resource Types

Resources include CPU, memory, power, storage, and network resources.

NOTE ESXi manages network bandwidth and disk resources on a per-host basis, using network traffic shaping and a proportional share mechanism, respectively.

Resource Providers

Hosts and clusters, including datastore clusters, are providers of physical resources.

For hosts, available resources are the host's hardware specification, minus the resources used by the virtualization software.

A cluster is a group of hosts. You can create a cluster using vSphere Client, and add multiple hosts to the cluster. vCenter Server manages these hosts' resources jointly: the cluster owns all of the CPU and memory of all hosts. You can enable the cluster for joint load balancing or failover. See Chapter 9, "Creating a DRS Cluster," on page 51 for more information.

A datastore cluster is a group of datastores. Like DRS clusters, you can create a datastore cluster using the vSphere Client, and add multiple datstores to the cluster. vCenter Server manages the datastore resources jointly. You can enable Storage DRS to balance I/O load and space utilization. See Chapter 11, "Creating a Datastore Cluster," on page 77.

Resource Consumers

Virtual machines are resource consumers.

The default resource settings assigned during creation work well for most machines. You can later edit the virtual machine settings to allocate a share-based percentage of the total CPU, memory, and storage I/O of the resource provider or a guaranteed reservation of CPU and memory. When you power on that virtual machine, the server checks whether enough unreserved resources are available and allows power on only if there are enough resources. This process is called admission control.

A resource pool is a logical abstraction for flexible management of resources. Resource pools can be grouped into hierarchies and used to hierarchically partition available CPU and memory resources. Accordingly, resource pools can be considered both resource providers and consumers. They provide resources to child resource pools and virtual machines, but are also resource consumers because they consume their parents' resources. See Chapter 8, "Managing Resource Pools," on page 43.

ESXi hosts allocate each virtual machine a portion of the underlying hardware resources based on a number of factors:

- Total available resources for the ESXi host (or the cluster).
- Number of virtual machines powered on and resource usage by those virtual machines.
- Overhead required to manage the virtualization.
- Resource limits defined by the user.

Goals of Resource Management

When managing your resources, you should be aware of what your goals are.

In addition to resolving resource overcommitment, resource management can help you accomplish the following:

- Performance Isolation—prevent virtual machines from monopolizing resources and guarantee predictable service rates.
- Efficient Utilization—exploit undercommitted resources and overcommit with graceful degradation.
- Easy Administration—control the relative importance of virtual machines, provide flexible dynamic partitioning, and meet absolute service-level agreements.

Configuring Resource Allocation Settings

2

When available resource capacity does not meet the demands of the resource consumers (and virtualization overhead), administrators might need to customize the amount of resources that are allocated to virtual machines or to the resource pools in which they reside.

Use the resource allocation settings (shares, reservation, and limit) to determine the amount of CPU, memory, and storage resources provided for a virtual machine. In particular, administrators have several options for allocating resources.

- Reserve the physical resources of the host or cluster.
- Ensure that a certain amount of memory for a virtual machine is provided by the physical memory of the ESXi machine.
- Guarantee that a particular virtual machine is always allocated a higher percentage of the physical resources than other virtual machines.
- Set an upper bound on the resources that can be allocated to a virtual machine.

This chapter includes the following topics:

Resource Allocation Shares

Shares specify the relative importance of a virtual machine (or resource pool). If a virtual machine has twice as many shares of a resource as another virtual machine, it is entitled to consume twice as much of that resource when these two virtual machines are competing for resources.

Shares are typically specified as **High**, **Normal**, or **Low** and these values specify share values with a 4:2:1 ratio, respectively. You can also select **Custom** to assign a specific number of shares (which expresses a proportional weight) to each virtual machine.

Specifying shares makes sense only with regard to sibling virtual machines or resource pools, that is, virtual machines or resource pools with the same parent in the resource pool hierarchy. Siblings share resources according to their relative share values, bounded by the reservation and limit. When you assign shares to a virtual machine, you always specify the priority for that virtual machine relative to other powered-on virtual machines.

The following table shows the default CPU and memory share values for a virtual machine. For resource pools, the default CPU and memory share values are the same, but must be multiplied as if the resource pool were a virtual machine with four virtual CPUs and 16 GB of memory.

Table 2-1. Share Values

Setting	CPU share values	Memory share values
High	2000 shares per virtual CPU	20 shares per megabyte of configured virtual machine memory.
Normal	1000 shares per virtual CPU	10 shares per megabyte of configured virtual machine memory.
Low	500 shares per virtual CPU	5 shares per megabyte of configured virtual machine memory.

For example, an SMP virtual machine with two virtual CPUs and 1GB RAM with CPU and memory shares set to **Normal** has 2x1000=2000 shares of CPU and 10x1024=10240 shares of memory.

NOTE Virtual machines with more than one virtual CPU are called SMP (symmetric multiprocessing) virtual machines. ESXi supports up to 32 virtual CPUs per virtual machine.

The relative priority represented by each share changes when a new virtual machine is powered on. This affects all virtual machines in the same resource pool. All of the virtual machines have the same number of virtual CPUs. Consider the following examples.

- Two CPU-bound virtual machines run on a host with 8GHz of aggregate CPU capacity. Their CPU shares are set to **Normal** and get 4GHz each.
- A third CPU-bound virtual machine is powered on. Its CPU shares value is set to **High**, which means it should have twice as many shares as the machines set to **Normal**. The new virtual machine receives 4GHz and the two other machines get only 2GHz each. The same result occurs if the user specifies a custom share value of 2000 for the third virtual machine.

Resource Allocation Reservation

A reservation specifies the guaranteed minimum allocation for a virtual machine.

vCenter Server or ESXi allows you to power on a virtual machine only if there are enough unreserved resources to satisfy the reservation of the virtual machine. The server guarantees that amount even when the physical server is heavily loaded. The reservation is expressed in concrete units (megahertz or megabytes).

For example, assume you have 2GHz available and specify a reservation of 1GHz for VM1 and 1GHz for VM2. Now each virtual machine is guaranteed to get 1GHz if it needs it. However, if VM1 is using only 500MHz, VM2 can use 1.5GHz.

Reservation defaults to 0. You can specify a reservation if you need to guarantee that the minimum required amounts of CPU or memory are always available for the virtual machine.

Resource Allocation Limit

Limit specifies an upper bound for CPU, memory, or storage I/O resources that can be allocated to a virtual machine.

A server can allocate more than the reservation to a virtual machine, but never allocates more than the limit, even if there are unused resources on the system. The limit is expressed in concrete units (megahertz, megabytes, or I/O operations per second).

CPU, memory, and storage I/O resource limits default to unlimited. When the memory limit is unlimited, the amount of memory configured for the virtual machine when it was created becomes its effective limit.

In most cases, it is not necessary to specify a limit. There are benefits and drawbacks:

- Benefits — Assigning a limit is useful if you start with a small number of virtual machines and want to manage user expectations. Performance deteriorates as you add more virtual machines. You can simulate having fewer resources available by specifying a limit.
- Drawbacks — You might waste idle resources if you specify a limit. The system does not allow virtual machines to use more resources than the limit, even when the system is underutilized and idle resources are available. Specify the limit only if you have good reasons for doing so.

Resource Allocation Settings Suggestions

Select resource allocation settings (shares, reservation, and limit) that are appropriate for your ESXi environment.

The following guidelines can help you achieve better performance for your virtual machines.

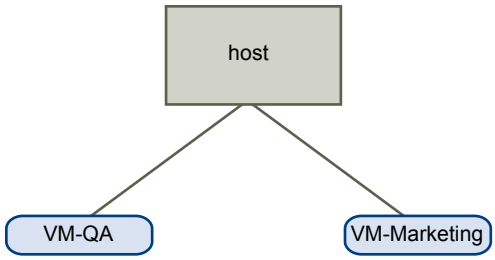
- If you expect frequent changes to the total available resources, use **Shares** to allocate resources fairly across virtual machines. If you use **Shares**, and you upgrade the host, for example, each virtual machine stays at the same priority (keeps the same number of shares) even though each share represents a larger amount of memory, CPU, or storage I/O resources.
- Use **Reservation** to specify the minimum acceptable amount of CPU or memory, not the amount you want to have available. The host assigns additional resources as available based on the number of shares, estimated demand, and the limit for your virtual machine. The amount of concrete resources represented by a reservation does not change when you change the environment, such as by adding or removing virtual machines.
- When specifying the reservations for virtual machines, do not commit all resources (plan to leave at least 10% unreserved). As you move closer to fully reserving all capacity in the system, it becomes increasingly difficult to make changes to reservations and to the resource pool hierarchy without violating admission control. 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.

Changing Resource Allocation Settings—Example

The following example illustrates how you can change resource allocation settings to improve virtual machine performance.

Assume that on an ESXi host, you have created two new virtual machines—one each for your QA (VM-QA) and Marketing (VM-Marketing) departments.

Figure 2-1. Single Host with Two Virtual Machines



In the following example, assume that VM-QA is memory intensive and accordingly you want to change the resource allocation settings for the two virtual machines to:

- Specify that, when system memory is overcommitted, VM-QA can use twice as much memory and CPU as the Marketing virtual machine. Set the memory shares and CPU shares for VM-QA to **High** and for VM-Marketing set them to **Normal**.

- Ensure that the Marketing virtual machine has a certain amount of guaranteed CPU resources. You can do so using a reservation setting.

Procedure

1 Start the vSphere Client and connect to a vCenter Server system.

2 Right-click **VM-QA**, the virtual machine for which you want to change shares, and select **Edit Settings**.

3 Select the **Resources** tab, and in the CPU panel, select **High** from the **Shares** drop-down menu.

4 In the Memory panel, select **High** from the **Shares** drop-down menu.

5 Click **OK**.

6 Right-click the marketing virtual machine (**VM-Marketing**) and select **Edit Settings**.

7 In the CPU panel, change the **Reservation** value to the desired number.

8 Click **OK**.

If you select the cluster's **Resource Allocation** tab and click **CPU**, you should see that shares for **VM-QA** are twice that of the other virtual machine. Also, because the virtual machines have not been powered on, the **Reservation Used** fields have not changed.

Admission Control

When you power on a virtual machine, the system checks the amount of CPU and memory resources that have not yet been reserved. Based on the available unreserved resources, the system determines whether it can guarantee the reservation for which the virtual machine is configured (if any). This process is called admission control.

If enough unreserved CPU and memory are available, or if there is no reservation, the virtual machine is powered on. Otherwise, an `Insufficient Resources` warning appears.

NOTE In addition to the user-specified memory reservation, for each virtual machine there is also an amount of overhead memory. This extra memory commitment is included in the admission control calculation.

When the vSphere DPM feature is enabled, hosts might be placed in standby mode (that is, powered off) to reduce power consumption. The unreserved resources provided by these hosts are considered available for admission control. If a virtual machine cannot be powered on without these resources, a recommendation to power on sufficient standby hosts is made.

CPU Virtualization Basics 3

CPU virtualization emphasizes performance and runs directly on the processor whenever possible. The underlying physical resources are used whenever possible and the virtualization layer runs instructions only as needed to make virtual machines operate as if they were running directly on a physical machine.

CPU virtualization is not the same thing as emulation. ESXi does not use emulation to run virtual CPUs. With emulation, all operations are run in software by an emulator. A software emulator allows programs to run on a computer system other than the one for which they were originally written. The emulator does this by emulating, or reproducing, the original computer's behavior by accepting the same data or inputs and achieving the same results. Emulation provides portability and runs software designed for one platform across several platforms.

When CPU resources are overcommitted, the ESXi host time-slices the physical processors across all virtual machines so each virtual machine runs as if it has its specified number of virtual processors. When an ESXi host runs multiple virtual machines, it allocates to each virtual machine a share of the physical resources. With the default resource allocation settings, all virtual machines associated with the same host receive an equal share of CPU per virtual CPU. This means that a single-processor virtual machines is assigned only half of the resources of a dual-processor virtual machine.

This chapter includes the following topics:

Software-Based CPU Virtualization

With software-based CPU virtualization, the guest application code runs directly on the processor, while the guest privileged code is translated and the translated code executes on the processor.

The translated code is slightly larger and usually executes more slowly than the native version. As a result, guest programs, which have a small privileged code component, run with speeds very close to native. Programs with a significant privileged code component, such as system calls, traps, or page table updates can run slower in the virtualized environment.

Hardware-Assisted CPU Virtualization

Certain processors provide hardware assistance for CPU virtualization.

When using this assistance, the guest can use a separate mode of execution called guest mode. The guest code, whether application code or privileged code, runs in the guest mode. On certain events, the processor exits out of guest mode and enters root mode. The hypervisor executes in the root mode, determines the reason for the exit, takes any required actions, and restarts the guest in guest mode.

When you use hardware assistance for virtualization, there is no need to translate the code. As a result, system calls or trap-intensive workloads run very close to native speed. Some workloads, such as those involving updates to page tables, lead to a large number of exits from guest mode to root mode. Depending on the number of such exits and total time spent in exits, hardware-assisted CPU virtualization can speed up execution significantly.

Virtualization and Processor-Specific Behavior

Although VMware software virtualizes the CPU, the virtual machine detects the specific model of the processor on which it is running.

Processor models might differ in the CPU features they offer, and applications running in the virtual machine can make use of these features. Therefore, it is not possible to use vMotion® to migrate virtual machines between systems running on processors with different feature sets. You can avoid this restriction, in some cases, by using Enhanced vMotion Compatibility (EVC) with processors that support this feature. See the *vCenter Server and Host Management* documentation for more information.

Performance Implications of CPU Virtualization

CPU virtualization adds varying amounts of overhead depending on the workload and the type of virtualization used.

An application is CPU-bound if it spends most of its time executing instructions rather than waiting for external events such as user interaction, device input, or data retrieval. For such applications, the CPU virtualization overhead includes the additional instructions that must be executed. This overhead takes CPU processing time that the application itself can use. CPU virtualization overhead usually translates into a reduction in overall performance.

For applications that are not CPU-bound, CPU virtualization likely translates into an increase in CPU use. If spare CPU capacity is available to absorb the overhead, it can still deliver comparable performance in terms of overall throughput.

ESXi supports up to 32 virtual processors (CPUs) for each virtual machine.

NOTE Deploy single-threaded applications on uniprocessor virtual machines, instead of on SMP virtual machines, for the best performance and resource use.

Single-threaded applications can take advantage only of a single CPU. Deploying such applications in dual-processor virtual machines does not speed up the application. Instead, it causes the second virtual CPU to use physical resources that other virtual machines could otherwise use.

---

4 Administering CPU Resources

You can configure virtual machines with one or more virtual processors, each with its own set of registers and control structures.

When a virtual machine is scheduled, its virtual processors are scheduled to run on physical processors. The VMkernel Resource Manager schedules the virtual CPUs on physical CPUs, thereby managing the virtual machine's access to physical CPU resources. ESXi supports virtual machines with up to 32 virtual CPUs.

This chapter includes the following topics:

View Processor Information

You can access information about current CPU configuration through the vSphere Client or using the vSphere SDK.

Procedure

1 In the vSphere Client, select the host and click the **Configuration** tab.

2 Select **Processors**.

You can view the information about the number and type of physical processors and the number of logical processors.

NOTE In hyperthreaded systems, each hardware thread is a logical processor. For example, a dual-core processor with hyperthreading enabled has two cores and four logical processors.

3 (Optional) You can also disable or enable hyperthreading by clicking **Properties**.

Specifying CPU Configuration

You can specify CPU configuration to improve resource management. However, if you do not customize CPU configuration, the ESXi host uses defaults that work well in most situations.

You can specify CPU configuration in the following ways:

- Use the attributes and special features available through the vSphere Client. The vSphere Client graphical user interface (GUI) allows you to connect to the ESXi host or a vCenter Server system.
- Use advanced settings under certain circumstances.
- Use the vSphere SDK for scripted CPU allocation.
- Use hyperthreading.

Multicore Processors

Multicore processors provide many advantages for a host performing multitasking of virtual machines.

Intel and AMD have each developed processors which combine two or more processor cores into a single integrated circuit (often called a package or socket). VMware uses the term socket to describe a single package which can have one or more processor cores with one or more logical processors in each core.

A dual-core processor, for example, can provide almost double the performance of a single-core processor, by allowing two virtual CPUs to execute at the same time. Cores within the same processor are typically configured with a shared last-level cache used by all cores, potentially reducing the need to access slower main memory. A shared memory bus that connects a physical processor to main memory can limit performance of its logical processors if the virtual machines running on them are running memory-intensive workloads which compete for the same memory bus resources.

Each logical processor of each processor core can be used independently by the ESXi CPU scheduler to execute virtual machines, providing capabilities similar to SMP systems. For example, a two-way virtual machine can have its virtual processors running on logical processors that belong to the same core, or on logical processors on different physical cores.

The ESXi CPU scheduler can detect the processor topology and the relationships between processor cores and the logical processors on them. It uses this information to schedule virtual machines and optimize performance.

The ESXi CPU scheduler can interpret processor topology, including the relationship between sockets, cores, and logical processors. The scheduler uses topology information to optimize the placement of virtual CPUs onto different sockets to maximize overall cache utilization, and to improve cache affinity by minimizing virtual CPU migrations.

In undercommitted systems, the ESXi CPU scheduler spreads load across all sockets by default. This improves performance by maximizing the aggregate amount of cache available to the running virtual CPUs. As a result, the virtual CPUs of a single SMP virtual machine are spread across multiple sockets (unless each socket is also a NUMA node, in which case the NUMA scheduler restricts all the virtual CPUs of the virtual machine to reside on the same socket.)

In some cases, such as when an SMP virtual machine exhibits significant data sharing between its virtual CPUs, this default behavior might be sub-optimal. For such workloads, it can be beneficial to schedule all of the virtual CPUs on the same socket, with a shared last-level cache, even when the ESXi host is undercommitted. In such scenarios, you can override the default behavior of spreading virtual CPUs across packages by including the following configuration option in the virtual machine's `.vmx` configuration file:
`sched.cpu.vsmpConsolidate="TRUE"`.

Hyperthreading

Hyperthreading technology allows a single physical processor core to behave like two logical processors. The processor can run two independent applications at the same time. To avoid confusion between logical and physical processors, Intel refers to a physical processor as a socket, and the discussion in this chapter uses that terminology as well.

Intel Corporation developed hyperthreading technology to enhance the performance of its Pentium IV and Xeon processor lines. Hyperthreading technology allows a single processor core to execute two independent threads simultaneously.

While hyperthreading does not double the performance of a system, it can increase performance by better utilizing idle resources leading to greater throughput for certain important workload types. An application running on one logical processor of a busy core can expect slightly more than half of the throughput that it obtains while running alone on a non-hyperthreaded processor. Hyperthreading performance improvements are highly application-dependent, and some applications might see performance degradation with hyperthreading because many processor resources (such as the cache) are shared between logical processors.

Note On processors with Intel Hyper-Threading technology, each core can have two logical processors which share most of the core's resources, such as memory caches and functional units. Such logical processors are usually called threads.

Many processors do not support hyperthreading and as a result have only one thread per core. For such processors, the number of cores also matches the number of logical processors. The following processors support hyperthreading and have two threads per core.

- Processors based on the Intel Xeon 5500 processor microarchitecture.
- Intel Pentium 4 (HT-enabled)
- Intel Pentium EE 840 (HT-enabled)

Hyperthreading and ESXi Hosts

A host that is enabled for hyperthreading should behave similarly to a host without hyperthreading. You might need to consider certain factors if you enable hyperthreading, however.

ESXi hosts manage processor time intelligently to guarantee that load is spread smoothly across processor cores in the system. Logical processors on the same core have consecutive CPU numbers, so that CPUs 0 and 1 are on the first core together, CPUs 2 and 3 are on the second core, and so on. Virtual machines are preferentially scheduled on two different cores rather than on two logical processors on the same core.

If there is no work for a logical processor, it is put into a halted state, which 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. The VMware scheduler properly accounts for this halt time, and charges a virtual machine running with the full resources of a core more than a virtual machine running on a half core. This approach to processor management ensures that the server does not violate any of the standard ESXi resource allocation rules.

Consider your resource management needs before you enable CPU affinity on hosts using hyperthreading. For example, if you bind a high priority virtual machine to CPU 0 and another high priority virtual machine to CPU 1, the two virtual machines have to share the same physical core. In this case, it can be impossible to meet the resource demands of these virtual machines. Ensure that any custom affinity settings make sense for a hyperthreaded system.

Enable Hyperthreading

To enable hyperthreading, you must first enable it in your system's BIOS settings and then turn it on in the vSphere Client. Hyperthreading is enabled by default.

Consult your system documentation to determine whether your CPU supports hyperthreading.

Procedure

1 Ensure that your system supports hyperthreading technology.

2 Enable hyperthreading in the system BIOS.

Some manufacturers label this option **Logical Processor**, while others call it **Enable Hyperthreading**.

3 Make sure that you turn on hyperthreading for the ESXi host.

a In the vSphere Client, select the host and click the **Configuration** tab.

b Select **Processors** and click **Properties**.

c In the dialog box, you can view hyperthreading status and turn hyperthreading off or on (default).

Hyperthreading is enabled.

Set Hyperthreading Sharing Options for a Virtual Machine

You can specify how the virtual CPUs of a virtual machine can share physical cores on a hyperthreaded system.

Two virtual CPUs share a core if they are running on logical CPUs of the core at the same time. You can set this for individual virtual machines.

Procedure

1 In the vSphere Client inventory panel, right-click the virtual machine and select **Edit Settings**.

2 Click the **Resources** tab, and click **Advanced CPU**.

3 Select a hyperthreading mode for this virtual machine from the **Mode** drop-down menu.

Hyperthreaded Core Sharing Options

You can set the hyperthreaded core sharing mode for a virtual machine using the vSphere Client.

Table 4-1. Hyperthreaded Core Sharing Modes

Option	Description
Any	The default for all virtual machines on a hyperthreaded system. The virtual CPUs of a virtual machine with this setting can freely share cores with other virtual CPUs from this or any other virtual machine at any time.
None	Virtual CPUs of a virtual machine should not share cores with each other or with virtual CPUs from other virtual machines. That is, each virtual CPU from this virtual machine should always get a whole core to itself, with the other logical CPU on that core being placed into the halted state.
Internal	This option is similar to none. Virtual CPUs from this virtual machine cannot share cores with virtual CPUs from other virtual machines. They can share cores with the other virtual CPUs from the same virtual machine. You can select this option only for SMP virtual machines. If applied to a uniprocessor virtual machine, the system changes this option to none.

These options have no effect on fairness or CPU time allocation. Regardless of a virtual machine's hyperthreading settings, it still receives CPU time proportional to its CPU shares, and constrained by its CPU reservation and CPU limit values.

For typical workloads, custom hyperthreading settings should not be necessary. The options can help in case of unusual workloads that interact badly with hyperthreading. For example, an application with cache thrashing problems might slow down an application sharing its physical core. You can place the virtual machine running the application in the none or internal hyperthreading status to isolate it from other virtual machines.

If a virtual CPU has hyperthreading constraints that do not allow it to share a core with another virtual CPU, the system might deschedule it when other virtual CPUs are entitled to consume processor time. Without the hyperthreading constraints, you can schedule both virtual CPUs on the same core.

The problem becomes worse on systems with a limited number of cores (per virtual machine). In such cases, there might be no core to which the virtual machine that is descheduled can be migrated. As a result, virtual machines with hyperthreading set to none or internal can experience performance degradation, especially on systems with a limited number of cores.

Quarantining

In certain rare circumstances, ESXi might detect that an application is interacting badly with the Pentium IV hyperthreading technology. (This does not apply to systems based on the Intel Xeon 5500 processor microarchitecture.) In such cases, quarantining, which is transparent to the user, might be necessary.

For example, certain types of self-modifying code can disrupt the normal behavior of the Pentium IV trace cache and can lead to substantial slowdowns (up to 90 percent) for an application sharing a core with the problematic code. In those cases, the ESXi host quarantines the virtual CPU running this code and places its virtual machine in the none or internal mode, as appropriate.

Using CPU Affinity

By specifying a CPU affinity setting for each virtual machine, you can restrict the assignment of virtual machines to a subset of the available processors in multiprocessor systems. By using this feature, you can assign each virtual machine to processors in the specified affinity set.

CPU affinity specifies virtual machine-to-processor placement constraints and is different from the relationship created by a VM-VM or VM-Host affinity rule, which specifies virtual machine-to-virtual machine host placement constraints.

In this context, the term CPU refers to a logical processor on a hyperthreaded system and refers to a core on a non-hyperthreaded system.

The CPU affinity setting for a virtual machine applies to all of the virtual CPUs associated with the virtual machine and to all other threads (also known as worlds) associated with the virtual machine. Such virtual machine threads perform processing required for emulating mouse, keyboard, screen, CD-ROM, and miscellaneous legacy devices.

In some cases, such as display-intensive workloads, significant communication might occur between the virtual CPUs and these other virtual machine threads. Performance might degrade if the virtual machine's affinity setting prevents these additional threads from being scheduled concurrently with the virtual machine's virtual CPUs. Examples of this include a uniprocessor virtual machine with affinity to a single CPU or a two-way SMP virtual machine with affinity to only two CPUs.

