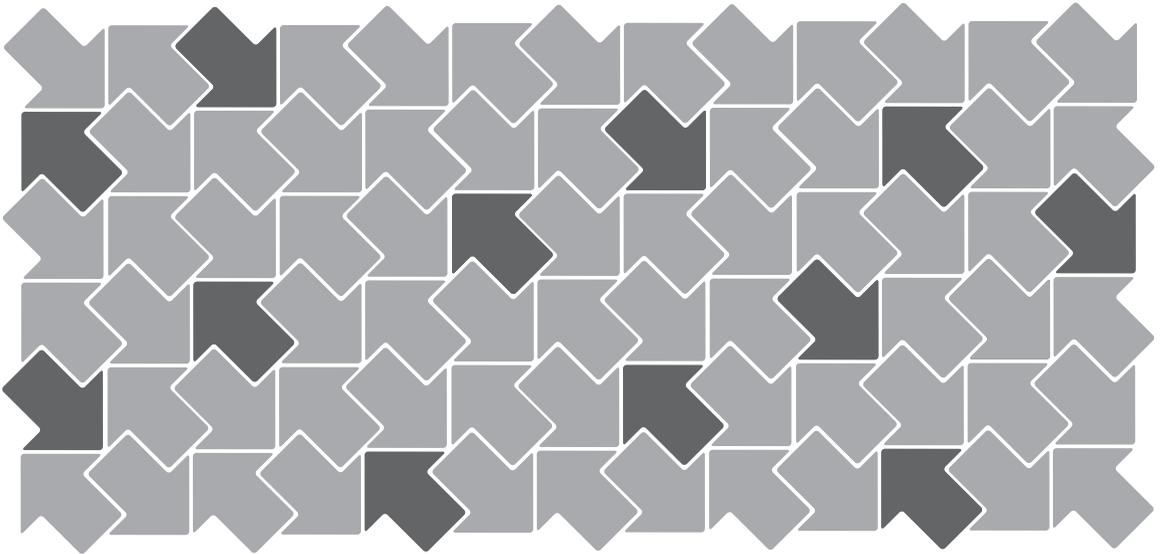


Resource Management Guide

ESX Server 3.0.1 and VirtualCenter 2.0.1



Resource Management Guide

Version: 2.0.1

Revision: 20090116

Item: VI-ENG-Q206-218

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

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

The VMware Web site also provides the latest product updates.

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

docfeedback@vmware.com

© 2006–2009 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>.

VMware, the VMware “boxes” logo and design, Virtual SMP and VMotion are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions.

All other marks and names mentioned herein may be trademarks of their respective companies.

VMware, Inc.

3401 Hillview Ave.

Palo Alto, CA 94304

www.vmware.com

Contents

1	Viewing Host Resource Information	13
	Understanding Virtual Machine Resource Allocation	18
	Reserving Host Resources	20
	Virtual Machine Attributes: Shares, Reservation, and Limit	20
	Shares	20
	Reservation	21
	Limit	21
	Admission Control	22
	Changing Virtual Machine Attributes	22
	Creating and Customizing Resource Pools	24
	Understanding Expandable Reservation	28
	Creating and Customizing Clusters	29
2	What Are Resources?	33
	Resource Providers and Consumers	33
	How ESX Server Manages Resources	35
	How Administrators Configure Resources	35
	Resource Utilization and Performance	36
	Understanding ESX Server Architecture	36
	VMkernel	36
	VMkernel Resource Manager	37
	VMkernel Hardware Interface Layer	37
	Virtual Machine Monitor	37
	Service Console	37
	How Administrators Can Affect CPU Management	38
	How Administrators Can Affect Memory Management	38
	Understanding CPU and Memory Virtualization	39
	CPU Virtualization Basics	39
	Memory Virtualization Basics	40
	Virtual Machine Memory	40
	Memory Overcommitment	41
	Memory Sharing	41

3	Introduction	43
	What Are Resource Pools?	44
	Why Use Resource Pools?	45
	Host Resource Pools and Cluster Resource Pools	46
	Resource Pool Admission Control	47
	Creating Resource Pools	48
	Understanding Expandable Reservations	50
	Expandable Reservations Example	50
	Viewing Resource Pool Information	51
	Resource Pool Summary Tab	51
	Resource Pool Resource Allocation Tab	52
	Changing Resource Pool Attributes	55
	Monitoring Resource Pool Performance	55
	Adding Virtual Machines to Resource Pools	56
	Removing Virtual Machines from Resource Pools	57
	Resource Pools and Clusters	57
	Clusters Enabled for DRS	58
	Clusters Not Enabled for DRS	59
4	Introduction to Clusters	61
	VMware DRS	62
	VMware HA	63
	Clusters and VirtualCenter Failure	64
	Understanding VMware DRS	65
	Initial Placement	66
	Virtual Machine Migration	66
	Migration Threshold	67
	Migration Recommendations	67
	DRS Clusters, Resource Pools, and ESX Server	68
	Maintenance Mode	68
	DRS Clusters and Maintenance Mode	69
	Understanding VMware HA	69
	Traditional and HA Failover Solutions	69
	Traditional Clustering Solutions	69
	VMware HA Solution	70
	VMware HA Features	71
	Failover Capacity	71
	Planning for HA Clusters	72
	VMware HA and Special Situations	73
	Primary and Secondary Hosts	74

	HA Clusters and Maintenance Mode	74
	HA Clusters and Disconnected Hosts	74
	HA Clusters and Host Isolation Timing Issue	75
	Using HA and DRS Together	76
	Valid, Yellow, and Red Clusters	76
	Valid Cluster	76
	Example 1: Valid Cluster, All Resource Pools of Type Fixed	77
	Example 2: Valid Cluster, Some Resource Pools of Type Expandable	78
	Yellow Cluster	79
	Red Cluster	80
	Red DRS Cluster	80
	Red HA Cluster	81
5	Cluster Prerequisites	83
	Clusters Enabled for HA	84
	VirtualCenter VMotion Requirements	84
	Shared Storage	84
	Shared VMFS Volume	84
	Processor Compatibility	85
	Other Requirements	85
	Cluster Creation Overview	86
	Creating a Cluster	87
	Choosing Cluster Features	87
	Selecting Automation Level	87
	Selecting High Availability Options (HA)	88
	Finishing Cluster Creation	88
	Viewing Cluster Information	89
	Summary Page	89
	DRS Resource Distribution Charts	91
	Migration Page	91
	Migration History	93
6	Introduction	95
	Adding Hosts to a DRS Cluster	96
	Adding Managed Hosts to a Cluster	96
	Adding Unmanaged Hosts to a Cluster	97
	Removing Hosts from Clusters	97
	Host Removal and Resource Pool Hierarchies	98
	Host Removal and Virtual Machines	98
	Host Removal and Invalid Clusters	99

	Applying DRS Migration Recommendations	99
	Reconfiguring DRS	100
	Using DRS Affinity Rules	101
	Understanding Rule Results	103
	Disabling or Deleting Rules	103
7	Introduction	105
	Adding Hosts to an HA Cluster	106
	Adding Managed Hosts to a Cluster	106
	Adding Unmanaged Hosts to a Cluster	106
	Results of Adding Hosts to a Cluster	107
	Configuring and Unconfiguring HA on a Host	108
	Working with VMware HA	108
8	Adding Virtual Machines to a Cluster	111
	Adding a Virtual Machine During Creation	111
	Migrating a Virtual Machine to a Cluster	112
	Adding a Host with Virtual Machines to a Cluster	112
	Powering On Virtual Machines in a Cluster	112
	DRS Enabled	112
	HA Enabled	113
	Removing Virtual Machines from a Cluster	113
	Migrating Virtual Machines out of a Cluster	113
	Removing a Host with Virtual Machines from a Cluster	114
	Customizing DRS for Virtual Machines	114
	Customizing HA for Virtual Machines	115
9	CPU Virtualization	118
	Virtualization Modes and Virtualization Overhead	118
	Performance Implications	119
	Virtualization and Processor-Specific Behavior	119
	Performance Implications	119
	Using CPU Affinity to Assign Virtual Machines to Specific Processors	120
	Potential Issues with Affinity	121
	Hyperthreading	122
	Introduction	122
	Enabling Hyperthreading	122
	Hyperthreading and ESX Server	123
	Advanced Server Configuration for Hyperthreading	123
	Quarantining	125

