vSphere Resource Management Guide

ESX 4.1

ESXi 4.1

vCenter Server 4.1

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-000317-02

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

Updated Information 5

About This Book 7

1 Getting Started with Resource Management 9
What Is Resource Management? 9
Configuring Resource Allocation Settings 10
Viewing Resource Allocation Information 13
Admission Control 16

2 Managing CPU Resources 17
CPU Virtualization Basics 17
Administering CPU Resources 18

3 Managing Memory Resources 27
Memory Virtualization Basics 27
Administering Memory Resources 30

4 Managing Storage I/O Resources 41
Storage I/O Control Requirements 41
Storage I/O Control Resource Shares and Limits 42
Set Storage I/O Control Resource Shares and Limits 43
Enable Storage I/O Control 43
Troubleshooting Storage I/O Control Events 44
Set Storage I/O Control Threshold Value 44

5 Managing Resource Pools 47
Why Use Resource Pools? 48
Create Resource Pools 49
Add Virtual Machines to a Resource Pool 50
Removing Virtual Machines from a Resource Pool 51
Resource Pool Admission Control 51

6 Creating a DRS Cluster 55
Admission Control and Initial Placement 56
Virtual Machine Migration 57
DRS Cluster Requirements 59
Create a DRS Cluster 60
Set a Custom Automation Level for a Virtual Machine 61
Disable DRS 62

7 Using DRS Clusters to Manage Resources 63
Adding Hosts to a Cluster 63
Adding Virtual Machines to a Cluster 64
Remove Hosts from a Cluster 65
Removing Virtual Machines from a Cluster 66
DRS Cluster Validity 66
Managing Power Resources 71
Using Affinity Rules 75

8 Viewing DRS Cluster Information 79
Viewing the Cluster Summary Tab 79
Using the DRS Tab 81

9 Using NUMA Systems with ESX/ESXi 85
What is NUMA? 85
How ESX/ESXi NUMA Scheduling Works 86
VMware NUMA Optimization Algorithms and Settings 87
Resource Management in NUMA Architectures 88
Specifying NUMA Controls 89

A Performance Monitoring Utilities: resxtop and esxtop 93
Using the esxtop Utility 93
Using the resxtop Utility 94
Using esxtop or resxtop in Interactive Mode 94
Using Batch Mode 108
Using Replay Mode 109

B Advanced Attributes 111
Set Advanced Host Attributes 111
Set Advanced Virtual Machine Attributes 113

Index 115

Updated Information

This *vSphere Resource Management Guide* is updated with each release of the product or when necessary.

This table provides the update history of the *vSphere Resource Management Guide*.

Revision	Description
EN-000317-02	Included a point in "Multicore Processors," on page 19 section.
EN-000317-01	Changed the value of maximum logical processors per host in Enable Hyperthreading section
EN-000317-00	Initial release.

About This Book

The *vSphere Resource Management Guide* describes resource management for VMware®ESX®, ESXi, and vCenter® Server environments.

This guide 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, VMware Distributed Resource Scheduler (DRS), VMware Distributed Power Management (DPM), and how to work with them
- Advanced resource management options
- Performance considerations

The *vSphere Resource Management Guide* covers ESX®, ESXi, and vCenter® Server.

Intended Audience

This manual 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, or VMware DPM.

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

VMware Technical Publications Glossary

VMware Technical Publications provides a glossary of terms that might be unfamiliar to you. For definitions of terms as they are used in VMware technical documentation, go to http://www.vmware.com/support/pubs.

Document Feedback

VMware welcomes your suggestions for improving our documentation. If you have comments, send your feedback to docfeedback@vmware.com.

vSphere Documentation

The vSphere documentation consists of the combined VMware vCenter Server and ESX/ESXi documentation set.

Technical Support and Education Resources

The following technical support resources are available to you. To access the current version of this book and other books, go to http://www.vmware.com/support/pubs.

Online and Telephone Support

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

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

Support Offerings

To find out how VMware support offerings can help meet your business needs, go to http://www.vmware.com/support/services.

VMware Professional Services

VMware Education Services courses offer extensive hands-on labs, case study examples, and course materials designed to be used as on-the-job reference tools. Courses are available onsite, in the classroom, and live online. For onsite pilot programs and implementation best practices, VMware Consulting Services provides offerings to help you assess, plan, build, and manage your virtual environment. To access information about education classes, certification programs, and consulting services, go to http://www.vmware.com/services.

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.

This chapter includes the following topics:

- "What Is Resource Management?," on page 9
- "Configuring Resource Allocation Settings," on page 10
- "Viewing Resource Allocation Information," on page 13
- "Admission Control," on page 16

What Is Resource Management?

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.

Resource Types

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

Resource management in this context focuses primarily on CPU and memory resources. Power resource consumption can also be reduced with the VMware® Distributed Power Management (DPM) feature.

NOTE ESX/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 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 VMware® vCenter Server, 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 6, "Creating a DRS Cluster," on page 55 for more information.

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 5, "Managing Resource Pools," on page 47.

An ESX/ESXi host allocates each virtual machine a portion of the underlying hardware resources based on a number of factors:

- Total available resources for the ESX/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

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 I/O 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 ESX/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.

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.

Table 1-1 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 VCPUs and 16 GB of memory.

Table 1-1. 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. ESX/ESXi supports up to eight virtual CPUs per virtual machine. This is also called eight-way SMP support.

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 VCPUs. 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 ESX/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.

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 ESX/ESXi environment.

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

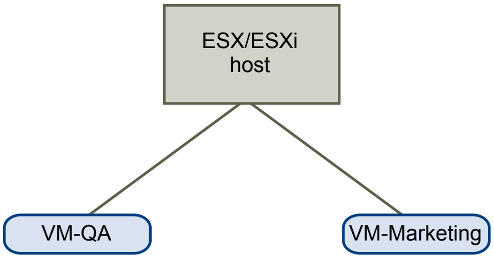
- If you expect frequent changes to the total available resources, use **Shares** to allocate resources fairly across virtual machines. If you use **Shares**, and you upgrade the host, for example, each virtual machine stays at the same priority (keeps the same number of shares) even though each share 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 ESX/ESXi host, you have created two new virtual machines—one each for your QA (VM-QA) and Marketing (VM-Marketing) departments.

Figure 1-1. 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.
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.

Viewing Resource Allocation Information

Using the vSphere Client, you can select a cluster, resource pool, standalone host, or a virtual machine in the inventory panel and view how its resources are being allocated by clicking the **Resource Allocation** tab.

This information can then be used to help inform your resource management decisions.

Cluster Resource Allocation Tab

The **Resource Allocation** tab is available when you select a cluster from the inventory panel.

The **Resource Allocation** tab displays information about the CPU and memory resources in the cluster.

CPU Section

The following information about CPU resource allocation is shown.

Table 1-2. CPU Resource Allocation

Field	Description
Total Capacity	Guaranteed CPU allocation, in megahertz (MHz), reserved for this object.
Reserved Capacity	Number of megahertz (MHz) of the reserved allocation that this object is using.
Available Capacity	Number of megahertz (MHz) not reserved.

Memory Section

The following information about memory resource allocation is shown.

Table 1-3. Memory Resource Allocation

Field	Description
Total Capacity	Guaranteed memory allocation, in megabytes (MB), for this object.
Reserved Capacity	Number of megabytes (MB) of the reserved allocation that this object is using.
Available Capacity	Number of megabytes (MB) not reserved.

NOTE Reservations for the root resource pool of a cluster that is enabled for VMware HA might be larger than the sum of the explicitly-used resources in the cluster. These reservations not only reflect the reservations for the running virtual machines and the hierarchically-contained (child) resource pools in the cluster, but also the reservations needed to support VMware HA failover. See the *vSphere Availability Guide*.

The **Resource Allocation** tab also displays a chart showing the resource pools and virtual machines in the DRS cluster with CPU, memory, or storage I/O resource usage information.

To view CPU or memory information, click the **CPU** button or **Memory** button, respectively.

Table 1-4. CPU or Memory Usage Information

Field	Description
Name	Name of the object.
Reservation - MHz	Guaranteed minimum CPU allocation, in megahertz (MHz), reserved for this object.
Reservation - MB	Guaranteed minimum memory allocation, in megabytes (MB), for this object.
Limit - MHz	Maximum amount of CPU the object can use.
Limit - MB	Maximum amount of memory the object can use.
Shares	A relative metric for allocating CPU or memory capacity. The values Low, Normal, High, and Custom are compared to the sum of all shares of all virtual machines in the enclosing resource pool.
Shares Value	Actual value based on resource and object settings.
% Shares	Percentage of cluster resources assigned to this object.
Worst Case Allocation	The amount of (CPU or memory) resource that is allocated to the virtual machine based on user-configured resource allocation policies (for example, reservation, shares and limit), and with the assumption that all virtual machines in the cluster consume their full amount of allocated resources. The values for this field must be updated manually by pressing the F5 key.
Type	Type of reserved CPU or memory allocation, either Expandable or Fixed.

To view storage I/O information, click the **Storage** button.

Table 1-5. Storage I/O Resource Usage Information

Field	Description
Name	Name of the object.
Disk	Name of the virtual machine's hard disk.
Datastore	Name of the datastore.
Limit - IOPS	Upper bound for storage resources that can be allocated to a virtual machine.
Shares	A relative metric for allocating storage I/O resources. The values Low, Normal, High, and Custom are compared to the sum of all shares of all virtual machines in the enclosing resource pool.
Shares Value	Actual value based on resource and object settings.
Datastore % Shares	Percentage of datastore resources assigned to this object.

Virtual Machine Resource Allocation Tab

A **Resource Allocation** tab is available when you select a virtual machine from the inventory panel.

The **Resource Allocation** tab displays information about the CPU and memory resources for the selected virtual machine.

CPU Section

These bars display the following information about host CPU usage:

Table 1-6. Host CPU

Field	Description
Consumed	Actual consumption of CPU resources by the virtual machine.
Active	Estimated amount of resources consumed by virtual machine if there is no resource contention. If you have set an explicit limit, this amount does not exceed that limit.

Table 1-7. Resource Settings

Field	Description
Reservation	Guaranteed minimum CPU allocation for this virtual machine.
Limit	Maximum CPU allocation for this virtual machine.
Shares	CPU shares for this virtual machine.
Worst Case Allocation	The amount of CPU resources allocated to the virtual machine based on user-configured resource allocation policies (for example, reservation, shares and limit), and with the assumption that all virtual machines in the cluster consume their full amount of allocated resources.

Memory Section

These bars display the following information about host memory usage:

Table 1-8. Host Memory

Field	Description
Consumed	Actual consumption of physical memory that has been allocated to the virtual machine.
Overhead Consumption	Amount of consumed memory being used for virtualization purposes. Overhead Consumption is included in the amount shown in Consumed.

These bars display the following information about guest memory usage:

Table 1-9. Guest Memory

Field	Description
Private	Amount of memory backed by host memory and not being shared.
Shared	Amount of memory being shared.
Swapped	Amount of memory reclaimed by swapping.
Compressed	Amount of memory stored in the virtual machine's compression cache.
Ballooned	Amount of memory reclaimed by ballooning.
Unaccessed	Amount of memory never referenced by the guest.
Active	Amount of memory recently accessed.

Table 1-10. Resource Settings

Field	Description
Reservation	Guaranteed memory allocation for this virtual machine.
Limit	Upper limit for this virtual machine’s memory allocation.
Shares	Memory shares for this virtual machine.
Configured	User-specified guest physical memory size.
Worst Case Allocation	The amount of memory resources allocated to the virtual machine based on user-configured resource allocation policies (for example, reservation, shares and limit), and with the assumption that all virtual machines in the cluster consume their full amount of allocated resources.
Overhead Reservation	The amount of memory that is being reserved for virtualization overhead.

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 VMware 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.

Managing CPU Resources 2

ESX/ESXi hosts support CPU virtualization. When you utilize CPU virtualization, you should understand how it works, its different types, and processor-specific behavior.

You also need to be aware of the performance implications of CPU virtualization.

This chapter includes the following topics:

- "CPU Virtualization Basics," on page 17
- "Administering CPU Resources," on page 18

CPU Virtualization Basics

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. 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 ESX/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 ESX/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.

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 (such as Intel VT and AMD SVM) 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, this can slow down 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 *VMware vSphere Datacenter Administration Guide* 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.

ESX/ESXi supports up to eight 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.

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. ESX/ESXi supports virtual machines with up to eight virtual processors.

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 ESX/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 an ESX/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 an ESX/ESXi 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 ESX 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 ESX 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 ESX 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 ESX 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 ESX/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"`.

To find out if a change in this parameter helps with performance, please do proper load testing. You cannot easily predict the effect of a change in this parameter. If you do not see a performance boost after changing the parameter, you have to revert the parameter to its default value.

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 ESX/ESXi Hosts

An ESX/ESXi host enabled for hyperthreading should behave similarly to a host without hyperthreading. You might need to consider certain factors if you enable hyperthreading, however.

ESX/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 ESX/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.

Some Intel processors, for example Xeon 5500 processors or those based on the P4 microarchitecture, support hyperthreading. Consult your system documentation to determine whether your CPU supports hyperthreading. ESX/ESXi cannot enable hyperthreading on a system with more than 32 physical cores, because ESX/ESXi has a logical limit of 128 CPUs.

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 your ESX/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 now 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.

The choices for this mode are listed in Table 2-1.

Table 2-1. 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, an ESX/ESXi host 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.

Certain types of self-modifying code, for example, 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 ESX/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 on which 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, ESX/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 ESX/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 an ESX/ESXi host's ability to schedule virtual machines on multicore or hyperthreaded processors to take full advantage of resources shared on such processors.

Using CPU Power Management Policies

ESX/ESXi provides up to four power management policies. You choose a power management policy depending on a host's hardware characteristics and BIOS support, which allows you to configure servers for specific levels of power efficiency and performance.

To improve CPU power efficiency, ESX/ESXi can take advantage of performance states (also known as P-states) to dynamically adjust CPU frequency to match the demand of running virtual machines. 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). ESX/ESXi attempts to adjust CPU frequencies so that virtual machine performance is not affected.

When a CPU is idle, ESX/ESXi can take advantage of power states (also known as C-states) and put the CPU in a deep sleep state. As a result, the CPU consumes as little power as possible and can quickly resume from sleep when necessary.

Table 2-2 shows the available power management policies. You select a policy for a host using the vSphere Client. If you do not select a policy, ESX/ESXi uses High Performance by default.

Table 2-2. 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 (Default)	VMkernel detected certain power management features, but will not use them unless the BIOS requests them for power capping or thermal events.
Balanced Performance	VMkernel is using all available power management features to reduce host energy consumption without compromising performance.
Low Power	VMkernel aggressively uses available power management features to reduce host energy consumption at the risk of lower performance.
Custom	VMkernel implements specific user-defined power management features based on the values of advanced configuration parameters. The parameters are set in the vSphere Client Advanced Settings dialog box.

Select a CPU Power Management Policy

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

Prerequisites

ESX/ESXi supports the Enhanced Intel SpeedStep and Enhanced AMD PowerNow! CPU power management technologies. For the VMkernel to take advantage of the power management capabilities provided by these technologies, you must enable power management, sometimes called Demand-Based Switching (DBS), in the BIOS.

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.

Managing Memory Resources 3

All modern operating systems provide support for virtual memory, allowing software to use more memory than the machine physically has. Similarly, the ESX/ESXi hypervisor provides support for overcommitting virtual machine memory, where the amount of guest memory configured for all virtual machines might be larger than the amount of physical host memory.