For the best performance, when you use manual affinity settings, VMware recommends that you include at least one additional physical CPU in the affinity setting to allow at least one of the virtual machine's threads to be scheduled at the same time as its virtual CPUs. Examples of this include a uniprocessor virtual machine with affinity to at least two CPUs or a two-way SMP virtual machine with affinity to at least three CPUs.

Assign a Virtual Machine to a Specific Processor

Using CPU affinity, you can assign a virtual machine to a specific processor. This allows you to restrict the assignment of virtual machines to a specific available processor in multiprocessor systems.

Procedure

1 In the vSphere Client inventory panel, select a virtual machine and select **Edit Settings**.

2 Select the **Resources** tab and select **Advanced CPU**.

3 Click the **Run on processor(s)** button.

4 Select the processors where you want the virtual machine to run and click **OK**.

Potential Issues with CPU Affinity

Before you use CPU affinity, you might need to consider certain issues.

Potential issues with CPU affinity include:

- For multiprocessor systems, ESXi systems perform automatic load balancing. Avoid manual specification of virtual machine affinity to improve the scheduler's ability to balance load across processors.
- Affinity can interfere with the ESXi host's ability to meet the reservation and shares specified for a virtual machine.
- Because CPU admission control does not consider affinity, a virtual machine with manual affinity settings might not always receive its full reservation.

  Virtual machines that do not have manual affinity settings are not adversely affected by virtual machines with manual affinity settings.
- When you move a virtual machine from one host to another, affinity might no longer apply because the new host might have a different number of processors.
- The NUMA scheduler might not be able to manage a virtual machine that is already assigned to certain processors using affinity.
- Affinity can affect the host's ability to schedule virtual machines on multicore or hyperthreaded processors to take full advantage of resources shared on such processors.

Host Power Management Policies

ESXi can take advantage of several power management features that the host hardware provides to adjust the trade-off between performance and power use. You can control how ESXi uses these features by selecting a power management policy.

In general, selecting a high-performance policy provides more absolute performance, but at lower efficiency (performance per watt). Lower-power policies provide less absolute performance, but at higher efficiency.

ESXi provides five power management policies. If the host does not support power management, or if the BIOS settings specify that the host operating system is not allowed to manage power, only the Not Supported policy is available.

You select a policy for a host using the vSphere Client. If you do not select a policy, ESXi uses Balanced by default.

Table 4-2. CPU Power Management Policies

Power Management Policy	Description
Not supported	The host does not support any power management features or power management is not enabled in the BIOS.
High Performance	The VMkernel detects certain power management features, but will not use them unless the BIOS requests them for power capping or thermal events.
Balanced (Default)	The VMkernel uses the available power management features conservatively to reduce host energy consumption with minimal compromise to performance.
Low Power	The VMkernel aggressively uses available power management features to reduce host energy consumption at the risk of lower performance.
Custom	The VMkernel bases its power management policy on the values of several advanced configuration parameters. You can set these parameters in the vSphere Client Advanced Settings dialog box.

When a CPU runs at lower frequency, it can also run at lower voltage, which saves power. This type of power management is typically called Dynamic Voltage and Frequency Scaling (DVFS). ESXi attempts to adjust CPU frequencies so that virtual machine performance is not affected.

When a CPU is idle, ESXi can take advantage of deep halt states (known as C-states). The deeper the C-state, the less power the CPU uses, but the longer it takes for the CPU to resume running. When a CPU becomes idle, ESXi applies an algorithm to predict how long it will be in an idle state and chooses an appropriate C-state to enter. In power management policies that do not use deep C-states, ESXi uses only the shallowest halt state (C1) for idle CPUs.

Select a CPU Power Management Policy

You set the CPU power management policy for a host using the vSphere Client.

Prerequisites

Verify that the BIOS settings on the host system allow the operating system to control power management (for example, **OS Controlled**).

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

If the host hardware does not allow the operating system to manage power, only the Not Supported policy is available. (On some systems, only the High Performance policy is available.)

Procedure

1 In the vSphere Client inventory panel, select a host and click the **Configuration** tab.

2 Under Hardware, select **Power Management** and select **Properties**.

3 Select a power management policy for the host and click **OK**.

The policy selection is saved in the host configuration and can be used again at boot time. You can change it at any time, and it does not require a server reboot.

Configure Custom Policy Parameters for Host Power Management

When you use the Custom policy for host power management, ESXi bases its power management policy on the values of several advanced configuration parameters.

Prerequisites

Select **Custom** for the power management policy, as described in "Select a CPU Power Management Policy," on page 23.

Procedure

1 In the vSphere Client inventory, select the host and click the **Configuration** tab.

2 Under Software, select **Advanced Settings**.

3 Click **Power** in the left pane.

4 In the right pane, you can edit the power management parameters that affect the Custom policy.

Power management parameters that affect the Custom policy have descriptions that begin with **In Custom policy**. All other power parameters affect all power management policies.

NOTE The default values of power management parameters match the Balanced policy.

Parameter	Description
Power.UsePStates	Use ACPI P-states to save power when the processor is busy.
Power.MaxCpuLoad	Use P-states to save power on a CPU only when the CPU is busy for less than the given percentage of real time.
Power.MinFreqPct	Do not use any P-states slower than the given percentage of full CPU speed.
Power.UseStallCtr	Use a deeper P-state when the processor is frequently stalled waiting for events such as cache misses.
Power.TimerHz	Controls how many times per second ESXi reevaluates which P-state each CPU should be in.
Power.UseCStates	Use deep ACPI C-states (C2 or below) when the processor is idle.
Power.CStateMaxLatency	Do not use C-states whose latency is greater than this value.
Power.CStateResidencyCoef	When a CPU becomes idle, choose the deepest C-state whose latency multiplied by this value is less than the host's prediction of how long the CPU will remain idle. Larger values make ESXi more conservative about using deep C-states, while smaller values are more aggressive.
Power.CStatePredictionCoef	A parameter in the ESXi algorithm for predicting how long a CPU that becomes idle will remain idle. Changing this value is not recommended.
Power.PerfBias	Performance Energy Bias Hint (Intel-only). Sets an MSR on Intel processors to an Intel-recommended value. Intel recommends 0 for high performance, 6 for balanced, and 15 for low power. Other values are undefined.

Memory Virtualization Basics 5

Before you manage memory resources, you should understand how they are being virtualized and used by ESXi.

The VMkernel manages all machine memory. The VMkernel dedicates part of this managed machine memory for its own use. The rest is available for use by virtual machines. Virtual machines use machine memory for two purposes: each virtual machine requires its own memory and the virtual machine monitor (VMM) requires some memory and a dynamic overhead memory for its code and data.

The virtual and physical memory space is divided into blocks called pages. When physical memory is full, the data for virtual pages that are not present in physical memory are stored on disk. Depending on processor architecture, pages are typically 4 KB or 2 MB. See "Advanced Memory Attributes," on page 100.

This chapter includes the following topics:

Virtual Machine Memory

Each virtual machine consumes memory based on its configured size, plus additional overhead memory for virtualization.

The configured size is a construct maintained by the virtualization layer for the virtual machine. It is the amount of memory that is presented to the guest operating system, but it is independent of the amount of physical RAM that is allocated to the virtual machine, which depends on the resource settings (shares, reservation, limit) explained below.

For example, consider a virtual machine with a configured size of 1GB. When the guest operating system boots, it detects that it is running on a dedicated machine with 1GB of physical memory. The actual amount of physical host memory allocated to the virtual machine depends on its memory resource settings and memory contention on the ESXi host. In some cases, the virtual machine might be allocated the full 1GB. In other cases, it might receive a smaller allocation. Regardless of the actual allocation, the guest operating system continues to behave as though it is running on a dedicated machine with 1GB of physical memory.

Shares Specify the relative priority for a virtual machine if more than the reservation is available.

Reservation Is a guaranteed lower bound on the amount of physical memory that the host reserves for the virtual machine, even when memory is overcommitted. Set the reservation to a level that ensures the virtual machine has sufficient memory to run efficiently, without excessive paging.

After a virtual machine has accessed its full reservation, it is allowed to retain that amount of memory and this memory is not reclaimed, even if the virtual machine becomes idle. For example, some guest operating systems (for example, Linux) might not access all of the configured memory immediately after booting. Until the virtual machines accesses its full reservation, VMkernel can allocate any unused portion of its reservation to other virtual machines. However, after the guest's workload increases and it consumes its full reservation, it is allowed to keep this memory.

Limit Is an upper bound on the amount of physical memory that the host can allocate to the virtual machine. The virtual machine's memory allocation is also implicitly limited by its configured size.

Overhead memory includes space reserved for the virtual machine frame buffer and various virtualization data structures.

Memory Overcommitment

For each running virtual machine, the system reserves physical memory for the virtual machine's reservation (if any) and for its virtualization overhead.

Because of the memory management techniques the ESXi host uses, your virtual machines can use more memory than the physical machine (the host) has available. For example, you can have a host with 2GB memory and run four virtual machines with 1GB memory each. In that case, the memory is overcommitted.

Overcommitment makes sense because, typically, some virtual machines are lightly loaded while others are more heavily loaded, and relative activity levels vary over time.

To improve memory utilization, the ESXi host transfers memory from idle virtual machines to virtual machines that need more memory. Use the Reservation or Shares parameter to preferentially allocate memory to important virtual machines. This memory remains available to other virtual machines if it is not in use.

In addition, memory compression is enabled by default on ESXi hosts to improve virtual machine performance when memory is overcommitted as described in "Memory Compression," on page 36.

Memory Sharing

Many workloads present opportunities for sharing memory across virtual machines.

For example, several virtual machines might be running instances of the same guest operating system, have the same applications or components loaded, or contain common data. ESXi systems use a proprietary page-sharing technique to securely eliminate redundant copies of memory pages.

With memory sharing, a workload consisting of multiple virtual machines often consumes less memory than it would when running on physical machines. As a result, the system can efficiently support higher levels of overcommitment.

The amount of memory saved by memory sharing depends on workload characteristics. A workload of many nearly identical virtual machines might free up more than thirty percent of memory, while a more diverse workload might result in savings of less than five percent of memory.

Software-Based Memory Virtualization

ESXi virtualizes guest physical memory by adding an extra level of address translation.

- The VMM for each virtual machine maintains a mapping from the guest operating system's physical memory pages to the physical memory pages on the underlying machine. (VMware refers to the underlying host physical pages as “machine” pages and the guest operating system’s physical pages as “physical” pages.)

  Each virtual machine sees a contiguous, zero-based, addressable physical memory space. The underlying machine memory on the server used by each virtual machine is not necessarily contiguous.
- The VMM intercepts virtual machine instructions that manipulate guest operating system memory management structures so that the actual memory management unit (MMU) on the processor is not updated directly by the virtual machine.
- The ESXi host maintains the virtual-to-machine page mappings in a shadow page table that is kept up to date with the physical-to-machine mappings (maintained by the VMM).
- The shadow page tables are used directly by the processor's paging hardware.

This approach to address translation allows normal memory accesses in the virtual machine to execute without adding address translation overhead, after the shadow page tables are set up. Because the translation look-aside buffer (TLB) on the processor caches direct virtual-to-machine mappings read from the shadow page tables, no additional overhead is added by the VMM to access the memory.

Performance Considerations

The use of two-page tables has these performance implications.

- No overhead is incurred for regular guest memory accesses.
- Additional time is required to map memory within a virtual machine, which might mean:
  - The virtual machine operating system is setting up or updating virtual address to physical address mappings.
  - The virtual machine operating system is switching from one address space to another (context switch).
- Like CPU virtualization, memory virtualization overhead depends on workload.

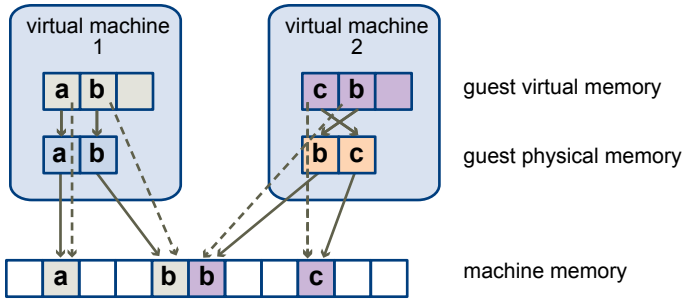
Hardware-Assisted Memory Virtualization

Some CPUs, such as AMD SVM-V and the Intel Xeon 5500 series, provide hardware support for memory virtualization by using two layers of page tables.

The first layer of page tables stores guest virtual-to-physical translations, while the second layer of page tables stores guest physical-to-machine translation. The TLB (translation look-aside buffer) is a cache of translations maintained by the processor's memory management unit (MMU) hardware. A TLB miss is a miss in this cache and the hardware needs to go to memory (possibly many times) to find the required translation. For a TLB miss to a certain guest virtual address, the hardware looks at both page tables to translate guest virtual address to host physical address.

The diagram illustrates the ESXi implementation of memory virtualization.

Figure 5-1. ESXi Memory Mapping



- The boxes represent pages, and the arrows show the different memory mappings.
- The arrows from guest virtual memory to guest physical memory show the mapping maintained by the page tables in the guest operating system. (The mapping from virtual memory to linear memory for x86-architecture processors is not shown.)
- The arrows from guest physical memory to machine memory show the mapping maintained by the VMM.
- The dashed arrows show the mapping from guest virtual memory to machine memory in the shadow page tables also maintained by the VMM. The underlying processor running the virtual machine uses the shadow page table mappings.

Because of the extra level of memory mapping introduced by virtualization, ESXi can effectively manage memory across all virtual machines. Some of the physical memory of a virtual machine might be mapped to shared pages or to pages that are unmapped, or swapped out.

A host performs virtual memory management without the knowledge of the guest operating system and without interfering with the guest operating system's own memory management subsystem.

Performance Considerations

When you use hardware assistance, you eliminate the overhead for software memory virtualization. In particular, hardware assistance eliminates the overhead required to keep shadow page tables in synchronization with guest page tables. However, the TLB miss latency when using hardware assistance is significantly higher. As a result, whether or not a workload benefits by using hardware assistance primarily depends on the overhead the memory virtualization causes when using software memory virtualization. If a workload involves a small amount of page table activity (such as process creation, mapping the memory, or context switches), software virtualization does not cause significant overhead. Conversely, workloads with a large amount of page table activity are likely to benefit from hardware assistance.

Administering Memory Resources 6

Using the vSphere Client you can view information about and make changes to memory allocation settings. To administer your memory resources effectively, you must also be familiar with memory overhead, idle memory tax, and how ESXi hosts reclaim memory.

When administering memory resources, you can specify memory allocation. If you do not customize memory allocation, the ESXi host uses defaults that work well in most situations.

You can specify memory allocation in several ways.

- Use the attributes and special features available through the vSphere Client. The vSphere Client user interface allows you to connect to the ESXi host or vCenter Server system.
- Use advanced settings.
- Use the vSphere SDK for scripted memory allocation.

This chapter includes the following topics:

Understanding Memory Overhead

Virtualization of memory resources has some associated overhead.

ESXi virtual machines can incur two kinds of memory overhead.

- The additional time to access memory within a virtual machine.
- The extra space needed by the ESXi host for its own code and data structures, beyond the memory allocated to each virtual machine.

ESXi memory virtualization adds little time overhead to memory accesses. Because the processor's paging hardware uses page tables (shadow page tables for software-based approach or nested page tables for hardware-assisted approach) directly, most memory accesses in the virtual machine can execute without address translation overhead.

The memory space overhead has two components.

- A fixed, system-wide overhead for the VMkernel.
- Additional overhead for each virtual machine.

Overhead memory includes space reserved for the virtual machine frame buffer and various virtualization data structures, such as shadow page tables. Overhead memory depends on the number of virtual CPUs and the configured memory for the guest operating system.

ESXi also provides optimizations such as memory sharing to reduce the amount of physical memory used on the underlying server. These optimizations can save more memory than is taken up by the overhead.

Overhead Memory on Virtual Machines

Virtual machines require a certain amount of available overhead memory to power on. You should be aware of the amount of this overhead.

The following table lists the amount of overhead memory a virtual machine requires to power on. After a virtual machine is running, the amount of overhead memory it uses might differ from the amount listed in the table. The sample values were collected with VMX swap enabled and hardware MMU enabled for the virtual machine. (VMX swap is enabled by default.)

NOTE The table provides a sample of overhead memory values and does not attempt to provide information about all possible configurations. You can configure a virtual machine to have up to 64 virtual CPUs, depending on the number of licensed CPUs on the host and the number of CPUs that the guest operating system supports.

Table 6-1. Sample Overhead Memory on Virtual Machines

Memory (MB)	1 VCPU	2 VCPUs	4 VCPUs	8 VCPUs
256	20.29	24.28	32.23	48.16
1024	25.90	29.91	37.86	53.82
4096	48.64	52.72	60.67	76.78
16384	139.62	143.98	151.93	168.60

How ESXi Hosts Allocate Memory

A host allocates the memory specified by the `Limit` parameter to each virtual machine, unless memory is overcommitted. ESXi never allocates more memory to a virtual machine than its specified physical memory size.

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

When memory is overcommitted, each virtual machine is allocated an amount of memory somewhere between what is specified by **Reservation** and what is specified by **Limit**. The amount of memory granted to a virtual machine above its reservation usually varies with the current memory load.

A host determines allocations for each virtual machine based on the number of shares allocated to it and an estimate of its recent working set size.

- Shares — ESXi hosts use a modified proportional-share memory allocation policy. Memory shares entitle a virtual machine to a fraction of available physical memory.

- Working set size — ESXi hosts estimate the working set for a virtual machine by monitoring memory activity over successive periods of virtual machine execution time. Estimates are smoothed over several time periods using techniques that respond rapidly to increases in working set size and more slowly to decreases in working set size.

  This approach ensures that a virtual machine from which idle memory is reclaimed can ramp up quickly to its full share-based allocation when it starts using its memory more actively.

  Memory activity is monitored to estimate the working set sizes for a default period of 60 seconds. To modify this default , adjust the `Mem.SamplePeriod` advanced setting. See "Set Advanced Host Attributes," on page 99.

VMX Swap Files

Virtual machine executable (VMX) swap files allow the host to greatly reduce the amount of overhead memory reserved for the VMX process.

NOTE VMX swap files are not related to the swap to host cache feature or to regular host-level swap files.

ESXi reserves memory per virtual machine for a variety of purposes. Memory for the needs of certain components, such as the virtual machine monitor (VMM) and virtual devices, is fully reserved when a virtual machine is powered on. However, some of the overhead memory that is reserved for the VMX process can be swapped. The VMX swap feature reduces the VMX memory reservation significantly (for example, from about 50MB or more per virtual machine to about 10MB per virtual machine). This allows the remaining memory to be swapped out when host memory is overcommitted, reducing overhead memory reservation for each virtual machine.

The host creates VMX swap files automatically, provided there is sufficient free disk space at the time a virtual machine is powered on.

Memory Tax for Idle Virtual Machines

If a virtual machine is not actively using all of its currently allocated memory, ESXi charges more for idle memory than for memory that is in use. This is done to help prevent virtual machines from hoarding idle memory.

The idle memory tax is applied in a progressive fashion. The effective tax rate increases as the ratio of idle memory to active memory for the virtual machine rises. (In earlier versions of ESXi that did not support hierarchical resource pools, all idle memory for a virtual machine was taxed equally.)

You can modify the idle memory tax rate with the `Mem.IdleTax` option. Use this option, together with the `Mem.SamplePeriod` advanced attribute, to control how the system determines target memory allocations for virtual machines. See "Set Advanced Host Attributes," on page 99.

NOTE In most cases, changes to `Mem.IdleTax` are not necessary nor appropriate.

Memory Reclamation

ESXi hosts can reclaim memory from virtual machines.

A host allocates the amount of memory specified by a reservation directly to a virtual machine. Anything beyond the reservation is allocated using the host's physical resources or, when physical resources are not available, handled using special techniques such as ballooning or swapping. Hosts can use two techniques for dynamically expanding or contracting the amount of memory allocated to virtual machines.

- ESXi systems use a memory balloon driver (`vmmemctl`), loaded into the guest operating system running in a virtual machine. See "Memory Balloon Driver," on page 32.

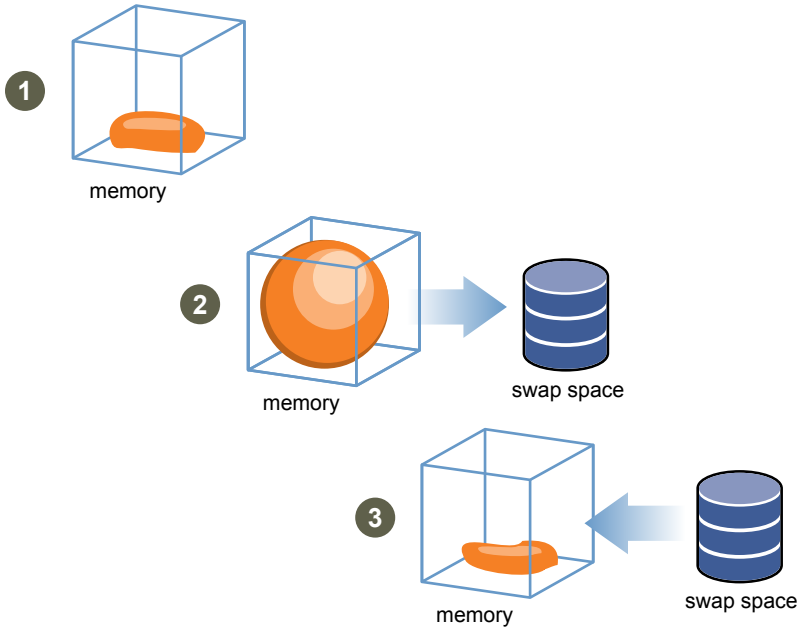
- ESXi systems page from a virtual machine to a server swap file without any involvement by the guest operating system. Each virtual machine has its own swap file.

Memory Balloon Driver

The memory balloon driver (`vmmemctl`) collaborates with the server to reclaim pages that are considered least valuable by the guest operating system.

The driver uses a proprietary ballooning technique that provides predictable performance that closely matches the behavior of a native system under similar memory constraints. This technique increases or decreases memory pressure on the guest operating system, causing the guest to use its own native memory management algorithms. When memory is tight, the guest operating system determines which pages to reclaim and, if necessary, swaps them to its own virtual disk.

Figure 6-1. Memory Ballooning in the Guest Operating System



NOTE You must configure the guest operating system with sufficient swap space. Some guest operating systems have additional limitations.

If necessary, you can limit the amount of memory `vmmemctl` reclaims by setting the **sched.mem.maxmemctl** parameter for a specific virtual machine. This option specifies the maximum amount of memory that can be reclaimed from a virtual machine in megabytes (MB). See "Set Advanced Virtual Machine Attributes," on page 102.

Using Swap Files

You can specify the location of your swap file, reserve swap space when memory is overcommitted, and delete a swap file.

ESXi hosts use swapping to forcibly reclaim memory from a virtual machine when the `vmmemctl` driver is not available or is not responsive.

- It was never installed.
- It is explicitly disabled.
- It is not running (for example, while the guest operating system is booting).
- It is temporarily unable to reclaim memory quickly enough to satisfy current system demands.

- It is functioning properly, but maximum balloon size is reached.

Standard demand-paging techniques swap pages back in when the virtual machine needs them.

Swap File Location

By default, the swap file is created in the same location as the virtual machine's configuration file.

A swap file is created by the ESXi host when a virtual machine is powered on. If this file cannot be created, the virtual machine cannot power on. Instead of accepting the default, you can also:

- Use per-virtual machine configuration options to change the datastore to another shared storage location.
- Use host-local swap, which allows you to specify a datastore stored locally on the host. This allows you to swap at a per-host level, saving space on the SAN. However, it can lead to a slight degradation in performance for vSphere vMotion because pages swapped to a local swap file on the source host must be transferred across the network to the destination host.

Enable Host-Local Swap for a DRS Cluster

Host-local swap allows you to specify a datastore stored locally on the host as the swap file location. You can enable host-local swap for a DRS cluster.

Procedure

1 In the vSphere Client, right-click the cluster in the inventory and select **Edit Settings**.

2 In the left pane of the cluster Settings dialog box, click **Swapfile Location**.

3 Select the **Store the swapfile in the datastore specified by the host** option and click **OK**.

4 In the vSphere Client inventory, select one of the hosts in the cluster and click the **Configuration** tab.

5 Under Software, select **Virtual Machine Swapfile Location**.

6 Select the local datastore to use and click **OK**.

7 Repeat Step 4 through Step 6 for each host in the cluster.

Host-local swap is now enabled for the DRS cluster.

Enable Host-Local Swap for a Standalone Host

Host-local swap allows you to specify a datastore stored locally on the host as the swap file location. You can enable host-local swap for a standalone host.

Procedure

1 In the vSphere Client, select the host in the inventory.

2 Click the **Configuration** tab.

3 Under Software, select **Virtual Machine Swapfile Location**.

4 Select **Store the swapfile in the swapfile datastore**.

5 Select a local datastore from the list and click **OK**.

Host-local swap is now enabled for the standalone host.