	Caveats: Hyperthreading and CPU Affinity	125
	Virtual Memory in Virtual Machines	125
	Virtual to Physical Memory Mapping	126
	Performance Implications	127
	Understanding Memory Overhead	128
	Memory Allocation and Idle Memory Tax	130
	How ESX Server Hosts Allocate Memory	130
	How Host Memory Is Used	130
	Memory Tax for Idle Virtual Machines	131
	How ESX Server Hosts Reclaim Memory	132
	Memory Balloon (vmmemctl) Driver	132
	Swap Space and Guest Operating Systems	133
	Swapping	134
	Swap Space and Memory Overcommitment	135
	Swap Files and ESX Server Failure	135
	Sharing Memory Across Virtual Machines	135
	Advanced Attributes and What They Do	136
	Setting Advanced Host Attributes	136
	Setting Advanced Virtual Machine Attributes	140
10	Resource Management Best Practices	143
	Creating and Deploying Virtual Machines	144
	Planning	144
	Creating Virtual Machines	144
	Deploying the Guest Operating System	145
	Deploying Guest Applications	145
	Configuring VMkernel Memory	145
A	Introduction	147
	What Is NUMA?	148
	NUMA Challenges for Operating Systems	148
	ESX Server NUMA Scheduling	149
	VMware NUMA Optimization Algorithms	150
	Home Nodes and Initial Placement	150
	Dynamic Load Balancing and Page Migration	151
	Transparent Page Sharing Optimized for NUMA	152
	Manual NUMA Controls	152
	IBM Enterprise X-Architecture Overview	153
	AMD Opteron-Based Systems Overview	153
	Retrieving NUMA Configuration Information	154

- Obtaining NUMA Statistics 154
- Determining the Amount of Memory for Each NUMA Node 155
- Determining the Amount of Memory for a Virtual Machine 155
- CPU Affinity for Associating Virtual Machines with a Single NUMA Node 155
- Memory Affinity for Associating Memory Allocations with a NUMA Node 156
- Example: Binding a Virtual Machine to a Single NUMA Node 157

- B** Using the esxtop Utility for Performance Monitoring 159
 - Configuration File 160
- Using the esxtop Utility in Interactive Mode 160
 - Interactive Mode Command-Line Options 160
 - Common Statistics Description 161
 - Interactive Mode Single-Key Commands 161
 - Statistics Columns and Order Pages 162
 - CPU Panel 163
 - Memory Panel 166
 - Memory Panel Interactive Commands 170
 - Storage Panel 170
 - Storage Panel Interactive Commands 171
 - Network Panel 173
 - Network Panel Statistics 173
 - Network Panel Interactive Commands 174
- Using the esxtop Utility in Batch Mode 174
 - Batch Mode Command-Line Options 175
- Using the esxtop Utility in Replay Mode 175
 - Replay Mode Command-Line Options 176

Preface

This preface describes the contents of the *Resource Management Guide* and provides pointers to VMware® technical and educational resources.

This preface contains the following topics:

- [“About This Book”](#) on page 9
- [“Technical Support and Education Resources”](#) on page 12

About This Book

The *Resource Management Guide* discusses resource management for Virtual Infrastructure environments. Its focus is on the following major topics:

- Resource allocation and resource management concepts
- Virtual machine attributes and admission control
- Resource pools and how to manage them
- Clusters, VMware DRS, VMware HA, and how to work with them
- Advanced resource management options
- Performance considerations

Revision History

This manual is revised with each release of the product or when necessary. A revised version can contain minor or major changes. [Table P-1](#) provides you with the revision history of this manual.

Table P-1. Revision History

Revision	Description
20060615	ESX Server 3.0 and VirtualCenter 2.0 version of the VMware Infrastructure <i>Resource Management Guide</i> . This is the first edition of this manual.
20060921	ESX Server 3.0.1 and VirtualCenter 2.0.1 version of the VMware Infrastructure 3 <i>Resource Management Guide</i> .

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 HA. Some of the information is relevant for all administrators of ESX Server 3 and VirtualCenter 2. Other sections and chapters are relevant only if you are using clusters enabled for DRS or HA, with the appropriate license.

This manual assumes you have a working knowledge of ESX Server and of the VirtualCenter Management Server.

Document Feedback

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

docfeedback@vmware.com

VMware Infrastructure Documentation

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

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

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

Conventions

[Table P-2](#) illustrates the typographic conventions used in this manual.

Table P-2. Conventions Used in This Manual

Style	Elements
Blue (online only)	Cross-references and email addresses
Blue boldface (online only)	Links
Black boldface	User interface elements such as button names and menu items
Monospace	Commands, filenames, directories, and paths
Monospace bold	User input
<i>Italic</i>	Document titles, glossary terms, and occasional emphasis
< Name >	Variable and parameter names

Abbreviations Used in Graphics

The graphics in this manual use the abbreviations listed in [Table P-3](#).

Table P-3. Abbreviations

Abbreviation	Description
VC	VirtualCenter
VI	Virtual Infrastructure Client
server	VirtualCenter Server
database	VirtualCenter database
host n	VirtualCenter managed hosts
VM#	Virtual machines on a managed host
user#	User with access permissions
dsk#	Storage disk for the managed host
datastore	Storage for the managed host
SAN	Storage area network type datastore shared between managed hosts
tplt	Template

Technical Support and Education Resources

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

Self-Service Support

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

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

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

Online and Telephone Support

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

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

Support Offerings

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

VMware Education Services

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

Resource Management Jumpstart

1

This chapter introduces basic resource management concepts using a simple example. The chapter steps you through resource allocation, first in a single-host environment, then in a more complex multi-host environment, as follows:

- [“Viewing Host Resource Information”](#) on page 13
- [“Understanding Virtual Machine Resource Allocation”](#) on page 18
- [“Changing Virtual Machine Attributes”](#) on page 22
- [“Creating and Customizing Resource Pools”](#) on page 24
- [“Understanding Expandable Reservation”](#) on page 28
- [“Creating and Customizing Clusters”](#) on page 29

Viewing Host Resource Information

In this section, you explore a host’s resources and learn how to find out who uses them.

Assume that a system administrator for a small company has just set up two virtual machines, VM-QA and VM-Marketing, on an ESX Server host.

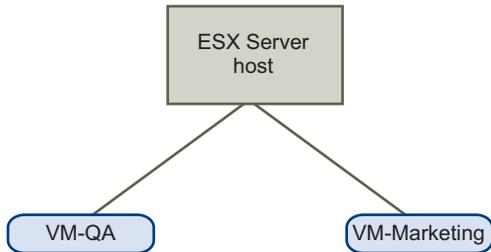


Figure 1-1. Single Host with Two Virtual Machines

To view information about a host

- 1 Start a Virtual Infrastructure Client (VI Client) and connect to a VirtualCenter Management Server (VirtualCenter Server).

NOTE You can also perform many of the tasks in this chapter using a VI Client connected to an ESX Server system or a Virtual Infrastructure Web Access Client connected to a server.

- 2 In the inventory panel on the left, select the host. With the Summary tab selected (the default), the panels display the following information about the host:

- General panel Shows information about processors, processor type, and so on.
- Commands panel Allows you to select commands to execute for the selected host.
- Resources panel Shows information about the total resources of the selected host. This panel includes information about the datastores connected to the host.

The screenshot displays the VMware Host Agent interface for a host named 'vcy169.eng.vmware.com'. The interface is divided into several panels:

- General Panel:**
 - Manufacturer: **Dell Computer Corporation**
 - Model: **PowerEdge 1850**
 - Processors: **2 CPU x 3.192 GHz, HT enabled**
 - Processor Type: **Intel(R) Xeon(TM) CPU 3.20GHz**
 - Number of Nics: **3**
 - State: **connected**
 - Virtual Machines: **6**
 - VMotion Enabled: **no**
 - Active Tasks: **no**
- Resources Panel:**
 - CPU usage: **47 MHz** (2 x 3 GHz)
 - Memory usage: **7 MB** (8 GB)
 - Datastore Table:**

Datastore	Capacity
storage1	60.25 GB
VCIUQA_Shared_5...	49.75 GB
VCIUQA_Shared_2...	199.75 GB
private_lun	99.75 GB
 - Network:** VM Network
- Commands Panel:**
 - New Virtual Machine
 - New Resource Pool
 - Enter Maintenance Mode
 - Reboot

- 3 For more detailed information about available memory, click the **Configuration** tab, then select **Memory**. The display lists total resources, how much is used by virtual machines, and how much is used by the service console.