If you intend to use memory virtualization, you should understand how ESX/ESXi hosts allocate, tax, and reclaim memory. Also, you need to be aware of the memory overhead incurred by virtual machines.

This chapter includes the following topics:

- "Memory Virtualization Basics," on page 27
- "Administering Memory Resources," on page 30

Memory Virtualization Basics

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

The VMkernel manages all machine memory. (An exception to this is the memory that is allocated to the service console in ESX.) 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 VMM requires some memory and a dynamic overhead memory for its code and data.

The virtual memory space is divided into blocks, typically 4KB, called pages. The physical memory is also divided into blocks, also typically 4KB. When physical memory is full, the data for virtual pages that are not present in physical memory are stored on disk. ESX/ESXi also provides support for large pages (2 MB). See "Advanced Memory Attributes," on page 111.

Virtual Machine Memory

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

Configured Size

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 ESX/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 ESX/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 ESX/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 ESX/ESXi hosts to improve virtual machine performance when memory is overcommitted as described in "Memory Compression," on page 37.

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. ESX/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

ESX/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 ESX/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.

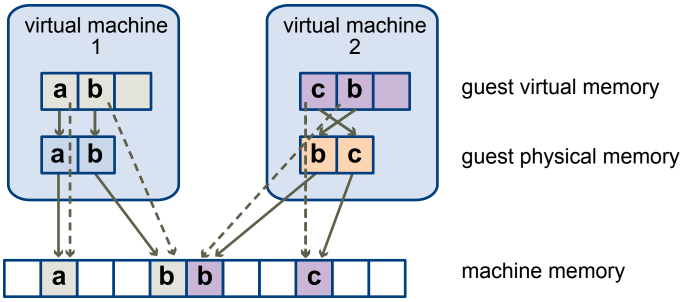
Hardware-Assisted Memory Virtualization

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 in Figure 3-1 illustrates the ESX/ESXi implementation of memory virtualization.

Figure 3-1. ESX/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, ESX/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.

An ESX/ESXi 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

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 ESX/ESXi hosts reclaim memory.

When administering memory resources, you can specify memory allocation. If you do not customize memory allocation, the ESX/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 GUI allows you to connect to an ESX/ESXi host or a vCenter Server system.
- Use advanced settings.
- Use the vSphere SDK for scripted memory allocation.

View Memory Allocation Information

You can use the vSphere Client to view information about current memory allocations.

You can view the information about the total memory and memory available to virtual machines. In ESX, you can also view memory assigned to the service console.

Procedure

1. In the vSphere Client, select a host and click the **Configuration** tab.
2. Click **Memory**.

You can view the information shown in "Host Memory Information," on page 31.

Host Memory Information

The vSphere Client shows information about host memory allocation.

The host memory fields are described in Table 3-1.

Table 3-1. Host Memory Information

Field	Description
Total	Total physical memory for this host.
System	Memory used by the ESX/ESXi system. ESX/ESXi uses at least 50MB of system memory for the VMkernel, and additional memory for device drivers. This memory is allocated when the ESX/ESXi is loaded and is not configurable. The actual required memory for the virtualization layer depends on the number and type of PCI (peripheral component interconnect) devices on a host. Some drivers need 40MB, which almost doubles base system memory. The ESX/ESXi host also attempts to keep some memory free at all times to handle dynamic allocation requests efficiently. ESX/ESXi sets this level at approximately six percent of the memory available for running virtual machines. An ESXi host uses additional system memory for management agents that run in the service console of an ESX host.
Virtual Machines	Memory used by virtual machines running on the selected host. Most of the host's memory is used for running virtual machines. An ESX/ESXi host manages the allocation of this memory to virtual machines based on administrative parameters and system load. The amount of physical memory the virtual machines can use is always less than what is in the physical host because the virtualization layer takes up some resources. For example, a host with a dual 3.2GHz CPU and 2GB of memory might make 6GHz of CPU power and 1.5GB of memory available for use by virtual machines.
Service Console	Memory reserved for the service console. Click **Properties** to change how much memory is available for the service console. This field appears only in ESX. ESXi does not provide a service console.

Understanding Memory Overhead

Virtualization of memory resources has some associated overhead.

ESX/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 ESX/ESXi host for its own code and data structures, beyond the memory allocated to each virtual machine.

ESX/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 and (for ESX only) the service console.
- Additional overhead for each virtual machine.

For ESX, the service console typically uses 272MB and the VMkernel uses a smaller amount of memory. The amount depends on the number and size of the device drivers that are being used.

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.

ESX/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 incur overhead memory. You should be aware of the amount of this overhead.

Table 3-2 lists the overhead memory (in MB) for each number of VCPUs.

Table 3-2. Overhead Memory on Virtual Machines

Memory (MB)	1 VCPU	2 VCPUs	3 VCPUs	4 VCPUs	5 VCPUs	6 VCPUs	7 VCPUs	8 VCPUs
256	113.17	159.43	200.53	241.62	293.15	334.27	375.38	416.50
512	116.68	164.96	206.07	247.17	302.75	343.88	385.02	426.15
1024	123.73	176.05	217.18	258.30	322.00	363.17	404.34	445.52
2048	137.81	198.20	239.37	280.53	360.46	401.70	442.94	484.18
4096	165.98	242.51	283.75	324.99	437.37	478.75	520.14	561.52
8192	222.30	331.12	372.52	413.91	591.20	632.86	674.53	716.19
16384	334.96	508.34	550.05	591.76	900.44	942.98	985.52	1028.07
32768	560.27	863.41	906.06	948.71	1515.75	1559.42	1603.09	1646.76
65536	1011.21	1572.29	1616.19	1660.09	2746.38	2792.30	2838.22	2884.14
131072	1912.48	2990.05	3036.46	3082.88	5220.24	5273.18	5326.11	5379.05
262144	3714.99	5830.60	5884.53	5938.46	10142.83	10204.79	10266.74	10328.69

How ESX/ESXi Hosts Allocate Memory

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

For example, a 1GB virtual machine might have the default limit (unlimited) or a user-specified limit (for example 2GB). In both cases, the ESX/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.

An ESX/ESXi 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 — ESX/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 —ESX/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 111.

Memory Tax for Idle Virtual Machines

If a virtual machine is not actively using all of its currently allocated memory, ESX/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 ESX 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 111.

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

Memory Reclamation

ESX/ESXi hosts can reclaim memory from virtual machines.

An ESX/ESXi 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.

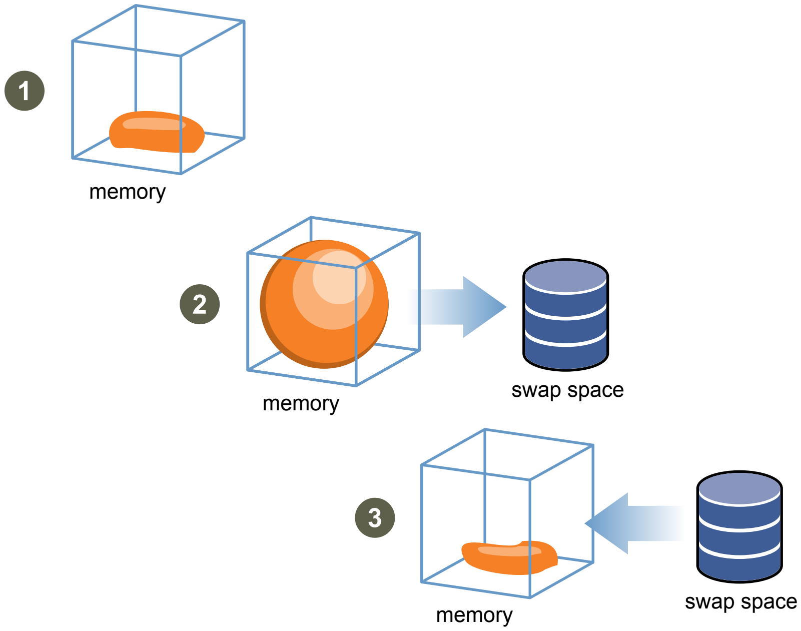
- ESX/ESXi systems use a memory balloon driver (`vmmemctl`), loaded into the guest operating system running in a virtual machine. See "Memory Balloon Driver," on page 33.
- ESX/ESXi systems page from a virtual machine to a server swap file without any involvement by the guest operating system. Each virtual 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. See Figure 3-2.

Figure 3-2. 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 113.

Using Swap Files

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

ESX/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 ESX/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 VMware 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. Right-click the cluster in the vSphere Client inventory panel and click **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. Select one of the cluster's hosts in the vSphere Client inventory panel and click the **Configuration** tab.
5. Select **Virtual Machine Swapfile Location**.
6. Click the **Swapfile Datastore** tab.
7. From the list provided, select the local datastore to use and click **OK**.
8. Repeat Step 4 through Step 7 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. Select the host in the vSphere Client inventory panel and click the **Configuration** tab.
2. Select **Virtual Machine Swapfile Location**.
3. In the **Swapfile location** tab of the Virtual Machine Swapfile Location dialog box, select **Store the swapfile in the swapfile datastore**.
4. Click the **Swapfile Datastore** tab.
5. From the list provided, select the local datastore to use 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 ESX/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 ESX/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.

Delete Swap Files

If an ESX/ESXi host fails, and that host had running virtual machines that were using swap files, those swap files continue to exist and take up disk space even after the ESX/ESXi host restarts. These swap files can consume many gigabytes of disk space so ensure that you delete them properly.

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.

Sharing Memory Across Virtual Machines

Many ESX/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, an ESX/ESXi 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 113.

ESX/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

ESX/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, ESX/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 ESX/ESXi allows you to overcommit memory without significantly hindering performance. When a virtual page needs to be swapped, ESX/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. Select the host in the vSphere Client inventory panel and click the **Configuration** tab.
2. Under Software, select **Advanced Settings**.
3. In the left pane, select **Mem** and locate Mem.MemZipEnable.
4. Enter 1 to enable or enter 0 to disable the memory compression cache.
5. 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, ESX/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, ESX/ESXi uses the default value of 10 percent.

Procedure

1. Select the host in the vSphere Client inventory panel and click the **Configuration** tab.
2. Under Software, select **Advanced Settings**.
3. 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.
4. 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.
5. Click **OK**.

Measuring and Differentiating Types of Memory Usage

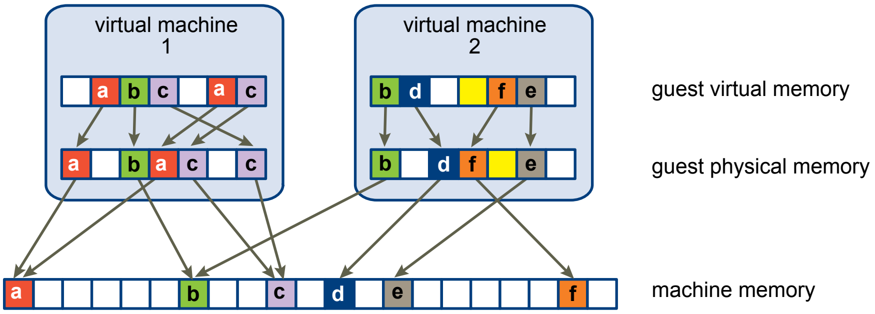
The **Performance** tab of the vSphere Client displays a number of metrics that can be used to analyze memory usage.

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 an ESX/ESXi host). To measure machine memory, however, use Memory Consumed (for a virtual machine) or Memory Shared Common (for an ESX/ESXi 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 an ESX/ESXi host differ.

Consider the example in the following figure. Two virtual machines are running on an ESX/ESXi host. Each block represents 4 KB of memory and each color/letter represents a different set of data on a block.

Figure 3-3. 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 ESX/ESXi 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.

Managing Storage I/O Resources 4

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, ESX/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.

This chapter includes the following topics:

- "Storage I/O Control Requirements," on page 41
- "Storage I/O Control Resource Shares and Limits," on page 42
- "Set Storage I/O Control Resource Shares and Limits," on page 43
- "Enable Storage I/O Control," on page 43
- "Troubleshooting Storage I/O Control Events," on page 44
- "Set Storage I/O Control Threshold Value," on page 44

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 and iSCSI-connected storage. NFS datastores and Raw Device Mapping (RDM) are 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 10 MB 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 Performance Charts online help.

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 MBs 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, ESX/ESXi monitors datastore latency and adjusts the I/O load sent to it, if datastore average latency exceeds the threshold.

Procedure

1 Select a datastore in the vSphere Client inventory 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.

Troubleshooting Storage I/O Control Events

In the vSphere Client, the alarm **Non-VI workload detected on the datastore** is triggered when vCenter Server detects that a workload from a non-vSphere host might be affecting performance.

An anomaly might be detected for one of the following reasons.

- The datastore is Storage I/O Control-enabled, but it cannot be fully controlled by Storage I/O Control because of the external workload. This can occur if the Storage I/O Control-enabled datastore is connected to an ESX/ESXi host that does not support Storage I/O Control. Ensure that all ESX/ESXi hosts that are connected to the datastore support Storage I/O Control.
- The datastore is Storage I/O Control-enabled and one or more of the hosts to which the datastore connects is not managed by vCenter Server. Ensure that all hosts to which the datastore is connected are managed by vCenter Server.
- The array is shared with non-vSphere workloads or the array is performing system tasks such as replication.

  vCenter Server does not reduce the total amount of I/O sent to the array, but continues to enforce shares.

For more information on alarms, see the *VMware vSphere Datacenter Administration Guide*.

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. Select a datastore in the vSphere Client inventory 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 5

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.

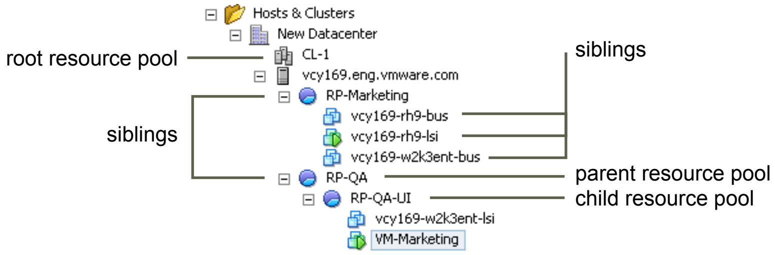
Each standalone host and each DRS cluster has an (invisible) root resource pool that groups the resources of that host or cluster. The root resource pool does not appear because the resources of the host (or cluster) and the root resource pool are always the same.

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 Figure 5-1, 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 5-1. 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?," on page 48
- "Create Resource Pools," on page 49
- "Add Virtual Machines to a Resource Pool," on page 50
- "Removing Virtual Machines from a Resource Pool," on page 51
- "Resource Pool Admission Control," on page 51

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.

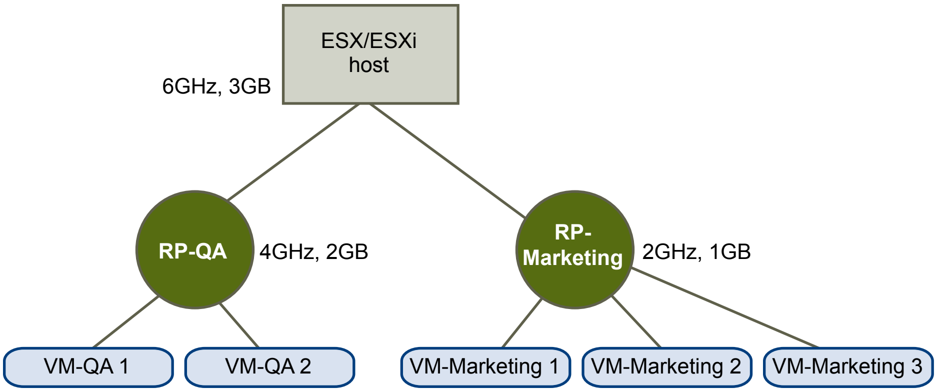
- Flexible hierarchical organization—Add, remove, or reorganize resource pools or change resource allocations as needed.
- Isolation between pools, sharing within pools—Top-level administrators can make a pool of resources available to a department-level administrator. Allocation changes that are internal 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.