Swap Space and Memory Overcommitment

You must reserve swap space for any unreserved virtual machine memory (the difference between the reservation and the configured memory size) on per-virtual machine swap files.

This swap reservation is required to ensure that the ESXi host is able to preserve virtual machine memory under any circumstances. In practice, only a small fraction of the host-level swap space might be used.

If you are overcommitting memory with ESXi, to support the intra-guest swapping induced by ballooning, ensure that your guest operating systems also have sufficient swap space. This guest-level swap space must be greater than or equal to the difference between the virtual machine's configured memory size and its Reservation.

CAUTION If memory is overcommitted, and the guest operating system is configured with insufficient swap space, the guest operating system in the virtual machine can fail.

To prevent virtual machine failure, increase the size of the swap space in your virtual machines.

- Windows guest operating systems— Windows operating systems refer to their swap space as paging files. Some Windows operating systems try to increase the size of paging files automatically, if there is sufficient free disk space.

  See your Microsoft Windows documentation or search the Windows help files for "paging files." Follow the instructions for changing the size of the virtual memory paging file.

- Linux guest operating system — Linux operating systems refer to their swap space as swap files. For information on increasing swap files, see the following Linux man pages:

  - `mkswap` — Sets up a Linux swap area.
  - `swapon` — Enables devices and files for paging and swapping.

Guest operating systems with a lot of memory and small virtual disks (for example, a virtual machine with 8GB RAM and a 2GB virtual disk) are more susceptible to having insufficient swap space.

NOTE Do not store swap files on thin-provisioned LUNs. Running a virtual machine with a swap file that is stored on a thin-provisioned LUN can cause swap file growth failure, which can lead to termination of the virtual machine.

When you create a large swap file (for example, larger than 100GB), the amount of time it takes for the virtual machine to power on can increase significantly. To avoid this, set a high reservation for large virtual machines.

You can also place swap files on less costly storage using host-local swap files.

Delete Swap Files

If a host fails, and that host had running virtual machines that were using swap files, those swap files continue to exist and consume many gigabytes of disk space. You can delete the swap files to eliminate this problem.

Procedure

1 Restart the virtual machine that was on the host that failed.

2 Stop the virtual machine.

The swap file for the virtual machine is deleted.

Swapping to Host Cache

Datastores that are created on solid state drives (SSD) can be used to allocate space for host cache. The host reserves a certain amount of space for swapping to host cache.

The host cache is made up of files on a low-latency disk that ESXi uses as a write back cache for virtual machine swap files. The cache is shared by all virtual machines running on the host. Host-level swapping of virtual machine pages makes the best use of potentially limited SSD space.

Using swap to host cache is not the same as placing regular swap files on SSD-backed datastores. Even if you enable swap to host cache, the host still needs to create regular swap files. However, when you use swap to host cache, the speed of the storage where the host places regular swap files is less important.

The Host Cache Configuration page allows you to view the amount of space on a datastore that a host can use to swap to host cache. Only SSD-backed datastores appear in the list of datastores on the Host Cache Configuration page.

Configure the Host Cache

You can change the percentage of space allocated for host cache or disable the host's ability to swap to host cache.

Prerequisites

You must have an SSD-backed datastore in your inventory.

Procedure

1 In the vSphere Client, select the host in the inventory.

2 Click the **Configuration** tab.

3 Under Software, click **Host Cache Configuration**.

4 Select the datastore in the list and click **Properties**.

5 Select a size for the host cache allocation on the drive.

6 To disable the ability for the host to swap to host cache on a per-datastore basis, deselect the **Allocate space for host cache** check box.

7 Click **OK**.

Sharing Memory Across Virtual Machines

Many ESXi workloads present opportunities for sharing memory across virtual machines (as well as within a single virtual machine).

For example, several virtual machines might be running instances of the same guest operating system, have the same applications or components loaded, or contain common data. In such cases, a host uses a proprietary transparent page sharing technique to securely eliminate redundant copies of memory pages. With memory sharing, a workload running in virtual machines often consumes less memory than it would when running on physical machines. As a result, higher levels of overcommitment can be supported efficiently.

Use the `Mem.ShareScanTime` and `Mem.ShareScanGHz` advanced settings to control the rate at which the system scans memory to identify opportunities for sharing memory.

You can also disable sharing for individual virtual machines by setting the `sched.mem.pshare.enable` option to **FALSE** (this option defaults to **TRUE**). See "Set Advanced Virtual Machine Attributes," on page 102.

ESXi memory sharing runs as a background activity that scans for sharing opportunities over time. The amount of memory saved varies over time. For a fairly constant workload, the amount generally increases slowly until all sharing opportunities are exploited.

To determine the effectiveness of memory sharing for a given workload, try running the workload, and use `resxtop` or `esxtop` to observe the actual savings. Find the information in the `PSHARE` field of the interactive mode in the Memory page.

Memory Compression

ESXi provides a memory compression cache to improve virtual machine performance when you use memory overcommitment. Memory compression is enabled by default. When a host's memory becomes overcommitted, ESXi compresses virtual pages and stores them in memory.

Because accessing compressed memory is faster than accessing memory that is swapped to disk, memory compression in ESXi allows you to overcommit memory without significantly hindering performance. When a virtual page needs to be swapped, ESXi first attempts to compress the page. Pages that can be compressed to 2 KB or smaller are stored in the virtual machine's compression cache, increasing the capacity of the host.

You can set the maximum size for the compression cache and disable memory compression using the Advanced Settings dialog box in the vSphere Client.

Enable or Disable the Memory Compression Cache

Memory compression is enabled by default. You can use the Advanced Settings dialog box in the vSphere Client to enable or disable memory compression for a host.

Procedure

1 In the vSphere Client, select the host in the inventory.

2 Click the **Configuration** tab.

3 Under Software, select **Advanced Settings**.

4 In the left pane, select **Mem** and locate Mem.MemZipEnable.

5 Enter 1 to enable or enter 0 to disable the memory compression cache.

6 Click **OK**.

Set the Maximum Size of the Memory Compression Cache

You can set the maximum size of the memory compression cache for the host's virtual machines.

You set the size of the compression cache as a percentage of the memory size of the virtual machine. For example, if you enter 20 and a virtual machine's memory size is 1000 MB, ESXi can use up to 200MB of host memory to store the compressed pages of the virtual machine.

If you do not set the size of the compression cache, ESXi uses the default value of 10 percent.

Procedure

1 In the vSphere Client, select the host in the inventory.

2 Click the **Configuration** tab.

3 Under Software, select **Advanced Settings**.

4 In the left pane, select **Mem** and locate Mem.MemZipMaxPct.

The value of this attribute determines the maximum size of the compression cache for the virtual machine.

5 Enter the maximum size for the compression cache.

The value is a percentage of the size of the virtual machine and must be between 5 and 100 percent.

6 Click **OK**.

Measuring and Differentiating Types of Memory Usage

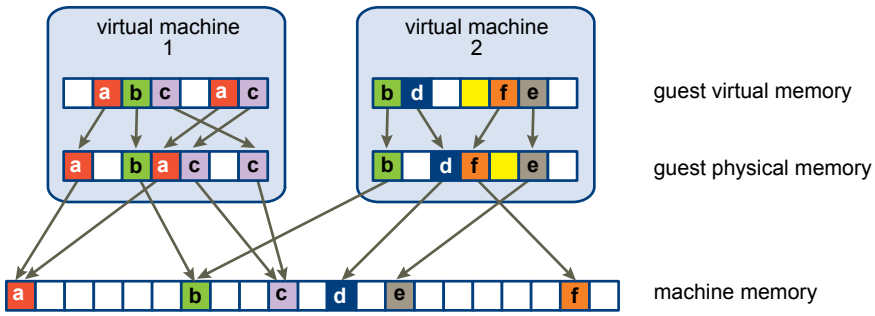
The **Performance** tab of the vSphere Client displays a number of metrics that can be used to analyze memory usage.

Some of these memory metrics measure guest physical memory while other metrics measure machine memory. For instance, two types of memory usage that you can examine using performance metrics are guest physical memory and machine memory. You measure guest physical memory using the Memory Granted metric (for a virtual machine) or Memory Shared (for a host). To measure machine memory, however, use Memory Consumed (for a virtual machine) or Memory Shared Common (for a host). Understanding the conceptual difference between these types of memory usage is important for knowing what these metrics are measuring and how to interpret them.

The VMkernel maps guest physical memory to machine memory, but they are not always mapped one-to-one. Multiple regions of guest physical memory might be mapped to the same region of machine memory (in the case of memory sharing) or specific regions of guest physical memory might not be mapped to machine memory (when the VMkernel swaps out or balloons guest physical memory). In these situations, calculations of guest physical memory usage and machine memory usage for an individual virtual machine or a host differ.

Consider the example in the following figure, which shows two virtual machines running on a host. Each block represents 4 KB of memory and each color/letter represents a different set of data on a block.

Figure 6-2. Memory Usage Example



The performance metrics for the virtual machines can be determined as follows:

- To determine Memory Granted (the amount of guest physical memory that is mapped to machine memory) for virtual machine 1, count the number of blocks in virtual machine 1's guest physical memory that have arrows to machine memory and multiply by 4 KB. Since there are five blocks with arrows, Memory Granted would be 20 KB.
- Memory Consumed is the amount of machine memory allocated to the virtual machine, accounting for savings from shared memory. First, count the number of blocks in machine memory that have arrows from virtual machine 1's guest physical memory. There are three such blocks, but one block is shared with virtual machine 2. So count two full blocks plus half of the third and multiply by 4 KB for a total of 10 KB Memory Consumed.

The important difference between these two metrics is that Memory Granted counts the number of blocks with arrows at the guest physical memory level and Memory Consumed counts the number of blocks with arrows at the machine memory level. The number of blocks differs between the two levels due to memory sharing and so Memory Granted and Memory Consumed differ. This is not problematic and shows that memory is being saved through sharing or other reclamation techniques.

A similar result is obtained when determining Memory Shared and Memory Shared Common for the host.

- Memory Shared for the host is the sum of each virtual machine's Memory Shared. Calculate this by looking at each virtual machine's guest physical memory and counting the number of blocks that have arrows to machine memory blocks that themselves have more than one arrow pointing at them. There are six such blocks in the example, so Memory Shared for the host is 24 KB.
- Memory Shared Common is the amount of machine memory that is shared by virtual machines. To determine this, look at the machine memory and count the number of blocks that have more than one arrow pointing at them. There are three such blocks, so Memory Shared Common is 12 KB.

Memory Shared is concerned with guest physical memory and looks at the origin of the arrows. Memory Shared Common, however, deals with machine memory and looks at the destination of the arrows.

The memory metrics that measure guest physical memory and machine memory might appear contradictory. In fact, they are measuring different aspects of a virtual machine's memory usage. By understanding the differences between these metrics, you can better utilize them to diagnose performance issues.

Memory Reliability

Memory reliability, also known as error insolation, allows ESXi to stop using parts of memory when it determines that a failure might occur, as well as when a failure did occur.

When enough corrected errors are reported at a particular address, ESXi stops using this address to prevent the corrected error from becoming an uncorrected error.

Memory reliability provides a better VMkernel reliability despite corrected and uncorrected errors in RAM. It also enables the system to avoid using memory pages that might contain errors.

Correct an Error Isolation Notification

With memory reliability, VMkernel stops using pages that receive an error isolation notification.

The user receives an event in the vSphere Client when VMkernel recovers from an uncorrectable memory error, when VMkernel retires a significant percentage of system memory due to a large number of correctable errors, or if there is a large number of pages that are unable to retire.

Procedure

1 Vacate the host.

2 Migrate the virtual machines.

3 Run tests.

Managing Storage I/O Resources 7

vSphere Storage I/O Control allows cluster-wide storage I/O prioritization, which allows better workload consolidation and helps reduce extra costs associated with over provisioning.

Storage I/O Control extends the constructs of shares and limits to handle storage I/O resources. You can control the amount of storage I/O that is allocated to virtual machines during periods of I/O congestion, which ensures that more important virtual machines get preference over less important virtual machines for I/O resource allocation.

When you enable Storage I/O Control on a datastore, ESXi begins to monitor the device latency that hosts observe when communicating with that datastore. When device latency exceeds a threshold, the datastore is considered to be congested and each virtual machine that accesses that datastore is allocated I/O resources in proportion to their shares. You set shares per virtual machine. You can adjust the number for each based on need.

Configuring Storage I/O Control is a two-step process:

1 Enable Storage I/O Control for the datastore.

2 Set the number of storage I/O shares and upper limit of I/O operations per second (IOPS) allowed for each virtual machine.

By default, all virtual machine shares are set to Normal (1000) with unlimited IOPS.

NOTE Storage I/O Control is enabled by default on Storage DRS-enabled datastore clusters.

This chapter includes the following topics:

Storage I/O Control Requirements

Storage I/O Control has several requirements and limitations.

- Datastores that are Storage I/O Control-enabled must be managed by a single vCenter Server system.
- Storage I/O Control is supported on Fibre Channel-connected, iSCSI-connected, and NFS-connected storage. Raw Device Mapping (RDM) is not supported.
- Storage I/O Control does not support datastores with multiple extents.

- Before using Storage I/O Control on datastores that are backed by arrays with automated storage tiering capabilities, check the *VMware Storage/SAN Compatibility Guide* to verify whether your automated tiered storage array has been certified to be compatible with Storage I/O Control.

  Automated storage tiering is the ability of an array (or group of arrays) to migrate LUNs/volumes or parts of LUNs/volumes to different types of storage media (SSD, FC, SAS, SATA) based on user-set policies and current I/O patterns. No special certification is required for arrays that do not have these automatic migration/tiering features, including those that provide the ability to manually migrate data between different types of storage media.

Storage I/O Control Resource Shares and Limits

You allocate the number of storage I/O shares and upper limit of I/O operations per second (IOPS) allowed for each virtual machine. When storage I/O congestion is detected for a datastore, the I/O workloads of the virtual machines accessing that datastore are adjusted according to the proportion of virtual machine shares each virtual machine has.

Storage I/O shares are similar to those used for memory and CPU resource allocation, which are described in "Resource Allocation Shares," on page 11. These shares represent the relative importance of a virtual machine with regard to the distribution of storage I/O resources. Under resource contention, virtual machines with higher share values have greater access to the storage array, which typically results in higher throughput and lower latency.

When you allocate storage I/O resources, you can limit the IOPS that are allowed for a virtual machine. By default, these are unlimited. If a virtual machine has more than one virtual disk, you must set the limit on all of its virtual disks. Otherwise, the limit will not be enforced for the virtual machine. In this case, the limit on the virtual machine is the aggregation of the limits for all virtual disks.

The benefits and drawbacks of setting resource limits are described in "Resource Allocation Limit," on page 12. If the limit you want to set for a virtual machine is in terms of MB per second instead of IOPS, you can convert MB per second into IOPS based on the typical I/O size for that virtual machine. For example, to restrict a backup application with 64KB IOs to 10MB per second, set a limit of 160 IOPS.

View Storage I/O Control Shares and Limits

You can view the shares and limits for all virtual machines running on a datastore. Viewing this information allows you to compare the settings of all virtual machines that are accessing the datastore, regardless of the cluster in which they are running.

Procedure

1 Select the datastore in the vSphere Client inventory.

2 Click the **Virtual Machines** tab.

  The tab displays each virtual machine running on the datastore and the associated shares value, IOPS limit, and percentage of datastore shares.

Monitor Storage I/O Control Shares

Use the datastore **Performance** tab to monitor how Storage I/O Control handles the I/O workloads of the virtual machines accessing a datastore based on their shares.

Datastore performance charts allow you to monitor the following information:

- Average latency and aggregated IOPS on the datastore
- Latency among hosts
- Queue depth among hosts
- Read/write IOPS among hosts

- Read/write latency among virtual machine disks
- Read/write IOPS among virtual machine disks

Procedure

1 Select the datastore in the vSphere Client inventory and click the **Performance** tab.

2 From the **View** drop-down menu, select **Performance**.

For more information, see the *vSphere Monitoring and Performance* documentation.

Set Storage I/O Control Resource Shares and Limits

Allocate storage I/O resources to virtual machines based on importance by assigning a relative amount of shares to the virtual machine.

Unless virtual machine workloads are very similar, shares do not necessarily dictate allocation in terms of I/O operations or megabytes per second. Higher shares allow a virtual machine to keep more concurrent I/O operations pending at the storage device or datastore compared to a virtual machine with lower shares. Two virtual machines might experience different throughput based on their workloads.

Procedure

1 Select a virtual machine in the vSphere Client inventory.

2 Click the **Summary** tab and click **Edit Settings**.

3 Click the **Resources** tab and select **Disk**.

4 Select a virtual hard disk from the list.

5 Click the **Shares** column to select the relative amount of shares to allocate to the virtual machine (Low, Normal, or High).

You can select **Custom** to enter a user-defined shares value.

6 Click the **Limit - IOPS** column and enter the upper limit of storage resources to allocate to the virtual machine.

IOPS are the number of I/O operations per second. By default, IOPS are unlimited. You select Low (500), Normal (1000), or High (2000), or you can select Custom to enter a user-defined number of shares.

7 Click **OK**.

Shares and limits are reflected on the **Resource Allocation** tab for the host and cluster.

Enable Storage I/O Control

When you enable Storage I/O Control, ESXi monitors datastore latency and adjusts the I/O load sent to it, if datastore average latency exceeds the threshold.

Procedure

1 In the vSphere Client inventory, select a datastore and click the **Configuration** tab.

2 Click **Properties**.

3 Under Storage I/O Control, select the **Enabled** check box.

4 Click **Close**.

On the Datastores tab, the Storage I/O Control column shows that Storage I/O Control is enabled for the datastore.

Set Storage I/O Control Threshold Value

The congestion threshold value for a datastore is the upper limit of latency that is allowed for a datastore before Storage I/O Control begins to assign importance to the virtual machine workloads according to their shares.

You do not need to adjust the threshold setting in most environments.

CAUTION Storage I/O Control will not function correctly unless all datatores that share the same spindles on the array have the same congestion threshold.

If you change the congestion threshold setting, set the value based on the following considerations.

- A higher value typically results in higher aggregate throughput and weaker isolation. Throttling will not occur unless the overall average latency is higher than the threshold.
- If throughput is more critical than latency, do not set the value too low. For example, for Fibre Channel disks, a value below 20 ms could lower peak disk throughput. A very high value (above 50 ms) might allow very high latency without any significant gain in overall throughput.
- A lower value will result in lower device latency and stronger virtual machine I/O performance isolation. Stronger isolation means that the shares controls are enforced more often. Lower device latency translates into lower I/O latency for the virtual machines with the highest shares, at the cost of higher I/O latency experienced by the virtual machines with fewer shares.
- If latency is more important, a very low value (lower than 20 ms) will result in lower device latency and better isolation among I/Os at the potential cost of a decrease in aggregate datastore throughput.

Prerequisites

Verify that Storage I/O Control is enabled.

Procedure

1 In the vSphere Client inventory, select a datastore and click the **Configuration** tab.

2 Click **Properties**.

3 Under Storage I/O Control, select the **Enabled** check box.

4 (Optional) Click **Advanced** to edit the congestion threshold value for the datastore.

The value must be between 10 ms and 100 ms.

5 (Optional) Click **Reset** to restore the congestion threshold setting to the default value (30 ms).

6 Click **OK** and click **Close**.

Managing Resource Pools

8

A resource pool is a logical abstraction for flexible management of resources. Resource pools can be grouped into hierarchies and used to hierarchically partition available CPU and memory resources.

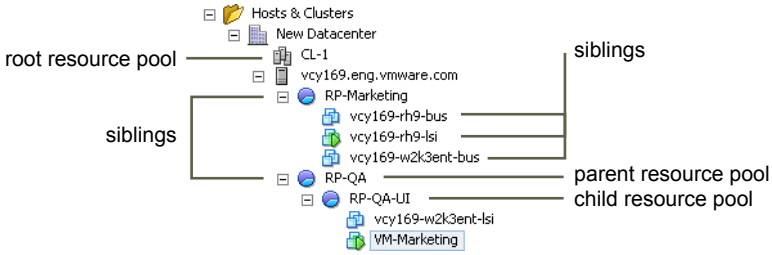
Each standalone host and each DRS cluster has an (invisible) root resource pool that groups the resources of that host or cluster. The root resource pool does not appear because the resources of the host (or cluster) and the root resource pool are always the same.

Users can create child resource pools of the root resource pool or of any user-created child resource pool. Each child resource pool owns some of the parent's resources and can, in turn, have a hierarchy of child resource pools to represent successively smaller units of computational capability.

A resource pool can contain child resource pools, virtual machines, or both. You can create a hierarchy of shared resources. The resource pools at a higher level are called parent resource pools. Resource pools and virtual machines that are at the same level are called siblings. The cluster itself represents the root resource pool. If you do not create child resource pools, only the root resource pools exist.

In the following example, RP-QA is the parent resource pool for RP-QA-UI. RP-Marketing and RP-QA are siblings. The three virtual machines immediately below RP-Marketing are also siblings.

Figure 8-1. Parents, Children, and Siblings in Resource Pool Hierarchy



For each resource pool, you specify reservation, limit, shares, and whether the reservation should be expandable. The resource pool resources are then available to child resource pools and virtual machines.

This chapter includes the following topics:

Why Use Resource Pools?

Resource pools allow you to delegate control over resources of a host (or a cluster), but the benefits are evident when you use resource pools to compartmentalize all resources in a cluster. Create multiple resource pools as direct children of the host or cluster and configure them. You can then delegate control over the resource pools to other individuals or organizations.

Using resource pools can result in the following benefits.

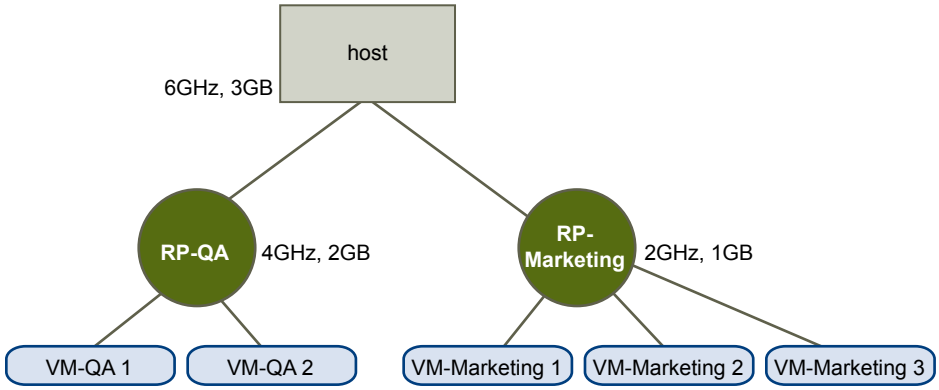
- Flexible hierarchical organization—Add, remove, or reorganize resource pools or change resource allocations as needed.
- Isolation between pools, sharing within pools—Top-level administrators can make a pool of resources available to a department-level administrator. Allocation changes that are internal to one departmental resource pool do not unfairly affect other unrelated resource pools.
- Access control and delegation—When a top-level administrator makes a resource pool available to a department-level administrator, that administrator can then perform all virtual machine creation and management within the boundaries of the resources to which the resource pool is entitled by the current shares, reservation, and limit settings. Delegation is usually done in conjunction with permissions settings.
- Separation of resources from hardware—If you are using clusters enabled for DRS, the resources of all hosts are always assigned to the cluster. That means administrators can perform resource management independently of the actual hosts that contribute to the resources. If you replace three 2GB hosts with two 3GB hosts, you do not need to make changes to your resource allocations.

  This separation allows administrators to think more about aggregate computing capacity and less about individual hosts.
- Management of sets of virtual machines running a multitier service— Group virtual machines for a multitier service in a resource pool. You do not need to set resources on each virtual machine. Instead, you can control the aggregate allocation of resources to the set of virtual machines by changing settings on their enclosing resource pool.

For example, assume a host has a number of virtual machines. The marketing department uses three of the virtual machines and the QA department uses two virtual machines. Because the QA department needs larger amounts of CPU and memory, the administrator creates one resource pool for each group. The administrator sets **CPU Shares** to **High** for the QA department pool and to **Normal** for the Marketing department pool so that the QA department users can run automated tests. The second resource pool with fewer CPU and memory resources is sufficient for the lighter load of the marketing staff. Whenever the QA department is not fully using its allocation, the marketing department can use the available resources.

The numbers in the following figure show the effective allocations to the resource pools.

Figure 8-2. Allocating Resources to Resource Pools



Create a Resource Pool

You can create a child resource pool of any ESXi host, resource pool, or DRS cluster.

NOTE If a host has been added to a cluster, you cannot create child resource pools of that host. If the cluster is enabled for DRS, you can create child resource pools of the cluster.

When you create a child resource pool, you are prompted for resource pool attribute information. The system uses admission control to make sure you cannot allocate resources that are not available.

Prerequisites

The vSphere Client is connected to the vCenter Server system. If you connect the vSphere Client directly to a host, you cannot create a resource pool.

Procedure

1 In the vSphere Client inventory, select a parent object for the resource pool (a host, another resource pool, or a DRS cluster).