The screenshot shows the VMware ESX Server 3.0.0, 23269 Configuration tab, Memory section. The left sidebar shows the navigation menu with 'Memory' selected under 'Hardware'. The main content area displays the 'Physical' memory usage:

Physical	
Total	8.00 GB
System	777.75 MB
Virtual Machines	6.97 GB
Service Console	272.00 MB

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.

- 4 For more detailed information on how the two virtual machines use the host's resources, click the **Resource Allocation** tab.

The screenshot shows the VMware ESX Server 3.0.0, 23269 Resource Allocation tab. The top summary shows:

CPU Reservation:	5426 MHz	Memory Reservation:	7142 MB
CPU Reservation Used:	0 MHz	Memory Reservation Used:	0 MB
CPU Reservation Unused:	5426 MHz	Memory Reservation Unused:	7142 MB

Below the summary, the 'View' section has 'CPU' and 'Memory' buttons. The table below shows the resource allocation for two virtual machines:

Name	Reservation - MHz	Limit - MHz	Shares	Shares Value	% Shares	Type
VM-Marketing	0	Unlimited	Normal	1000	50	N/A
VM-QA	0	Unlimited	Normal	1000	50	N/A

- 5 Look first at the information at the top. You can see the **CPU Reservation** and **Memory Reservation** (previously called minimum), how much of the reservation is used, and how much is available.

NOTE In the illustration above, no virtual machines are running, so no CPU or memory is used. You will later revisit this tab after powering on a virtual machine.

The fields display the following information:

Table 1-1. (SEE UPDATE)

Field	Description
CPU Reservation	Total CPU resources available for this host.
CPU Reservation Used	Total CPU resources of this host that are reserved by running virtual machines. Note: Virtual machines that aren't powered on don't consume CPU resources. For powered-on virtual machines, the system reserves CPU resources according to each virtual machine's Reservation setting.
CPU Reservation Unused	Total CPU resources of this host that are not currently reserved. Consider a virtual machine with reservation=2GHz that is totally idle. It has 2GHz reserved, but it is not actually using any of its reservation. <ul style="list-style-type: none"> ■ Other virtual machines <i>cannot reserve</i> these 2GHz. ■ Other virtual machines <i>can use</i> these 2GHz, that is, idle CPU reservations are not wasted.
Memory Reservation	Total memory resources available for this host.
Memory Reservation Used	Total memory resources of this host that are reserved by a running virtual machine and virtualization overhead. Note: Virtual machines that aren't powered on don't consume memory resources. For powered-on virtual machines, the system reserves memory resources according to each virtual machine's Reservation setting and overhead.
Memory Reservation Unused	Total memory resources of this host that are not currently reserved. See "CPU Reservation Unused" on page 17 for a discussion.

- 6 Next, look at the information about the virtual machines. You can click the **Memory** or **CPU** button depending on the information in which you're interested.

View: CPU Memory

Name	Reservation - MHz	Limit - MHz	Shares	Shares Value	% Shares	Type
 VM-Marketing	0	Unlimited	Normal	1000	50	N/A
 VM-QA	0	Unlimited	Normal	1000	50	N/A

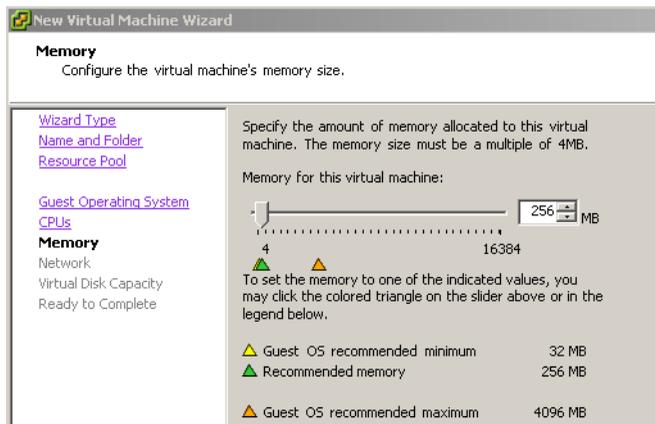
Table 1-2.

Field	Description
Name	Name of the virtual machine.
Reservation — MHz/MB	Amount of CPU or memory specified as reservation for this virtual machine. By default, no reservation is specified and 0 is displayed. See “Reservation” on page 21.
Limit	Amount of CPU or memory specified as upper limit for this virtual machine. By default, no limit is specified and Unlimited is displayed. See “Limit” on page 21.
Shares	Shares specified for this virtual machine. Each virtual machine is entitled to resources in proportion to its specified shares, bounded by its reservation and limit. A virtual machine with twice as many shares as another is entitled to twice as many resources. Shares default to Normal. For detailed information, see “Shares” on page 20.
Shares Value	Number of shares allocated to this virtual machine. See “Shares” on page 20.
% Shares	Percentage of shares allocated to this virtual machine.
Type	For resource pools, either Expandable or Fixed . See “Understanding Expandable Reservation” on page 28.

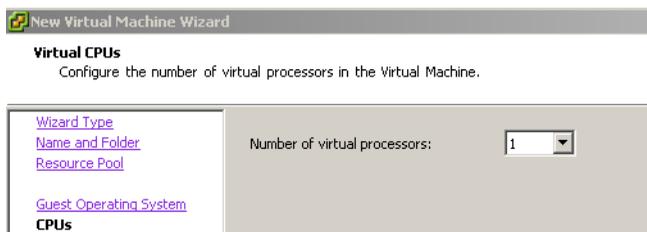
Understanding Virtual Machine Resource Allocation

When you create a virtual machine, the New Virtual Machine wizard prompts you for the memory size for this virtual machine. This amount of memory is the same as the amount of memory you install in a physical machine.

NOTE The ESX Server host makes this memory available to virtual machines. The host allocates the number of MB specified by the reservation directly to the virtual machine. Anything beyond the reservation is allocated using the host's physical resources or, when physical resources aren't available, handled using special techniques such as ballooning or swapping. See [“How ESX Server Hosts Reclaim Memory”](#) on page 132 for more information.



The system also prompts for the number of virtual processors (CPUs) if the operating system you have chosen supports more than one.



When CPU resources are overcommitted, the ESX Server host time-slices the physical processors across all virtual machines so each virtual machine runs as if it has the specified number of processors.

When an ESX Server host runs multiple virtual machines, it allocates each virtual machine a fair share of the physical resources. By default, all virtual machines associated with the same host receive:

- An equal share of CPU per virtual CPU. That means single-processor virtual machines are assigned only half of the resources of a dual-processor virtual machine.
- An equal share per MB of virtual memory size. That means an 8GB virtual machine is entitled to eight times as much memory as a 1GB virtual machine.

Reserving Host Resources

In some situations, system administrators want to know that a certain amount of memory for a virtual machine comes directly from the physical resources of the ESX Server machine. Similarly, the administrator might want to guarantee that a certain virtual machine always receives a higher percentage of the physical resources than other virtual machines.

You can reserve physical resources of the host using each virtual machine's attributes, discussed in the next section.

NOTE In many cases, it makes sense to use the default settings. See [Chapter 10, "Best Practices,"](#) on page 143 for information on how to best use custom resource allocations.

Virtual Machine Attributes: Shares, Reservation, and Limit

For each virtual machine, you can specify shares, reservation (minimum), and limit (maximum). This section explains what it means to specify these attributes.

Shares

Shares specify the **relative priority** or importance of a virtual machine. If a virtual machine has twice as many shares of a resource as another virtual machine, it is entitled to consume twice as much of that resource. Shares are typically specified as high, normal, or low. High, normal, and low specify share values with a 4:2:1 ratio. You can also choose **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. See ["What Are Resource Pools?"](#) on page 44 for an explanation of the hierarchy and sibling concepts.

When you assign shares to a virtual machine, you always specify the relative priority for that virtual machine. This is like handing out pieces of a pie: assume you get one piece of a pie and your sister gets two pieces of pie. How much pie each of you actually gets depends completely on the size of the pie and on the total number of pieces of the pie.