This scenario is shown in Figure 5-2. The numbers show the effective allocations to the resource pools.

Figure 5-2. Allocating Resources to Resource Pools



Create Resource Pools

You can create a child resource pool of any ESX/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. You can create child resource pools of the cluster if the cluster is enabled for DRS.

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.

Procedure

1. Select the intended parent and select **File > New > Resource Pool** (or click **New Resource Pool** in the Commands panel of the **Summary** tab).
2. In the Create Resource Pool dialog box, provide the required information for your resource pool.
3. After you have made all selections, click **OK**.

   vCenter Server creates the resource pool and displays it in the inventory panel. A yellow triangle appears if any of the selected values are not legal values because of limitations on total available CPU and memory.

After a resource pool has been created, 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.

Resource Pool Attributes

You can use resource allocation settings to manage a resource pool.

Table 5-1 is a summary of the attributes that you can specify for a resource pool.

Table 5-1. Resource Pool Attributes

Field	Description
Name	Name of the new resource pool.
Shares	Number of CPU or memory shares the resource pool has with respect to the parent's total. Sibling resource pools share resources according to their relative share values bounded by the reservation and limit. You can select **Low**, **Normal**, or **High**, or select **Custom** to specify a number that assigns a share value.
Reservation	Guaranteed CPU or memory allocation for this resource pool. 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. Defaults to 0.
Expandable Reservation	Indicates whether expandable reservations are considered during admission control. If you power on a virtual machine in this resource pool, and the reservations of the virtual machines combined are larger than the reservation of the resource pool, the resource pool can use resources from its parent or ancestors if this check box is selected (the default).
Limit	Upper limit for the amount of CPU or memory the host makes available to this resource pool. Default is **Unlimited**. To specify a limit, deselect the **Unlimited**check box.

Resource Pool Creation Example

This procedure example demonstrates how you can create a resource pool with the ESX/ESXi host as the parent resource.

Assume that you have an ESX/ESXi 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 procedure demonstrates how to create a resource pool, with the ESX/ESXi host as the parent resource.

Procedure

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** to exit.

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.

Change Resource Pool Attributes

After a resource pool is created, you can change its attributes.

Procedure

1 Select the resource pool in the vSphere Client inventory panel.

2 In the **Summary** tab Command panel, select **Edit Settings**.

3 In the Edit Settings dialog box, you can change all attributes of the selected resource pool.

Add Virtual Machines 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 Select the preexisting virtual machine from any location 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 object you want.

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 explains the situation. The error dialog box compares available and requested resources, so you can consider whether an adjustment might resolve the issue.

Removing Virtual Machines 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.

Moving a Virtual Machine to a Different Resource Pool

You can drag the virtual machine to another resource pool. You do not need to power off a virtual machine if you only move 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.

Removing a Virtual Machine from the Inventory or Deleting it from the Disk

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

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

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, check the CPU **Unreserved** and memory **Unreserved** fields in the resource pool's **Resource Allocation** tab to determine whether sufficient resources are available.

How **Unreserved** CPU and memory are computed and whether actions are performed depends on the **Reservation Type**, as described in Table 5-2.

Table 5-2. 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.

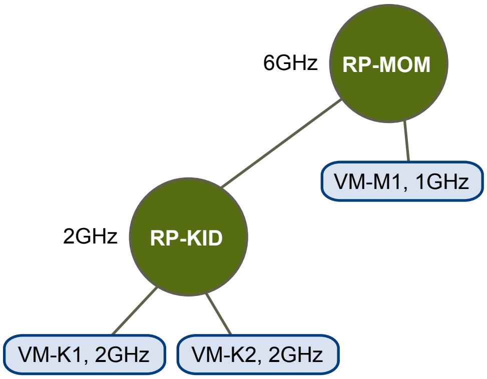
Expandable Reservations Example 2

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

Assume the following scenario (shown in Figure 5-3).

- 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.

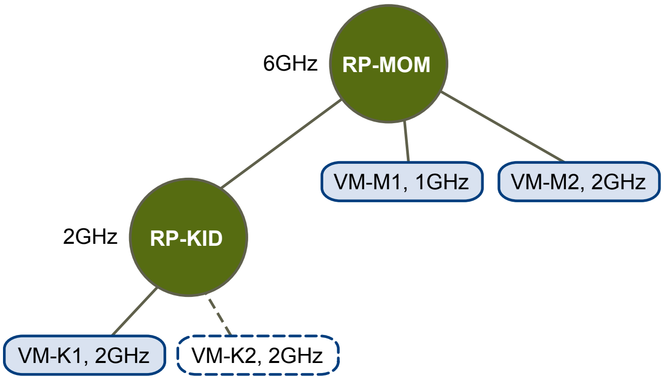
Figure 5-3. Admission Control with Expandable Resource Pools: Successful Power-On



Now, consider another scenario with VM-M1 and VM-M2 (shown in Figure 5-4):

- 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 5-4. Admission Control with Expandable Resource Pools: Power-On Prevented



Creating a DRS Cluster 6

A DRS cluster is a collection of ESX/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 57. 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 56

Power management

When the VMware Distributed Power Management 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 71.

Affinity Rules

You can control the placement of virtual machines on hosts within a cluster, by assigning affinity rules. See "Using Affinity Rules," on page 75.

NOTE If you are using VMware Fault Tolerance in your cluster, DRS provides load balancing and initial placement recommendations for fault tolerant virtual machines if Enhanced vMotion Compatibility (EVC) is also enabled. If EVC is not enabled, DRS is unable to load balance those virtual machines and also treats Primary VMs as DRS disabled and Secondary VMs as fully automated.

This chapter includes the following topics:

- "Admission Control and Initial Placement," on page 56
- "Virtual Machine Migration," on page 57
- "DRS Cluster Requirements," on page 59

- "Create a DRS Cluster," on page 60
- "Set a Custom Automation Level for a Virtual Machine," on page 61
- "Disable DRS," on page 62

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.

For more information about DRS recommendations and applying them, see "DRS Recommendations Page," on page 81.

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.

Group Power-On Example

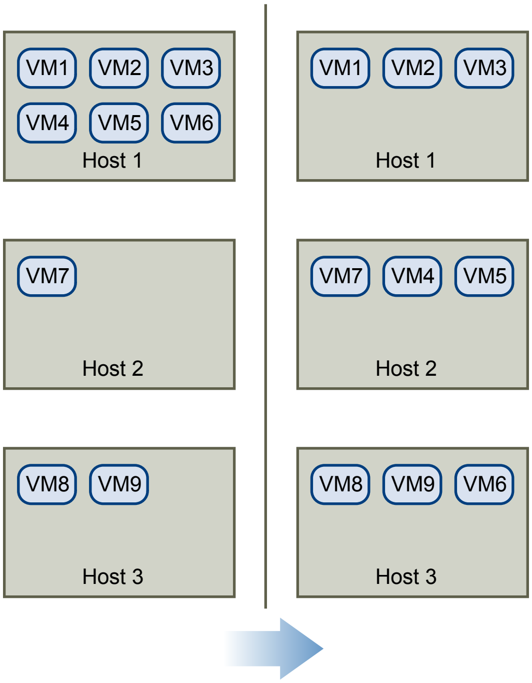
The user selects three virtual machines in the same datacenter for a group power-on attempt. The first two virtual machines (VM1 and VM2) are in the same DRS cluster (Cluster1), while the third virtual machine (VM3) is on a standalone host. VM1 is in automatic mode and VM2 is in manual mode. For this 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, see Figure 6-1. The three hosts on the left side of this 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 6-1. 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 VMware Distributed Power Management 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

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 *iSCSI SAN Configuration Guide* and the *Fibre Channel SAN Configuration Guide* for more information about SAN and the *ESX Configuration Guide* or *ESXi Configuration Guide* for information about other shared storage.

Shared VMFS Volume

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.
- Set access mode for the shared VMFS to public.
- 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

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 ESX/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 such as Xeon MP and Opteron 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 *VMware vSphere Datacenter Administration Guide*.

- 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

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:

- The virtual machine configuration file for ESX/ESXi hosts must reside on a VMware Virtual Machine File System (VMFS).
- 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 DRS (or VMware 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 **VMware DRS** box.

   You can also enable the VMware HA feature by clicking **VMware 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 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 63 and "Removing Virtual Machines from a Cluster," on page 66.

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.

Procedure

1 Select the cluster in the vSphere Client inventory.

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

3 In the Cluster Settings dialog box, under **VMware DRS** select **Virtual Machine Options**.

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

5 Select an individual virtual machine, or select multiple virtual machines.

6 Right-click and select an automation mode.

7 Click **OK**.

NOTE Other VMware products or features, such as VMware vApp and VMware 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 (see "Using Affinity Rules," on page 75) 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 VMware DRS**check box.

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

Using DRS Clusters to Manage Resources

7

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 VMware HA.

This chapter includes the following topics:

- "Adding Hosts to a Cluster," on page 63
- "Adding Virtual Machines to a Cluster," on page 64
- "Remove Hosts from a Cluster," on page 65
- "Removing Virtual Machines from a Cluster," on page 66
- "DRS Cluster Validity," on page 66
- "Managing Power Resources," on page 71
- "Using Affinity Rules," on page 75

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.

Remove Hosts from a Cluster

You can remove hosts from a cluster.

Prerequisites

Before you remove a host from a DRS cluster, consider the issues involved.

- 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.

Procedure

1. Select the host and select **Enter Maintenance Mode** from the right-click menu.
2. After the host is in maintenance mode, drag it to a different inventory location, either the top-level datacenter or a different 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.

After you move the host, you can:

- Remove the host from vCenter Server. (Select **Remove** from the right-click menu.)
- Run the host as a standalone host under vCenter Server. (Select **Exit Maintenance Mode** from the right-click menu.)
- Move the host into another cluster.

Using 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 VMware HA failover level would be violated after the host enters the requested mode.

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 VMware 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.

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.

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 an ESX/ESXi 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 an ESX/ESXi 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.

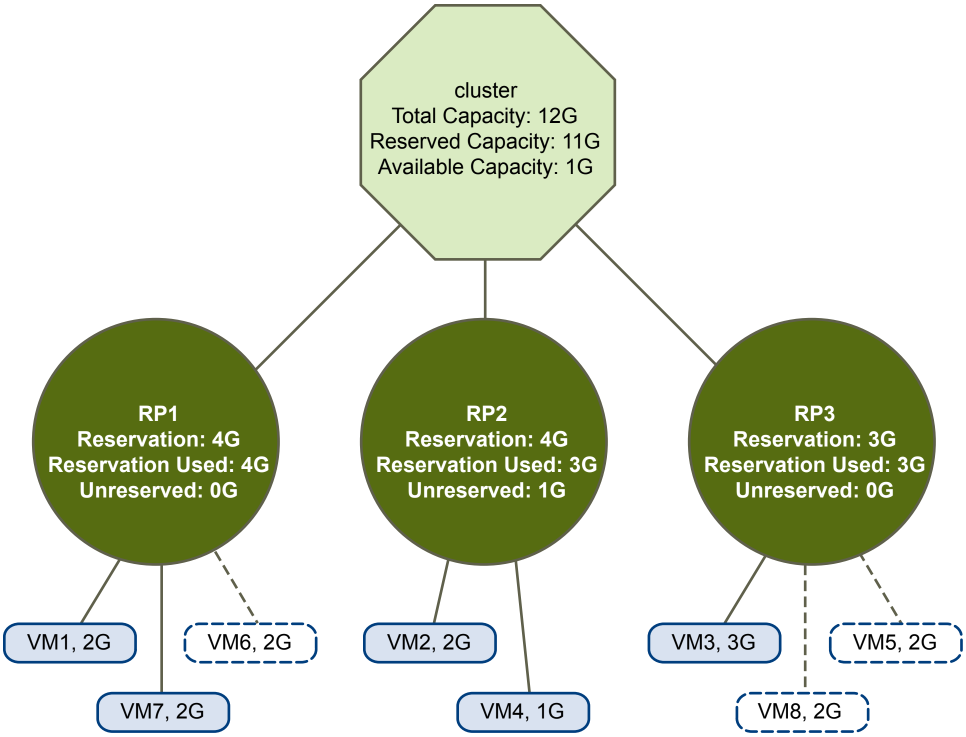
- Nonexpandable resource pools: Reservation minus reservation used.
- Expandable resource pools: (Reservation minus reservation used) plus any unreserved resources that can be borrowed from its ancestor resource pools.

Valid DRS Clusters

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

Figure 7-1 shows an example of a valid cluster with fixed resource pools and how its CPU and memory resources are computed.

Figure 7-1. Valid Cluster with Fixed Resource Pools



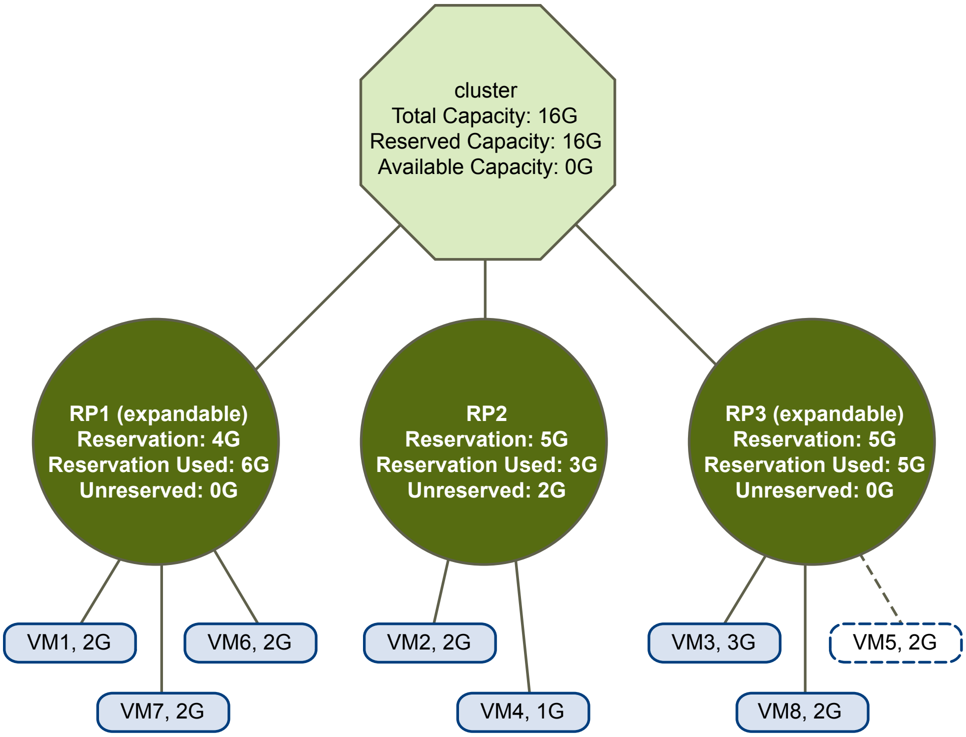
The cluster has the following characteristics:

- 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.