2 Select **File > New > Resource Pool**.

3 Type a name to identify the resource pool.

4 Specify how to allocate CPU and memory resources.

The CPU resources for your resource pool are the guaranteed physical resources the host reserves for a resource pool. Normally, you accept the default and let the host handle resource allocation.

Option	Description
Shares	Specify shares for this resource pool with respect to the parent's total resources. Sibling resource pools share resources according to their relative share values bounded by the reservation and limit. ■ Select **Low**, **Normal**, or **High** to specify share values respectively in a 1:2:4 ratio. ■ Select **Custom** to give each virtual machine a specific number of shares, which expresses a proportional weight.
Reservation	Specify a guaranteed CPU or memory allocation for this resource pool. Defaults to 0. A nonzero reservation is subtracted from the unreserved resources of the parent (host or resource pool). The resources are considered reserved, regardless of whether virtual machines are associated with the resource pool.
Expandable Reservation	When the check box is selected (default), expandable reservations are considered during admission control. If you power on a virtual machine in this resource pool, and the combined reservations of the virtual machines are larger than the reservation of the resource pool, the resource pool can use resources from its parent or ancestors.
Limit	Specify the upper limit for this resource pool's CPU or memory allocation. You can usually accept the default (**Unlimited**). To specify a limit, deselect the **Unlimited** check box.

5 Click **OK**.

After you create a resource pool, you can add virtual machines to it. A virtual machine's shares are relative to other virtual machines (or resource pools) with the same parent resource pool.

Example: Creating Resource Pools

Assume that you have a host that provides 6GHz of CPU and 3GB of memory that must be shared between your marketing and QA departments. You also want to share the resources unevenly, giving one department (QA) a higher priority. This can be accomplished by creating a resource pool for each department and using the **Shares** attribute to prioritize the allocation of resources.

The example shows how to create a resource pool with the ESXi host as the parent resource.

1 In the Create Resource Pool dialog box, type a name for the QA department's resource pool (for example, RP-QA).

2 Specify **Shares** of **High** for the CPU and memory resources of RP-QA.

3 Create a second resource pool, RP-Marketing.

Leave Shares at **Normal** for CPU and memory.

4 Click **OK**.

If there is resource contention, RP-QA receives 4GHz and 2GB of memory, and RP-Marketing 2GHz and 1GB. Otherwise, they can receive more than this allotment. Those resources are then available to the virtual machines in the respective resource pools.

Edit a Resource Pool

After you create the resource pool, you can edit its CPU and memory resource settings.

Procedure

1 In the vSphere Client, right-click the resource pool in the inventory and select **Edit Settings**.

2 In the Edit Settings dialog box, you can change all attributes of the selected resource pool as described in "Create a Resource Pool," on page 45.

3 Click **OK** to save your changes.

Add a Virtual Machine to a Resource Pool

When you create a virtual machine, the New Virtual Machine wizard allows you to specify a resource pool location as part of the creation process. You can also add an existing virtual machine to a resource pool.

When you move a virtual machine to a new resource pool:

- The virtual machine's reservation and limit do not change.
- If the virtual machine's shares are high, medium, or low, %Shares adjusts to reflect the total number of shares in use in the new resource pool.
- If the virtual machine has custom shares assigned, the share value is maintained.

NOTE Because share allocations are relative to a resource pool, you might have to manually change a virtual machine's shares when you move it into a resource pool so that the virtual machine's shares are consistent with the relative values in the new resource pool. A warning appears if a virtual machine would receive a very large (or very small) percentage of total shares.

- The information displayed in the Resource Allocation tab about the resource pool's reserved and unreserved CPU and memory resources changes to reflect the reservations associated with the virtual machine (if any).

NOTE If a virtual machine has been powered off or suspended, it can be moved but overall available resources (such as reserved and unreserved CPU and memory) for the resource pool are not affected.

Procedure

1 In the vSphere Client, select the virtual machine in the inventory.

The virtual machine can be associated with a standalone host, a cluster, or a different resource pool.

2 Drag the virtual machine (or machines) to the resource pool.

If a virtual machine is powered on, and the destination resource pool does not have enough CPU or memory to guarantee the virtual machine's reservation, the move fails because admission control does not allow it. An error dialog box displays available and requested resources, so you can consider whether an adjustment might resolve the issue.

Remove a Virtual Machine from a Resource Pool

You can remove a virtual machine from a resource pool either by moving the virtual machine to another resource pool or deleting it.

When you remove a virtual machine from a resource pool, the total number of shares associated with the resource pool decreases, so that each remaining share represents more resources. For example, assume you have a pool that is entitled to 6GHz, containing three virtual machines with shares set to **Normal**. Assuming the virtual machines are CPU-bound, each gets an equal allocation of 2GHz. If one of the virtual machines is moved to a different resource pool, the two remaining virtual machines each receive an equal allocation of 3GHz.

Procedure

1 In the vSphere Client, right-click the cluster in the inventory and select **Edit Settings**.

2 Choose one of the following methods to remove the virtual machine from a resource pool.

- Drag the virtual machine to another resource pool.

  You do not need to power off the virtual machine before you move it.

- Right-click the virtual machine and select **Remove from Inventory** or **Delete from Disk**.

  You must power off the virtual machine before you can completely remove it.

Remove a Resource Pool

You can remove a resource pool from the inventory.

Procedure

1 In the vSphere Client, right-click the resource pool and select **Remove**.

A confirmation dialog box appears.

2 Click **Yes** to remove the resource pool.

Resource Pool Admission Control

When you power on a virtual machine in a resource pool, or try to create a child resource pool, the system performs additional admission control to ensure the resource pool's restrictions are not violated.

Before you power on a virtual machine or create a resource pool, ensure that sufficient resources are available using the Resource Allocation tab in the vSphere Client. The **Available Reservation** value for CPU and memory displays resources that are unreserved.

How available CPU and memory resources are computed and whether actions are performed depends on the **Reservation Type**.

Table 8-1. Reservation Types

Reservation Type	Description
Fixed	The system checks whether the selected resource pool has sufficient unreserved resources. If it does, the action can be performed. If it does not, a message appears and the action cannot be performed.
Expandable (default)	The system considers the resources available in the selected resource pool and its direct parent resource pool. If the parent resource pool also has the **Expandable Reservation** option selected, it can borrow resources from its parent resource pool. Borrowing resources occurs recursively from the ancestors of the current resource pool as long as the **Expandable Reservation** option is selected. Leaving this option selected offers more flexibility, but, at the same time provides less protection. A child resource pool owner might reserve more resources than you anticipate.

The system does not allow you to violate preconfigured **Reservation** or **Limit** settings. Each time you reconfigure a resource pool or power on a virtual machine, the system validates all parameters so all service-level guarantees can still be met.

Expandable Reservations Example 1

This example shows you how a resource pool with expandable reservations works.

Assume an administrator manages pool P, and defines two child resource pools, S1 and S2, for two different users (or groups).

The administrator knows that users want to power on virtual machines with reservations, but does not know how much each user will need to reserve. Making the reservations for S1 and S2 expandable allows the administrator to more flexibly share and inherit the common reservation for pool P.

Without expandable reservations, the administrator needs to explicitly allocate S1 and S2 a specific amount. Such specific allocations can be inflexible, especially in deep resource pool hierarchies and can complicate setting reservations in the resource pool hierarchy.

Expandable reservations cause a loss of strict isolation. S1 can start using all of P's reservation, so that no memory or CPU is directly available to S2.

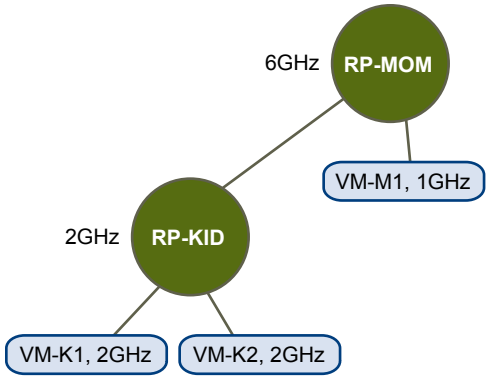
Expandable Reservations Example 2

This example shows how a resource pool with expandable reservations works.

Assume the following scenario, as shown in the figure.

- Parent pool RP-MOM has a reservation of 6GHz and one running virtual machine VM-M1 that reserves 1GHz.
- You create a child resource pool RP-KID with a reservation of 2GHz and with **Expandable Reservation** selected.
- You add two virtual machines, VM-K1 and VM-K2, with reservations of 2GHz each to the child resource pool and try to power them on.
- VM-K1 can reserve the resources directly from RP-KID (which has 2GHz).
- No local resources are available for VM-K2, so it borrows resources from the parent resource pool, RP-MOM. RP-MOM has 6GHz minus 1GHz (reserved by the virtual machine) minus 2GHz (reserved by RP-KID), which leaves 3GHz unreserved. With 3GHz available, you can power on the 2GHz virtual machine.

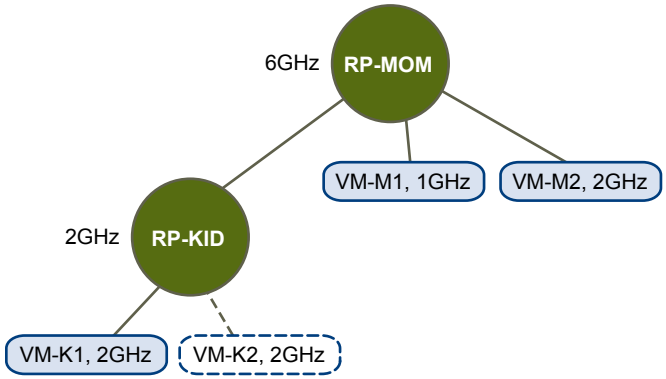
Figure 8-3. Admission Control with Expandable Resource Pools: Successful Power-On



Now, consider another scenario with VM-M1 and VM-M2.

- Power on two virtual machines in RP-MOM with a total reservation of 3GHz.
- You can still power on VM-K1 in RP-KID because 2GHz are available locally.
- When you try to power on VM-K2, RP-KID has no unreserved CPU capacity so it checks its parent. RP-MOM has only 1GHz of unreserved capacity available (5GHz of RP-MOM are already in use—3GHz reserved by the local virtual machines and 2GHz reserved by RP-KID). As a result, you cannot power on VM-K2, which requires a 2GHz reservation.

Figure 8-4. Admission Control with Expandable Resource Pools: Power-On Prevented



Creating a DRS Cluster

9

A DRS cluster is a collection of ESXi hosts and associated virtual machines with shared resources and a shared management interface. Before you can obtain the benefits of cluster-level resource management you must create a DRS cluster.

When you add a host to a DRS cluster, the host's resources become part of the cluster's resources. In addition to this aggregation of resources, with a DRS cluster you can support cluster-wide resource pools and enforce cluster-level resource allocation policies. The following cluster-level resource management capabilities are also available.

Load Balancing The distribution and usage of CPU and memory resources for all hosts and virtual machines in the cluster are continuously monitored. DRS compares these metrics to an ideal resource utilization given the attributes of the cluster's resource pools and virtual machines, the current demand, and the imbalance target. It then performs (or recommends) virtual machine migrations accordingly. See "Virtual Machine Migration," on page 53. When you first power on a virtual machine in the cluster, DRS attempts to maintain proper load balancing by either placing the virtual machine on an appropriate host or making a recommendation. See "Admission Control and Initial Placement," on page 52.

Power management When the vSphere Distributed Power Management (DPM) feature is enabled, DRS compares cluster- and host-level capacity to the demands of the cluster's virtual machines, including recent historical demand. It places (or recommends placing) hosts in standby power mode if sufficient excess capacity is found or powering on hosts if capacity is needed. Depending on the resulting host power state recommendations, virtual machines might need to be migrated to and from the hosts as well. See "Managing Power Resources," on page 67.

Affinity Rules You can control the placement of virtual machines on hosts within a cluster, by assigning affinity rules. See "Using DRS Affinity Rules," on page 71.

Depending on whether or not Enhanced vMotion Compatibility (EVC) is enabled, DRS behaves differently when you use vSphere Fault Tolerance (vSphere FT) virtual machines in your cluster.

Table 9-1. DRS Behavior with vSphere FT Virtual Machines and EVC

EVC	DRS (Load Balancing)	DRS (Initial Placement)
Enabled	Enabled (Primary and Secondary VMs)	Enabled (Primary and Secondary VMs)
Disabled	Disabled (Primary and Secondary VMs)	Disabled (Primary VMs) Fully Automated (Secondary VMs)

This chapter includes the following topics:

Admission Control and Initial Placement

When you attempt to power on a single virtual machine or a group of virtual machines in a DRS-enabled cluster, vCenter Server performs admission control. It checks that there are enough resources in the cluster to support the virtual machine(s).

If the cluster does not have sufficient resources to power on a single virtual machine, or any of the virtual machines in a group power-on attempt, a message appears. Otherwise, for each virtual machine, DRS generates a recommendation of a host on which to run the virtual machine and takes one of the following actions

- Automatically executes the placement recommendation.
- Displays the placement recommendation, which the user can then choose to accept or override.

NOTE No initial placement recommendations are given for virtual machines on standalone hosts or in non-DRS clusters. When powered on, they are placed on the host where they currently reside.

Single Virtual Machine Power On

In a DRS cluster, you can power on a single virtual machine and receive initial placement recommendations.

When you power on a single virtual machine, you have two types of initial placement recommendations:

- A single virtual machine is being powered on and no prerequisite steps are needed.

  The user is presented with a list of mutually exclusive initial placement recommendations for the virtual machine. You can select only one.

- A single virtual machine is being powered on, but prerequisite actions are required.

  These actions include powering on a host in standby mode or the migration of other virtual machines from one host to another. In this case, the recommendations provided have multiple lines, showing each of the prerequisite actions. The user can either accept this entire recommendation or cancel powering on the virtual machine.

Group Power On

You can attempt to power on multiple virtual machines at the same time (group power on).

Virtual machines selected for a group power-on attempt do not have to be in the same DRS cluster. They can be selected across clusters but must be within the same datacenter. It is also possible to include virtual machines located in non-DRS clusters or on standalone hosts. These are powered on automatically and not included in any initial placement recommendation.

The initial placement recommendations for group power-on attempts are provided on a per-cluster basis. If all of the placement-related actions for a group power-on attempt are in automatic mode, the virtual machines are powered on with no initial placement recommendation given. If placement-related actions for any of the virtual machines are in manual mode, the powering on of all of the virtual machines (including those that are in automatic mode) is manual and is included in an initial placement recommendation.

For each DRS cluster that the virtual machines being powered on belong to, there is a single recommendation, which contains all of the prerequisites (or no recommendation). All such cluster-specific recommendations are presented together under the Power On Recommendations tab.

When a nonautomatic group power-on attempt is made, and virtual machines not subject to an initial placement recommendation (that is, those on standalone hosts or in non-DRS clusters) are included, vCenter Server attempts to power them on automatically. If these power ons are successful, they are listed under the Started Power-Ons tab. Any virtual machines that fail to power on are listed under the Failed Power-Ons tab.

Example: Group Power On

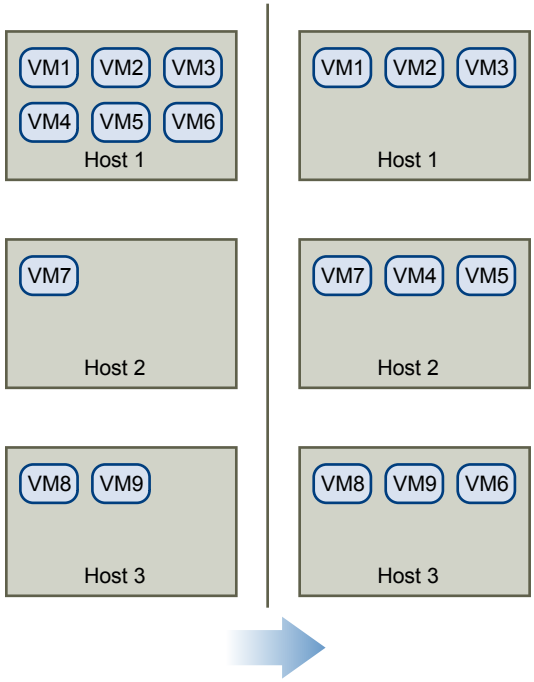
The user selects three virtual machines in the same datacenter for a group power-on attempt. The first two virtual machines (VM1 and VM2) are in the same DRS cluster (Cluster1), while the third virtual machine (VM3) is on a standalone host. VM1 is in automatic mode and VM2 is in manual mode. For this scenario, the user is presented with an initial placement recommendation for Cluster1 (under the Power On Recommendations tab) which consists of actions for powering on VM1 and VM2. An attempt is made to power on VM3 automatically and, if successful, it is listed under the Started Power-Ons tab. If this attempt fails, it is listed under the Failed Power-Ons tab.

Virtual Machine Migration

Although DRS performs initial placements so that load is balanced across the cluster, changes in virtual machine load and resource availability can cause the cluster to become unbalanced. To correct such imbalances, DRS generates migration recommendations.

If DRS is enabled on the cluster, load can be distributed more uniformly to reduce the degree of this imbalance. For example, the three hosts on the left side of the following figure are unbalanced. Assume that Host 1, Host 2, and Host 3 have identical capacity, and all virtual machines have the same configuration and load (which includes reservation, if set). However, because Host 1 has six virtual machines, its resources might be overused while ample resources are available on Host 2 and Host 3. DRS migrates (or recommends the migration of) virtual machines from Host 1 to Host 2 and Host 3. On the right side of the diagram, the properly load balanced configuration of the hosts that results appears.

Figure 9-1. Load Balancing



When a cluster becomes unbalanced, DRS makes recommendations or migrates virtual machines, depending on the default automation level:

- If the cluster or any of the virtual machines involved are manual or partially automated, vCenter Server does not take automatic actions to balance resources. Instead, the Summary page indicates that migration recommendations are available and the DRS Recommendations page displays recommendations for changes that make the most efficient use of resources across the cluster.
- If the cluster and virtual machines involved are all fully automated, vCenter Server migrates running virtual machines between hosts as needed to ensure efficient use of cluster resources.

NOTE Even in an automatic migration setup, users can explicitly migrate individual virtual machines, but vCenter Server might move those virtual machines to other hosts to optimize cluster resources.

By default, automation level is specified for the whole cluster. You can also specify a custom automation level for individual virtual machines.

DRS Migration Threshold

The DRS migration threshold allows you to specify which recommendations are generated and then applied (when the virtual machines involved in the recommendation are in fully automated mode) or shown (if in manual mode). This threshold is also a measure of how much cluster imbalance across host (CPU and memory) loads is acceptable.

You can move the threshold slider to use one of five settings, ranging from Conservative to Aggressive. The five migration settings generate recommendations based on their assigned priority level. Each setting you move the slider to the right allows the inclusion of one more lower level of priority. The Conservative setting generates only priority-one recommendations (mandatory recommendations), the next level to the right generates priority-two recommendations and higher, and so on, down to the Aggressive level which generates priority-five recommendations and higher (that is, all recommendations.)

A priority level for each migration recommendation is computed using the load imbalance metric of the cluster. This metric is displayed as Current host load standard deviation in the cluster's Summary tab in the vSphere Client. A higher load imbalance leads to higher-priority migration recommendations. For more information about this metric and how a recommendation priority level is calculated, see the VMware Knowledge Base article "Calculating the priority level of a VMware DRS migration recommendation."

After a recommendation receives a priority level, this level is compared to the migration threshold you set. If the priority level is less than or equal to the threshold setting, the recommendation is either applied (if the relevant virtual machines are in fully automated mode) or displayed to the user for confirmation (if in manual or partially automated mode.)

Migration Recommendations

If you create a cluster with a default manual or partially automated mode, vCenter Server displays migration recommendations on the DRS Recommendations page.

The system supplies as many recommendations as necessary to enforce rules and balance the resources of the cluster. Each recommendation includes the virtual machine to be moved, current (source) host and destination host, and a reason for the recommendation. The reason can be one of the following:

- Balance average CPU loads or reservations.
- Balance average memory loads or reservations.
- Satisfy resource pool reservations.
- Satisfy an affinity rule.

- Host is entering maintenance mode or standby mode.

NOTE If you are using the vSphere Distributed Power Management (DPM) feature, in addition to migration recommendations, DRS provides host power state recommendations.

DRS Cluster Requirements

Hosts that are added to a DRS cluster must meet certain requirements to use cluster features successfully.

Shared Storage Requirements

A DRS cluster has certain shared storage requirements.

Ensure that the managed hosts use shared storage. Shared storage is typically on a SAN, but can also be implemented using NAS shared storage.

See the *vSphere Storage* documentation for information about other shared storage.

Shared VMFS Volume Requirements

A DRS cluster has certain shared VMFS volume requirements.

Configure all managed hosts to use shared VMFS volumes.

- Place the disks of all virtual machines on VMFS volumes that are accessible by source and destination hosts.
- Ensure the VMFS volume is sufficiently large to store all virtual disks for your virtual machines.
- Ensure all VMFS volumes on source and destination hosts use volume names, and all virtual machines use those volume names for specifying the virtual disks.

NOTE Virtual machine swap files also need to be on a VMFS accessible to source and destination hosts (just like `.vmdk` virtual disk files). This requirement does not apply if all source and destination hosts are ESX Server 3.5 or higher and using host-local swap. In that case, vMotion with swap files on unshared storage is supported. Swap files are placed on a VMFS by default, but administrators might override the file location using advanced virtual machine configuration options.

Processor Compatibility Requirements

A DRS cluster has certain processor compatibility requirements.

To avoid limiting the capabilities of DRS, you should maximize the processor compatibility of source and destination hosts in the cluster.

vMotion transfers the running architectural state of a virtual machine between underlying ESXi hosts. vMotion compatibility means that the processors of the destination host must be able to resume execution using the equivalent instructions where the processors of the source host were suspended. Processor clock speeds and cache sizes might vary, but processors must come from the same vendor class (Intel versus AMD) and the same processor family to be compatible for migration with vMotion.

Processor families are defined by the processor vendors. You can distinguish different processor versions within the same family by comparing the processors' model, stepping level, and extended features.

Sometimes, processor vendors have introduced significant architectural changes within the same processor family (such as 64-bit extensions and SSE3). VMware identifies these exceptions if it cannot guarantee successful migration with vMotion.

vCenter Server provides features that help ensure that virtual machines migrated with vMotion meet processor compatibility requirements. These features include:

- Enhanced vMotion Compatibility (EVC) – You can use EVC to help ensure vMotion compatibility for the hosts in a cluster. EVC ensures that all hosts in a cluster present the same CPU feature set to virtual machines, even if the actual CPUs on the hosts differ. This prevents migrations with vMotion from failing due to incompatible CPUs.

  Configure EVC from the Cluster Settings dialog box. The hosts in a cluster must meet certain requirements for the cluster to use EVC. For information about EVC and EVC requirements, see the *vCenter Server and Host Management* documentation.

- CPU compatibility masks – vCenter Server compares the CPU features available to a virtual machine with the CPU features of the destination host to determine whether to allow or disallow migrations with vMotion. By applying CPU compatibility masks to individual virtual machines, you can hide certain CPU features from the virtual machine and potentially prevent migrations with vMotion from failing due to incompatible CPUs.

vMotion Requirements for DRS Clusters

A DRS cluster has certain vMotion requirements.

To enable the use of DRS migration recommendations, the hosts in your cluster must be part of a vMotion network. If the hosts are not in the vMotion network, DRS can still make initial placement recommendations.

To be configured for vMotion, each host in the cluster must meet the following requirements:

- vMotion does not support raw disks or migration of applications clustered using Microsoft Cluster Service (MSCS).
- vMotion requires a private Gigabit Ethernet migration network between all of the vMotion enabled managed hosts. When vMotion is enabled on a managed host, configure a unique network identity object for the managed host and connect it to the private migration network.

Create a DRS Cluster

Create a DRS cluster using the New Cluster wizard in the vSphere Client.

Prerequisites

You can create a cluster without a special license, but you must have a license to enable a cluster for vSphere DRS (or vSphere HA).

Procedure

1 Right-click a datacenter or folder in the vSphere Client and select **New Cluster**.

2 Name the cluster in the **Name** text box.

  This name appears in the vSphere Client inventory panel.

3 Enable the DRS feature by clicking the **vSphere DRS** box.

  You can also enable the vSphere HA feature by clicking **vSphere HA**.

4 Click **Next**.

5 Select a default automation level for DRS.

Automation Level	Action
Manual	■ Initial placement: Recommended host(s) is displayed. ■ Migration: Recommendation is displayed.
Partially Automated	■ Initial placement: Automatic. ■ Migration: Recommendation is displayed.
Fully Automated	■ Initial placement: Automatic. ■ Migration: Recommendation is executed automatically.

6 Set the migration threshold for DRS.

7 Click **Next**.

8 Specify the default power management setting for the cluster.

If you enable power management, select a vSphere DPM threshold setting.