By default, you can choose high, normal, and low. High means twice as many shares as normal, and normal means twice as many shares as low.

Share values default to:

- **High** — 2000 shares per virtual CPU, 20 shares per MB of virtual machine memory

- **Normal** — 1000 shares per virtual CPU, 10 shares per MB of virtual machine memory
- **Low** — 500 shares per virtual CPU, 5 shares per MB of virtual machine memory

You can also specify a custom share value.

For example, an SMP virtual machine with two virtual CPUs and 1GB RAM with CPU and memory shares set to **Normal** has $2 \times 1000 = 2000$ shares of CPU and $10 \times 1024 = 10240$ shares of memory.

NOTE Virtual machines with more than one virtual CPU are called SMP (symmetric multiprocessing) virtual machines.

The amount of resources represented by each share changes when a new virtual machine is powered on. This affects all virtual machines. For example:

- Two virtual machines run on a host with 8GHz. Both are set to **Normal** and get 4GHz each.
- A third virtual machine is powered on. It 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.

Reservation

Reservation specifies the **guaranteed reservation** for a virtual machine. The server allows you to power on a virtual machine only if the CPU and memory reservation is available. The server guarantees that amount even when the physical server is heavily loaded. The reservation is expressed in concrete units (MHz or MB). When resources are not used, the ESX Server host makes them available to other virtual machines.

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, then VM2 can use 1.5GHz.

Reservation defaults to 0. It is often a good idea to specify a reservation to guarantee that the necessary CPU or memory are always available for the virtual machine.

Limit

Limit specifies the **upper limit** for CPU or memory for a virtual machine. A server can allocate more than the reservation to a virtual machine, but never allocates more than the limit. The limit is expressed in concrete units (MHz or MB).

CPU and memory limit default to unlimited. In that case, the amount of memory you assigned to the virtual machine when you created it becomes the virtual machine's memory.

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

- **Benefits** — Assigning a limit can be useful if you start with a small number of virtual machines and want to manage user expectations. Performance will deteriorate somewhat 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. You should therefore specify the limit only if there are good reasons for doing so.

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 has been 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.

Changing Virtual Machine Attributes

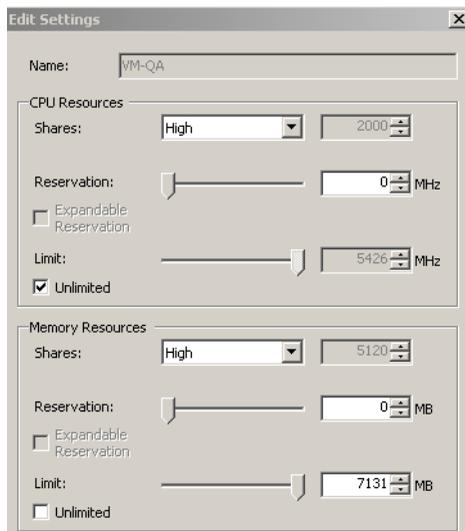
Earlier in this chapter, you viewed hosts and virtual machines and their resource allocation. You did not specify shares, reservation, and limit for the virtual machines. In this example, assume:

- The QA virtual machine is memory intensive. You therefore want to specify that, when system memory is overcommitted, VM-QA can use twice as much memory and CPU as the Marketing virtual machine. You can do that by setting memory shares to **High**.
- You want to make sure that the Marketing virtual machine has a certain amount of guaranteed CPU resources. You can do so using a Reservation setting.

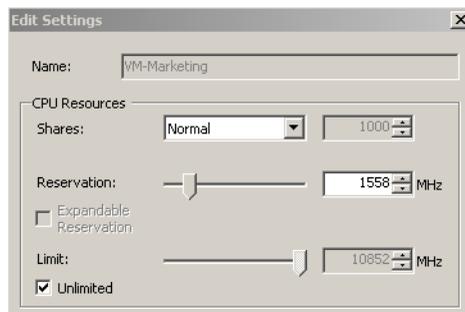
To edit a virtual machine's resource allocation

- 1 Start a VI Client and connect to a VirtualCenter Server.
- 2 Select the host in the inventory panel and choose the **Resource Allocation** tab.

- 3 Right-click **VM-QA**, the virtual machine for which you want to change shares, and choose **Edit Resource Settings**.
- 4 In the CPU Resources panel, choose **High** from the Shares drop-down menu.
- 5 Repeat these steps in the **Memory Resources** panel, then click **OK**.



- 6 Right-click the marketing virtual machine (**VM-Marketing**).
- 7 Move the slider in the **Reservation** field to the desired number, then click **OK**.



- 8 Click **OK** when you're done.

Now when you select the host's **Resource Allocation** tab and click **CPU**, you see that shares for VM-QA are twice that of the other virtual machine.

Summary | Virtual Machines | **Resource Allocation** | Performance | Configuration | Tasks & Events | Alarms

CPU Reservation: **5426 MHz** Memory Reservation: **7142 MB**
 CPU Reservation Used: **0 MHz** Memory Reservation Used: **0 MB**
 CPU Reservation Unused: **5426 MHz** Memory Reservation Unused: **7142 MB**

View: CPU Memory

Name	Reservation - MHz	Limit - MHz	Shares	Shares Value	% Shares	Type
VM-Marketing	1612	Unlimited	Normal	1000	33	N/A
VM-QA	0	Unlimited	High	2000	66	N/A

↑
VM-QA has twice the number of CPU shares

Notice that because the virtual machines have not been powered on, the **Reservation Used** fields have not changed.

- Power on one of the virtual machines and see how the **CPU Reservation Used** and **CPU Unreserved** fields change.

Summary | Virtual Machines | **Resource Allocation** | Performance | Configuration | Tasks & Events | Alarms

CPU Reservation: **5426 MHz** Memory Reservation: **7129 MB**
 CPU Reservation Used: **1558 MHz** Memory Reservation Used: **84.88 MB**
 CPU Unreserved: **3868 MHz** Memory Unreserved: **7044.12 MB**

View: CPU Memory

Name	Reservation - MHz	Limit - MHz	Shares	Shares Value	% Shares
VM-QA	0	Unlimited	High	2000	13
VM-Marketing	1558	Unlimited	Normal	1000	6

Creating and Customizing Resource Pools

As companies grow, they can afford faster and better systems and allocate more resources to the different departments. In this section, you learn how you can use resource pools to divide a host's resources. Resource pools can also be used in conjunction with VMware clusters, where they allow you to manage the resources of all hosts in a cluster as one pool of resources. Clusters are discussed in the next section, [“Creating and Customizing Clusters”](#) on page 29.

When you create a resource pool, you can specify the following attributes:

- Reservation, limit, and shares work just as they do for virtual machines. See [“Changing Virtual Machine Attributes”](#) on page 22.

- The **Reservation Type** attribute allows you to set up the resource pool so the pool can reserve available resources from its parent if it doesn't have enough resources available locally. See [“Understanding Expandable Reservation”](#) on page 28.

See [Chapter 3, “Understanding and Managing Resource Pools,”](#) on page 43 for more information.

Assume you continue with the example from above. You've decided you no longer want to assign one virtual machine each to your QA and Marketing departments but want to give each department a predefined chunk of resources. Depending on departmental needs, the department administrator can then create virtual machines for the department.

For example, if you started with a host that provides 6GHz of CPU and 3GB of memory, you could decide to choose shares allocations of **High** for RP-QA and shares allocations of **Normal** for RP-Marketing. That results in approximately 4GHz and 2GB of memory for RP-QA, and 2GHz and 1GB for RP-Marketing. Those resources are then available to the virtual machines in the respective resource pools.