Figure 7-2 shows an example of a valid cluster with some resource pools (RP1 and RP3) using reservation type **Expandable**.

Figure 7-2. 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.

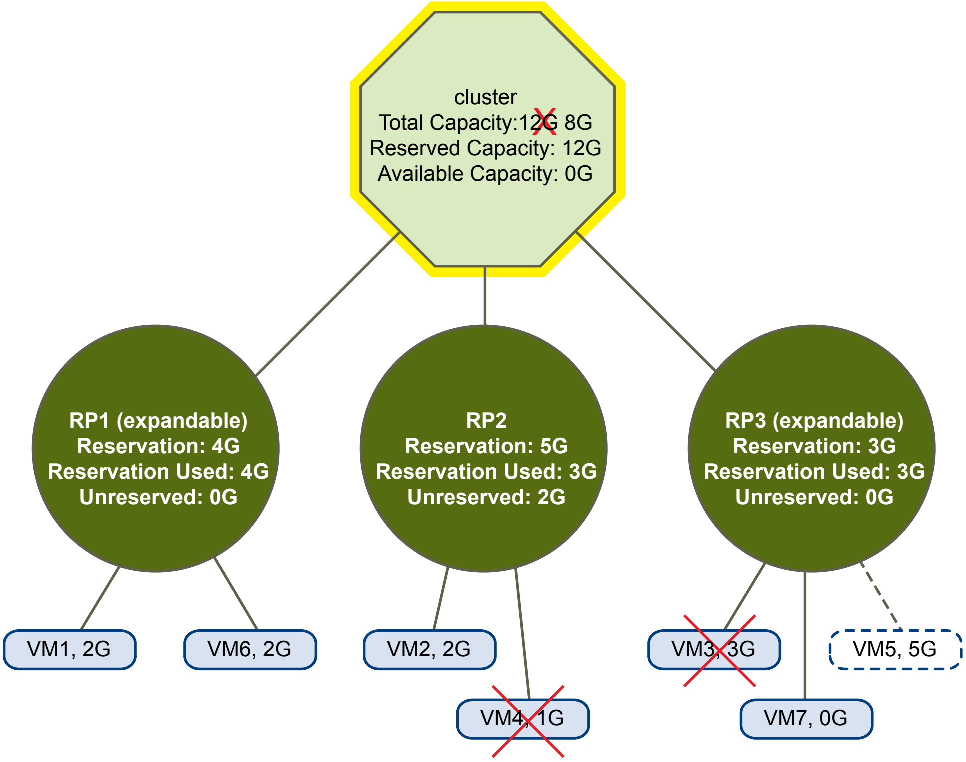
Overcommitted DRS Clusters

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.

Consider the following example, as shown in Figure 7-3.

Figure 7-3. 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.

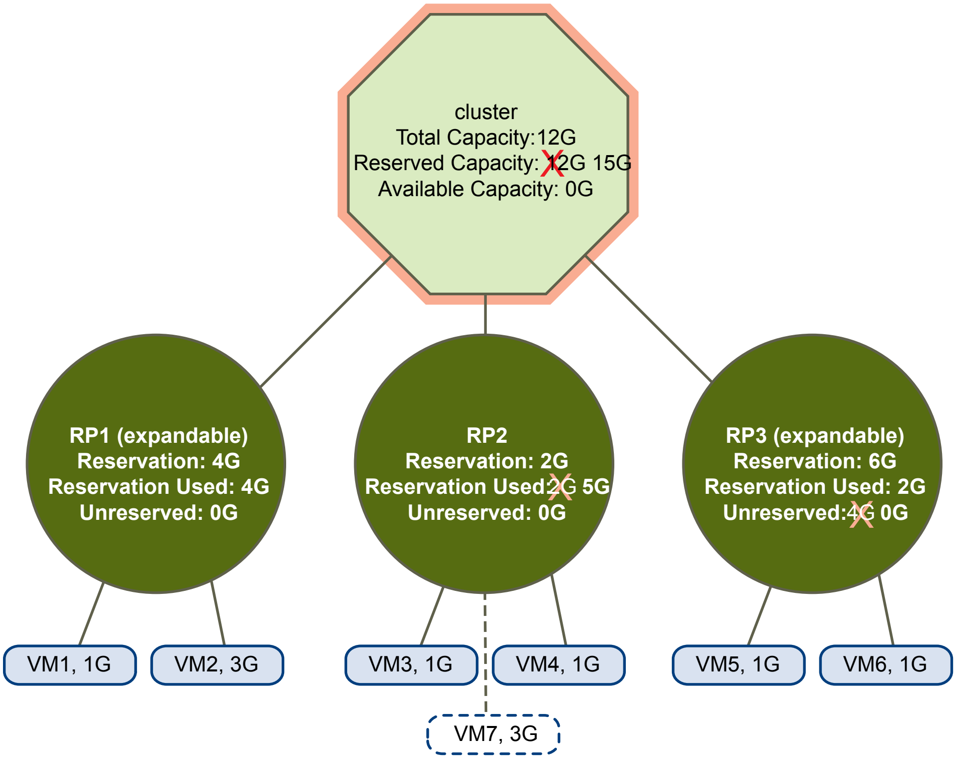
The total amount of resources in the cluster does not affect whether the cluster is red. A cluster can be red, even if enough resources exist at the root level, if there is an inconsistency at a child level.

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.

As is shown in the example in Figure 7-4, 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.

Figure 7-4. Red Cluster



Managing Power Resources

The VMware 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.

VMware 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, VMware 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, VMware DPM considers not only current demand, but it also honors any user-specified virtual machine resource reservations.

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

VMware 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 VMware 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 VMware 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.

VMware 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 VMware DPM

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

CAUTION Ensure that any host being added to a VMware 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, VMware 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, VMware DPM does not consider that host a candidate for being powered off.

Enabling VMware 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 VMware 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 VMware 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. The options available are:

- 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.

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

VMware DPM Threshold

The power state (host power on or off) recommendations generated by the VMware 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 VMware 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, VMware 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 VMware 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 VMware DPM in a DRS cluster, by default all hosts in the cluster inherit its VMware 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 VMware 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 VMware DPM

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

The most serious potential error you face when using VMware 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 VMware 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 VMware DPM activity, you can also create alarms for the following vCenter Server events as shown in Table 7-1.

Table 7-1. 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 *VMware vSphere Datacenter Administration Guide*.

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 VMware 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 VMware DPM and does not trigger these false alarms.

Using Affinity Rules

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

The following list describes the two types of affinity rules.

- VM-Host affinity rules are used to specify affinity (or anti-affinity) between a group of virtual machines and a group of hosts. See "VM-Host Affinity Rules," on page 75 for information about creating and using this type of rule.
- VM-VM affinity rules are used to specify affinity (or anti-affinity) between individual virtual machines. See "VM-VM Affinity Rules," on page 77 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.

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 Right-click the cluster in the inventory and select **Edit Settings**.

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

3 Click **Add**.

4 In the Virtual Machine 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; that is whether it is a requirement ("must") or preference ("should") and whether it is affinity ("run on") or anti-affinity ("not run on").

8 Click **OK**.

9 Click **OK** to save the rule.

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 ESX/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, VMware HA, and VMware 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, VMware HA, and VMware 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.
- VMware HA does not perform failovers.
- VMware 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, VMware HA, and VMware 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 on creating and editing alarms, see the *VMware vSphere Datacenter Administration Guide*.

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.

Procedure

1 Right-click the cluster in the inventory and select **Edit Settings**.

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

3 Click **Add**.

4 In the Virtual Machine 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** to save the rule.

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.

Viewing DRS Cluster Information 8

You can view information about a DRS cluster using the cluster **Summary** and **DRS** tabs in the vSphere Client. You can also apply the DRS recommendations that appear in the **DRS** tab.

This chapter includes the following topics:

- "Viewing the Cluster Summary Tab," on page 79
- "Using the DRS Tab," on page 81

Viewing the Cluster Summary Tab

You can access a cluster's **Summary** tab from the inventory panel of the vSphere Client.

The General, VMware DRS, and VMware DRS Resource Distribution sections of this tab display useful information about the configuration and operation of your cluster. The following sections describe the fields that appear in those sections.

Cluster Summary Tab General Section

The General section of the cluster's Summary tab provides general information about your cluster.

Table 8-1. General Section

Field	Description
VMware DRS	Indicates whether VMware DRS is on or off.
VMware HA	Indicates whether VMware HA is on or off.
VMware EVC Mode	Indicates whether Enhanced vMotion Compatibility is enabled or disabled.
Total CPU Resources	Total CPU resources assigned to this cluster.
Total Memory	Total memory resources assigned to this cluster.
Number of Hosts	Number of hosts in this cluster.
Total Processors	Number of processors in all of the hosts in this cluster.
Number of Virtual Machines	Number of virtual machines in this cluster.
Total Migrations using vMotion	Number of migrations performed in the cluster.

Cluster Summary Tab VMware DRS Section

The VMware DRS section appears in the cluster's **Summary** tab only if VMware DRS is enabled.

Table 8-2. VMware DRS Section

Field	Description
Migration Automation Level	Manual, Partially Automated, Fully Automated.
Power Management Automation Level	Off, Manual, Automatic.
DRS Recommendations	Number of DRS migration recommendations awaiting user confirmation. If the value is nonzero, opens the Recommendations page of the cluster's **DRS** tab.
DRS Faults	Number of DRS faults currently outstanding. If the value is nonzero, opens the Faults page of the cluster's **DRS** tab.
Migration Threshold	Indicates the priority level of migration recommendations to apply or generate.
Target host load standard deviation	A value derived from the migration threshold setting that indicates the value under which load imbalance is to be kept.
Current host load standard deviation	A value indicating the current load imbalance in the cluster. This value should be less than the target host load standard deviation unless unapplied DRS recommendations or constraints preclude attaining that level.
View Resource Distribution Chart	Opens the Resource Distribution chart that provides CPU and memory utilization information.
View DRS Troubleshooting Guide	Opens the DRS Troubleshooting Information guide that provides definitions for DRS faults and details about cluster, host, and virtual machine problems.

VMware DRS Resource Distribution Chart

The VMware DRS Resource Distribution chart displays CPU and memory utilization information.

Open this chart by clicking the View Resource Distribution Chart link on the **Summary** tab for a VMware DRS cluster.

CPU Utilization

CPU utilization is displayed on a per-virtual machine basis, grouped by host. The chart shows information for each virtual machine as a colored box, which symbolizes the percentage of entitled resources (as computed by DRS) that are delivered to it. If the virtual machine is receiving its entitlement, this box should be green. If it is not green for an extended time, you might want to investigate what is causing this shortfall (for example, unapplied recommendations).

If you hold the pointer over the box for a virtual machine, its utilization information (Consumed versus Entitlement) appears.

You can toggle the display of CPU resources between % and MHz by clicking the appropriate button.

Memory Utilization

Memory utilization is displayed on a per-virtual machine basis, grouped by host.

If you hold the pointer over the box for a virtual machine, its utilization information (Consumed versus Entitlement) appears.

You can toggle the display of memory resources between % and MB by clicking the appropriate button.

Using the DRS Tab

The **DRS** tab is available when you select a DRS cluster object from the inventory panel in the vSphere Client.

This tab displays information about the DRS recommendations made for the cluster, faults that have occurred in applying such recommendations, and the history of DRS actions. You can access three pages from this tab. These pages are named Recommendations, Faults, and History.

DRS Recommendations Page

The DRS Recommendations page displays information about your cluster's use of DRS. Additionally, this page displays the current set of recommendations generated for optimizing resource utilization in the cluster through either migrations or power management. Only manual recommendations awaiting user confirmation appear on this list.

To view the DRS Recommendations page, click the **Recommendations** button on the **DRS** tab.

The DRS Recommendations page displays the cluster properties that appear in Table 8-3.

Table 8-3. DRS Recommendations Page

Field	Description
Migration Automation Level	Automation level for DRS virtual machine migration recommendations. **Fully Automated**, **Partially Automated**, or **Manual**.
Power Management Automation Level	Automation level for VMware DPM recommendations. **Off**, **Manual**, or **Automatic**.
Migration Threshold	Priority level (or higher) of DRS recommendations to apply.
Power Management Threshold	Priority level (or higher) of VMware DPM recommendations to apply.

Table 8-4 shows the information that DRS provides for each recommendation.

Table 8-4. DRS Recommendations Information

Column	Description
Priority	Priority level (1-5) for the recommendation. Priority one, the highest, indicates a mandatory move because of a host entering maintenance or standby mode or affinity rule violations. Other priority ratings denote how much the recommendation would improve the cluster's performance, from priority two (significant improvement) to priority five (slight). Prior to ESX/ESXi 4.1, recommendations received a star rating (1 to 5 stars) instead of a priority level. The higher the star rating, the more desirable the move. See the VMware knowledge base article at http://kb.vmware.com/kb/1007485 for information on priority level calculation.
Recommendation	The action recommended by DRS. What appears in this column depends on the type of recommendation. ■ For virtual machine migrations: the name of the virtual machine to migrate, the source host (on which the virtual machine is currently running), and the destination host (to which the virtual machine is migrated). ■ For host power state changes: the name of the host to power on or off.
Reason	Reason for the recommendation. why DRS recommends that you migrate the virtual machine or transition the power state of the host. Reasons can be related to any of the following. ■ Balance average CPU or memory loads. ■ Satisfy an affinity rule. ■ Host is entering maintenance. ■ Decrease power consumption. ■ Power off a specific host. ■ Increase cluster capacity. ■ Balance CPU or memory reservations. ■ Maintain unreserved capacity.

Actions that you can take from the DRS Recommendations page:

- To refresh the recommendations, click **Run DRS** and the recommendations update. This command appears on all three DRS pages.
- To apply all recommendations, click **Apply Recommendations**.
- To apply a subset of the recommendations, select the **Override DRS recommendations** check box. This activates the **Apply** check boxes next to each recommendation. Select the check box next to each recommendation and click **Apply Recommendations**.

DRS recommendations are configurable only using vCenter Server. Migrations are not available when you connect the vSphere Client directly to ESX/ESXi hosts. To use the migrations function, have vCenter Server manage the host.

DRS Faults Page

The Faults page of the **DRS** tab displays faults that prevented the recommendation of a DRS action (in manual mode) or the application of a DRS recommendation (in automatic mode).

You can reach this page by clicking the **Faults** button on the **DRS** tab.

You can customize the display of problems using the **Contains** text box. Select the search criteria (Time, Problem, Target) from the drop-down box next to the text box and enter a relevant text string.

You can click on a problem to display additional details about it, including specific faults and the recommendations it prevented. If you click on a fault name, a detailed description of that fault is provided by the DRS Troubleshooting Guide. You can also access this guide from the Faults page by clicking **View DRS Troubleshooting Guide**.

For each fault, DRS provides the information shown in Table 8-5.

Table 8-5. DRS Faults Page

Field	Description
Time	Timestamp of when the fault occurred.
Problem	Description of the condition that prevented the recommendation from being made or applied. When you select this field, more detailed information about its associated faults displays in the Problem Details box.
Target	Target of the intended action.

DRS History Page

The History page of the **DRS** tab displays recent actions taken as a result of DRS recommendations.

You can reach this page by clicking the **History** button on the **DRS** tab.

For each action, DRS provides the information shown in Table 8-6.