9 Click **Next**.

10 If appropriate, enable Enhanced vMotion Compatibility (EVC) and select the mode it should operate in.

11 Click **Next**.

12 Select a location for the swapfiles of your virtual machines.

You can either store a swapfile in the same directory as the virtual machine itself, or a datastore specified by the host (host-local swap)

13 Click **Next**.

14 Review the summary page that lists the options you selected.

15 Click **Finish** to complete cluster creation, or click **Back** to go back and make modifications to the cluster setup.

A new cluster does not include any hosts or virtual machines.

To add hosts and virtual machines to the cluster see "Adding Hosts to a Cluster," on page 59 and "Removing Virtual Machines from a Cluster," on page 61.

Set a Custom Automation Level for a Virtual Machine

After you create a DRS cluster, you can customize the automation level for individual virtual machines to override the cluster's default automation level.

For example, you can select **Manual** for specific virtual machines in a cluster with full automation, or **Partially Automated** for specific virtual machines in a manual cluster.

If a virtual machine is set to **Disabled**, vCenter Server does not migrate that virtual machine or provide migration recommendations for it. This is known as pinning the virtual machine to its registered host.

NOTE If you have not enabled Enhanced vMotion Compatibility (EVC) for the cluster, fault tolerant virtual machines are set to DRS disabled. They appear on this screen, but you cannot assign an automation mode to them.

Procedure

1 In the vSphere Client, right-click the cluster in the inventory and select **Edit Settings**.

2 In the left pane under vSphere DRS, select **Virtual Machine Options**.

3 Select the **Enable individual virtual machine automation levels** check box.

4 (Optional) To temporarily disable any individual virtual machine overrides, deselect the **Enable individual virtual machine automation levels** check box.

   Virtual machine settings are restored when the check box is selected again.

5 (Optional) To temporarily suspend all vMotion activity in a cluster, put the cluster in manual mode and deselect the **Enable individual virtual machine automation levels** check box.

6 Select one or more virtual machines.

7 Click the **Automation Level** column and select an automation level from the drop-down menu.

Option	Description
Manual	Placement and migration recommendations are displayed, but do not run until you manually apply the recommendation.
Fully Automated	Placement and migration recommendations run automatically.
Partially Automated	Initial placement is performed automatically. Migration recommendations are displayed, but do not run.
Disabled	vCenter Server does not migrate the virtual machine or provide migration recommendations for it.

8 Click **OK**.

NOTE Other VMware products or features, such as vSphere vApp and vSphere Fault Tolerance, might override the automation levels of virtual machines in a DRS cluster. Refer to the product-specific documentation for details.

Disable DRS

You can turn off DRS for a cluster.

When DRS is disabled, the cluster's resource pool hierarchy and affinity rules are not reestablished when DRS is turned back on. So if you disable DRS, the resource pools are removed from the cluster. To avoid losing the resource pools, instead of disabling DRS, you should suspend it by changing the DRS automation level to manual (and disabling any virtual machine overrides). This prevents automatic DRS actions, but preserves the resource pool hierarchy.

Procedure

1 Select the cluster in the vSphere Client inventory.

2 Right click and select **Edit Settings**.

3 In the left panel, select **General**, and deselect the **Turn On vSphere DRS** check box.

4 Click **OK** to turn off DRS.

Using DRS Clusters to Manage Resources

10

After you create a DRS cluster, you can customize it and use it to manage resources.

To customize your DRS cluster and the resources it contains you can configure affinity rules and you can add and remove hosts and virtual machines. When a cluster's settings and resources have been defined, you should ensure that it is and remains a valid cluster. You can also use a valid DRS cluster to manage power resources and interoperate with vSphere HA.

This chapter includes the following topics:

Adding Hosts to a Cluster

The procedure for adding hosts to a cluster is different for hosts managed by the same vCenter Server (managed hosts) than for hosts not managed by that server.

After a host has been added, the virtual machines deployed to the host become part of the cluster and DRS can recommend migration of some virtual machines to other hosts in the cluster.

Add a Managed Host to a Cluster

When you add a standalone host already being managed by vCenter Server to a DRS cluster, the host's resources become associated with the cluster.

You can decide whether you want to associate existing virtual machines and resource pools with the cluster's root resource pool or graft the resource pool hierarchy.

NOTE If a host has no child resource pools or virtual machines, the host's resources are added to the cluster but no resource pool hierarchy with a top-level resource pool is created.

Procedure

1 Select the host from either the inventory or list view.

2 Drag the host to the target cluster object.

3 Select what to do with the host's virtual machines and resource pools.

- **Put this host's virtual machines in the cluster's root resource pool**

  vCenter Server removes all existing resource pools of the host and the virtual machines in the host's hierarchy are all attached to the root. Because share allocations are relative to a resource pool, you might have to manually change a virtual machine's shares after selecting this option, which destroys the resource pool hierarchy.

- **Create a resource pool for this host's virtual machines and resource pools**

  vCenter Server creates a top-level resource pool that becomes a direct child of the cluster and adds all children of the host to that new resource pool. You can supply a name for that new top-level resource pool. The default is **Grafted from <host_name>**.

The host is added to the cluster.

Add an Unmanaged Host to a Cluster

You can add an unmanaged host to a cluster. Such a host is not currently managed by the same vCenter Server system as the cluster and it is not visible in the vSphere Client.

Procedure

1 Select the cluster to which to add the host and select **Add Host** from the right-click menu.

2 Enter the host name, user name, and password, and click **Next**.

3 View the summary information and click **Next**.

4 Select what to do with the host's virtual machines and resource pools.

- **Put this host's virtual machines in the cluster's root resource pool**

  vCenter Server removes all existing resource pools of the host and the virtual machines in the host's hierarchy are all attached to the root. Because share allocations are relative to a resource pool, you might have to manually change a virtual machine's shares after selecting this option, which destroys the resource pool hierarchy.

- **Create a resource pool for this host's virtual machines and resource pools**

  vCenter Server creates a top-level resource pool that becomes a direct child of the cluster and adds all children of the host to that new resource pool. You can supply a name for that new top-level resource pool. The default is **Grafted from <host_name>**.

The host is added to the cluster.

Adding Virtual Machines to a Cluster

You can add a virtual machine to a cluster in three ways.

- When you add a host to a cluster, all virtual machines on that host are added to the cluster.
- When a virtual machine is created, the New Virtual Machine wizard prompts you for the location to place the virtual machine. You can select a standalone host or a cluster and you can select any resource pool inside the host or cluster.
- You can migrate a virtual machine from a standalone host to a cluster or from a cluster to another cluster using the Migrate Virtual Machine wizard. To start this wizard either drag the virtual machine object on top of the cluster object or right-click the virtual machine name and select **Migrate**.

  NOTE You can drag a virtual machine directly to a resource pool within a cluster. In this case, the Migrate Virtual Machine wizard is started but the resource pool selection page does not appear. Migrating directly to a host within a cluster is not allowed because the resource pool controls the resources.

Removing Virtual Machines from a Cluster

You can remove virtual machines from a cluster.

You can remove a virtual machine from a cluster in two ways:

- When you remove a host from a cluster, all of the powered-off virtual machines that you do not migrate to other hosts are removed as well. You can remove a host only if it is in maintenance mode or disconnected. If you remove a host from a DRS cluster, the cluster can become yellow because it is overcommitted.
- You can migrate a virtual machine from a cluster to a standalone host or from a cluster to another cluster using the Migrate Virtual Machine wizard. To start this wizard either drag the virtual machine object on top of the cluster object or right-click the virtual machine name and select **Migrate**.

  If the virtual machine is a member of a DRS cluster rules group, vCenter Server displays a warning before it allows the migration to proceed. The warning indicates that dependent virtual machines are not migrated automatically. You have to acknowledge the warning before migration can proceed.

Removing a Host from a Cluster

When you remove a host from a DRS cluster, you affect resource pool hierarchies, virtual machines, and you might create invalid clusters. Consider the affected objects before you remove the host.

- Resource Pool Hierarchies – When you remove a host from a cluster, the host retains only the root resource pool, even if you used a DRS cluster and decided to graft the host resource pool when you added the host to the cluster. In that case, the hierarchy remains with the cluster. You can create a host-specific resource pool hierarchy.

  NOTE Ensure that you remove the host from the cluster by first placing it in maintenance mode. If you instead disconnect the host before removing it from the cluster, the host retains the resource pool that reflects the cluster hierarchy.

- Virtual Machines – A host must be in maintenance mode before you can remove it from the cluster and for a host to enter maintenance mode all powered-on virtual machines must be migrated off that host. When you request that a host enter maintenance mode, you are also asked whether you want to migrate all the powered-off virtual machines on that host to other hosts in the cluster.
- Invalid Clusters – When you remove a host from a cluster, the resources available for the cluster decrease. If the cluster has enough resources to satisfy the reservations of all virtual machines and resource pools in the cluster, the cluster adjusts resource allocation to reflect the reduced amount of resources. If the cluster does not have enough resources to satisfy the reservations of all resource pools, but there are enough resources to satisfy the reservations for all virtual machines, an alarm is issued and the cluster is marked yellow. DRS continues to run.

Place a Host in Maintenance Mode

You place a host in maintenance mode when you need to service it, for example, to install more memory. A host enters or leaves maintenance mode only as the result of a user request.

Virtual machines that are running on a host entering maintenance mode need to be migrated to another host (either manually or automatically by DRS) or shut down. The host is in a state of **Entering Maintenance Mode** until all running virtual machines are powered down or migrated to different hosts. You cannot power on virtual machines or migrate virtual machines to a host entering maintenance mode.

When no more running virtual machines are on the host, the host's icon changes to include **under maintenance** and the host's Summary panel indicates the new state. While in maintenance mode, the host does not allow you to deploy or power on a virtual machine.

NOTE DRS does not recommend (or perform, in fully automated mode) any virtual machine migrations off of a host entering maintenance or standby mode if the vSphere HA failover level would be violated after the host enters the requested mode.

Procedure

1 In the vSphere Client inventory, right-click a host and select **Enter Maintenance Mode**.

- If the host is part of a partially automated or manual DRS cluster, a list of migration recommendations for virtual machines running on the host appears.
- If the host is part of an automated DRS cluster, virtual machines are migrated to different hosts when the host enters maintenance mode.

2 If applicable, click **Apply Recommendations**.

The host is in maintenance mode until you select **Exit Maintenance Mode**.

Remove a Host from a Cluster

You can remove hosts from a cluster.

Procedure

1 In the vSphere Client, right-click the host in the inventory and select **Enter Maintenance Mode**.

2 When the host is in maintenance mode, drag it to a different inventory location, either the top-level datacenter or to a different cluster.

After you remove a host from a cluster, you can perform the following tasks.

- Remove the host from vCenter Server: Right-click the host and select **Remove**.
- Run the host as a standalone host under vCenter Server: Right-click the host and select **Exit Maintenance Mode**.
- Drag the host into another cluster.

When you move the host, its resources are removed from the cluster. If you grafted the host's resource pool hierarchy onto the cluster, that hierarchy remains with the cluster.

Using Standby Mode

When a host machine is placed in standby mode, it is powered off.

Normally, hosts are placed in standby mode by the vSphere DPM feature to optimize power usage. You can also place a host in standby mode manually. However, DRS might undo (or recommend undoing) your change the next time it runs. To force a host to remain off, place it in maintenance mode and power it off.

DRS Cluster Validity

The vSphere Client indicates whether a DRS cluster is valid, overcommitted (yellow), or invalid (red).

DRS clusters become overcommitted or invalid for several reasons.

- A cluster might become overcommitted if a host fails.
- A cluster becomes invalid if vCenter Server is unavailable and you power on virtual machines using a vSphere Client connected directly to a host.

- A cluster becomes invalid if the user reduces the reservation on a parent resource pool while a virtual machine is in the process of failing over.
- If changes are made to hosts or virtual machines using a vSphere Client connected to a host while vCenter Server is unavailable, those changes take effect. When vCenter Server becomes available again, you might find that clusters have turned red or yellow because cluster requirements are no longer met.

When considering cluster validity scenarios, you should understand these terms.

Reservation — A fixed, guaranteed allocation for the resource pool input by the user.

Reservation Used — The sum of the reservation or reservation used (whichever is larger) for each child resource pool, added recursively.

Unreserved — This nonnegative number differs according to resource pool type.

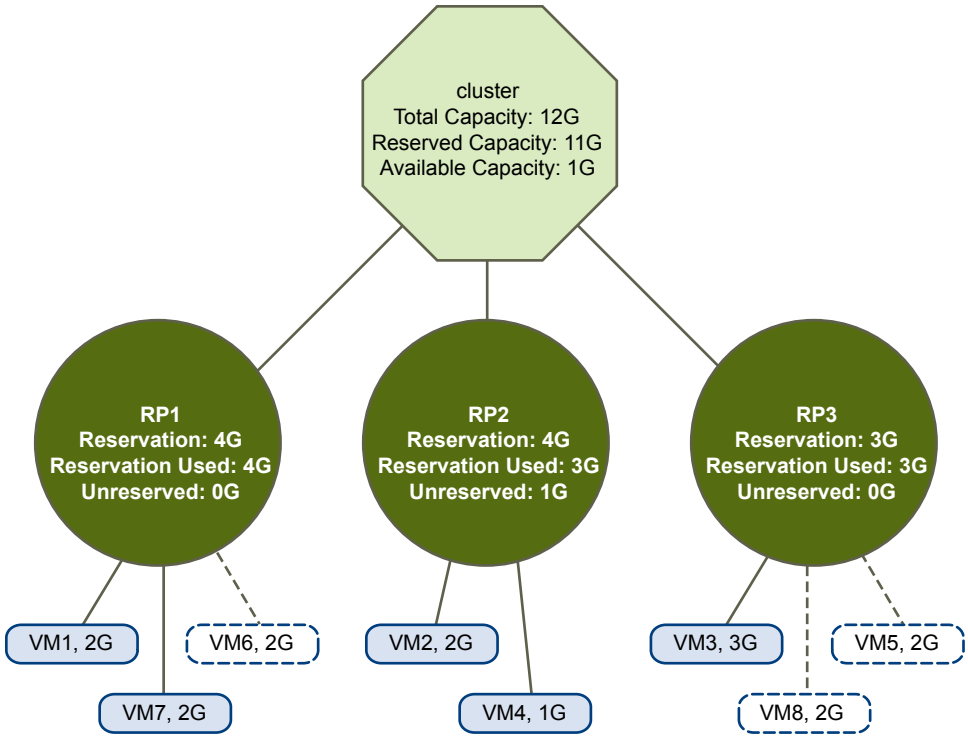
- Nonexpandable resource pools: Reservation minus reservation used.
- Expandable resource pools: (Reservation minus reservation used) plus any unreserved resources that can be borrowed from its ancestor resource pools.

Valid DRS Clusters

A valid cluster has enough resources to meet all reservations and to support all running virtual machines.

The following figure shows an example of a valid cluster with fixed resource pools and how its CPU and memory resources are computed.

Figure 10-1. Valid Cluster with Fixed Resource Pools



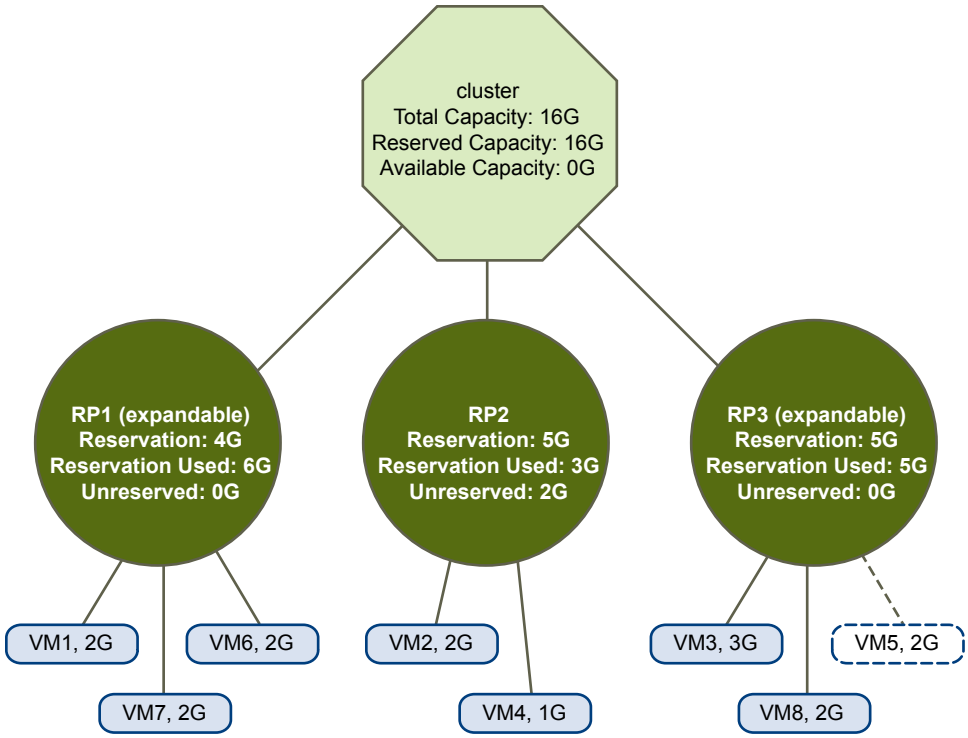
The cluster has the following characteristics:

- A cluster with total resources of 12GHz.
- Three resource pools, each of type **Fixed** (**Expandable Reservation** is not selected).

- The total reservation of the three resource pools combined is 11GHz (4+4+3 GHz). The total is shown in the **Reserved Capacity** field for the cluster.
- RP1 was created with a reservation of 4GHz. Two virtual machines. (VM1 and VM7) of 2GHz each are powered on (**Reservation Used**: 4GHz). No resources are left for powering on additional virtual machines. VM6 is shown as not powered on. It consumes none of the reservation.
- RP2 was created with a reservation of 4GHz. Two virtual machines of 1GHz and 2GHz are powered on (**Reservation Used**: 3GHz). 1GHz remains unreserved.
- RP3 was created with a reservation of 3GHz. One virtual machine with 3GHz is powered on. No resources for powering on additional virtual machines are available.

The following figure shows an example of a valid cluster with some resource pools (RP1 and RP3) using reservation type **Expandable**.

Figure 10-2. Valid Cluster with Expandable Resource Pools



A valid cluster can be configured as follows:

- A cluster with total resources of 16GHz.
- RP1 and RP3 are of type **Expandable**, RP2 is of type Fixed.
- The total reservation used of the three resource pools combined is 16GHz (6GHz for RP1, 5GHz for RP2, and 5GHz for RP3). 16GHz shows up as the **Reserved Capacity** for the cluster at top level.
- RP1 was created with a reservation of 4GHz. Three virtual machines of 2GHz each are powered on. Two of those virtual machines (for example, VM1 and VM7) can use RP1's reservations, the third virtual machine (VM6) can use reservations from the cluster's resource pool. (If the type of this resource pool were **Fixed**, you could not power on the additional virtual machine.)
- RP2 was created with a reservation of 5GHz. Two virtual machines of 1GHz and 2GHz are powered on (**Reservation Used**: 3GHz). 2GHz remains unreserved.

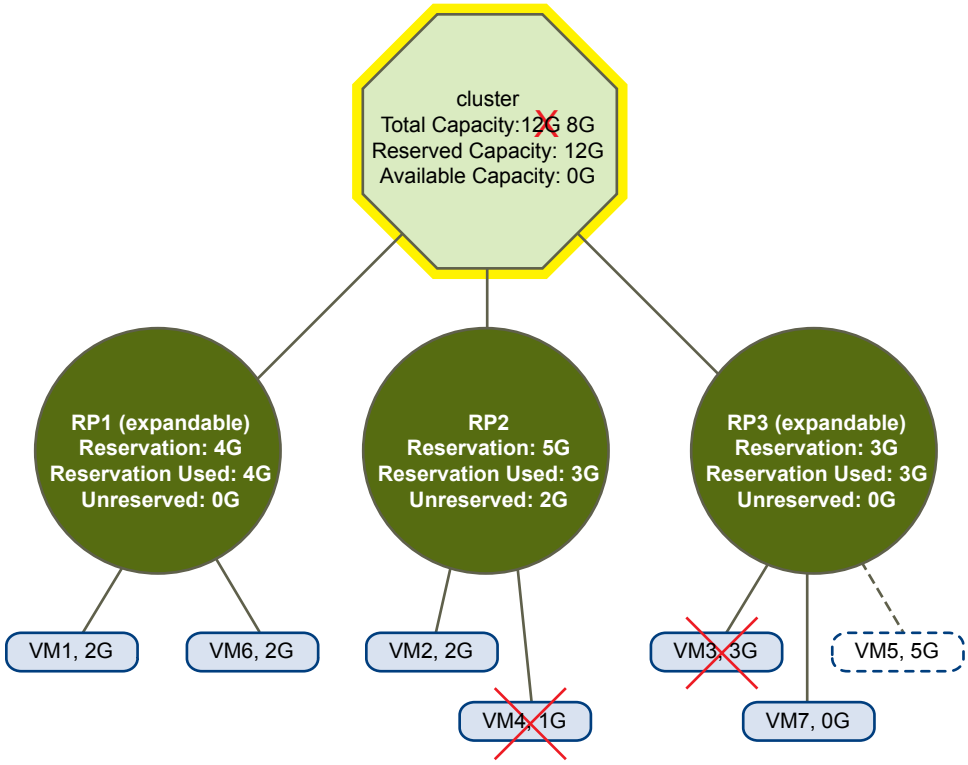
  RP3 was created with a reservation of 5GHz. Two virtual machines of 3GHz and 2GHz are powered on. Even though this resource pool is of type **Expandable**, no additional 2GHz virtual machine can be powered on because the parent's extra resources are already used by RP1.

Overcommitted DRS Clusters

A cluster becomes overcommitted (yellow) when the tree of resource pools and virtual machines is internally consistent but the cluster does not have the capacity to support all resources reserved by the child resource pools.

There will always be enough resources to support all running virtual machines because, when a host becomes unavailable, all its virtual machines become unavailable. A cluster typically turns yellow when cluster capacity is suddenly reduced, for example, when a host in the cluster becomes unavailable. VMware recommends that you leave adequate additional cluster resources to avoid your cluster turning yellow.

Figure 10-3. Yellow Cluster



In this example:

- A cluster with total resources of 12GHz coming from three hosts of 4GHz each.
- Three resource pools reserving a total of 12GHz.
- The total reservation used by the three resource pools combined is 12GHz (4+5+3 GHz). That shows up as the **Reserved Capacity** in the cluster.
- One of the 4GHz hosts becomes unavailable, so total resources reduce to 8GHz.
- At the same time, VM4 (1GHz) and VM3 (3GHz), which were running on the host that failed, are no longer running.
- The cluster is now running virtual machines that require a total of 6GHz. The cluster still has 8GHz available, which is sufficient to meet virtual machine requirements.

  The resource pool reservations of 12GHz can no longer be met, so the cluster is marked as yellow.

Invalid DRS Clusters

A cluster enabled for DRS becomes invalid (red) when the tree is no longer internally consistent, that is, resource constraints are not observed.

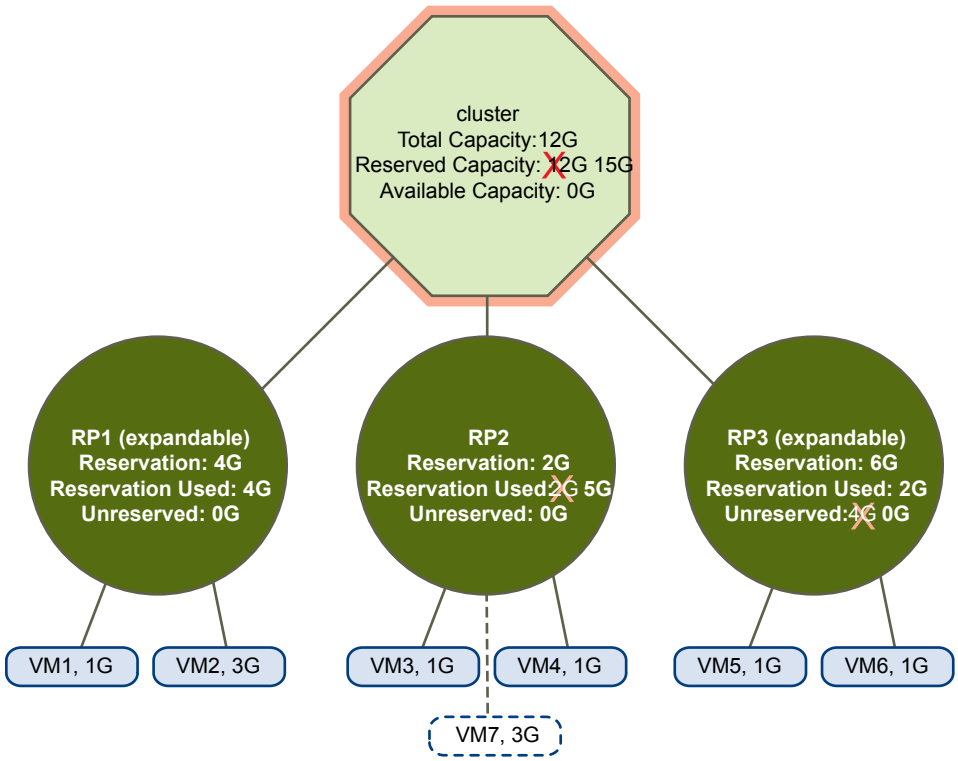
The total amount of resources in the cluster does not affect whether the cluster is red. A cluster can be red, even if enough resources exist at the root level, if there is an inconsistency at a child level.