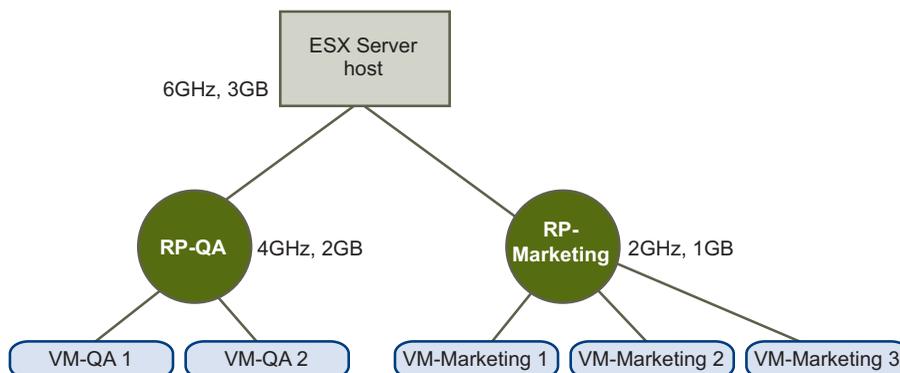
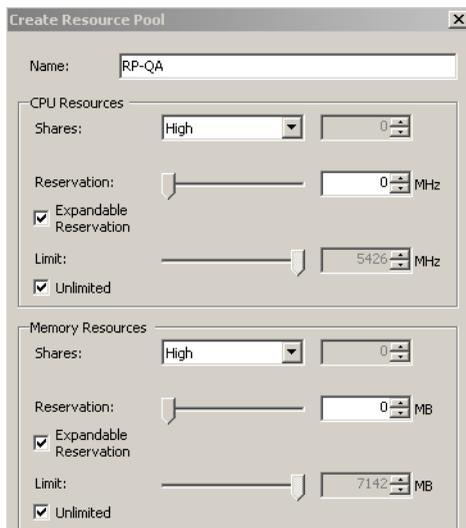


Figure 1-2. ESX Server Host with Two Resource Pools

To create and customize resource pools

- 1 Start a VI Client and connect to a VirtualCenter Server.
- 2 In the inventory panel on the left, select a host and choose **New Resource Pool** in the Commands panel on the right.
- 3 In the dialog box that appears, name the resource pool (for example, RP-QA).

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



- Create a second resource pool, RP-Marketing:
 - Leave **Shares** at **Normal** for both CPU and memory.
 - Specify a Reservation for both CPU and memory.
 - Click **OK** to exit.
- Select the host in the inventory panel and click the **Resource Allocation** tab.

The resource pools have been added to the display. In the top panel, the Reservation for the second resource pool has been subtracted from the unreserved resources. In the second panel, resource pool information, including the resource pool type, is now available.

Resource pool reservation
has been subtracted from total

The screenshot shows the VMware ESX Server interface with the 'Resource Allocation' tab selected. It displays the following statistics:

- CPU Reservation: **5426 MHz**
- CPU Reservation Used: **2256 MHz**
- CPU Reservation Unused: **3170 MHz**
- Memory Reservation: **7142 MB**
- Memory Reservation Used: **0 MB**
- Memory Reservation Unused: **7142 MB**

Below the statistics is a table of resource pools. The table has columns for Name, Reservation - MHz, Limit - MHz, Shares, Shares Value, % Shares, and Type. Two rows are highlighted with arrows: 'VM-Marketing' and 'RP-QA'.

Name	Reservation - MHz	Limit - MHz	Shares	Shares Value	% Shares	Type
VM-Marketing	1612	Unlimited	Normal	1000	6	N/A
VM-QA	0	Unlimited	High	2000	13	N/A
RP-QA	0	Unlimited	High	8000	53	Expa...
RP-Marketing	2256	Unlimited	Normal	4000	26	Expa...

Here's a summary of the values you can specify for a resource pool.

Table 1-3. Resource Pool Attributes

Field	Description
CPU Shares	Allows you to specify the shares for this resource pool. The basic principles are the same as for virtual machines, discussed in “ Shares ” on page 20.
Memory Shares	
Reservation	Displays the amount of CPU or memory the host reserves for this resource pool. Defaults to 0. A non-zero reservation is subtracted from the unreserved resources of the parent (host or resource pool). The resources are considered reserved, regardless of whether virtual machines are associated with the resource pool.
Expandable reservation	If this check box is selected (the default), and if the resource pool needs to make a reservation that is higher than its own reservation (for example, to power on a virtual machine), then the resource pool can use resources of a parent and reserve those resources. See “ Understanding Expandable Reservation ” on page 28.
Limit	Displays the upper limit on the CPU or memory that the host allocates to the selected resource pool. Default is unlimited. This default avoids wasting idle resources. Deselect the Unlimited check box to specify a different limit. Resource pool limits are useful, for example, if you want to assign a certain amount of resources to a group administrator. The group administrator can then create virtual machines for the group as needed, but never use more resources than specified by the limit.

After the resource pools have been created, you can add virtual machines to each resource pool. The shares you allocate for virtual machines are now relative to the resource pool's total shares.

NOTE After you've added virtual machines to the resource pool, you can select the resource pool's **Resource Allocation** tab for information on reserved and unreserved resources.

Understanding Expandable Reservation

How expandable reservations work is easiest to understand using an example.

Assume the following scenario:

- 1 Parent pool RP-MOM has a reservation of 6GHz, and one running virtual machine VM-M1 that reserves 1GHz.
- 2 You create a child resource pool RP-KID with a reservation of 2GHz and with **Expandable Reservation** selected.
- 3 You add two virtual machines, VM-K1 and VM-K2, with reservations of 2GHz each to the child resource pool and attempt to power them on.

Reservations are checked only when you power on a virtual machine.

- 4 VM-K1 can reserve the resources directly from RP-KID (which has 2GHz).
- 5 There are no local resources 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.

NOTE The reservation in the resource pool is meant to be used by the virtual machine and does not count as an extra reservation.

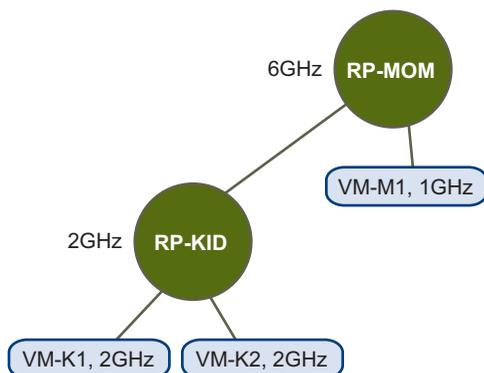


Figure 1-3. Admission Control with Expandable Resource Pools — 1

Now, the scenario changes:

- 1 You power on two virtual machines in RP-MOM with a total reservation of 3GHz.
- 2 You can still power on VM-K1 in RP-KID because 2GHz are available locally.
- 3 When you try to power on VM-K2, the system finds that 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 the second virtual machine.

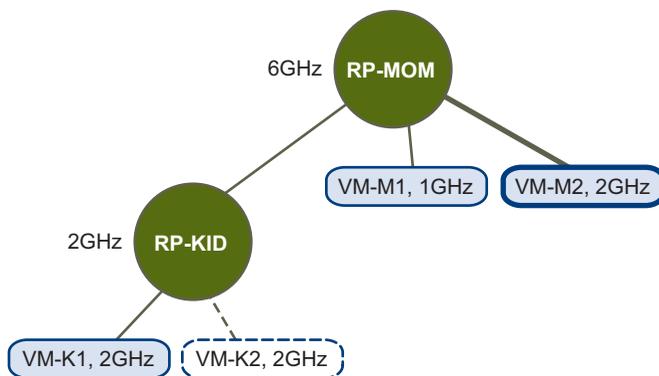


Figure 1-4. Admission Control with Expandable Resource Pools — 2

Creating and Customizing Clusters

In the previous section, you set up two resource pools that shared the resources of a single host. A cluster pools a set of hosts. If DRS is enabled, the cluster supports shared resource pools and performs placement and dynamic load balancing for virtual

machines in the cluster. If HA is enabled, the cluster supports failover. When a host fails, all associated virtual machines are restarted on different hosts.

NOTE You must be licensed to use cluster features. See the *Installation and Upgrade Guide* for information on licensing.

This section steps you through creating a cluster, and explains basic cluster functionality. Focus is on the default behavior of basic clusters. For more information, see the following chapters:

- [Chapter 4, “Understanding Clusters,”](#) on page 61
- [Chapter 5, “Creating a VMware Cluster,”](#) on page 83
- [Chapter 6, “Managing VMware DRS,”](#) on page 95
- [Chapter 7, “Managing VMware HA,”](#) on page 105

Assume you have a cluster that consists of three physical hosts. Each host provides 3GHz and 1.5GB, with a total of 9GHz and 4.5GB available. If you enable the cluster for DRS, you can create resource pools with different reservation or shares to group virtual machines with similar allocation requirements.