Table 8-6. DRS History Page

Field	Description
DRS Actions	Details of the action taken.
Time	Timestamp of when the action occurred.

By default, the information on this page is maintained for four hours and it is preserved across sessions (you can log out and when you log back in, the information is still available).

You can customize the display of recent actions using the **Contains** text box. Select the search criteria (DRS Actions, Time) from the drop-down box next to the text box and enter a relevant text string.

Using NUMA Systems with ESX/ESXi 9

ESX/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 ESX/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?," on page 85
- "How ESX/ESXi NUMA Scheduling Works," on page 86
- "VMware NUMA Optimization Algorithms and Settings," on page 87
- "Resource Management in NUMA Architectures," on page 88
- "Specifying NUMA Controls," on page 89

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. For example, Windows 2000 is fully supported on the IBM x440, although it is not designed for use with NUMA.

There are many disadvantages to using such an operating system on a NUMA platform. 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. Finally, processor clocks might not be synchronized between multiple nodes, so applications that read the clock directly might behave incorrectly.

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 ESX/ESXi NUMA Scheduling Works

ESX/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 ESX/ESXi host preferentially allocates it from the home node.
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 ESX/ESXi NUMA scheduler. For example, if you manually set the processor 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 ESX/ESXi NUMA optimizations.

The NUMA scheduling and memory placement policies in ESX/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. ESX/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 low locality. For information on configuring the behavior of this feature, see “Advanced Virtual Machine Attributes,” on page 114.

VMware NUMA Optimization Algorithms and Settings

This section describes the algorithms and settings used by ESX/ESXi to maximize application performance while still maintaining resource guarantees.

Home Nodes and Initial Placement

When a virtual machine is powered on, ESX/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.

New virtual machines are initially assigned to home nodes in a round robin fashion, with the first virtual machine going to the first node, the second virtual machine to the second node, and so forth. This policy ensures that memory is evenly used throughout all nodes of the system.

Several operating systems, such as Windows Server 2003, provide this level of NUMA support, which is known as initial placement. It might be sufficient for systems that run only a single workload, such as a benchmarking configuration, which does not change over the course of the system's uptime. However, initial placement is not sophisticated enough to guarantee good performance and fairness for a datacenter-class system that is expected to support changing workloads.

To understand the weaknesses of an initial-placement-only system, consider the following example: an administrator starts four virtual machines and the system places two of them on the first node. The second two virtual machines are placed on the second node. If both virtual machines on the second node are stopped, or if they become idle, the system becomes completely imbalanced, with the entire load placed on the first node. Even if the system allows one of the remaining virtual machines to run remotely on the second node, it suffers a serious performance penalty because all its memory remains on its original node.

Dynamic Load Balancing and Page Migration

ESX/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. ESX/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 ESX/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 ESX/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 ESX/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, ESX/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 ESX/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.

Memory Page Sharing Across and Within NUMA Nodes

The VMkernel.Boot.sharePerNode option controls whether memory pages can be shared (de-duplicated) only within a single NUMA node or across multiple NUMA nodes.

VMkernel.Boot.sharePerNode is turned on by default, and identical pages are shared only within the same NUMA node. This improves memory locality, because all accesses to shared pages use local memory.

NOTE This default behavior is the same in all previous versions of ESX.

When you turn off the VMkernel.Boot.sharePerNode option, identical pages can be shared across different NUMA nodes. This increases the amount of sharing and de-duplication, which reduces overall memory consumption at the expense of memory locality. In memory-constrained environments, such as VMware View deployments, many similar virtual machines present an opportunity for de-duplication, and page sharing across NUMA nodes could be very beneficial.

Resource Management in NUMA Architectures

You can perform resource management with different types of NUMA architecture. The systems that offer a NUMA platform to support industry-standard operating systems include those based on either AMD CPUs or the IBM Enterprise X-Architecture.

IBM Enterprise X-Architecture

One architecture that supports NUMA is the IBM Enterprise X-Architecture.

The IBM Enterprise X-Architecture supports servers with up to four nodes (also called CECs or SMP Expansion Complexes in IBM terminology). Each node can contain up to four Intel Xeon MP processors for a total of 16 CPUs. The next generation IBM eServer x445 uses an enhanced version of the Enterprise X-Architecture, and scales to eight nodes with up to four Xeon MP processors for a total of 32 CPUs. The third-generation IBM eServer x460 provides similar scalability but also supports 64-bit Xeon MP processors. The high scalability of all these systems stems from the Enterprise X-Architecture's NUMA design that is shared with IBM high end POWER4-based pSeries servers.

AMD Opteron-Based Systems

AMD Opteron-based systems, such as the HP ProLiant DL585 Server, also provide NUMA support.

The BIOS setting for node interleaving determines whether the system behaves more like a NUMA system or more like a Uniform Memory Architecture (UMA) system. See the HP ProLiant DL585 Server technology brief. See also the *HP ROM-Based Setup Utility User Guide* at the HP Web site.

By default, node interleaving is disabled, so each processor has its own memory. The BIOS builds a System Resource Allocation Table (SRAT), so the ESX/ESXi host detects the system as NUMA and applies NUMA optimizations. If you enable node interleaving (also known as interleaved memory), the BIOS does not build an SRAT, so the ESX/ESXi host does not detect the system as NUMA.

Currently shipping Opteron processors have up to four cores per socket. When node memory is enabled, the memory on the Opteron processors is divided such that each socket has some local memory, but memory for other sockets is remote. The single-core Opteron systems have a single processor per NUMA node and the dual-core Opteron systems have two processors for each NUMA node.

SMP virtual machines (having two virtual processors) cannot reside within a NUMA node that has a single core, such as the single-core Opteron processors. This also means they cannot be managed by the ESX/ESXi NUMA scheduler. Virtual machines that are not managed by the NUMA scheduler still run correctly. However, those virtual machines don't benefit from the ESX/ESXi NUMA optimizations. Uniprocessor virtual machines (with a single virtual processor) can reside within a single NUMA node and are managed by the ESX/ESXi NUMA scheduler.

NOTE For small Opteron systems, NUMA rebalancing is now disabled by default to ensure scheduling fairness. Use the Numa.RebalanceCoresTotal and Numa.RebalanceCoresNode options to change this behavior.

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.

This 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, an ESX/ESXi host's automatic NUMA optimizations result in good performance.

ESX/ESXi provides two sets of controls for NUMA placement, so that administrators can control memory and processor placement of a virtual machine.

The vSphere Client allows you to specify two options.

CPU Affinity	A virtual machine should use only the processors on a given node.
Memory Affinity	The server should allocate memory only on the specified node.

If you set both options before a virtual machine starts, the virtual machine runs only on the selected node and all of its memory is allocated locally.

An administrator can also manually move a virtual machine to another node after the virtual machine has started running. In this case, the page migration rate of the virtual machine must be set manually, so that memory from the virtual machine's previous node can be moved to its new node.

Manual NUMA placement might interfere with the ESX/ESXi resource management algorithms, which try to give each virtual machine a fair share of the system's processor resources. For example, if ten virtual machines with processor-intensive workloads are manually placed on one node, and only two virtual machines are manually placed on another node, it is impossible for the system to give all twelve virtual machines equal shares of the system's resources.

NOTE You can view NUMA configuration information in the Memory panel of the `resxtop` (or `esxtop`) utility.

Associate Virtual Machines with a Single NUMA Node Using CPU Affinity

You might be able to improve the performance of the applications on a virtual machine by associating it to the CPU numbers on a single NUMA node (manual CPU affinity).

Procedure

1. Using a vSphere Client, right-click a virtual machine and select **Edit Settings**.
2. In the Virtual Machine Properties dialog box, select the **Resources** tab and select **Advanced CPU**.
3. In the Scheduling Affinity panel, set CPU affinity for different NUMA nodes.

   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 a NUMA Node Using Memory Affinity

You can specify that all future memory allocations on a virtual machine use pages associated with a single NUMA node (also known as manual memory affinity). When the virtual machine uses local memory, the performance improves on that virtual machine.

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. Using a vSphere Client, right-click a virtual machine and select **Edit Settings**.
2. In the Virtual Machine Properties dialog box, 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.

Performance Monitoring Utilities: resxtop and esxtop

A

The `resxtop` and `esxtop` command-line utilities provide a detailed look at how ESX/ESXi uses resources in real time. You can start either utility in one of three modes: interactive (default), batch, or replay.

The fundamental difference between `resxtop` and `esxtop` is that you can use `resxtop` remotely, whereas you can start `esxtop` only through the service console of a local ESX host.

This appendix includes the following topics:

- "Using the esxtop Utility," on page 93
- "Using the resxtop Utility," on page 94
- "Using esxtop or resxtop in Interactive Mode," on page 94
- "Using Batch Mode," on page 108
- "Using Replay Mode," on page 109

Using the esxtop Utility

The `esxtop` utility runs only on the ESX host's service console and to use it you must have root user privileges.

Type the command, using the options you want:

```
esxtop [-] [h] [v] [b] [s] [a] [c filename] [R vm-support_dir_path]  [d delay] [n iter]
```

The `esxtop` utility reads its default configuration from `.esxtop41rc`. This configuration file consists of nine lines.

The first eight lines contain lowercase and uppercase letters to specify which fields appear in which order on the CPU, memory, storage adapter, storage device, virtual machine storage, network, interrupt, and CPU power panels. The letters correspond to the letters in the Fields or Order panels for the respective `esxtop` panel.

The ninth line contains information on the other options. Most important, if you saved a configuration in secure mode, you do not get an insecure `esxtop` without removing the `s` from the seventh line of your `.esxtop41rc` file. A number specifies the delay time between updates. As in interactive mode, typing `c`, `m`, `d`, `u`, `v`, `n`, `I`, or `p` determines the panel with which `esxtop` starts.

Note Do not edit the `.esxtop41rc` file. Instead, select the fields and the order in a running `esxtop` process, make changes, and save this file using the `W` interactive command.

Using the resxtop Utility

The resxtop utility is a vSphere CLI command.

Before you can use any vSphere CLI commands, you must either download and install a vSphere CLI package or deploy the vSphere Management Assistant (vMA) to your ESX/ESXi host or vCenter Server system.

After it is set up, start resxtop from the command line. For remote connections, you can connect to an ESX/ESXi host either directly or through vCenter Server.

The command-line options listed in Table A-1 are the same as for esxtop (except for the R option) with additional connection options.

NOTE resxtop does not use all the options shared by other vSphere CLI commands.

Table A-1. resxtop Command-Line Options

Option	Description
[server]	Name of the remote host to connect to (required). If connecting directly to the ESX/ESXi host, use the name of that host. If your connection to the ESX/ESXi host is indirect (that is, through vCenter Server), use the name of the vCenter Server system for this option.
[vihost]	If you connect indirectly (through vCenter Server), this option should contain the name of the ESX/ESXi host you connect to. If you connect directly to the ESX/ESXi host, this option is not used. Note that the host name needs to be the same as what appears in the vSphere Client.
[portnumber]	Port number to connect to on the remote server. The default port is 443, and unless this is changed on the server, this option is not needed.
[username]	User name to be authenticated when connecting to the remote host. The remote server prompts you for a password.

You can also use resxtop on a local ESX/ESXi host by omitting the server option on the command line. The command defaults to localhost.

Using esxtop or resxtop in Interactive Mode

By default, resxtop and esxtop run in interactive mode. Interactive mode displays statistics in different panels.

A help menu is available for each panel.

Interactive Mode Command-Line Options

You can use various command-line options with esxtop and resxtop in interactive mode.

Table A-2 lists the command-line options available in interactive mode.

Table A-2. Interactive Mode Command-Line Options

Option	Description
h	Prints help for resxtop (or esxtop) command-line options.
v	Prints resxtop (or esxtop) version number.
s	Calls resxtop (or esxtop) in secure mode. In secure mode, the -d command, which specifies delay between updates, is disabled.
d	Specifies the delay between updates. The default is five seconds. The minimum is two seconds. Change this with the interactive command s. If you specify a delay of less than two seconds, the delay is set to two seconds.
n	Number of iterations. Updates the display n times and exits. Default value is 10000.

Table A-2. Interactive Mode Command-Line Options (Continued)

Option	Description
`server`	The name of the remote server host to connect to (required for `resxtop` only).
`vihost`	If you connect indirectly (through vCenter Server), this option should contain the name of the ESX/ESXi host you connect to. If you connect directly to the ESX/ESXi host, this option is not used. Note that the host name needs to be the same as what is displayed in the vSphere Client.
`portnumber`	The port number to connect to on the remote server. The default port is 443, and unless this is changed on the server, this option is not needed. (`resxtop` only)
`username`	The user name to be authenticated when connecting to the remote host. The remote server prompts you for a password, as well (`resxtop` only).
`a`	Show all statistics. This option overrides configuration file setups and shows all statistics. The configuration file can be the default ~/.esxtop41rc configuration file or a user-defined configuration file.
`c`*filename*	Load a user-defined configuration file. If the -c option is not used, the default configuration filename is ~/.esxtop41rc. Create your own configuration file, specifying a different filename, using the `W` single-key interactive command.

Common Statistics Description

Several statistics appear on the different panels while `resxtop` (or `esxtop`) is running in interactive mode. These statistics are common across all four panels.

The Uptime line, found at the top of each of the four `resxtop` (or `esxtop`) panels, displays the current time, time since last reboot, number of currently running worlds and load averages. A world is an ESX/ESXi VMkernel schedulable entity, similar to a process or thread in other operating systems.

Below that the load averages over the past one, five, and fifteen minutes appear. Load averages take into account both running and ready-to-run worlds. A load average of 1.00 means that there is full utilization of all physical CPUs. A load average of 2.00 means that the ESX/ESXi system might need twice as many physical CPUs as are currently available. Similarly, a load average of 0.50 means that the physical CPUs on the ESX/ESXi system are half utilized.

Statistics Columns and Order Pages

You can define the order of fields displayed in interactive mode.

If you press `f`, `F`, `o`, or `O`, the system displays a page that specifies the field order on the top line and short descriptions of the field contents. If the letter in the field string corresponding to a field is uppercase, the field is displayed. An asterisk in front of the field description indicates whether a field is displayed.

The order of the fields corresponds to the order of the letters in the string.

From the Field Select panel, you can:

- Toggle the display of a field by pressing the corresponding letter.
- Move a field to the left by pressing the corresponding uppercase letter.
- Move a field to the right by pressing the corresponding lowercase letter.

Interactive Mode Single-Key Commands

When running in interactive mode, `resxtop` (or `esxtop`) recognizes several single-key commands.

All interactive mode panels recognize the commands listed in Table A-3. The command to specify the delay between updates is disabled if the `s` option is given on the command line. All sorting interactive commands sort in descending order.

Table A-3. Interactive Mode Single-Key Commands

Key	Description
h or ?	Displays a help menu for the current panel, giving a brief summary of commands, and the status of secure mode.
space	Immediately updates the current panel.
^L	Erases and redraws the current panel.
f or F	Displays a panel for adding or removing statistics columns (fields) to or from the current panel.
o or O	Displays a panel for changing the order of statistics columns on the current panel.
#	Prompts you for the number of statistics rows to display. Any value greater than 0 overrides automatic determination of the number of rows to show, which is based on window size measurement. If you change this number in one `resxtop` (or `esxtop`) panel, the change affects all four panels.
s	Prompts you for the delay between updates, in seconds. Fractional values are recognized down to microseconds. The default value is five seconds. The minimum value is two seconds. This command is not available in secure mode.
W	Write the current setup to an esxtop (or resxtop) configuration file. This is the recommended way to write a configuration file. The default filename is the one specified by -c option, or `~/.esxtop41rc` if the -c option is not used. You can also specify a different filename on the prompt generated by this W command.
q	Quit interactive mode.
c	Switch to the CPU resource utilization panel.
p	Switch to the CPU Power utilization panel.
m	Switch to the memory resource utilization panel.
d	Switch to the storage (disk) adapter resource utilization panel.
u	Switch to storage (disk) device resource utilization screen.
v	Switch to storage (disk) virtual machine resource utilization screen.
n	Switch to the network resource utilization panel.
i	Switch to the interrupt panel.