You can resolve a red DRS cluster problem either by powering off one or more virtual machines, moving virtual machines to parts of the tree that have sufficient resources, or editing the resource pool settings in the red part. Adding resources typically helps only when you are in the yellow state.

A cluster can also turn red if you reconfigure a resource pool while a virtual machine is failing over. A virtual machine that is failing over is disconnected and does not count toward the reservation used by the parent resource pool. You might reduce the reservation of the parent resource pool before the failover completes. After the failover is complete, the virtual machine resources are again charged to the parent resource pool. If the pool's usage becomes larger than the new reservation, the cluster turns red.

If a user is able to start a virtual machine (in an unsupported way) with a reservation of 3GHz under resource pool 2, the cluster would become red, as shown in the following figure.

Figure 10-4. Red Cluster



Managing Power Resources

The vSphere Distributed Power Management (DPM) feature allows a DRS cluster to reduce its power consumption by powering hosts on and off based on cluster resource utilization.

vSphere DPM monitors the cumulative demand of all virtual machines in the cluster for memory and CPU resources and compares this to the total available resource capacity of all hosts in the cluster. If sufficient excess capacity is found, vSphere DPM places one or more hosts in standby mode and powers them off after migrating their virtual machines to other hosts. Conversely, when capacity is deemed to be inadequate, DRS brings hosts out of standby mode (powers them on) and uses vMotion to migrate virtual machines to them. When making these calculations, vSphere DPM considers not only current demand, but it also honors any user-specified virtual machine resource reservations.

NOTE ESXi hosts cannot automatically be brought out of standby mode unless they are running in a cluster managed by vCenter Server.

vSphere DPM can use one of three power management protocols to bring a host out of standby mode: Intelligent Platform Management Interface (IPMI), Hewlett-Packard Integrated Lights-Out (iLO), or Wake-On-LAN (WOL). Each protocol requires its own hardware support and configuration. If a host does not support any of these protocols it cannot be put into standby mode by vSphere DPM. If a host supports multiple protocols, they are used in the following order: IPMI, iLO, WOL.

NOTE Do not disconnect a host in standby mode or move it out of the DRS cluster without first powering it on, otherwise vCenter Server is not able to power the host back on.

Configure IPMI or iLO Settings for vSphere DPM

IPMI is a hardware-level specification and Hewlett-Packard iLO is an embedded server management technology. Each of them describes and provides an interface for remotely monitoring and controlling computers.

You must perform the following procedure on each host.

Prerequisites

Both IPMI and iLO require a hardware Baseboard Management Controller (BMC) to provide a gateway for accessing hardware control functions, and allow the interface to be accessed from a remote system using serial or LAN connections. The BMC is powered-on even when the host itself is powered-off. If properly enabled, the BMC can respond to remote power-on commands.

If you plan to use IPMI or iLO as a wake protocol, you must configure the BMC. BMC configuration steps vary according to model. See your vendor's documentation for more information. With IPMI, you must also ensure that the BMC LAN channel is configured to be always available and to allow operator-privileged commands. On some IPMI systems, when you enable "IPMI over LAN" you must configure this in the BIOS and specify a particular IPMI account.

vSphere DPM using only IPMI supports MD5- and plaintext-based authentication, but MD2-based authentication is not supported. vCenter Server uses MD5 if a host's BMC reports that it is supported and enabled for the Operator role. Otherwise, plaintext-based authentication is used if the BMC reports it is supported and enabled. If neither MD5 nor plaintext authentication is enabled, IPMI cannot be used with the host and vCenter Server attempts to use Wake-on-LAN.

Procedure

1 Select the host in the vSphere Client inventory.

2 Click the **Configuration** tab.

3 Click **Power Management**.

4 Click **Properties**.

5 Enter the following information.

- User name and password for a BMC account. (The user name must have the ability to remotely power the host on.)
- IP address of the NIC associated with the BMC, as distinct from the IP address of the host. The IP address should be static or a DHCP address with infinite lease.
- MAC address of the NIC associated with the BMC.

6 Click **OK**.

Test Wake-on-LAN for vSphere DPM

The use of Wake-on-LAN (WOL) for the vSphere DPM feature is fully supported, if you configure and successfully test it according to the VMware guidelines. You must perform these steps before enabling vSphere DPM for a cluster for the first time or on any host that is being added to a cluster that is using vSphere DPM.

Prerequisites

Before testing WOL, ensure that your cluster meets the prerequisites.

- Your cluster must contain at least two ESX 3.5 (or ESX 3i version 3.5) or later hosts.
- Each host's vMotion networking link must be working correctly. The vMotion network should also be a single IP subnet, not multiple subnets separated by routers.
- The vMotion NIC on each host must support WOL. To check for WOL support, first determine the name of the physical network adapter corresponding to the VMkernel port by selecting the host in the inventory panel of the vSphere Client, selecting the **Configuration** tab, and clicking **Networking**. After you have this information, click on **Network Adapters** and find the entry corresponding to the network adapter. The **Wake On LAN Supported** column for the relevant adapter should show Yes.
- To display the WOL-compatibility status for each NIC on a host, select the host in the inventory panel of the vSphere Client, select the **Configuration** tab, and click **Network Adapters**. The NIC must show Yes in the **Wake On LAN Supported** column.
- The switch port that each WOL-supporting vMotion NIC is plugged into should be set to auto negotiate the link speed, and not set to a fixed speed (for example, 1000 Mb/s). Many NICs support WOL only if they can switch to 100 Mb/s or less when the host is powered off.

After you verify these prerequisites, test each ESXi host that is going to use WOL to support vSphere DPM. When you test these hosts, ensure that the vSphere DPM feature is disabled for the cluster.

CAUTION Ensure that any host being added to a vSphere DPM cluster that uses WOL as a wake protocol is tested and disabled from using power management if it fails the testing. If this is not done, vSphere DPM might power off hosts that it subsequently cannot power back up.

Procedure

1 Click the **Enter Standby Mode** command on the host's **Summary** tab in the vSphere Client.

This action powers down the host.

2 Try to bring the host out of standby mode by clicking the **Power On**command on the host's **Summary** tab.

3 Observe whether or not the host successfully powers back on.

4 For any host that fails to exit standby mode successfully, select the host in the cluster Settings dialog box's Host Options page and change its **Power Management** setting to Disabled.

After you do this, vSphere DPM does not consider that host a candidate for being powered off.

Enabling vSphere DPM for a DRS Cluster

After you have performed configuration or testing steps required by the wake protocol you are using on each host, you can enable vSphere DPM.

Configure the power management automation level, threshold, and host-level overrides. These settings are configured under **Power Management** in the cluster's Settings dialog box.

You can also create scheduled tasks to enable and disable DPM for a cluster using the Schedule Task: Change Cluster Power Settings wizard.

NOTE If a host in your DRS cluster has USB devices connected, disable DPM for that host. Otherwise, DPM might turn off the host and sever the connection between the device and the virtual machine that was using it.

Automation Level

Whether the host power state and migration recommendations generated by vSphere DPM are executed automatically or not depends upon the power management automation level selected for the feature.

The automation level is configured under **Power Management** in the cluster's Settings dialog box.

NOTE The power management automation level is not the same as the DRS automation level.

Table 10-1. Power Management Automation Level

Option	Description
Off	The feature is disabled and no recommendations will be made.
Manual	Host power operation and related virtual machine migration recommendations are made, but not automatically executed. These recommendations appear on the cluster's **DRS** tab in the vSphere Client.
Automatic	Host power operations are automatically executed if related virtual machine migrations can all be executed automatically.

vSphere DPM Threshold

The power state (host power on or off) recommendations generated by the vSphere DPM feature are assigned priorities that range from priority-one recommendations to priority-five recommendations.

These priority ratings are based on the amount of over- or under-utilization found in the DRS cluster and the improvement that is expected from the intended host power state change. A priority-one recommendation is mandatory, while a priority-five recommendation brings only slight improvement.

The threshold is configured under **Power Management** in the cluster's Settings dialog box. Each level you move the vSphere DPM Threshold slider to the right allows the inclusion of one more lower level of priority in the set of recommendations that are executed automatically or appear as recommendations to be manually executed. At the Conservative setting, vSphere DPM only generates priority-one recommendations, the next level to the right only priority-two and higher, and so on, down to the Aggressive level which generates priority-five recommendations and higher (that is, all recommendations.)

NOTE The DRS threshold and the vSphere DPM threshold are essentially independent. You can differentiate the aggressiveness of the migration and host-power-state recommendations they respectively provide.

Host-Level Overrides

When you enable vSphere DPM in a DRS cluster, by default all hosts in the cluster inherit its vSphere DPM automation level.

You can override this default for an individual host by selecting the Host Options page of the cluster's Settings dialog box and clicking its **Power Management** setting. You can change this setting to the following options:

- Disabled
- Manual
- Automatic

NOTE Do not change a host's Power Management setting if it has been set to Disabled due to failed exit standby mode testing.

After enabling and running vSphere DPM, you can verify that it is functioning properly by viewing each host's **Last Time Exited Standby** information displayed on the Host Options page in the cluster Settings dialog box and on the **Hosts** tab for each cluster. This field shows a timestamp and whether vCenter Server Succeeded or Failed the last time it attempted to bring the host out of standby mode. If no such attempt has been made, the field displays Never.

NOTE Times for the **Last Time Exited Standby** text box are derived from the vCenter Server event log. If this log is cleared, the times are reset to Never.

Monitoring vSphere DPM

You can use event-based alarms in vCenter Server to monitor vSphere DPM.

The most serious potential error you face when using vSphere DPM is the failure of a host to exit standby mode when its capacity is needed by the DRS cluster. You can monitor for instances when this error occurs by using the preconfigured **Exit Standby Error** alarm in vCenter Server. If vSphere DPM cannot bring a host out of standby mode (vCenter Server event `DrsExitStandbyModeFailedEvent`), you can configure this alarm to send an alert email to the administrator or to send notification using an SNMP trap. By default, this alarm is cleared after vCenter Server is able to successfully connect to that host.

To monitor vSphere DPM activity, you can also create alarms for the following vCenter Server events.

Table 10-2. vCenter Server Events

Event Type	Event Name
Entering Standby mode (about to power off host)	`DrsEnteringStandbyModeEvent`
Successfully entered Standby mode (host power off succeeded)	`DrsEnteredStandbyModeEvent`
Exiting Standby mode (about to power on the host)	`DrsExitingStandbyModeEvent`
Successfully exited Standby mode (power on succeeded)	`DrsExitedStandbyModeEvent`

For more information about creating and editing alarms, see the *vSphere Monitoring and Performance* documentation.

If you use monitoring software other than vCenter Server, and that software triggers alarms when physical hosts are powered off unexpectedly, you might have a situation where false alarms are generated when vSphere DPM places a host into standby mode. If you do not want to receive such alarms, work with your vendor to deploy a version of the monitoring software that is integrated with vCenter Server. You could also use vCenter Server itself as your monitoring solution, because starting with vSphere 4.x, it is inherently aware of vSphere DPM and does not trigger these false alarms.

Using DRS Affinity Rules

You can control the placement of virtual machines on hosts within a cluster by using affinity rules.

You can create two types of rules.

- Used to specify affinity or anti-affinity between a group of virtual machines and a group of hosts. An affinity rule specifies that the members of a selected virtual machine DRS group can or must run on the members of a specific host DRS group. An anti-affinity rule specifies that the members of a selected virtual machine DRS group cannot run on the members of a specific host DRS group.

  See "VM-Host Affinity Rules," on page 73 for information about creating and using this type of rule.

- Used to specify affinity or anti-affinity between individual virtual machines. A rule specifying affinity causes DRS to try to keep the specified virtual machines together on the same host, for example, for performance reasons. With an anti-affinity rule, DRS tries to keep the specified virtual machines apart, for example, so that when a problem occurs with one host, you do not lose both virtual machines.

  See "VM-VM Affinity Rules," on page 72 for information about creating and using this type of rule.

When you add or edit an affinity rule, and the cluster's current state is in violation of the rule, the system continues to operate and tries to correct the violation. For manual and partially automated DRS clusters, migration recommendations based on rule fulfillment and load balancing are presented for approval. You are not required to fulfill the rules, but the corresponding recommendations remain until the rules are fulfilled.

To check whether any enabled affinity rules are being violated and cannot be corrected by DRS, select the cluster's **DRS** tab and click **Faults**. Any rule currently being violated has a corresponding fault on this page. Read the fault to determine why DRS is not able to satisfy the particular rule. Rules violations also produce a log event.

NOTE VM-VM and VM-Host affinity rules are different from an individual host's CPU affinity rules.

Create a Host DRS Group

A VM-Host affinity rule establishes an affinity (or anti-affinity) relationship between a virtual machine DRS group with a host DRS group. You must create both of these groups before you can create a rule that links them.

Procedure

1. In the vSphere Client, right-click the cluster in the inventory and select **Edit Settings**.
2. In the left pane of the cluster Settings dialog box under **vSphere DRS**, select **DRS Groups Manager**.
3. In the Host DRS Groups section, click **Add**.
4. In the DRS Group dialog box, type a name for the group.
5. In the left pane, select a host and click **>>** to add it to the group. Continue this process until all desired hosts have been added.

   You can also remove hosts from the group by selecting them in the right pane and clicking **<<**.
6. Click **OK**.

What to do next

Using this host DRS group, you can create a VM-Host affinity rule that establishes an affinity (or anti-affinity) relationship with an appropriate virtual machine DRS group.

"Create a Virtual Machine DRS Group," on page 72

"Create a VM-Host Affinity Rule," on page 73

Create a Virtual Machine DRS Group

Affinity rules establish an affinity (or anti-affinity) relationship between DRS groups. You must create DRS groups before you can create a rule that links them.

Procedure

1 In the vSphere Client, right-click the cluster in the inventory and select **Edit Settings**.

2 In the left pane of the cluster Settings dialog box under **vSphere DRS**, select **DRS Groups Manager**.

3 In the Virtual Machines DRS Groups section, click **Add**.

4 In the DRS Group dialog box, type a name for the group.

5 In the left pane, select a host and click **>>** to add it to the group. Continue this process until all desired hosts have been added.

You can also remove hosts from the group by selecting them in the right pane and clicking **<<**.

6 Click **OK**.

What to do next

"Create a Host DRS Group," on page 71

"Create a VM-Host Affinity Rule," on page 73

"Create a VM-VM Affinity Rule," on page 72

VM-VM Affinity Rules

A VM-VM affinity rule specifies whether selected individual virtual machines should run on the same host or be kept on separate hosts. This type of rule is used to create affinity or anti-affinity between individual virtual machines that you select.

When an affinity rule is created, DRS tries to keep the specified virtual machines together on the same host. You might want to do this, for example, for performance reasons.

With an anti-affinity rule, DRS tries to keep the specified virtual machines apart. You could use such a rule if you want to guarantee that certain virtual machines are always on different physical hosts. In that case, if a problem occurs with one host, not all virtual machines would be placed at risk.

Create a VM-VM Affinity Rule

You can create VM-VM affinity rules in the Cluster Settings dialog box to specify whether selected individual virtual machines should run on the same host or be kept on separate hosts.

NOTE If you use the vSphere HA Specify Failover Hosts admission control policy and designate multiple failover hosts, VM-VM affinity rules are not supported.

Procedure

1 In the vSphere Client, right-click the cluster in the inventory and select **Edit Settings**.

2 In the left pane of the Cluster Settings dialog box under **vSphere DRS**, select **Rules**.

3 Click **Add**.

4 In the Rule dialog box, type a name for the rule.

5 From the **Type** menu, select either **Keep Virtual Machines Together** or **Separate Virtual Machines**.

6 Click **Add**.

7 Select at least two virtual machines to which the rule will apply and click **OK**.

8 Click **OK**.

VM-VM Affinity Rule Conflicts

You can create and use multiple VM-VM affinity rules, however, this might lead to situations where the rules conflict with one another.

If two VM-VM affinity rules are in conflict, you cannot enable both. For example, if one rule keeps two virtual machines together and another rule keeps the same two virtual machines apart, you cannot enable both rules. Select one of the rules to apply and disable or remove the conflicting rule.

When two VM-VM affinity rules conflict, the older one takes precedence and the newer rule is disabled. DRS only tries to satisfy enabled rules and disabled rules are ignored. DRS gives higher precedence to preventing violations of anti-affinity rules than violations of affinity rules.

VM-Host Affinity Rules

A VM-Host affinity rule specifies whether or not the members of a selected virtual machine DRS group can run on the members of a specific host DRS group.

Unlike a VM-VM affinity rule, which specifies affinity (or anti-affinity) between individual virtual machines, a VM-Host affinity rule specifies an affinity relationship between a group of virtual machines and a group of hosts. There are 'required' rules (designated by "must") and 'preferential' rules (designated by "should".)

A VM-Host affinity rule includes the following components.

- One virtual machine DRS group.
- One host DRS group.
- A designation of whether the rule is a requirement ("must") or a preference ("should") and whether it is affinity ("run on") or anti-affinity ("not run on").

Because VM-Host affinity rules are cluster-based, the virtual machines and hosts that are included in a rule must all reside in the same cluster. If a virtual machine is removed from the cluster, it loses its DRS group affiliation, even if it is later returned to the cluster.

Create a VM-Host Affinity Rule

You can create VM-Host affinity rules in the Cluster Settings dialog box to specify whether or not the members of a selected virtual machine DRS group can run on the members of a specific host DRS group.

Prerequisites

Create the virtual machine and host DRS groups to which the VM-Host affinity rule applies.

Procedure

1 In the vSphere Client, right-click the cluster in the inventory and select **Edit Settings**.

2 In the left pane of the Cluster Settings dialog box under vSphere DRS, select **Rules**.

3 Click **Add**.

4 In the Rule dialog box, type a name for the rule.

5 From the **Type** menu, select **Virtual Machines to Hosts**.

6 Select the virtual machine DRS group and the host DRS group to which the rule applies.

7 Select a specification for the rule.

- **Must run on hosts in group**. Virtual machines in VM Group 1 must run on hosts in Host Group A.
- **Should run on hosts in group**. Virtual machines in VM Group 1 should, but are not required, to run on hosts in Host Group A.
- **Must not run on hosts in group**. Virtual machines in VM Group 1 must never run on host in Host Group A.
- **Should not run on hosts in group**. Virtual machines in VM Group 1 should not, but might, run on hosts in Host Group A.

8 Click **OK**.

Using VM-Host Affinity Rules

You use a VM-Host affinity rule to specify an affinity relationship between a group of virtual machines and a group of hosts. When using VM-Host affinity rules, you should be aware of when they could be most useful, how conflicts between rules are resolved, and the importance of caution when setting required affinity rules.

One use case where VM-Host affinity rules are helpful is when the software you are running in your virtual machines has licensing restrictions. You can place such virtual machines into a DRS group and then create a rule that requires them to run on a host DRS group that contains only host machines that have the required licenses.

NOTE When you create a VM-Host affinity rule that is based on the licensing or hardware requirements of the software running in your virtual machines, you are responsible for ensuring that the groups are properly set up. The rule does not monitor the software running in the virtual machines nor does it know what non-VMware licenses are in place on which ESXi hosts.

If you create more than one VM-Host affinity rule, the rules are not ranked, but are applied equally. Be aware that this has implications for how the rules interact. For example, a virtual machine that belongs to two DRS groups, each of which belongs to a different required rule, can run only on hosts that belong to both of the host DRS groups represented in the rules.

When you create a VM-Host affinity rule, its ability to function in relation to other rules is not checked. So it is possible for you to create a rule that conflicts with the other rules you are using. When two VM-Host affinity rules conflict, the older one takes precedence and the newer rule is disabled. DRS only tries to satisfy enabled rules and disabled rules are ignored.

DRS, vSphere HA, and vSphere DPM never take any action that results in the violation of required affinity rules (those where the virtual machine DRS group 'must run on' or 'must not run on' the host DRS group). Accordingly, you should exercise caution when using this type of rule because of its potential to adversely affect the functioning of the cluster. If improperly used, required VM-Host affinity rules can fragment the cluster and inhibit the proper functioning of DRS, vSphere HA, and vSphere DPM.

A number of cluster functions are not performed if doing so would violate a required affinity rule.

- DRS does not evacuate virtual machines to place a host in maintenance mode.
- DRS does not place virtual machines for power-on or load balance virtual machines.
- vSphere HA does not perform failovers.
- vSphere DPM does not optimize power management by placing hosts into standby mode.

To avoid these situations, exercise caution when creating more than one required affinity rule or consider using VM-Host affinity rules that are preferential only (those where the virtual machine DRS group 'should run on' or 'should not run on' the host DRS group). Ensure that the number of hosts in the cluster with which each virtual machine is affined is large enough that losing a host does not result in a lack of hosts on which the virtual machine can run. Preferential rules can be violated to allow the proper functioning of DRS, vSphere HA, and vSphere DPM.

NOTE You can create an event-based alarm that is triggered when a virtual machine violates a VM-Host affinity rule. In the vSphere Client, add a new alarm for the virtual machine and select **VM is violating VM-Host Affinity Rule** as the event trigger. For more information about creating and editing alarms, see the *vSphere Monitoring and Performance* documentation.

11 Creating a Datastore Cluster

A datastore cluster is a collection of datastores with shared resources and a shared management interface. Datastore clusters are to datastores what clusters are to hosts. When you create a datastore cluster, you can use vSphere Storage DRS to manage storage resources.

NOTE Datastore clusters are referred to as storage pods in the vSphere API.

When you add a datastore to a datastore cluster, the datastore's resources become part of the datastore cluster's resources. As with clusters of hosts, you use datastore clusters to aggregate storage resources, which enables you to support resource allocation policies at the datastore cluster level. The following resource management capabilities are also available per datastore cluster.

Space utilization load balancing	You can set a threshold for space use. When space use on a datastore exceeds the threshold, Storage DRS generates recommendations or performs Storage vMotion migrations to balance space use across the datastore cluster.
I/O latency load balancing	You can set an I/O latency threshold for bottleneck avoidance. When I/O latency on a datastore exceeds the threshold, Storage DRS generates recommendations or performs Storage vMotion migrations to help alleviate high I/O load.
Anti-affinity rules	You can create anti-affinity rules for virtual machine disks. For example, the virtual disks of a certain virtual machine must be kept on different datastores. By default, all virtual disks for a virtual machine are placed on the same datastore.

This chapter includes the following topics:

Initial Placement and Ongoing Balancing

Storage DRS provides initial placement and ongoing balancing recommendations to datastores in a Storage DRS-enabled datastore cluster.

Initial placement occurs when Storage DRS selects a datastore within a datastore cluster on which to place a virtual machine disk. This happens when the virtual machine is being created or cloned, when a virtual machine disk is being migrated to another datastore cluster, or when you add a disk to an existing virtual machine.

Initial placement recommendations are made in accordance with space constraints and with respect to the goals of space and I/O load balancing. These goals aim to minimize the risk of over-provisioning one datastore, storage I/O bottlenecks, and performance impact on virtual machines.

Storage DRS is invoked at the configured frequency (by default, every eight hours) or when one or more datastores in a datastore cluster exceeds the user-configurable space utilization thresholds. When Storage DRS is invoked, it checks each datastore's space utilization and I/O latency values against the threshold. For I/O latency, Storage DRS uses the 90th percentile I/O latency measured over the course of a day to compare against the threshold.

Storage Migration Recommendations

vCenter Server displays migration recommendations on the Storage DRS Recommendations page for datastore clusters that have manual automation mode.

The system provides as many recommendations as necessary to enforce Storage DRS rules and to balance the space and I/O resources of the datastore cluster. Each recommendation includes the virtual machine name, the virtual disk name, the name of the datastore cluster, the source datastore, the destination datastore, and a reason for the recommendation.

- Balance datastore space use
- Balance datastore I/O load

Storage DRS makes mandatory recommendations for migration in the following situations:

- The datastore is out of space.
- Anti-affinity or affinity rules are being violated.
- The datastore is entering maintenance mode and must be evacuated.

In addition, optional recommendations are made when a datastore is close to running out of space or when adjustments should be made for space and I/O load balancing.

Storage DRS considers moving virtual machines that are powered off or powered on for space balancing. Storage DRS includes powered-off virtual machines with snapshots in these considerations.

Create a Datastore Cluster

You can manage datastore cluster resources using Storage DRS.

Procedure

1. In the Datastores and Datastore Clusters view of the vSphere Client inventory, right-click the Datacenter object and select **New Datastore Cluster**.
2. Follow the prompts to complete the Create Datastore Cluster wizard.

Enable and Disable Storage DRS

Storage DRS allows you to manage the aggregated resources of a datastore cluster. When Storage DRS is enabled, it provides recommendations for virtual machine disk placement and migration to balance space and I/O resources across the datastores in the datastore cluster.

When you enable Storage DRS, you enable the following functions.

- Space load balancing among datastores within a datastore cluster.
- I/O load balancing among datastores within a datastore cluster.
- Initial placement for virtual disks based on space and I/O workload.