For DRS-enabled clusters, the system places virtual machines on the most suitable physical hosts (or makes recommendations for placement) when virtual machines are powered on. The exact behavior depends on the cluster’s automation level.

To create and customize a cluster

- 1 Start a VI Client and connect to a VirtualCenter Server.
- 2 In the inventory panel on the left, right-click a datacenter, and choose **New Cluster**.
- 3 Name the cluster. For this example, enable it for both HA and DRS.
- 4 Keep the default, fully automated, for DRS.
- 5 Keep the defaults for host failures and admission control for HA.
- 6 Click **Finish**. The VirtualCenter Server creates a new cluster with the specified attributes.

For detailed information on DRS, HA, and available attributes, see [Chapter 5, “Creating a VMware Cluster,”](#) on page 83.

The next task is to add a number of hosts to the cluster. Using clusters enabled for DRS makes sense even if you have only two hosts in the cluster.

For an HA cluster, the number of hosts to add depends on the number of host failures the cluster should tolerate, and on the number of powered on virtual machines you are

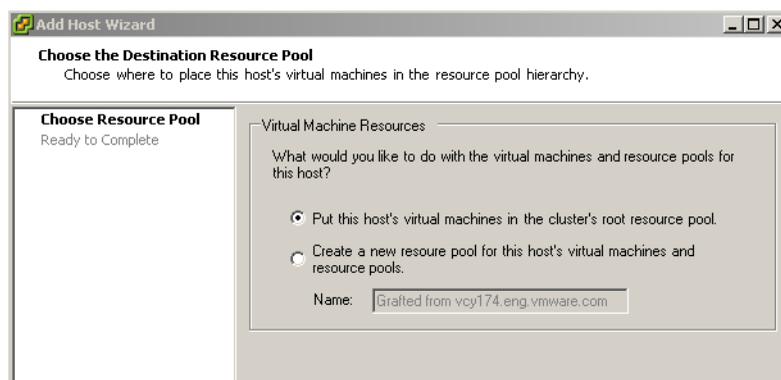
planning for HA can support a maximum of 4 concurrent host failures. In the following steps, you add a host to the cluster that is managed by the same VirtualCenter Server.

To add a host to the cluster

- 1 In the left panel of the VI Client, select the host and drag it over the cluster's icon.

If the cluster is enabled for DRS, you are prompted whether you want to add the host's virtual machines directly to the cluster's (invisible) root resource pool or whether you want to create a new resource pool to represent that host. The root resource pool is at top level and is not displayed because the resources are effectively owned by the cluster.

If the cluster is not enabled for DRS, all resource pools are removed.



- 2 Choose the appropriate option. If you choose the first option, the resource pool hierarchy that was on the host you are adding to the cluster is collapsed and all resources will be managed by the cluster. Consider choosing the second option if you created resource pools for the host.

NOTE If you are using a cluster enabled for HA, that cluster might be marked with a red warning icon until you have added enough hosts to satisfy the specified failover capacity. See [“Valid, Yellow, and Red Clusters”](#) on page 76.

You can now add more hosts, then look at the resource allocation information for the cluster by selecting the cluster and choosing its **Resource Allocation** tab.

Resource Management Concepts

2

This chapter introduces resource management concepts. It discusses the following topics:

- [“What Are Resources?”](#) on page 33
- [“Understanding ESX Server Architecture”](#) on page 36
- [“Understanding CPU and Memory Virtualization”](#) on page 39

What Are Resources?

Resources include CPU, memory, disk, and network resources. This manual focusses primarily on CPU and memory resources. For information about disk and network resources, see the *Server Configuration Guide*.

This section introduces resources by discussing the following topics:

- [“Resource Providers and Consumers”](#) on page 33
- [“How ESX Server Manages Resources”](#) on page 35
- [“How Administrators Configure Resources”](#) on page 35
- [“Resource Utilization and Performance”](#) on page 36

Resource Providers and Consumers

Within a virtual infrastructure environment, it’s helpful to think of resource providers and consumers.

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 VirtualCenter, and then add multiple hosts to it. VirtualCenter 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 4, “Understanding Clusters,”](#) on page 61 for an introduction to clusters.

Resource pools are a logical abstraction for flexible management of resources. Resource pools can be grouped into hierarchies. They can be considered both resource providers and consumers.

- Resource pools provide resources to child resource pools and virtual machines.
- Resource pools are also resource consumers because they consume their parent’s resources. See [Chapter 3, “Understanding and Managing Resource Pools,”](#) on page 43.

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 and memory of the resource provider or a guaranteed reservation of CPU and memory. When you power on that virtual machine, the server checks whether or not enough unreserved resources are available and allows power on only if there are enough resources.

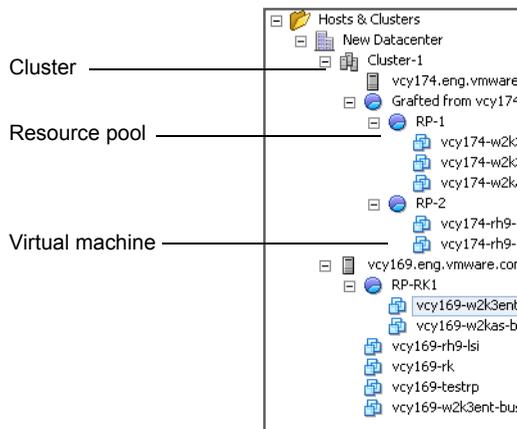


Figure 2-1. Clusters, Resource Pools, and Virtual Machines in VI Client

How ESX Server Manages Resources

Each virtual machine consumes a portion of the CPU, memory, network bandwidth, and storage resources of the ESX Server host. The host guarantees each virtual machine its share of the underlying hardware resources based on a number of factors:

- Available resources for the ESX Server host (or the cluster).
- Reservation, limit, or shares of the virtual machine. These attributes of a virtual machine have default values that you can change to customize resource allocation. See [“Understanding Virtual Machine Resource Allocation”](#) on page 18.
- Number of virtual machines powered on, and resource utilization by those virtual machines.
- Reservation, limit, and shares the administrator assigned to the resource pools in the resource pool hierarchy.
- Overhead required to manage the virtualization.

The server manages different resources differently:

The server manages **CPU and memory resources** based on the total available resources and the factors listed above.

The server manages **network and disk resources** on a per-host basis. A VMware server:

- Manages disk resources using a proportional share mechanism.
- Controls network bandwidth with network-traffic shaping.

NOTE The *Server Configuration Guide* is the best resource for information on disk and network resources. The *SAN Configuration Guide* gives background and setup information for using ESX Server with SAN storage.

How Administrators Configure Resources

In many cases, the defaults the system uses when you create a virtual machine are entirely appropriate. However, in some cases, you might find it useful to customize virtual machines so that the system allocates more or fewer resources to them.

Virtual machine and resource pool attributes, and how to customize them, are discussed throughout this guide. See [“How Administrators Can Affect CPU Management”](#) on page 38 and [“How Administrators Can Affect Memory Management”](#) on page 38 for an introduction.

Resource Utilization and Performance

Resource utilization is the key to performance. The best way to get the highest performance from your virtual infrastructure components is to make sure no resource is a bottleneck. See [Chapter 10, “Best Practices,”](#) on page 143 for information. See [Appendix B, “Using the esxtop Utility,”](#) on page 159 for information on the esxtop performance measurement tool.

Understanding ESX Server Architecture

The different components of an ESX Server system work together to run virtual machines and give them access to resources. This section briefly describes the ESX Server architecture.

NOTE You can skip this section if your interest is the practical application of resource management.

The following illustration shows the main components of an ESX Server host.