CPU Panel

The CPU panel displays server-wide statistics as well as statistics for individual world, resource pool, and virtual machine CPU utilization.

Resource pools, virtual machines that are running, or other worlds are at times called groups. For worlds belonging to a virtual machine, statistics for the virtual machine that is running are displayed. All other worlds are logically aggregated into the resource pools that contain them.

Table A-4 lists statistics that appear in the CPU Panel.

Table A-4. CPU Panel Statistics

Line	Description
PCPU USED(%)	A PCPU refers to a physical hardware execution context. This can be a physical CPU core if hyperthreading is unavailable or disabled, or a logical CPU (LCPU or SMT thread) if hyperthreading is enabled. PCPU USED(%) displays the following percentages: ■ percentage of CPU usage per PCPU ■ percentage of CPU usage averaged over all PCPUs CPU Usage (%USED) is the percentage of PCPU nominal frequency that was used since the last screen update. It equals the total sum of %USED for Worlds that ran on this PCPU. **Note** If a PCPU is running at frequency that is higher than its nominal (rated) frequency, then PCPU USED(%) can be greater than 100%. If a PCPU and its partner are busy when hyperthreading is enabled, each PCPU accounts for half of the CPU usage.
PCPU UTIL(%)	A PCPU refers to a physical hardware execution context. This can be a physical CPU core if hyperthreading is unavailable or disabled, or a logical CPU (LCPU or SMT thread) if hyperthreading is enabled. PCPU UTIL(%) represents the percentage of real time that the PCPU was not idle (raw PCPU utilization) and it displays the percentage CPU utilization per PCPU, and the percentage CPU utilization averaged over all PCPUs. **Note** PCPU UTIL(%) might differ from PCPU USED(%) due to power management technologies or hyperthreading.
CCPU(%)	Percentages of total CPU time as reported by the ESX service console. This field does not appear if you are using ESXi. ■ `us` — Percentage user time. ■ `sy` — Percentage system time. ■ `id` — Percentage idle time. ■ `wa` — Percentage wait time. ■ `cs/sec` — Context switches per second recorded by the service console.
ID	Resource pool ID or virtual machine ID of the resource pool or virtual machine of the world that is running, or world ID of the world that is running.
GID	Resource pool ID of the resource pool or virtual machine of the world that is running.
NAME	Name of the resource pool or virtual machine of the world that is running, or name of the world that is running.
NWLD	Number of members in the resource pool or virtual machine of the world that is running. If a Group is expanded using the interactive command `e`, then NWLD for all the resulting worlds is 1. (Some resource pools like the console resource pool have only one member.)
%STATE TIMES	Set of CPU statistics made up of the following percentages. For a world, the percentages are a percentage of one physical CPU core.
%USED	Percentage of physical CPU core cycles used by the resource pool, virtual machine, or world. %USED might depend on the frequency with which the CPU core is running. When running with lower CPU core frequency, %USED can be smaller than %RUN. On CPUs which support turbo mode, CPU frequency can also be higher than the nominal (rated) frequency, and %USED can be larger than %RUN.
%SYS	Percentage of time spent in the ESX/ESXi VMkernel on behalf of the resource pool, virtual machine, or world to process interrupts and to perform other system activities. This time is part of the time used to calculate %USED.
%WAIT	Percentage of time the resource pool, virtual machine, or world spent in the blocked or busy wait state. This percentage includes the percentage of time the resource pool, virtual machine, or world was idle.

Table A-4. CPU Panel Statistics (Continued)

Line	Description
%IDLE	Percentage of time the resource pool, virtual machine, or world was idle. Subtract this percentage from %WAIT to see the percentage of time the resource pool, virtual machine, or world was waiting for some event. The difference, %WAIT- %IDLE, of the VCPU worlds can be used to estimate guest I/O wait time. To find the VCPU worlds, use the single-key command `e` to expand a virtual machine and search for the world NAME starting with "vcpu". (The VCPU worlds might wait for other events in addition to I/O events, so this measurement is only an estimate.)
%RDY	Percentage of time the resource pool, virtual machine, or world was ready to run, but was not provided CPU resources on which to execute.
%MLMTD (max limited)	Percentage of time the ESX/ESXi VMkernel deliberately did not run the resource pool, virtual machine, or world because doing so would violate the resource pool, virtual machine, or world's limit setting. Because the resource pool, virtual machine, or world is ready to run when it is prevented from running in this way, the %MLMTD (max limited) time is included in %RDY time.
%SWPWT	Percentage of time a resource pool or world spends waiting for the ESX/ESXi VMkernel to swap memory. The %SWPWT (swap wait) time is included in the %WAIT time.
EVENT COUNTS/s	Set of CPU statistics made up of per second event rates. These statistics are for VMware internal use only.
CPU ALLOC	Set of CPU statistics made up of the following CPU allocation configuration parameters.
AMIN	Resource pool, virtual machine, or world attribute Reservation.
AMAX	Resource pool, virtual machine, or world attribute Limit. A value of -1 means unlimited.
ASHRS	Resource pool, virtual machine, or world attribute Shares.
SUMMARY STATS	Set of CPU statistics made up of the following CPU configuration parameters and statistics. These statistics apply only to worlds and not to virtual machines or resource pools.
AFFINITY BIT MASK	Bit mask showing the current scheduling affinity for the world.
HTSHARING	Current hyperthreading configuration.
CPU	The physical or logical processor on which the world was running when `resxtop` (or `esxtop`) obtained this information.
HTQ	Indicates whether the world is currently quarantined or not. N means no and Y means yes.
TIMER/s	Timer rate for this world.
%OVRLP	Percentage of system time spent during scheduling of a resource pool, virtual machine, or world on behalf of a different resource pool, virtual machine, or world while the resource pool, virtual machine, or world was scheduled. This time is not included in %SYS. For example, if virtual machine A is currently being scheduled and a network packet for virtual machine B is processed by the ESX/ESXi VMkernel, the time spent appears as %OVRLP for virtual machine A and %SYS for virtual machine B.
%RUN	Percentage of total time scheduled. This time does not account for hyperthreading and system time. On a hyperthreading enabled server, the %RUN can be twice as large as %USED.
%CSTP	Percentage of time a resource pool spends in a ready, co-deschedule state. **NOTE** You might see this statistic displayed, but it is intended for VMware use only.
POWER	Current CPU power consumption for a resource pool (in Watts).
%LAT_C	Percentage of time the resource pool or world was ready to run but was not scheduled to run because of CPU resource contention.
%LAT_M	Percentage of time the resource pool or world was ready to run but was not scheduled to run because of memory resource contention.
%DMD	CPU demand in percentage. It represents the average active CPU load in the past minute.

You can change the display using single-key commands as described in Table A-5.

Table A-5. CPU Panel Single-Key Commands

Command	Description
e	Toggles whether CPU statistics are displayed expanded or unexpanded. The expanded display includes CPU resource utilization statistics broken down by individual worlds belonging to a resource pool or virtual machine. All percentages for the individual worlds are percentage of a single physical CPU. Consider these examples: ■ If the %Used by a resource pool is 30% on a two-way server, the resource pool is utilizing 30 percent of one physical core. ■ If the %Used by a world belonging to a resource pool is 30 percent on a two-way server, that world is utilizing 30% of one physical core.
U	Sorts resource pools, virtual machines, and worlds by the resource pool's or virtual machine's %Used column. This is the default sort order.
R	Sorts resource pools, virtual machines, and worlds by the resource pool's or virtual machine's %RDY column.
N	Sorts resource pools, virtual machines, and worlds by the GID column.
V	Displays virtual machine instances only.
L	Changes the displayed length of the NAME column.

CPU Power Panel

The CPU Power panel displays CPU Power utilization statistics.

On the CPU Power panel, statistics are arranged per PCPU. A PCPU is a physical hardware execution context -- a physical CPU core if hyper-threading is unavailable or disabled, or a logical CPU (LCPU or SMT thread) if hyper-threading is enabled.

Table A-6. CPU Power Panel Statistics

Line	Description
Power Usage	Current total power usage (in Watts).
Power Cap	Total power cap (in Watts).
%USED	Percentage of PCPU nominal frequency used since the last screen update. It is the same as PCPU USED(%) shown in the CPU Screen.
%UTIL	Raw PCPU utilization, that is the percentage of real time that PCPU was not idle. It is the same as PCPU UTIL(%) shown in the CPU Screen.
%Cx	Percentage of time the PCPU spent in C-State 'x'.
%Px	Percentage of time the PCPU spent in P-State 'x'.
%Tx	Percentage of time the PCPU spent in T-State 'x'.

Memory Panel

The Memory panel displays server-wide and group memory utilization statistics. As on the CPU panel, groups correspond to resource pools, running virtual machines, or other worlds that are consuming memory.

The first line, found at the top of the Memory panel, displays the current time, time since last reboot, number of currently running worlds, and memory overcommitment averages. The memory overcommitment averages over the past one, five, and fifteen minutes appear. Memory overcommitment of 1.00 means a memory overcommitment of 100 percent. See "Memory Overcommitment," on page 28.

Table A-7. Memory Panel Statistics

Field	Description	
PMEM (MB)	Displays the machine memory statistics for the server. All numbers are in megabytes.	
	total	Total amount of machine memory in the server.
	cos	Amount of machine memory allocated to the ESX service console.
	vmk	Amount of machine memory being used by the ESX/ESXi VMkernel.
	other	Amount of machine memory being used by everything other than the ESX service console and ESX/ESXi VMkernel.
	free	Amount of machine memory that is free.
VMKMEM (MB)	Displays the machine memory statistics for the ESX/ESXi VMkernel. All numbers are in megabytes.	
	managed	Total amount of machine memory managed by the ESX/ESXi VMkernel.
	min free	Minimum amount of machine memory that the ESX/ESXi VMkernel aims to keep free.
	rsvd	Total amount of machine memory currently reserved by resource pools.
	ursvd	Total amount of machine memory currently unreserved.
	state	Current machine memory availability state. Possible values are high, soft, hard and low. High means that the machine memory is not under any pressure and low means that it is.
COSMEM (MB)	Displays the memory statistics as reported by the ESX service console. All numbers are in megabytes. This field does not appear if you are using ESXi.	
	free	Amount of idle memory.
	swap_t	Total swap configured.
	swap_f	Amount of swap free.
	r/s is	Rate at which memory is swapped in from disk.
	w/s	Rate at which memory is swapped to disk.
NUMA (MB)	Displays the ESX/ESXi NUMA statistics. This line appears only if the ESX/ESXi host is running on a NUMA server. All numbers are in megabytes. For each NUMA node in the server, two statistics are displayed: ■ The total amount of machine memory in the NUMA node that is managed by ESX/ESXi. ■ The amount of machine memory in the node that is currently free (in parentheses).	
PSHARE (MB)	Displays the ESX/ESXi page-sharing statistics. All numbers are in megabytes.	
	shared	Amount of physical memory that is being shared.
	common	Amount of machine memory that is common across worlds.
	saving	Amount of machine memory that is saved because of page sharing.

Table A-7. Memory Panel Statistics (Continued)

Field	Description
SWAP (MB)	Displays the ESX/ESXi swap usage statistics. All numbers are in megabytes. **curr** Current swap usage. **rclmtgt** Where the ESX/ESXi system expects the reclaimed memory to be. Memory can be reclaimed by swapping or compression. **r/s** Rate at which memory is swapped in by the ESX/ESXi system from disk. **w/s** Rate at which memory is swapped to disk by the ESX/ESXi system.
ZIP (MB)	Displays the ESX/ESXi memory compression statistics. All numbers are in megabytes. **zipped** Total compressed physical memory. **saved** Saved memory by compression. See "Memory Compression," on page 37.
MEMCTL (MB)	Displays the memory balloon statistics. All numbers are in megabytes. **curr** Total amount of physical memory reclaimed using the `vmmemctl` module. **target** Total amount of physical memory the ESX/ESXi host attempts to reclaim using the `vmmemctl` module. **max** Maximum amount of physical memory the ESX/ESXi host can reclaim using the `vmmemctl` module.
AMIN	Memory reservation for this resource pool or virtual machine.
AMAX	Memory limit for this resource pool or virtual machine. A value of -1 means Unlimited.
ASHRS	Memory shares for this resource pool or virtual machine.
NHN	Current home node for the resource pool or virtual machine. This statistic is applicable only on NUMA systems. If the virtual machine has no home node, a dash (-) appears.
NRMEM (MB)	Current amount of remote memory allocated to the virtual machine or resource pool. This statistic is applicable only on NUMA systems.
N% L	Current percentage of memory allocated to the virtual machine or resource pool that is local.
MEMSZ (MB)	Amount of physical memory allocated to a resource pool or virtual machine.
GRANT (MB)	Amount of guest physical memory mapped to a resource pool or virtual machine. The consumed host machine memory is equal to GRANT - SHRDSVD.
SZTGT (MB)	Amount of machine memory the ESX/ESXi VMkernel wants to allocate to a resource pool or virtual machine.
TCHD (MB)	Working set estimate for the resource pool or virtual machine.
%ACTV	Percentage of guest physical memory that is being referenced by the guest. This is an instantaneous value.
%ACTVS	Percentage of guest physical memory that is being referenced by the guest. This is a slow moving average.
%ACTVF	Percentage of guest physical memory that is being referenced by the guest. This is a fast moving average.
%ACTVN	Percentage of guest physical memory that is being referenced by the guest. This is an estimation. (You might see this statistic displayed, but it is intended for VMware use only.)
MCTL?	Memory balloon driver is installed or not. **N** means no, **Y** means yes.

Table A-7. Memory Panel Statistics (Continued)

Field	Description
MCTLSZ (MB)	Amount of physical memory reclaimed from the resource pool by way of ballooning.
MCTLTGT (MB)	Amount of physical memory the ESX/ESXi system attempts to reclaim from the resource pool or virtual machine by way of ballooning.
MCTLMAX (MB)	Maximum amount of physical memory the ESX/ESXi system can reclaim from the resource pool or virtual machine by way of ballooning. This maximum depends on the guest operating system type.
SWCUR (MB)	Current swap usage by this resource pool or virtual machine.
SWTGT (MB)	Target where the ESX/ESXi host expects the swap usage by the resource pool or virtual machine to be.
SWR/s (MB)	Rate at which the ESX/ESXi host swaps in memory from disk for the resource pool or virtual machine.
SWW/s (MB)	Rate at which the ESX/ESXi host swaps resource pool or virtual machine memory to disk.
CPTRD (MB)	Amount of data read from checkpoint file.
CPTTGT (MB)	Size of checkpoint file.
ZERO (MB)	Resource pool or virtual machine physical pages that are zeroed.
SHRD (MB)	Resource pool or virtual machine physical pages that are shared.
SHRDSVD (MB)	Machine pages that are saved because of resource pool or virtual machine shared pages.
OVHD (MB)	Current space overhead for resource pool.
OVHDMAX (MB)	Maximum space overhead that might be incurred by resource pool or virtual machine.
OVHDUW (MB)	Current space overhead for a user world. (You might see this statistic displayed, but it is intended for VMware use only.)
GST_NDx (MB)	Guest memory allocated for a resource pool on NUMA node x. This statistic is applicable on NUMA systems only.
OVD_NDx (MB)	VMM overhead memory allocated for a resource pool on NUMA node x. This statistic is applicable on NUMA systems only.
TCHD_W (MB)	Write working set estimate for resource pool.
CACHESZ (MB)	Compression memory cache size.
CACHEUSD (MB)	Used compression memory cache.
ZIP/s (MB/s)	Compressed memory per second.
UNZIP/s (MB/s)	Decompressed memory per second.