The Enable Storage DRS check box in the Datastore Cluster Settings dialog box enables or disables all of these components at once. If necessary, you can disable I/O-related functions of Storage DRS independently of space balancing functions.

When you disable Storage DRS on a datastore cluster, Storage DRS settings are preserved. When you enable Storage DRS, the settings for the datastore cluster are restored to the point where Storage DRS was disabled.

Procedure

1 In the vSphere Client inventory, right-click a datastore cluster and select **Edit Settings**.

2 Click **General**.

3 Select **Turn on Storage DRS** and click **OK**.

4 (Optional) To disable only I/O-related functions of Storage DRS, leaving space-related controls enabled, perform the following steps.

   a Select **SDRS Runtime Rules**.

   b Deselect the **Enable I/O metric for Storage DRS** check box.

5 Click **OK**.

Set the Automation Level for Datastore Clusters

The automation level for a datastore cluster specifies whether or not placement and migration recommendations from Storage DRS are applied automatically.

Procedure

1 In the vSphere Client inventory, right-click a datastore cluster and select **Edit Settings**.

2 Select **SDRS Automation**.

3 Select an automation level.

   Manual is the default automation level.

Option	Description
No Automation (Manual Mode)	Placement and migration recommendations are displayed, but do not run until you manually apply the recommendation.
Fully Automated	Placement and migration recommendations run automatically.

4 Click **OK**.

Setting the Aggressiveness Level for Storage DRS

The aggressiveness of Storage DRS is determined by specifying thresholds for space used and I/O latency.

Storage DRS collects resource usage information for the datastores in a datastore cluster. vCenter Server uses this information to generate recommendations for placement of virtual disks on datastores.

When you set a low aggressiveness level for a datastore cluster, Storage DRS recommends Storage vMotion migrations only when absolutely necessary, for example, if when I/O load, space utilization, or their imbalance is high. When you set a high aggressiveness level for a datastore cluster, Storage DRS recommends migrations whenever the datastore cluster can benefit from space or I/O load balancing.

In the vSphere Client, you can use the following thresholds to set the aggressiveness level for Storage DRS:

Space Utilization — Storage DRS generates recommendations or performs migrations when the percentage of space utilization on the datastore is greater than the threshold you set in the vSphere Client.

I/O Latency — Storage DRS generates recommendations or performs migrations when the 90th percentile I/O latency measured over a day for the datastore is greater than the threshold.

You can also set advanced options to further configure the aggressiveness level of Storage DRS.

Space utilization difference — This threshold ensures that there is some minimum difference between the space utilization of the source and the destination. For example, if the space used on datastore A is 82% and datastore B is 79%, the difference is 3. If the threshold is 5, Storage DRS will not make migration recommendations from datastore A to datastore B.

I/O load balancing invocation interval — After this interval, Storage DRS runs to balance I/O load.

I/O imbalance threshold — Lowering this value makes I/O load balancing less aggressive. Storage DRS computes an I/O fairness metric between 0 and 1, which 1 being the fairest distribution. I/O load balancing runs only if the computed metric is less than 1 - (I/O imbalance threshold / 100).

Set Storage DRS Runtime Rules

Set Storage DRS triggers and configure advanced options for the datastore cluster.

Procedure

1 (Optional) Select or deselect the **Enable I/O metric for SDRS recommendations** check box to enable or disable I/O metric inclusion.

When you disable this option, vCenter Server does not consider I/O metrics when making Storage DRS recommendations. When you disable this option, you disable the following elements of Storage DRS:

- I/O load balancing among datastores within a datastore cluster.
- Initial placement for virtual disks based on I/O workload. Initial placement is based on space only.

2 (Optional) Set Storage DRS thresholds.

You set the aggressiveness level of Storage DRS by specifying thresholds for used space and I/O latency.

- Use the Utilized Space slider to indicate the maximum percentage of consumed space allowed before Storage DRS is triggered. Storage DRS makes recommendations and performs migrations when space use on the datastores is higher than the threshold.
- Use the I/O Latency slider to indicate the maximum I/O latency allowed before Storage DRS is triggered. Storage DRS makes recommendations and performs migrations when latency is higher than the threshold.

NOTE The Storage DRS I/O Latency threshold for the datastore cluster should be lower than or equal to the Storage I/O Control congestion threshold.

3 (Optional) Configure advanced options.

- No recommendations until utilization difference between source and destination is: Use the slider to specify the space utilization difference threshold. Utilization is usage * 100/capacity.

  This threshold ensures that there is some minimum difference between the space utilization of the source and the destination. For example, if the space used on datastore A is 82% and datastore B is 79%, the difference is 3. If the threshold is 5, Storage DRS will not make migration recommendations from datastore A to datastore B.
- Evaluate I/O load every: Specify how often Storage DRS should assess space and I/O load balancing.
- I/O imbalance threshold: Use the slider to indicate the aggressiveness of I/O load balancing. Lowering this value makes I/O load balancing less aggressive. Storage DRS computes an I/O fairness metric between 0 and 1, which 1 being the fairest distribution. I/O load balancing runs only if the computed metric is less than 1 - (I/O imbalance threshold / 100).

4 Click **Next**.

Datastore Cluster Requirements

Datastores and hosts that are associated with a datastore cluster must meet certain requirements to use datastore cluster features successfully.

Follow these guidelines when you create a datastore cluster.

- Datastore clusters must contain similar or interchangeable datastores.

  A datastore cluster can contain a mix of datastores with different sizes and I/O capacities, and can be from different arrays and vendors. However, the following types of datastores cannot coexist in a datastore cluster.

  - NFS and VMFS datastores cannot be combined in the same datastore cluster.
  - Replicated datastores cannot be combined with non-replicated datastores in the same Storage-DRS-enabled datastore cluster.
- All hosts attached to the datastores in a datastore cluster must be ESXi 5.0 and later. If datastores in the datastore cluster are connected to ESX/ESXi 4.x and earlier hosts, Storage DRS does not run.
- Datastores shared across multiple datacenters cannot be included in a datastore cluster.
- As a best practice, do not include datastores that have hardware acceleration enabled in the same datastore cluster as datastores that do not have hardware acceleration enabled. Datastores in a datastore cluster must be homogeneous to guarantee hardware acceleration-supported behavior.

Adding and Removing Datastores from a Datastore Cluster

You add and remove datastores to and from an existing datastore cluster by dragging them in the vSphere Client inventory.

You can add to a datastore cluster any datastore that is mounted on a host in the vSphere Client inventory, with the following exceptions:

- All hosts attached to the datastore must be ESXi 5.0 and later.
- The datastore cannot be in more than one datacenter in the same instance of the vSphere Client.

When you remove a datastore from a datastore cluster, the datastore remains in the vSphere Client inventory and is not unmounted from the host.

Using Datastore Clusters to Manage Storage Resources 12

After you create a datastore cluster, you can customize it and use it to manage storage I/O and space utilization resources.

This chapter includes the following topics:

Using Storage DRS Maintenance Mode

You place a datastore in maintenance mode when you need to take it out of use to service it. A datastore enters or leaves maintenance mode only as the result of a user request.

Maintenance mode is available to datastores within a Storage DRS-enabled datastore cluster. Standalone datastores cannot be placed in maintenance mode.

Virtual disks that are located on a datastore that is entering maintenance mode must be migrated to another datastore, either manually or using Storage DRS. When you attempt to put a datastore in maintenance mode, the **Placement Recommendations** tab displays a list of migration recommendations, datastores within the same datastore cluster where virtual disks can be migrated. On the **Faults** tab, vCenter Server displays a list of the disks that cannot be migrated and the reasons why. If Storage DRS affinity or anti-affinity rules prevent disks from being migrated, you can choose to enable the Ignore Affinity Rules for Maintenance option.

The datastore is in a state of Entering Maintenance Mode until all virtual disks have been migrated.

Place a Datastore in Maintenance Mode

If you need to take a datastore out of service, you can place the datastore in Storage DRS maintenance mode.

Prerequisites

Storage DRS is enabled on the datastore cluster that contains the datastore that is entering maintenance mode.

No CD-ROM image files are stored on the datastore.

There are at least two datastores in the datastore cluster.

Procedure

1 In the vSphere Client inventory, right-click a datastore in a datastore cluster and select **Enter SDRS Maintenance Mode**.

A list of recommendations appears for datastore maintenance mode migration.

2 (Optional) On the Placement Recommendations tab, deselect any recommendations you do not want to apply.

NOTE The datastore cannot enter maintenance mode without evacuating all disks. If you deselect recommendations, you must manually move the affected virtual machines.

3 If necessary, click **Apply Recommendations**.

vCenter Server uses Storage vMotion to migrate the virtual disks from the source datastore to the destination datastore and the datastore enters maintenance mode.

The datastore icon might not be immediately updated to reflect the datastore's current state. To update the icon immediately, click **Refresh**.

Ignore Storage DRS Affinity Rules for Maintenance Mode

Storage DRS affinity or anti-affinity rules might prevent a datastore from entering maintenance mode. You can ignore these rules when you put a datastore in maintenance mode.

When you enable the Ignore Affinity Rules for Maintenance option for a datastore cluster, vCenter Server ignores Storage DRS affinity and anti-affinity rules that prevent a datastore from entering maintenance mode.

Storage DRS rules are ignored only for evacuation recommendations. vCenter Server does not violate the rules when making space and load balancing recommendations or initial placement recommendations.

Procedure

1 In the vSphere Client inventory, right-click a datastore cluster and select **Edit Settings**.

2 In the right pane of the Edit Datastore Cluster dialog box, select **SDRS Automation**.

3 Click **Advanced Options**.

4 Select **IgnoreAffinityRulesForMaintenance**.

5 In the Value column, type **1** to enable the option.

Type **0** to disable the option.

6 Click **OK**.

The Ignore Affinity Rules for Maintenance Mode option is applied to the datastore cluster.

Applying Storage DRS Recommendations

Storage DRS collects resource usage information for all datastores in a datastore cluster. Storage DRS uses the information to generate recommendations for virtual machine disk placement on datastores in a datastore cluster.

Storage DRS recommendations appear on the **Storage DRS** tab in the vSphere Client datastore view. Recommendations also appear when you attempt to put a datastore into Storage DRS maintenance mode. When you apply Storage DRS recommendations, vCenter Server uses Storage vMotion to migrate virtual machine disks to other datastores in the datastore cluster to balance the resources.

You can apply a subset of the recommendations by selecting the Override Suggested DRS Recommendations check box and selecting each recommendation to apply.

Table 12-1. Storage DRS Recommendations

Label	Description
Priority	Priority level (1-5) of the recommendation. (Hidden by default.)
Recommendation	Action being recommended by Storage DRS.
Reason	Why the action is needed.
Space Utilization % Before (source) and (destination)	Percentage of space used on the source and destination datastores before migration.
Space Utilization % After (source) and (destination)	Percentage of space used on the source and destination datastores after migration.
I/O Latency Before (source)	Value of I/O latency on the source datastore before migration.
I/O Latency Before (destination)	Value of I/O latency on the destination datastore before migration.

Refresh Storage DRS Recommendations

Storage DRS migration recommendations appear on the **Storage DRS** tab in the vSphere Client. You can refresh these recommendations by running Storage DRS.

Prerequisites

At least one datastore cluster must exist in the vSphere Client inventory.

Enable Storage DRS for the datastore cluster. The **Storage DRS** tab appears only if Storage DRS is enabled.

Procedure

1 In the vSphere Client datastore view, select the datastore cluster and click the **Storage DRS** tab.

2 Select the **Recommendations** view and click the **Run Storage DRS** link in the upper right corner.

The recommendations are updated. The Last Updated timestamp displays the time when Storage DRS recommendations were refreshed.

Change Storage DRS Automation Level for a Virtual Machine

You can override the datastore cluster-wide automation level for individual virtual machines. You can also override default virtual disk affinity rules.

Procedure

1 In the vSphere Client inventory, right-click a datastore cluster and select **Edit Settings**.

2 Select **Virtual Machine Settings**.

3 Select a virtual machine.

4 In the Automation Level column, select an automation level for the virtual machine.

Option	Description
Default (Manual)	Placement and migration recommendations are displayed, but do not run until you manually apply the recommendation.
Fully Automated	Placement and migration recommendations run automatically.
Disabled	vCenter Server does not migrate the virtual machine or provide migration recommendations for it.

5 In the **Keep VMDKs together** column, deselect the check box to override default VMDK affinity.

See "Override VMDK Affinity Rules," on page 89.

6 Click **OK**.

Set Up Off-Hours Scheduling for Storage DRS

You can create a scheduled task to change Storage DRS settings for a datastore cluster so that migrations for fully automated datastore clusters are more likely to occur during off-peak hours.

You can create a scheduled task to change the automation level and aggressiveness level for a datastore cluster. For example, you might configure Storage DRS to run less aggressively during peak hours, when performance is a priority, to minimize the occurrence of storage migrations. During non-peak hours, Storage DRS can run in a more aggressive mode and be invoked more frequently.

Prerequisites

Enable Storage DRS.

Procedure

1 In the vSphere Client inventory, right-click a datastore cluster and select **Edit Settings**.

2 In the Edit Datastore Cluster dialog box, click **SDRS Scheduling**.

3 Click **Add**.

4 Type the time and select the days for the task to run.

5 Click **Next**.

6 Specify the start settings for the task.

a Type a description for the start settings.

For example, **`Change SDRS Configuration`**.

b Select an automation level.

c To disable I/O metrics for Storage DRS recommendations, select the check box.

When you disable I/O metrics for Storage DRS recommendations, I/O metrics are not considered as part of Storage DRS recommendations or automated migrations for the datastore cluster.

d Set the Utilized Space threshold.

Use the Utilized Space slider to indicate the maximum percentage of consumed space allowed before Storage DRS is triggered. Storage DRS makes recommendations and performs migrations when space use on the datastores is higher than the threshold.

e Set the I/O latency threshold.

Use the I/O Latency slider to indicate the maximum I/O latency allowed before Storage DRS is triggered. Storage DRS makes recommendations and performs migrations when latency is higher than the threshold.

NOTE The Storage DRS I/O Latency threshold for the datastore cluster should be lower than or equal to the Storage I/O Control congestion threshold.

---

f Set the I/O imbalance threshold.

Use the I/O Imbalance Threshold slider to indicate the aggressiveness of I/O load balancing. Storage DRS makes recommendations and performs migrations if the I/O load imbalance level exceeds the threshold.

7 Click **Next**.

8 Specify the end settings for the task.

- To restore the Storage DRS settings to the pre-task configuration, select the **Restore settings** check box.
- To specify settings other than the pre-task configuration, deselect the **Restore settings** check box.

9 Review the Ready to Complete page and click **Finish**.

The scheduled task runs at the specified time.

Storage DRS Anti-Affinity Rules

You can create Storage DRS anti-affinity rules to control which virtual disks should not be placed on the same datastore within a datastore cluster. By default, a virtual machine's virtual disks are kept together on the same datastore.

When you create an anti-affinity rule, it applies to the relevant virtual disks in the datastore cluster. Anti-affinity rules are enforced during initial placement and Storage DRS-recommendation migrations, but are not enforced when a migration is initiated by a user.

NOTE Anti-affinity rules do not apply to CD-ROM ISO image files that are stored on a datastore in a datastore cluster, nor do they apply to swapfiles that are stored in user-defined locations.

Inter-VM Anti-Affinity Rules	Specify which virtual machines should never be kept on the same datastore. See "Create Inter-VM Anti-Affinity Rules," on page 88.
Intra-VM Anti-Affinity Rules	Specify which virtual disks associated with a particular virtual machine must be kept on different datastores. See "Create Intra-VM Anti-Affinity Rules," on page 88.

If you move a virtual disk out of the datastore cluster, the affinity or anti-affinity rule no longer applies to that disk.

When you move virtual disk files into a datastore cluster that has existing affinity and anti-affinity rules, the following behavior applies:

- Datastore Cluster B has an intra-VM affinity rule. When you move a virtual disk out of Datastore Cluster A and into Datastore Cluster B, any rule that applied to the virtual disk for a given virtual machine in Datastore Cluster A no longer applies. The virtual disk is now subject to the intra-VM affinity rule in Datastore Cluster B.
- Datastore Cluster B has an inter-VM anti-affinity rule. When you move a virtual disk out of Datastore Cluster A and into Datastore Cluster B, any rule that applied to the virtual disk for a given virtual machine in Datastore Cluster A no longer applies. The virtual disk is now subject to the inter-VM anti-affinity rule in Datastore Cluster B.
- Datastore Cluster B has an intra-VM anti-affinity rule. When you move a virtual disk out of Datastore Cluster A and into Datastore Cluster B, the intra-VM anti-affinity rule does not apply to the virtual disk for a given virtual machine because the rule is limited to only specified virtual disks in Datastore Cluster B.

NOTE Storage DRS rules might prevent a datastore from entering maintenance mode. You can choose to ignore Storage DRS rules for maintenance mode by enabling the Ignore Affinity Rules for Maintenance option.

Create Inter-VM Anti-Affinity Rules

You can create an anti-affinity rule to indicate that all virtual disks of certain virtual machines must be kept on different datastores. The rule applies to individual datastore clusters.

Virtual machines that participate in an inter-VM anti-affinity rule in a datastore cluster must be associated with an intra-VM affinity rule in the datastore cluster. The virtual machines must also comply with the intra-VM affinity rule.

If a virtual machine is subject to an inter-VM anti-affinity rule, the following behavior applies:

- Storage DRS places the virtual machine's virtual disks according to the rule.
- Storage DRS migrates the virtual disks using vMotion according to the rule, even if the migration is for a mandatory reason such as putting a datastore in maintenance mode.
- If the virtual machine's virtual disk violates the rule, Storage DRS makes migration recommendations to correct the error or reports the violation as a fault if it cannot make a recommendation that will correct the error.

No inter-VM anti-affinity rules are defined by default.

Procedure

1 In the vSphere Client inventory, right-click a datastore cluster and select **Edit Settings**.

2 In the left pane of the Edit Datastore Cluster dialog box, select **Rules**.

3 Click **Add**.

4 Type a name for the rule.

5 From the Type menu, select **VM anti-affinity**.

6 Click **Add**.

7 Click **Select Virtual Machine**.

8 Select at least two virtual machines and click **OK**.

9 Click **OK** to save the rule.

Create Intra-VM Anti-Affinity Rules

You can create a VMDK anti-affinity rule for a virtual machine that indicates which of its virtual disks must be kept on different datastores.

VMDK anti-affinity rules apply to the virtual machine for which the rule is defined, not to all virtual machines. The rule is expressed as a list of virtual disks that are to be separated from one another.

If you attempt to set an intra-VM anti-affinity rule and an intra-VM affinity rule for a virtual machine, vCenter Server rejects the most recently defined rule.

If a virtual machine is subject to a VMDK anti-affinity rule, the following behavior applies:

- Storage DRS places the virtual machine's virtual disks according to the rule.
- Storage DRS migrates the virtual disks using vMotion according to the rule, even if the migration is for a mandatory reason such as putting a datastore in maintenance mode.
- If the virtual machine's virtual disk violates the rule, Storage DRS makes migration recommendations to correct the error or reports the violation as a fault if it cannot make a recommendation that will correct the error.

No intra-VM anti-affinity rules are defined by default.

Procedure

1 In the vSphere Client inventory, right-click a datastore cluster and select **Edit Settings**.

2 In the left pane of the Edit Datastore Cluster dialog box, select **Rules**.

3 Click **Add**.

4 Type a name for the rule.

5 From the Type menu, select **VMDK anti-affinity**.

6 Click **Add**.

7 Click **Select Virtual Machine**.

8 Select a virtual machine and click **OK**.

9 Select at least two virtual disks to which the rule applies and click **OK**.

10 Click **OK** to save the rule.

Override VMDK Affinity Rules

VMDK affinity rules indicate that all virtual disks in a datastore cluster that are associated with a particular virtual machine are located on the same datastore in the datastore cluster. The rules apply to individual datastore clusters.

VMDK affinity rules are enabled by default for all virtual machines that are in a datastore cluster. You can override the default setting for the datastore cluster or for individual virtual machines.

Virtual machines that are subject to VMDK affinity rules have the following behavior:

- Storage DRS places the virtual machine's virtual disks according to the rule.
- Storage DRS migrates the virtual disks using vMotion according to the rule, even if the migration is for a mandatory reason such as putting a datastore in maintenance mode.
- If the virtual machine's virtual disk violates the rule, Storage DRS makes migration recommendations to correct the error or reports the violation as a fault if it cannot make a recommendation that will correct the error.

When you add a datastore to a datastore cluster that is enabled for Storage DRS, the VMDK affinity rule is disabled for any virtual machine that has virtual disks on that datastore if it also has virtual disks on other datastores.

Procedure

1 In the vSphere Client inventory, right-click a datastore cluster and select **Edit Settings**.

2 Click **Virtual Machine Settings**.

3 Deselect the **Keep VMDKs together** check box for the virtual machine.

4 Click **OK**.

Clear Storage DRS Statistics

To diagnose problems with Storage DRS, you can clear Storage DRS statistics before you manually run Storage DRS.

IMPORTANT When you enable the option to clear Storage DRS statistics, statistics are cleared every time Storage DRS runs until you disable the option. Always disable the option after you diagnose the Storage DRS problem.

Prerequisites

Enable Storage DRS for the datastore cluster.

Procedure

1 Enable the **ClearIoStatsOnSdrsRun** option.

   a In the vSphere Client, right-click the datastore cluster and select **Edit Settings**.

   b Select **SDRS Automation Level** and click **Advanced Options**.

   c In the Option text box, type **ClearIoStatsOnSdrsRun**.

   d In the corresponding Value text box, type **1**.

   e Click **OK**, then click **OK** again to dismiss the settings dialog box.

2 In the vSphere Client inventory, select a datastore cluster.

3 Click the **Storage DRS** tab and select **Run DRS** in the upper right corner of the page.

   The current Storage DRS statistics for all datastores and virtual disks in all datastore clusters in the vSphere Client inventory are cleared, but no new statistics are collected.

4 Change the **ClearIoStatsOnSdrsRun** flag value to **0** to disable it.

5 Run Storage DRS again.

   Storage DRS runs normally. Allow several hours for the new setting to take effect.

Storage vMotion Compatibility with Datastore Clusters

A datastore cluster has certain vSphere Storage vMotion® requirements.

- The host must be running a version of ESXi that supports Storage vMotion.
- The host must have write access to both the source datastore and the destination datastore.
- The host must have enough free memory resources to accommodate Storage vMotion.
- The destination datastore must have sufficient disk space.
- The destination datastore must not be in maintenance mode or entering maintenance mode.

13 Using NUMA Systems with ESXi

ESXi supports memory access optimization for Intel and AMD Opteron processors in server architectures that support NUMA (non-uniform memory access).

After you understand how ESXi NUMA scheduling is performed and how the VMware NUMA algorithms work, you can specify NUMA controls to optimize the performance of your virtual machines.

This chapter includes the following topics:

What is NUMA?

NUMA systems are advanced server platforms with more than one system bus. They can harness large numbers of processors in a single system image with superior price to performance ratios.

For the past decade, processor clock speed has increased dramatically. A multi-gigahertz CPU, however, needs to be supplied with a large amount of memory bandwidth to use its processing power effectively. Even a single CPU running a memory-intensive workload, such as a scientific computing application, can be constrained by memory bandwidth.

This problem is amplified on symmetric multiprocessing (SMP) systems, where many processors must compete for bandwidth on the same system bus. Some high-end systems often try to solve this problem by building a high-speed data bus. However, such a solution is expensive and limited in scalability.

NUMA is an alternative approach that links several small, cost-effective nodes using a high-performance connection. Each node contains processors and memory, much like a small SMP system. However, an advanced memory controller allows a node to use memory on all other nodes, creating a single system image. When a processor accesses memory that does not lie within its own node (remote memory), the data must be transferred over the NUMA connection, which is slower than accessing local memory. Memory access times are not uniform and depend on the location of the memory and the node from which it is accessed, as the technology's name implies.

Challenges for Operating Systems

Because a NUMA architecture provides a single system image, it can often run an operating system with no special optimizations.

The high latency of remote memory accesses can leave the processors under-utilized, constantly waiting for data to be transferred to the local node, and the NUMA connection can become a bottleneck for applications with high-memory bandwidth demands.

Furthermore, performance on such a system can be highly variable. It varies, for example, if an application has memory located locally on one benchmarking run, but a subsequent run happens to place all of that memory on a remote node. This phenomenon can make capacity planning difficult.

Some high-end UNIX systems provide support for NUMA optimizations in their compilers and programming libraries. This support requires software developers to tune and recompile their programs for optimal performance. Optimizations for one system are not guaranteed to work well on the next generation of the same system. Other systems have allowed an administrator to explicitly decide on the node on which an application should run. While this might be acceptable for certain applications that demand 100 percent of their memory to be local, it creates an administrative burden and can lead to imbalance between nodes when workloads change.

Ideally, the system software provides transparent NUMA support, so that applications can benefit immediately without modifications. The system should maximize the use of local memory and schedule programs intelligently without requiring constant administrator intervention. Finally, it must respond well to changing conditions without compromising fairness or performance.