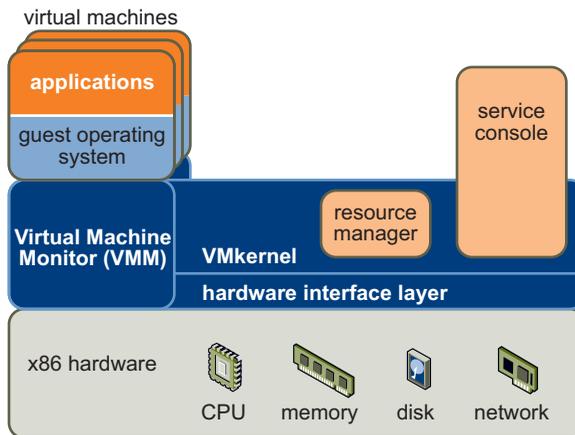


Figure 2-2. ESX Server Host Components

VMkernel

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

- Memory
- Physical processors

- Storage and networking controllers

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

VMkernel Resource Manager

The resource manager partitions the physical resources of the underlying server. It uses a proportional share mechanism to allocate CPU, memory, and disk resources to virtual machines that are powered on. See [Chapter 9, “Advanced Resource Management,”](#) on page 117 for more detailed information about resource allocation.

Users can specify shares, reservations, and limits for each virtual machine. The resource manager takes that information into account when it allocates CPU and memory to each virtual machine. See [“How ESX Server Manages Resources”](#) on page 35 for more information.

VMkernel Hardware Interface Layer

The hardware interface hides hardware differences from ESX Server (and virtual machine) users. It enables hardware-specific service delivery and includes:

- Device drivers.
- Virtual Machine File System (VMFS)— Distributed file system. Optimized for very large files like virtual machine disks and swap files.

NOTE Device drivers and VMFS are discussed in the *Server Configuration Guide*.

Virtual Machine Monitor

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

Service Console

The service console is a limited distribution of Linux based on Red Hat Enterprise Linux 3, Update 6 (RHEL 3 U6). The service console provides an execution environment for monitoring and administering an ESX Server system.

NOTE In most cases, administrators use a VI Client connected to either an ESX Server system or a VirtualCenter Server to monitor and administer ESX Server systems.

How Administrators Can Affect CPU Management

You have access to information about current CPU allocation through the VI Client or using the Virtual Infrastructure SDK.

You can specify CPU allocation in these ways:

- Use the attributes and special features available through the VI Client. The VI Client graphical user interface allows you to connect to an ESX Server host or a VirtualCenter Server. See [Chapter 1, “Resource Management Jumpstart,”](#) on page 13 for an introduction.
- Use advanced settings under certain circumstances. See [Chapter 9, “Advanced Resource Management,”](#) on page 117.
- Use the Virtual Infrastructure SDK for scripted CPU allocation.
- Use hyperthreading, as discussed in [“Hyperthreading”](#) on page 122.
- Specify a CPU affinity for a certain host. See [“Using CPU Affinity to Assign Virtual Machines to Specific Processors”](#) on page 120.

NOTE CPU affinity is not usually recommended. See [“Using CPU Affinity to Assign Virtual Machines to Specific Processors”](#) on page 120 for information on CPU affinity and potential problems with it.

If you don't customize CPU allocation, the ESX Server host uses defaults that work well in most situations.

How Administrators Can Affect Memory Management

You have access to information about current memory allocations and other status information through the VI Client or using the Virtual Infrastructure SDK.

You can specify memory allocation in these ways:

- Use the attributes and special features available through the VI Client. The VI Client graphical user interface allows you to connect to an ESX Server host or a VirtualCenter Server. See [Chapter 1, “Resource Management Jumpstart,”](#) on page 13 for an introduction.
- Use advanced settings under certain circumstances. See [Chapter 9, “Advanced Resource Management,”](#) on page 117.
- Use the Virtual Infrastructure SDK for scripted memory allocation.

If you don't customize memory allocation, the ESX Server host uses defaults that work well in most situations.

For servers with NUMA architecture, see [Appendix A, “Using NUMA Systems with ESX Server,”](#) on page 147. For information on supported NUMA platforms, see the white paper at www.vmware.com/pdf/esx2_NUMA.pdf.

Understanding CPU and Memory Virtualization

This section discusses virtualization and what it means for the resources available for the virtual machines.

CPU Virtualization Basics

Each virtual machine appears to run on a dedicated CPU, or set of CPUs, with each CPU having its own registers and control structures. VMware uses the terms **virtual CPU** for a processor within the virtual machine and **physical CPU** for an underlying physical x86-based processor. ESX Server systems support virtual machines with up to four virtual CPUs.

To view information about physical and logical processors

- 1 In the VI Client, select the host and click the **Configuration** tab, then select **Processors**.

The screenshot shows the VMware ESX Server configuration interface. The top navigation bar includes tabs for Summary, Virtual Machines, Resource Allocation, Performance, Configuration (selected), Tasks & Events, Alarms, and Permissions. The left sidebar shows a tree view with Hardware and Software sections. Under Hardware, Processors is selected. The main content area displays the configuration for Processors, divided into General and System sections.

Processors	
General	
Model	Intel(R) Xeon(TM) CPU 3.20GHz
Processor Speed	3.2 GHz
Hyperthreading	Enabled
Physical Processors	2
Total Processor Cores	2
Logical Processors	4
System	
Manufacturer	Dell Computer Corporation
Model	PowerEdge 1850

- 2 You can now view the information about the number and type of physical processors and the number of logical processors. You can also disable or enable hyper-threading by clicking **Properties**.

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

Memory Virtualization Basics

The VMkernel manages all machine memory, except for the memory that is allocated to the service console. 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 for its code and data.

To view information on how a host's memory is being used

- 1 In the VI Client, select the host, click the **Configuration** tab, and select **Memory**.

The screenshot shows the VMware ESX Server Configuration page for host vcy174.eng.vmware.com. The 'Configuration' tab is selected, and the 'Memory' section is expanded. The 'Physical' memory usage is displayed as follows:

Physical	
Total	8.00 GB
System	777.75 MB
Virtual Machines	6.97 GB
Service Console	272.00 MB

- 2 You can now view the information about the total memory, memory assigned to the service console, and memory available to virtual machines.

The following section discusses the following memory virtualization topics:

- [“Virtual Machine Memory”](#) on page 40
- [“Memory Overcommitment”](#) on page 41
- [“Memory Sharing”](#) on page 41

Virtual Machine Memory

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

- **Configured Size** — The dynamic memory allocation for a virtual machine is based on the shares that are specified, and is bounded by its reservation and limit.
- **Shares** — Specify the relative priority for a virtual machine if more than the reservation is available. See [“Shares”](#) on page 20.
- **Reservation** — Is a guaranteed lower bound on the amount of memory that the host reserves for the virtual machine, even when memory is overcommitted. The reservation should be set to a level that ensures the virtual machine has sufficient memory to run efficiently, without excessive paging.
- **Limit** — Is the upper limit on memory the host might make available to virtual machines.

Overhead memory includes space reserved for the virtual machine frame buffer and various virtualization data structures. See [“Understanding Memory Overhead”](#) on page 128 for more information.

Memory Overcommitment

For each running virtual machine, the system reserves physical memory for both the virtual machine’s reservation (if any) and for its virtualization overhead. Because of the memory management techniques the ESX Server host employs, however, your virtual machines can use more memory than the physical machine (the host) has available. For example, you can have a host with 2GB memory and run four virtual machines with 1GB memory each. In that case, the memory is overcommitted.

Overcommitment 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 Server host automatically transfers memory from idle virtual machines to virtual machines that need more memory. You can use the **Reservation** or **Shares** parameter to preferentially allocate memory to important virtual machines. This memory remains available to other virtual machines if it’s not in use. See [“Understanding Virtual Machine Resource Allocation”](#) on page 18.

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 Server 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.

Understanding and Managing Resource Pools

3

This chapter introduces resource pools and explains how Virtual Infrastructure allows you to view and manipulate them. It discusses these topics:

- [“Introduction”](#) on page 43
- [“Resource Pool Admission Control”](#) on page 47
- [“Creating Resource Pools”](#) on page 48
- [“Viewing Resource Pool Information”](#) on page 51
- [“Changing Resource Pool Attributes”](#) on page 55
- [“Monitoring Resource Pool Performance”](#) on page 55
- [“Adding Virtual Machines to Resource Pools”](#) on page 56
- [“Removing Virtual Machines from Resource Pools”](#) on page 57
- [“Resource Pools and Clusters”](#) on page 57

NOTE All tasks assume you have permission to perform them. See the online Help for information on permissions and how to set them.