Table A-8 displays the interactive commands that you can use with the memory panel.

Table A-8. Memory Panel Interactive Commands

Command	Description
M	Sort resource pools or virtual machines by MEMSZ column. This is the default sort order.
B	Sort resource pools or virtual machines by Group Memctl column.
N	Sort resource pools or virtual machines by GID column.
V	Display virtual machine instances only.
L	Changes the displayed length of the NAME column.

Storage Adapter Panel

Statistics in the Storage Adapter panel are aggregated per storage adapter by default. Statistics can also be viewed per storage path.

The Storage Adapter panel displays the information shown in Table A-9.

Table A-9. Storage Adapter Panel Statistics

Column	Description
ADAPTR	Name of the storage adapter.
PATH	Storage path name. This name is only visible if the corresponding adapter is expanded. See interactive command `e` in Table A-10.
NPTHS	Number of paths.
AQLEN	Current queue depth of the storage adapter.
CMDS/s	Number of commands issued per second.
READS/s	Number of read commands issued per second.
WRITES/s	Number of write commands issued per second.
MBREAD/s	Megabytes read per second.
MBWRTN/s	Megabytes written per second.
RESV/s	Number of SCSI reservations per second.
CONS/s	Number of SCSI reservation conflicts per second.
DAVG/cmd	Average device latency per command, in milliseconds.
KAVG/cmd	Average ESX/ESXi VMkernel latency per command, in milliseconds.
GAVG/cmd	Average virtual machine operating system latency per command, in milliseconds.
QAVG/cmd	Average queue latency per command, in milliseconds.
DAVG/rd	Average device read latency per read operation, in milliseconds.
KAVG/rd	Average ESX/ESXi VMkernel read latency per read operation, in milliseconds.
GAVG/rd	Average guest operating system read latency per read operation, in milliseconds.
QAVG/rd	Average queue latency per read operation, in milliseconds.
DAVG/wr	Average device write latency per write operation, in milliseconds.
KAVG/wr	Average ESX/ESXi VMkernel write latency per write operation, in milliseconds.
GAVG/wr	Average guest operating system write latency per write operation, in milliseconds.
QAVG/wr	Average queue latency per write operation, in milliseconds.
ABRTS/s	Number of commands aborted per second.
RESETS/s	Number of commands reset per second.
PAECMD/s	The number of PAE (Physical Address Extension) commands per second.
PAECP/s	The number of PAE copies per second.
SPLTCMD/s	The number of split commands per second.
SPLTCP/s	The number of split copies per second.

Table A-10 displays the interactive commands you can use with the storage adapter panel.

Table A-10. Storage Adapter Panel Interactive Commands

Command	Description
`e`	Toggles whether storage adapter statistics appear expanded or unexpanded. Allows you to view storage resource utilization statistics broken down by individual paths belonging to an expanded storage adapter. You are prompted for the adapter name.
`r`	Sorts by READS/s column.
`w`	Sorts by WRITES/s column.
`R`	Sorts by MBREAD/s read column.
`T`	Sorts by MBWRTN/s written column.
`N`	Sorts first by ADAPTR column, then by PATH column. This is the default sort order.

Storage Device Panel

The storage device panel displays server-wide storage utilization statistics.

By default, the information is grouped per storage device. You can also group the statistics per path, per world, or per partition.

Table A-11. Storage Device Panel Statistics

Column	Description
DEVICE	Name of the storage device.
PATH	Path name. This name is visible only if the corresponding device is expanded to paths. See the interactive command `p` in Table A-12.
WORLD	World ID. This ID is visible only if the corresponding device is expanded to worlds. See the interactive command `e` in Table A-12. The world statistics are per world per device.
PARTITION	Partition ID. This ID is visible only if the corresponding device is expanded to partitions. See interactive command `t` in Table A-12.
NPH	Number of paths.
NWD	Number of worlds.
NPN	Number of partitions.
SHARES	Number of shares. This statistic is applicable only to worlds.
BLKSZ	Block size in bytes.
NUMBLKS	Number of blocks of the device.
DQLEN	Current device queue depth of the storage device.
WQLEN	World queue depth. This is the maximum number of ESX/ESXi VMkernel active commands that the world is allowed to have. This is a per device maximum for the world. It is valid only if the corresponding device is expanded to worlds.
ACTV	Number of commands in the ESX/ESXi VMkernel that are currently active. This statistic applies to only worlds and devices.
QUED	Number of commands in the ESX/ESXi VMkernel that are currently queued. This statistic applies to only worlds and devices.
%USD	Percentage of the queue depth used by ESX/ESXi VMkernel active commands. This statistic applies to only worlds and devices.
LOAD	Ratio of ESX/ESXi VMkernel active commands plus ESX/ESXi VMkernel queued commands to queue depth. This statistic applies to only worlds and devices.
CMDS/s	Number of commands issued per second.
READS/s	Number of read commands issued per second.

Table A-11. Storage Device Panel Statistics (Continued)

Column	Description
WRITES/s	Number of write commands issued per second.
MBREAD/s	Megabytes read per second.
MBWRTN/s	Megabytes written per second.
DAVG/cmd	Average device latency per command in milliseconds.
KAVG/cmd	Average ESX/ESXi VMkernel latency per command in milliseconds.
GAVG/cmd	Average guest operating system latency per command in milliseconds.
QAVG/cmd	Average queue latency per command in milliseconds.
DAVG/rd	Average device read latency per read operation in milliseconds.
KAVG/rd	Average ESX/ESXi VMkernel read latency per read operation in milliseconds.
GAVG/rd	Average guest operating system read latency per read operation in milliseconds.
QAVG/rd	Average queue read latency per read operation in milliseconds.
DAVG/wr	Average device write latency per write operation in milliseconds.
KAVG/wr	Average ESX/ESXi VMkernel write latency per write operation in milliseconds.
GAVG/wr	Average guest operating system write latency per write operation in milliseconds.
QAVG/wr	Average queue write latency per write operation in milliseconds.
ABRTS/s	Number of commands aborted per second.
RESETS/s	Number of commands reset per second.
PAECMD/s	Number of PAE commands per second. This statistic applies to only paths.
PAECP/s	Number of PAE copies per second. This statistic applies to only paths.
SPLTCMD/s	Number of split commands per second. This statistic applies to only paths.
SPLTCP/s	Number of split copies per second. This statistic applies to only paths.

Table A-12 displays the interactive commands you can use with the storage device panel.

Table A-12. Storage Device Panel Interactive Commands

Command	Description
`e`	Expand or roll up storage world statistics. This command allows you to view storage resource utilization statistics separated by individual worlds belonging to an expanded storage device. You are prompted for the device name. The statistics are per world per device.
`P`	Expand or roll up storage path statistics. This command allows you to view storage resource utilization statistics separated by individual paths belonging to an expanded storage device. You are prompted for the device name.
`t`	Expand or roll up storage partition statistics. This command allows you to view storage resource utilization statistics separated by individual partitions belonging to an expanded storage device. You are prompted for the device name.
`r`	Sort by READS/s column.
`w`	Sort by WRITES/s column.
`R`	Sort by MBREAD/s column.
`T`	Sort by MBWRTN column.
`N`	Sort first by DEVICE column, then by PATH, WORLD, and PARTITION column. This is the default sort order.
`L`	Changes the displayed length of the DEVICE column.

Virtual Machine Storage Panel

This panel displays virtual machine-centric storage statistics.

By default, statistics are aggregated on a per-resource-pool basis. One virtual machine has one corresponding resource pool, so the panel displays statistics on a per-virtual-machine basis as shown in Table A-13. You can also view statistics on per-VSCSI-device basis.

Table A-13. Virtual Machine Storage Panel Statistics

Column	Description
ID	Resource pool ID or VSCSI ID of VSCSI device.
GID	Resource pool ID.
VMNAME	Name of the resource pool.
VSCSINAME	Name of the VSCSI device.
NDK	Number of VSCSI devices
CMDS/s	Number of commands issued per second.
READS/s	Number of read commands issued per second.
WRITES/s	Number of write commands issued per second.
MBREAD/s	Megabytes read per second.
MBWRTN/s	Megabytes written per second.
LAT/rd	Average latency (in milliseconds) per read.
LAT/wr	Average latency (in milliseconds) per write.

Table A-14 displays the interactive commands you can use with the virtual machine storage panel.

Table A-14. Virtual Machine Storage Panel Interactive Commands

Command	Description
`e`	Expand or roll up storage VSCSI statistics. Allows you to view storage resource utilization statistics broken down by individual VSCSI devices belonging to a group. You are prompted to enter the group ID. The statistics are per VSCSI device.
`r`	Sort by READS/s column.
`w`	Sort by WRITES/s column.
`R`	Sort by MBREAD/s column.
`T`	Sort by MBWRTN/s column.
`N`	Sort first by VMNAME column, and then by VSCSINAME column. This is the default sort order.

Network Panel

The Network panel displays server-wide network utilization statistics.

Statistics are arranged by port for each virtual network device configured. For physical network adapter statistics, see the row in Table A-15 that corresponds to the port to which the physical network adapter is connected. For statistics on a virtual network adapter configured in a particular virtual machine, see the row corresponding to the port to which the virtual network adapter is connected.

Table A-15. Network Panel Statistics

Column	Description
PORT-ID	Virtual network device port ID.
UPLINK	Y means the corresponding port is an uplink. N means it is not.
UP	Y means the corresponding link is up. N means it is not.
SPEED	Link speed in Megabits per second.
FDUPLX	Y means the corresponding link is operating at full duplex. N means it is not.
USED-BY	Virtual network device port user.
DTYP	Virtual network device type. H means HUB and S means switch.
DNAME	Virtual network device name.
PKTTX/s	Number of packets transmitted per second.
PKTRX/s	Number of packets received per second.
MbTX/s	MegaBits transmitted per second.
MbRX/s	MegaBits received per second.
%DRPTX	Percentage of transmit packets dropped.
%DRPRX	Percentage of receive packets dropped.
TEAM-PNIC	Name of the physical NIC used for the team uplink.
PKTTXMUL/s	Number of multicast packets transmitted per second.
PKTRXMUL/s	Number of multicast packets received per second.
PKTTXBRD/s	Number of broadcast packets transmitted per second.
PKTRXBRD/s	Number of broadcast packets received per second.

Table A-16 displays the interactive commands you can use with the network panel.

Table A-16. Network Panel Interactive Commands

Command	Description
`T`	Sorts by Mb Tx column.
`R`	Sorts by Mb Rx column.
`t`	Sorts by Packets Tx column.
`r`	Sorts by Packets Rx column.
`N`	Sorts by PORT-ID column. This is the default sort order.
`L`	Changes the displayed length of the DNAME column.

Interrupt Panel

The interrupt panel displays information about the use of interrupt vectors.

Table A-17. Interrupt Panel Statistics

Column	Description
VECTOR	Interrupt vector ID.
COUNT/s	Total number of interrupts per second. This value is cumulative of the count for every CPU.
COUNT_x	Interrupts per second on CPU x.
TIME/int	Average processing time per interrupt (in microseconds).

Table A-17. Interrupt Panel Statistics (Continued)

Column	Description
TIME_x	Average processing time per interrupt on CPU x (in microseconds).
DEVICES	Devices that use the interrupt vector. If the interrupt vector is not enabled for the device, its name is enclosed in angle brackets (< and >).

Using Batch Mode

Batch mode allows you to collect and save resource utilization statistics in a file.

After you prepare for batch mode, you can use `esxtop` or `resxtop` in this mode.

Prepare for Batch Mode

To run in batch mode, you must first prepare for batch mode.

Procedure

1 Run `resxtop` (or `esxtop`) in interactive mode.

2 In each of the panels, select the columns you want.

3 Save this configuration to a file (by default `~/.esxtop41rc`) using the `W` interactive command.

You can now use `resxtop` (or `esxtop`) in batch mode.

Use esxtop or resxtop in Batch Mode

After you have prepared for batch mode, you can use `esxtop` or `resxtop` in this mode.

Procedure

1 Start `resxtop` (or `esxtop`) to redirect the output to a file.

For example:

```
esxtop -b > my_file.csv
```

The filename must have a `.csv` extension. The utility does not enforce this, but the post-processing tools require it.

2 Process statistics collected in batch mode using tools such as Microsoft Excel and Perfmon.

In batch mode, `resxtop` (or `esxtop`) does not accept interactive commands. In batch mode, the utility runs until it produces the number of iterations requested (see command-line option `n`, below, for more details), or until you end the process by pressing Ctrl+c.

Batch Mode Command-Line Options

You can use batch mode with command-line options.

The command-line options in Table A-18 are available in batch mode.

Table A-18. Command-Line Options in Batch Mode

Option	Description
a	Show all statistics. This option overrides configuration file setups and shows all statistics. The configuration file can be the default `~/.esxtop41rc` configuration file or a user-defined configuration file.
b	Runs `resxtop` (or `esxtop`) in batch mode.

Table A-18. Command-Line Options in Batch Mode (Continued)

Option	Description
`c` *filename*	Load a user-defined configuration file. If the `-c` option is not used, the default configuration filename is ~/.esxtop41rc. Create your own configuration file, specifying a different filename, using the `W` single-key interactive command.
`d`	Specifies the delay between statistics snapshots. The default is five seconds. The minimum is two seconds. If a delay of less than two seconds is specified, the delay is set to two seconds.
`n`	Number of iterations. `resxtop` (or `esxtop`) collects and saves statistics this number of times, and then exits.
`server`	The name of the remote server host to connect to (required, `resxtop` only).
`vihost`	If you connect indirectly (through vCenter Server), this option should contain the name of the ESX/ESXi host you connect to. If you connect directly to the ESX/ESXi host, this option is not used. Note that the host name needs to be the same as what appears in the vSphere Client.
`portnumber`	The port number to connect to on the remote server. The default port is 443, and unless this is changed on the server, this option is not needed. (`resxtop` only)
`username`	The user name to be authenticated when connecting to the remote host. You are prompted by the remote server for a password, as well (`resxtop` only).

Using Replay Mode

In replay mode, `esxtop` replays resource utilization statistics collected using `vm-support`.

After you prepare for replay mode, you can use esxtop in this mode. See the `vm-support` man page.

In replay mode, `esxtop` accepts the same set of interactive commands as in interactive mode and runs until no more snapshots are collected by `vm-support` to be read or until the requested number of iterations are completed.

Prepare for Replay Mode

To run in replay mode, you must prepare for replay mode.

Procedure

1 Run `vm-support` in snapshot mode on the ESX service console.

Use the following command.

```
vm-support -S -d duration -I interval
```

2 Unzip and untar the resulting tar file so that `esxtop` can use it in replay mode.

You can now use `esxtop` in replay mode.