How ESXi NUMA Scheduling Works

ESXi uses a sophisticated NUMA scheduler to dynamically balance processor load and memory locality or processor load balance.

1. Each virtual machine managed by the NUMA scheduler is assigned a home node. A home node is one of the system's NUMA nodes containing processors and local memory, as indicated by the System Resource Allocation Table (SRAT).
2. When memory is allocated to a virtual machine, the ESXi host preferentially allocates it from the home node. The virtual CPUs of the virtual machine are constrained to run on the home node to maximize memory locality.
3. The NUMA scheduler can dynamically change a virtual machine's home node to respond to changes in system load. The scheduler might migrate a virtual machine to a new home node to reduce processor load imbalance. Because this might cause more of its memory to be remote, the scheduler might migrate the virtual machine's memory dynamically to its new home node to improve memory locality. The NUMA scheduler might also swap virtual machines between nodes when this improves overall memory locality.

Some virtual machines are not managed by the ESXi NUMA scheduler. For example, if you manually set the processor or memory affinity for a virtual machine, the NUMA scheduler might not be able to manage this virtual machine. Virtual machines that are not managed by the NUMA scheduler still run correctly. However, they don't benefit from ESXi NUMA optimizations.

The NUMA scheduling and memory placement policies in ESXi can manage all virtual machines transparently, so that administrators do not need to address the complexity of balancing virtual machines between nodes explicitly.

The optimizations work seamlessly regardless of the type of guest operating system. ESXi provides NUMA support even to virtual machines that do not support NUMA hardware, such as Windows NT 4.0. As a result, you can take advantage of new hardware even with legacy operating systems.

A virtual machine that has more virtual processors than the number of physical processor cores available on a single hardware node can be managed automatically. The NUMA scheduler accommodates such a virtual machine by having it span NUMA nodes. That is, it is split up as multiple NUMA clients, each of which is assigned to a node and then managed by the scheduler as a normal, non-spanning client. This can improve the performance of certain memory-intensive workloads with high locality. For information on configuring the behavior of this feature, see "Advanced Virtual Machine Attributes," on page 103.

ESXi 5.0 introduces support for exposing virtual NUMA topology to guest operating systems. For more information about virtual NUMA control, see "Using Virtual NUMA," on page 94.

VMware NUMA Optimization Algorithms and Settings

This section describes the algorithms and settings used by ESXi to maximize application performance while still maintaining resource guarantees.

Home Nodes and Initial Placement

When a virtual machine is powered on, ESXi assigns it a home node. A virtual machine runs only on processors within its home node, and its newly allocated memory comes from the home node as well.

Unless a virtual machine's home node changes, it uses only local memory, avoiding the performance penalties associated with remote memory accesses to other NUMA nodes.

When a virtual machine is powered on, it is assigned an initial home node so that the overall CPU and memory load among NUMA nodes remains balanced. Because internode latencies in a large NUMA system can vary greatly, ESXi determines these internode latencies at boot time and uses this information when initially placing virtual machines that are wider than a single NUMA node. These wide virtual machines are placed on NUMA nodes that are close to each other for lowest memory access latencies.

Initial placement-only approaches are usually sufficient for systems that run only a single workload, such as a benchmarking configuration that remains unchanged as long as the system is running. However, this approach is unable to guarantee good performance and fairness for a datacenter-class system that supports changing workloads. Therefore, in addition to initial placement, ESXi 5.0 does dynamic migration of virtual CPUs and memory between NUMA nodes for improving CPU balance and increasing memory locality.

Dynamic Load Balancing and Page Migration

ESXi combines the traditional initial placement approach with a dynamic rebalancing algorithm. Periodically (every two seconds by default), the system examines the loads of the various nodes and determines if it should rebalance the load by moving a virtual machine from one node to another.

This calculation takes into account the resource settings for virtual machines and resource pools to improve performance without violating fairness or resource entitlements.

The rebalancer selects an appropriate virtual machine and changes its home node to the least loaded node. When it can, the rebalancer moves a virtual machine that already has some memory located on the destination node. From that point on (unless it is moved again), the virtual machine allocates memory on its new home node and it runs only on processors within the new home node.

Rebalancing is an effective solution to maintain fairness and ensure that all nodes are fully used. The rebalancer might need to move a virtual machine to a node on which it has allocated little or no memory. In this case, the virtual machine incurs a performance penalty associated with a large number of remote memory accesses. ESXi can eliminate this penalty by transparently migrating memory from the virtual machine's original node to its new home node:

1. The system selects a page (4KB of contiguous memory) on the original node and copies its data to a page in the destination node.
2. The system uses the virtual machine monitor layer and the processor's memory management hardware to seamlessly remap the virtual machine's view of memory, so that it uses the page on the destination node for all further references, eliminating the penalty of remote memory access.

When a virtual machine moves to a new node, the ESXi host immediately begins to migrate its memory in this fashion. It manages the rate to avoid overtaxing the system, particularly when the virtual machine has little remote memory remaining or when the destination node has little free memory available. The memory migration algorithm also ensures that the ESXi host does not move memory needlessly if a virtual machine is moved to a new node for only a short period.

When initial placement, dynamic rebalancing, and intelligent memory migration work in conjunction, they ensure good memory performance on NUMA systems, even in the presence of changing workloads. When a major workload change occurs, for instance when new virtual machines are started, the system takes time to readjust, migrating virtual machines and memory to new locations. After a short period, typically seconds or minutes, the system completes its readjustments and reaches a steady state.

Transparent Page Sharing Optimized for NUMA

Many ESXi workloads present opportunities for sharing memory across virtual machines.

For example, several virtual machines might be running instances of the same guest operating system, have the same applications or components loaded, or contain common data. In such cases, ESXi systems use a proprietary transparent page-sharing technique to securely eliminate redundant copies of memory pages. With memory sharing, a workload running in virtual machines often consumes less memory than it would when running on physical machines. As a result, higher levels of overcommitment can be supported efficiently.

Transparent page sharing for ESXi systems has also been optimized for use on NUMA systems. On NUMA systems, pages are shared per-node, so each NUMA node has its own local copy of heavily shared pages. When virtual machines use shared pages, they don't need to access remote memory.

Note This default behavior is the same in all previous versions of ESX and ESXi.

Resource Management in NUMA Architectures

You can perform resource management with different types of NUMA architecture.

With the proliferation of highly multicore systems, NUMA architectures are becoming more popular as these architectures allow better performance scaling of memory intensive workloads. All modern Intel and AMD systems have NUMA support built into the processors. Additionally, there are traditional NUMA systems like the IBM Enterprise X-Architecture that extend Intel and AMD processors with NUMA behavior with specialized chipset support.

Typically, you can use BIOS settings to enable and disable NUMA behavior. For example, in AMD Opteron-based HP Proliant servers, NUMA can be disabled by enabling node interleaving in the BIOS. If NUMA is enabled, the BIOS builds a system resource allocation table (SRAT) which ESXi uses to generate the NUMA information used in optimizations. For scheduling fairness, NUMA optimizations are not enabled for systems with too few cores per NUMA node or too few cores overall. You can modify the `numa.rebalancecorestotal` and `numa.rebalancecoresnode` options to change this behavior.

Using Virtual NUMA

vSphere 5.0 introduces support for exposing virtual NUMA topology to guest operating systems, which can improve performance by facilitating guest operating system and application NUMA optimizations.

Virtual NUMA topology is available to hardware version 8 virtual machines and is enabled by default when the number of virtual CPUs is greater than eight. You can also manually influence virtual NUMA topology using advanced configuration options.

You can affect the virtual NUMA topology with two settings in the vSphere Client: number of virtual sockets and number of cores per socket for a virtual machine. If the number of cores per socket (cpuid.coresPerSocket) is greater than one, and the number of virtual cores in the virtual machine is greater than 8, the virtual NUMA node size matches the virtual socket size. If the number of cores per socket is less than or equal to one, virtual NUMA nodes are created to match the topology of the first physical host where the virtual machine is powered on.

When the number of virtual CPUs and the amount of memory used grow proportionately, you can use the default values. For virtual machines that consume a disproportionally large amount of memory, you can override the default values in one of the following ways:

- Increase the number of virtual CPUs, even if this number of virtual CPUs is not used. See "Change the Number of Virtual CPUs," on page 95.
- Use advanced options to control virtual NUMA topology and its mapping over physical NUMA topology. See "Virtual NUMA Controls," on page 95.

Change the Number of Virtual CPUs

You can configure a virtual machine that runs on an ESXi host to have up to 32 virtual CPUs.

IMPORTANT When you configure your virtual machine for multicore virtual CPU settings, you must ensure that your configuration complies with the requirements of the guest operating system EULA.

Procedure

1 In the vSphere Client, right-click the virtual machine in the inventory and select **Edit Settings**.

2 Click the **Hardware** tab and select **CPUs**.

3 Select a value from the **Number of virtual sockets** drop-down menu.

4 Select a value from the **Number of cores per socket** drop-down menu.

The resulting total number of cores is a number equal to or less than the number of logical CPUs on the host.

5 Click **OK**.

Virtual NUMA Controls

For virtual machines with disproportionately large memory consumption, you can use advanced options to manually override the default virtual CPU settings.

You can add these advanced options to the virtual machine configuration file.

Table 13-1. Advanced Options for Virtual NUMA Controls

Option	Description	Default Value
`cpuid.coresPerSocket`	Determines the number of virtual cores per virtual CPU socket. If the value is greater than 1, also determines the size of virtual NUMA nodes if a virtual machine has a virtual NUMA topology. You can set this option if you know the exact virtual NUMA topology for each physical host.	1
`numa.vcpu.maxPerVirtualNode`	If `cpuid.coresPerSocket` is too restrictive as a power of two, you can set `numa.vcpu.maxPerVirtualNode` directly. In this case, do not set `cpuid.coresPerSocket`.	8
`numa.autosize`	When you set this option, the virtual NUMA topology has the same number of virtual CPUs per virtual node as there are cores on each physical node.	FALSE

Table 13-1. Advanced Options for Virtual NUMA Controls (Continued)

Option	Description	Default Value
numa.autosize.once	When you create a virtual machine template with these settings, the settings are guaranteed to remain the same every time you subsequently power on the virtual machine. The virtual NUMA topology will be reevaluated if the configured number of virtual CPUs on the virtual machine is modified.	TRUE
numa.vcpu.min	Minimum number of virtual CPUs in a virtual machine that are required in order to generate a virtual NUMA topology.	9

NOTE When you set numa.autosize to TRUE, and if the configuration is set up manually or with a script, some guests might not be able to handle dynamic changes.

For example, a Linux application configured with the numactl system utility is set up and tested on one physical host with four cores per node. The host requires two NUMA nodes for a virtual machine with eight virtual CPUs. If the same virtual machine is run on a system with eight cores per node, which might occur during a vMotion operation, and numa.autosize is set to TRUE, only one virtual NUMA node will be created (rather than two virtual NUMA nodes). When numactl references the second virtual NUMA node, the operation will fail.

To avoid this, scripts should be intelligent enough to first query numactl --hardware. Otherwise, you must set the NUMA topology explicitly or allow the default numa.autosize.once setting to take effect.

Specifying NUMA Controls

If you have applications that use a lot of memory or have a small number of virtual machines, you might want to optimize performance by specifying virtual machine CPU and memory placement explicitly.

Specifying controls is useful if a virtual machine runs a memory-intensive workload, such as an in-memory database or a scientific computing application with a large data set. You might also want to optimize NUMA placements manually if the system workload is known to be simple and unchanging. For example, an eight-processor system running eight virtual machines with similar workloads is easy to optimize explicitly.

NOTE In most situations, the ESXi host's automatic NUMA optimizations result in good performance.

ESXi provides three sets of controls for NUMA placement, so that administrators can control memory and processor placement of a virtual machine.

The vSphere Client lets you specify the following options.

NUMA Node Affinity When you set this option, NUMA can schedule a virtual machine only on the nodes specified in the affinity.

CPU Affinity When you set this option, a virtual machine uses only the processors specified in the affinity.

Memory Affinity When you set this option, the server allocates memory only on the specified nodes.

A virtual machine is still managed by NUMA when you specify NUMA node affinity, but its virtual CPUs can be scheduled only on the nodes specified in the NUMA node affinity. Likewise, memory can be obtained only from the nodes specified in the NUMA node affinity. When you specify CPU or memory affinities, a virtual machine ceases to be managed by NUMA. NUMA management of these virtual machines is effective when you remove the CPU and memory affinity constraints.

Manual NUMA placement might interfere with ESXi resource management algorithms, which distribute processor resources fairly across a system. For example, if you manually place 10 virtual machines with processor-intensive workloads on one node, and manually place only 2 virtual machines on another node, it is impossible for the system to give all 12 virtual machines equal shares of systems resources.

Associate Virtual Machines with Specific Processors

You might be able to improve the performance of the applications on a virtual machine by pinning its virtual CPUs to fixed processors. This allows you to prevent the virtual CPUs from migrating across NUMA nodes.

Procedure

1 In the vSphere Client, right-click the virtual machine in the inventory and select **Edit Settings**.

2 Select the **Resources** tab, and select **Advanced CPU**.

3 In the Scheduling Affinity panel, set the CPU affinity to the preferred processors.

NOTE You must manually select the boxes for all processors in the NUMA node. CPU affinity is specified on a per-processor, not on a per-node, basis.

Associate Memory Allocations with Specific NUMA Nodes Using Memory Affinity

You can specify that all future memory allocations on a virtual machine use pages associated with specific NUMA nodes (also known as manual memory affinity).

NOTE Specify nodes to be used for future memory allocations only if you have also specified CPU affinity. If you make manual changes only to the memory affinity settings, automatic NUMA rebalancing does not work properly.

Procedure

1 In the vSphere Client, right-click the virtual machine in the inventory and select **Edit Settings**.

2 Select the **Resources** tab, and select **Memory**.

3 In the NUMA Memory Affinity panel, set memory affinity.

Example: Binding a Virtual Machine to a Single NUMA Node

The following example illustrates manually binding the last four physical CPUs to a single NUMA node for a two-way virtual machine on an eight-way server.

The CPUs (for example, 4, 5, 6, and 7) are the physical CPU numbers.

1 In the vSphere Client inventory panel, select the virtual machine and select **Edit Settings**.

2 Select **Options** and click **Advanced**.

3 Click the **Configuration Parameters** button.

4 In the vSphere Client, turn on CPU affinity for processors 4, 5, 6, and 7.

Then, you want this virtual machine to run only on node 1.

1 In the vSphere Client inventory panel, select the virtual machine and select **Edit Settings**.

2 Select **Options** and click **Advanced**.

3 Click the **Configuration Parameters** button.

4 In the vSphere Client, set memory affinity for the NUMA node to 1.

Completing these two tasks ensures that the virtual machine runs only on NUMA node 1 and, when possible, allocates memory from the same node.

Associate Virtual Machines with Specified NUMA Nodes

When you associate a NUMA node with a virtual machine to specify NUMA node affinity, you constrain the set of NUMA nodes on which NUMA can schedule a virtual machine's virtual CPU and memory.

NOTE When you constrain NUMA node affinities, you might interfere with the ability of the ESXi NUMA scheduler to rebalance virtual machines across NUMA nodes for fairness. Specify NUMA node affinity only after you consider the rebalancing issues.

Procedure

1 In the vSphere Client, right-click the virtual machine in the inventory and select **Edit Settings**.

2 Click the **Options** tab.

3 Select **Advanced > General**.

4 Click **Configuration Parameters**.

5 Click **Add Row** to add a new option.

6 In the Name column, enter **`numa.nodeAffinity`**.

7 In the Value column, enter the NUMA nodes where the virtual machine can be scheduled.

Use a comma-separated list for multiple nodes. For example, enter **`0,1`** to constrain the virtual machine resource scheduling to NUMA nodes 0 and 1.

8 Click **OK**.

9 Click **OK** to close the Virtual Machine Properties dialog box.

14 Advanced Attributes

You can set advanced attributes for hosts or individual virtual machines to help you customize resource management.

In most cases, adjusting the basic resource allocation settings (reservation, limit, shares) or accepting default settings results in appropriate resource allocation. However, you can use advanced attributes to customize resource management for a host or a specific virtual machine.

This chapter includes the following topics:

Set Advanced Host Attributes

You can set advanced attributes for a host.

CAUTION Changing advanced options is considered unsupported unless VMware technical support or a KB article instruct you to do so. In all other cases, changing these options is considered unsupported. In most cases, the default settings produce the optimum result.

Procedure

1. In the vSphere Client, select the host in the inventory.
2. Click the **Configuration** tab.
3. Under **Software**, click **Advanced Settings**.
4. In the Advanced Settings dialog box, select the appropriate item (for example, **CPU** or **Mem**).
5. Locate the attribute in the right panel and edit the value.
6. Click **OK**.

Advanced Memory Attributes

You can use the advanced memory attributes to customize memory resource usage.

Table 14-1. Advanced Memory Attributes

Attribute	Description	Default
Mem.SamplePeriod	Specifies the periodic time interval, measured in seconds of the virtual machine's execution time, over which memory activity is monitored to estimate working set sizes.	60
Mem.BalancePeriod	Specifies the periodic time interval, in seconds, for automatic memory reallocations. Significant changes in the amount of free memory also trigger reallocations.	15
Mem.IdleTax	Specifies the idle memory tax rate, as a percentage. This tax effectively charges virtual machines more for idle memory than for memory they are actively using. A tax rate of 0 percent defines an allocation policy that ignores working sets and allocates memory strictly based on shares. A high tax rate results in an allocation policy that allows idle memory to be reallocated away from virtual machines that are unproductively hoarding it.	75
Mem.ShareScanGHz	Specifies the maximum amount of memory pages to scan (per second) for page sharing opportunities for each GHz of available host CPU resource. For example, defaults to 4 MB/sec per 1 GHz.	4
Mem.ShareScanTime	Specifies the time, in minutes, within which an entire virtual machine is scanned for page sharing opportunities. Defaults to 60 minutes.	60
Mem.CtlMaxPercent	Limits the maximum amount of memory reclaimed from any virtual machine using the memory balloon driver (`vmmemctl`), based on a percentage of its configured memory size. Specify 0 to disable reclamation for all virtual machines.	65
Mem.AllocGuestLargePage	Enables backing of guest large pages with host large pages. Reduces TLB misses and improves performance in server workloads that use guest large pages. 0=disable.	1
Mem.AllocUsePSharePool and Mem.AllocUseGuestPool	Reduces memory fragmentation by improving the probability of backing guest large pages with host large pages. If host memory is fragmented, the availability of host large pages is reduced. 0 = disable.	15
Mem.MemZipEnable	Enables memory compression for the host. 0 = disable.	1
Mem.MemZipMaxPct	Specifies the maximum size of the compression cache in terms of the maximum percentage of each virtual machine's memory that can be stored as compressed memory.	10
LPage.LPageDefragEnable	Enables large page defragmentation. 0 = disable.	1
LPage.LPageDefragRateVM	Maximum number of large page defragmentation attempts per second per virtual machine. Accepted values range from 1 to 1024.	32
LPage.LPageDefragRateTotal	Maximum number of large page defragmentation attempts per second. Accepted values range from 1 to 10240.	256
LPage.LPageAlwaysTryForNPT	Try to allocate large pages for nested page tables (called 'RVI' by AMD or 'EPT' by Intel). If you enable this option, all guest memory is backed with large pages in machines that use nested page tables (for example, AMD Barcelona). If NPT is not available, only some portion of guest memory is backed with large pages. 0= disable.	1

Advanced NUMA Attributes

You can use the advanced NUMA attributes to customize NUMA usage.

Table 14-2. Advanced NUMA Attributes

Attribute	Description	Default
Numa.RebalancePeriod	Controls the frequency of rebalance periods, specified in milliseconds. More frequent rebalancing can increase CPU overheads, particularly on machines with a large number of running virtual machines. More frequent rebalancing can also improve fairness.	2000
Numa.MigImbalanceThreshold	The NUMA rebalancer computes the CPU imbalance between nodes, accounting for the difference between each virtual machine's CPU time entitlement and its actual consumption. This option controls the minimum load imbalance between nodes needed to trigger a virtual machine migration, in percent.	10
Numa.RebalanceEnable	Enable NUMA rebalancing and scheduling. Set this option to 0 to disable all NUMA rebalancing and initial placement of virtual machines, effectively disabling the NUMA scheduling system.	1
Numa.RebalanceCoresTotal	Specifies the minimum number of total processor cores on the host required to enable the NUMA rebalancer.	4
Numa.RebalanceCoresNode	Specifies the minimum number of processor cores per node required to enable the NUMA rebalancer. This option and Numa.RebalanceCoresTotal are useful when disabling NUMA rebalancing on small NUMA configurations (for example, two-way Opteron hosts), where the small number of total or per-node processors can compromise scheduling fairness when you enable NUMA rebalancing.	2
Numa.AutoMemAffinity	Automatically set memory affinity for virtual machines that have CPU affinity set.	1
Numa.PageMigEnable	Automatically migrate pages between NUMA nodes to improve memory locality. Page migration rates set manually are still in effect.	1

Advanced Virtual NUMA Attributes

You can use the advanced virtual NUMA attributes to customize virtual NUMA usage.

Table 14-3. Advanced NUMA Attributes

Attribute	Description	Default
cpuid.coresPerSocket	Determines the number of virtual cores per virtual CPU socket. If the value is greater than 1, also determines the size of virtual NUMA nodes if a virtual machine has a virtual NUMA topology. You can set this option if you know the exact virtual NUMA topology for each physical host.	1
numa.autosize	When you set this option, the virtual NUMA topology has the same number of virtual CPUs per virtual node as there are cores on each physical node.	FALSE

Table 14-3. Advanced NUMA Attributes (Continued)

Attribute	Description	Default
numa.autosize.once	When you create a virtual machine template with these settings, the settings are guaranteed to remain the same every time you subsequently power on the virtual machine. The virtual NUMA topology will be reevaluated if the configured number of virtual CPUs on the virtual machine is modified.	TRUE
numa.vcpu.maxPerVirtualNode	If `cpuid.coresPerSocket` is too restrictive as a power of two, you can set `numa.vcpu.maxPerVirtualNode` directly. In this case, do not set `cpuid.coresPerSocket`.	8
numa.vcpu.min	Minimum number of virtual CPUs in a virtual machine that are required in order to generate a virtual NUMA topology.	9
numa.vcpu.maxPerMachineNode	Maximum number of virtual CPUs that belong to the same virtual machine that can be scheduled on a NUMA node at the same time. Use this attribute to ensure maximum bandwidth, by forcing different NUMA clients on different NUMA nodes.	Number of cores per node on the physical host where a virtual machine is running.
numa.vcpu.maxPerClient	Maximum number of virtual CPUs in a NUMA client. A client is a group of virtual CPUs that are NUMA-managed as a single entity. By default, each virtual NUMA node is a NUMA client, but if a virtual NUMA node is larger than a physical NUMA node, a single virtual NUMA node can be backed by multiple NUMA clients.	Equals `numa.vcpu.maxPerMachineNode`
numa.nodeAffinity	Constrains the set of NUMA nodes on which a virtual machine's virtual CPU and memory can be scheduled. **NOTE** When you constrain NUMA node affinities, you might interfere with the ability of the NUMA scheduler to rebalance virtual machines across NUMA nodes for fairness. Specify NUMA node affinity only after you consider the rebalancing issues.	
numa.mem.interleave	Specifies whether the memory allocated to a virtual machine is statically interleaved across all the NUMA nodes on which its constituent NUMA clients are running and there is no virtual NUMA topology exposed.	True

Set Advanced Virtual Machine Attributes

You can set advanced attributes for a virtual machine.

Procedure

1 In the vSphere Client, right-click the virtual machine in the inventory and select **Edit Settings**.

2 Click **Options** and click **Advanced > General**.

3 Click **Configuration Parameters**.

4 In the dialog box that appears, click **Add Row** to enter a new parameter and its value.

5 Click **OK**.

Advanced Virtual Machine Attributes

You can use the advanced virtual machine attributes to customize virtual machine configuration.

Table 14-4. Advanced Virtual Machine Attributes

Attribute	Description	Default
sched.mem.maxmemctl	Maximum amount of memory reclaimed from the selected virtual machine by ballooning, in megabytes (MB). If the ESXi host needs to reclaim additional memory, it is forced to swap. Swapping is less desirable than ballooning.	-1 (Unlimited)
sched.mem.pshare.enable	Enables memory sharing for a selected virtual machine. This boolean value defaults to True. If you set it to False for a virtual machine, this turns off memory sharing.	True
sched.swap.persist	Specifies whether the virtual machine's swap files should persist or be deleted when the virtual machine is powered off. By default, the system creates the swap file for a virtual machine when the virtual machine is powered on, and deletes the swap file when the virtual machine is powered off.	False
sched.swap.dir	VMFS directory location of the virtual machine's swap file. Defaults to the virtual machine's working directory, that is, the VMFS directory that contains its configuration file. This directory must remain on a host that is accessible to the virtual machine. If you move the virtual machine (or any clones created from it), you might need to reset this attribute.	Equals `workingDir`

Index

W

Y