Use esxtop in Replay Mode

You can use `esxtop` in replay mode.

You do not have to run replay mode on the ESX service console. Replay mode can be run to produce output in the same style as batch mode (see the command-line option `b`, below).

Procedure

◆ To activate replay mode, enter the following at the command-line prompt.

```
esxtop -R vm-support_dir_path
```

Replay Mode Command-Line Options

You can use replay mode with command-line options.

Table A-19 lists the command-line options available for `esxtop` replay mode.

Table A-19. Command-Line Options in Replay Mode

Option	Description
`R`	Path to the vm-support collected snapshot's directory.
`a`	Show all statistics. This option overrides configuration file setups and shows all statistics. The configuration file can be the default `~/.esxtop41rc` configuration file or a user-defined configuration file.
`b`	Runs `esxtop` in Batch mode.
`c`*filename*	Load a user-defined configuration file. If the `-c` option is not used, the default configuration filename is ~/.esxtop41rc. Create your own configuration file and specify a different filename using the `W` single-key interactive command.
`d`	Specifies the delay between panel updates. The default is five seconds. The minimum is two seconds. If a delay of less than two seconds is specified, the delay is set to two seconds.
`n`	Number of iterations `esxtop` updates the display this number of times and then exits.

Advanced Attributes

B

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 appendix includes the following topics:

- "Set Advanced Host Attributes," on page 111
- "Set Advanced Virtual Machine Attributes," on page 113

Set Advanced Host Attributes

You can set advanced attributes for a host.

CAUTION VMware recommends that only advanced users set advanced host attributes. In most cases, the default settings produce the optimum result.

Procedure

1 In the vSphere Client inventory panel, select the host to customize.

2 Click the **Configuration** tab.

3 In the **Software** menu, click **Advanced Settings**.

4 In the Advanced Settings dialog box select the appropriate item (for example, **CPU** or **Memory**), and scroll in the right panel to find and change the attribute.

Advanced Memory Attributes

You can use the advanced memory attributes to customize memory resource usage.

Table B-1. Advanced Memory Attributes

Attribute	Description	Default
Mem.CtlMaxPercent	Limits the maximum amount of memory reclaimed from any virtual machine using `vmmemctl`, based on a percentage of its configured memory size. Specify 0 to disable reclamation using `vmmemctl` for all virtual machines.	65
Mem.ShareScanTime	Specifies the time, in minutes, within which an entire virtual machine is scanned for page sharing opportunities. Defaults to 60 minutes.	60

Table B-1. Advanced Memory Attributes (Continued)

Attribute	Description	Default
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. Defaults to 4 MB/sec per 1GHz.	4
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.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.AllocGuestLargePage	Set this option to 1 to enable 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	Set these options to 1 to reduce memory fragmentation. If host memory is fragmented, the availability of host large pages is reduced. These options improve the probability of backing guest large pages with host large pages. 0 = disable.	1
Mem.MemZipEnable	Set this option to 1 to enable memory compression for the host. Set the option to 0 to disable memory compression.	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	Set this option to 1 to enable 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	Set this option to 1 to enable always try to allocate large pages for nested page tables (called 'RVI' by AMD or 'EPT' by Intel). 0= disable. 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.	1

Advanced NUMA Attributes

You can use the advanced NUMA attributes to customize NUMA usage.

Table B-2. Advanced NUMA Attributes

Attribute	Description	Default
Numa.RebalanceEnable	Set this option to 0 to disable all NUMA rebalancing and initial placement of virtual machines, effectively disabling the NUMA scheduling system.	1
Numa.PageMigEnable	If you set this option to 0, the system does not automatically migrate pages between nodes to improve memory locality. Page migration rates set manually are still in effect.	1
Numa.AutoMemAffinity	If you set this option to 0, the system does not automatically set memory affinity for virtual machines with CPU affinity set.	1
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.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.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
VMkernel.Boot.sharePerNode	Controls whether memory pages can be shared (de-duplicated) only within a single NUMA node or across multiple NUMA nodes. Unlike the other NUMA options, this option appears under "VMkernel" in the Advanced Settings dialog box. This is because, unlike the other NUMA options shown here which can be changed while the system is running, VMkernel.Boot.sharePerNode is a boot-time option that only takes effect after a reboot.	True (selected)

Set Advanced Virtual Machine Attributes

You can set advanced attributes for a virtual machine.

Procedure

1 Select the virtual machine in the vSphere Client inventory panel, and select **Edit Settings** from the right-click menu.

2 Click **Options** and click **Advanced > General**.

3 Click the **Configuration Parameters** button.

4 In the dialog box that appears, click **Add Row** to enter a new parameter and its value.

Advanced Virtual Machine Attributes

You can use the advanced virtual machine attributes to customize virtual machine configuration.

Table B-3. Advanced Virtual Machine Attributes

Attribute	Description
sched.mem.maxmemctl	Maximum amount of memory reclaimed from the selected virtual machine by ballooning, in megabytes (MB). If the ESX/ESXi host needs to reclaim additional memory, it is forced to swap. Swapping is less desirable than ballooning.
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.
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.
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.
numa.vcpu.maxPerMachineNode	Maximum number of the virtual machine's virtual CPUs that can reside on a single NUMA node. By default, the maximum is the number of physical cores present in a NUMA node.
numa.vcpu.maxPerClient	Maximum number of the virtual machine's virtual CPUs that are rebalanced as a single unit (NUMA client) by the NUMA scheduler. By default there is no limit and all virtual CPUs belong to the same NUMA client. However, if the number of virtual CPUs in a NUMA client exceeds the number of physical cores on the smallest NUMA node in the cluster, the client is not managed by the NUMA scheduler.
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. By default, the value is TRUE.

Index

A

admission control
- CPU **23**
- resource pools **51**
- with expandable resource pools **52**

advanced attributes
- hosts **111**
- memory **111**
- NUMA **113**
- Storage I/O Control **44**
- virtual machines **113**

affinity rules, creating **75, 77**

alarms **74**

AMD Opteron-based systems **85, 88, 113**

applications
- CPU-bound **18**
- single-threaded **18**

automation modes, virtual machines **61**

B

ballooning, memory **33**

Baseboard Management Controller (BMC) **71**

batch mode
- command-line options **108**
- preparing for **108**

C

cluster settings, affinity rules **75, 77**

CPU
- admission control **23**
- managing allocation **17, 18**
- overcommitment **17**

CPU affinity
- hyperthreading **21**
- NUMA nodes **90**
- potential issues **23**

CPU panel
- esxtop **96**
- resxtop **96**

CPU power efficiency **24**

CPU Power panel
- esxtop **99**
- resxtop **99**

CPU virtualization
- hardware-assisted **18**
- software-based **17**

CPU-bound applications **18**

custom automation mode **61**

D

DPM
- and admission control **16**
- automation level **73**
- enabling **73**
- Last Time Exited Standby **74**
- monitoring **74**
- overrides **74**
- threshold **73**

DRS
- disabling **62**
- fully automated **60**
- group power on **56**
- information **80**
- initial placement **55, 56**
- load balancing **55**
- manual **60**
- migration **55**
- migration recommendations **58**
- partially automated **60**
- single virtual machine power on **56**
- virtual machine migration **57**
- vMotion network **59**

DRS actions, history **81**

DRS Cluster Summary tab **79**

DRS clusters
- adding managed hosts **63**
- adding unmanaged hosts **64**
- as resource providers **9**
- creating **60**
- general information **79**
- managing resources with **63**
- prerequisites **59**
- processor compatibility **59**
- shared storage **59**
- shared VMFS volume **59**
- validity **66**
- viewing information **79**

DRS faults **81**

DRS groups **75, 76**

DRS migration threshold **58**

DRS recommendations
  priority **81**
  reasons **81**
DRS Resource Distribution Chart **80**
DRS tab
  Faults page **82**
  History page **83**
  Recommendations page **81**
  using **81**
DRS Troubleshooting Guide **82**
dual-processor virtual machine **17**
dynamic load balancing, NUMA **87**
Dynamic Voltage and Frequency Scaling (DVFS) **24**

E
educational support **7**
Enhanced vMotion Compatibility (EVC) **18, 59, 60, 79**
entering maintenance mode **65**
ESX/ESXi
  memory allocation **32**
  memory reclamation **33**
esxtop
  batch mode **108**
  common statistics description **95**
  CPU panel **96**
  CPU Power panel **99**
  interactive mode **94**
  interactive mode command-line options **94**
  interactive mode single-key commands **95**
  interrupt panel **107**
  memory panel **99**
  network panel **106**
  order pages **95**
  performance monitoring **93**
  replay mode **109**
  statistics column **95**
  storage adapter panel **103**
  storage device panel **104**
  virtual machine storage panel **106**
Exit Standby Error **74**
expandable reservations, example **52**

F
fully automated DRS **60**

G
grafted, resource pool **63, 64**
group power on **56**

H
home nodes, NUMA **87**
host-local swap
  DRS cluster **35**
  standalone host **35**
hosts
  adding to DRS clusters **63, 64**
  advanced attributes **111**
  as resource providers **9**
  entering maintenance mode **65**
  removing from a DRS cluster **65**
hyperthreading
  and ESX/ESXi **21**
  core sharing modes **22**
  CPU affinity **21**
  disabling **19**
  enabling **21**
  performance implications **20**
  quarantining **22**
  server configuration for **21**
hyperthreading modes **21**

I
IBM Enterprise X-Architecture **88**
idle memory tax **33**
iLO, configuring **71**
initial placement, NUMA **87**
Intelligent Platform Management Interface (IPMI), configuring **71**
invalid DRS clusters **70**

L
Last Time Exited Standby **74**
limit **12**
load balancing, virtual machines **57**
logical processors **19**
LPage.LPageAlwaysTryForNPT **111**
LPage.LPageDefragEnable **111**
LPage.LPageDefragRateTotal **111**
LPage.LPageDefragRateVM **111**

M
maintenance mode, entering **65**
manual DRS **60**
Mem.AllocGuestLargePage **111**
Mem.AllocUseGuestPool **111**
Mem.AllocUsePSharePool **111**
Mem.BalancePeriod **111**
Mem.CtlMaxPercent **111**
Mem.IdleTax **33, 111**
Mem.MemZipEnable **111**
Mem.MemZipMaxPct **111**
Mem.SamplePeriod **32, 111**

Mem.ShareScanGHz **36, 111**
Mem.ShareScanTime **36, 111**
memory
  advanced attributes **111**
  balloon driver **33**
  managing allocation **27, 30**
  overcommitment **28, 35**
  overhead **27**
  overhead, understanding **31**
  reclaiming unused **33**
  service console **27**
  sharing **28**
  sharing across virtual machines **36**
  virtual machines **33**
  virtualization **27**
memory affinity, NUMA nodes **90**
memory compression **37**
memory compression cache
  disabling **37**
  enabling **37**
  set size **37**
memory idle tax **33**
memory usage **38**
memory virtualization
  hardware-assisted **29**
  software-based **29**
migration recommendations **58**
monitoring, Storage I/O Control **42, 44**
monitoring software **74**
multicore processors **19**

N
NUMA
  advanced attributes **113**
  AMD Opteron-based systems **88**
  CPU affinity **90**
  description **85**
  dynamic load balancing **87**
  home nodes **87**
  home nodes and initial placement **87**
  IBM Enterprise X-Architecture **88**
  manual controls **89**
  memory page sharing **87**
  optimization algorithms **87**
  page migration **87**
  scheduling **86**
  supported architectures **88**
  transparent page sharing **87**
Numa.AutoMemAffinity **113**
Numa.MigImbalanceThreshold **113**
Numa.PageMigEnable **113**
Numa.RebalanceCoresNode **113**
Numa.RebalanceCoresTotal **113**
Numa.RebalanceEnable **113**
Numa.RebalancePeriod **113**

O
Opteron **88**
overcommitted DRS clusters **69**
overhead memory **27**

P
page migration, NUMA **87**
parent resource pool **47**
partially automated DRS **60**
performance, CPU-bound applications **18**
performance monitoring **93**
physical memory usage **31**
physical processors **19**
policies, CPU power management **24**
power management policies
  CPU **24**
  setting **24**
power on, single virtual machine **56**
processor-specific behavior **18**

Q
quarantining, hyperthreading **22**

R
red DRS clusters **70**
replay mode
  command-line options **110**
  preparing for **109**
reservation **11**
resource allocation settings
  changing **12**
  limit **12**
  reservation **11**
  shares **11**
  suggestions **12**
resource consumers **10**
resource management
  customizing **111**
  defined **9**
  goals **10**
  information **13**
resource pools
  adding virtual machines **50**
  admission control **51**
  advantages **48**
  attributes **49**
  changing attributes of **50**
  creating **49, 50**
  grafted **63, 64**

parent **47**
removing virtual machines **51**
root resource pool **47**
siblings **47**
resource providers **9**
resource types **9**
resxtop
batch mode **108**
common statistics description **95**
CPU panel **96**
CPU Power panel **99**
interactive mode **94**
interactive mode command-line options **94**
interactive mode single-key commands **95**
interrupt panel **107**
memory panel **99**
network panel **106**
options **94**
order pages **95**
performance monitoring **93**
statistics column **95**
storage adapter panel **103**
storage device panel **104**
virtual machine storage panel **106**
root resource pool **47**

S

sched.mem.maxmemctl **33, 114**
sched.mem.pshare.enable **114**
sched.swap.dir **114**
sched.swap.persist **114**
server configuration for hyperthreading **21**
service console, memory use **27**
shares, Storage I/O Control **41**
shares and limits, Storage I/O Control **42**
sharing memory **28**
siblings **47**
single virtual machine power on **56**
single-processor virtual machine **17**
single-threaded applications **18**
SMP virtual machines **18**
standby mode, Last Time Exited Standby **74**
statistics, esxtop **95**
statistics, resxtop **95**
Storage I/O Control
enabling **43**
limitations **41**
monitoring **42, 44**
requirements **41**
shares and limits **41–43**
threshold **44**
swap file
deleting **36**
location **34**
using **34**
swap space **35**
System Resource Allocation Table (SRAT) **86**

T

technical support **7**
threshold, Storage I/O Control **44**

U

updated information **5**

V

valid DRS clusters **67**
vCenter Server events **74**
virtual machine affinity **75–77**
virtual machine anti-affinity **75–77**
Virtual Machine File System (VMFS) **59, 114**
virtual machines
adding to DRS clusters **64**
adding to resource pools **50**
advanced attributes **113**
as resource consumers **10**
assigning to a specific processor **23**
automation modes **61**
configuration file **59**
memory **27, 33**
migration **57**
monitor **29**
number of virtual processors **18**
overhead memory **32**
removing from a DRS cluster **66**
removing from resource pools **51**
sharing memory across **36**
VM-Host affinity rule
conflicts **76**
using **76**
VM-VM affinity rule, conflicts **77**
vMA **94**
VMFS (Virtual Machine File System) **59, 114**
VMkernel.Boot.sharePerNode **87, 113**
VMM **27, 29**
vmmemctl
Mem.CtlMaxPercent **111**
sched.mem.maxmemctl **114**
VMware DRS, creating rules **75, 77**
VMware HA **13, 60, 63, 65, 79**
vSphere CLI **94**
vSphere Client **13, 15, 19, 31, 79, 81**
vSphere Management Assistant **94**

vSphere SDK **19**

W

wake protocols **71**
Wake-on-LAN (WOL), testing **72**
working set size **32**

Y

yellow DRS clusters **69**