Introduction

This section introduces resource pools. It discusses the following topics:

- [“What Are Resource Pools?”](#) on page 44
- [“Why Use Resource Pools?”](#) on page 45
- [“Host Resource Pools and Cluster Resource Pools”](#) on page 46

What Are Resource Pools?

Resource pools can be 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 is not displayed because the resources of the host (or cluster) and the root resource pool are always the same.

NOTE VMware DRS helps you balance resources across virtual machines. It is discussed in [“Understanding VMware DRS”](#) on page 65.

If you don’t create child resource pools, only the root resource pools exist.

- 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 therefore 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.

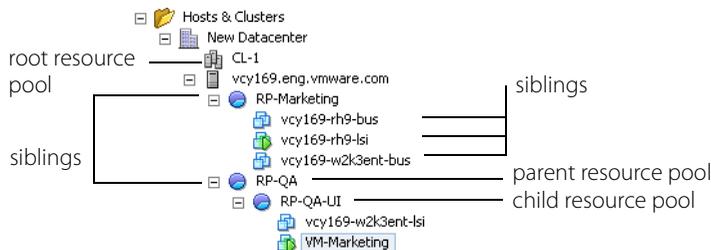


Figure 3-1. Parents, Children, and Siblings in Resource Pool Hierarchy

In the example above, 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.

For each resource pool, you can 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.

Why Use Resource Pools?

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

- **Flexible hierarchical organization** — You can 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, which are discussed in the *Introduction to Virtual Infrastructure*.
- **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 the resources. If you replace three 2GB hosts with two 3GB hosts, you don't 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 multi-tier service** — You don't 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. Three of the virtual machines are used by the marketing department, and two are used by the QA department. Because the QA department needs larger amounts of CPU and memory, the administrator creates one resource pool for each group, and 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 still sufficient for the lighter load of the

marketing staff. Whenever the QA department isn't fully using its allocation, the marketing department can use the available resources.

This scenario is shown in [Figure 3-2](#). The numbers show the effective allocations to the resource pools.

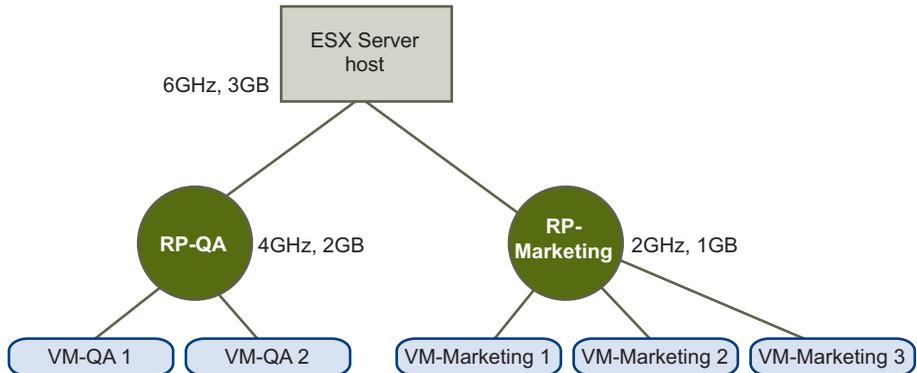


Figure 3-2. Allocating Resources to Resource Pools

Host Resource Pools and Cluster Resource Pools

You can create child resource pools of standalone ESX Server hosts or of DRS clusters (see [“Understanding VMware DRS”](#) on page 65).

- For standalone ESX Server hosts, you create and manage resource pools as children of the host. Each host supports its own hierarchy of resource pools.
- For clusters enabled for DRS, the resources of all hosts are assigned to the cluster.

When you add a host with resource pools to a DRS cluster, you are prompted to decide on resource pool placement. By default, the resource pool hierarchy is discarded and the host is added at the same level as the virtual machines. You can choose to graft the host's resource pools onto the cluster's resource pool hierarchy and choose a name for the top-level resource pool. See [“Resource Pools and Clusters”](#) on page 57.

Because all resources are combined, you no longer manage resources for individual hosts but manage all resources in the context of the cluster. You assign virtual machines to resource pools with predefined characteristics. If you later change capacity by adding, removing, or upgrading hosts, you might not have to change the resource allocations you made for the resource pools.

If the VirtualCenter Server becomes unavailable, you can make changes using a VI Client connected to an ESX Server host. However, if you do, the cluster may

become yellow (overcommitted) or red (invalid) when the VirtualCenter Server becomes available again. See “Valid, Yellow, and Red Clusters” on page 76. If your cluster is in automatic mode, VirtualCenter reapplies the last known cluster configuration (and potentially undoes your changes) when the VirtualCenter Server becomes available again.

- If you add a host to a cluster that is not enabled for DRS, the host’s resource pool hierarchy is discarded, and no resource pool hierarchy can be created.

Resource Pool Admission Control

When you power on virtual machines on an ESX Server host, the host first performs basic admission control, as discussed in “Admission Control” on page 22. When you power on a virtual machine inside a resource pool, or attempt to create a child resource pool, the system performs additional admission control to assure the resource pool’s restrictions are not violated.

Before you power on a virtual machine or create a resource pool, you can check the CPU Unreserved and Memory Unreserved fields in the resource pool’s Resource Allocation tab to see whether sufficient resources are available.

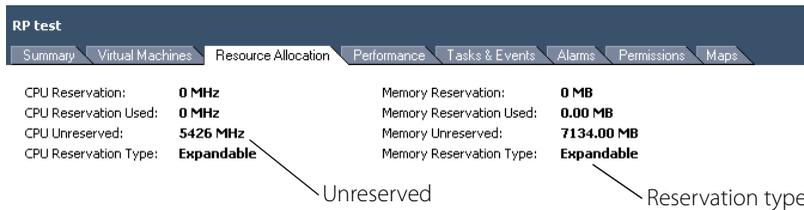


Figure 3-3. Resource Pool Reservation Information

How unreserved CPU and memory are computed depends on the reservation type:

- If the reservation type is **Fixed**, the system checks that the 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.
- If the reservation type is **Expandable**, the system first checks that the resource pool has sufficient resources to fulfill the requirements.
 - If there are sufficient resources, the action is performed.
 - If there are not sufficient resources, the managing server checks whether resources are available in a parent resource pool (direct parent or ancestor). If they are, the action is performed and the parent resource pool resources are reserved. If no resources are available, a message appears and the action is not

performed. See [“Understanding Expandable Reservation”](#) on page 28 for more information.

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.

Creating Resource Pools

You can create a child resource pool of any ESX Server 3.0 host, resource pool, or DRS cluster.

NOTE If a host has been added to a cluster, you can no longer 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 can't allocate resources that aren't available. See [“Resource Pool Admission Control”](#) on page 47.

To create a resource pool

- 1 Select the intended parent and choose **File > New > New Resource Pool** (or click **New Resource Pool** in the Commands panel of the Summary tab).

NOTE Pressing Control-R also starts resource pool creation.

- 2 In the New Resource Pool dialog box, provide the following information for your resource pool.

Table 3-1. New Resource Pool fields

Field	Description
Name	Name of the new resource pool.
CPU Resources	
Shares	Number of CPU 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 choose Low , Normal , or High , or choose Custom to specify a number that assigns a share value. See “Shares” on page 20.
Reservation	Guaranteed CPU allocation for this resource pool.

Table 3-1. New Resource Pool fields (Continued)

Field	Description
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 a parent's or ancestor's resources if this check box is selected (the default). See "Understanding Expandable Reservations" on page 50.
Limit	Upper limit for the amount of CPU the host makes available to this resource pool. Default is Unlimited. To specify a limit, deselect the Unlimited check box and type in the number.
Memory Resources	
Shares	Number of 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 choose Low , Normal , or High , or choose Custom to specify a number that assigns a share value. See "Shares" on page 20.
Reservation	Guaranteed memory allocation for this resource pool.
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 a parent's or ancestor's resources if this check box is selected (the default). See "Understanding Expandable Reservations" on page 50.
Limit	Upper limit for this resource pool's memory allocation. Default is Unlimited . To specify a different limit, deselect the Unlimited check box.

- 3 After you've made all choices, click **OK**. VirtualCenter creates the resource pool and displays it in the inventory panel.

NOTE A yellow triangle is displayed if any of the selected values are not legal values because of limitations on total available CPU and memory. For example, if you have a resource pool with a reservation of 10GB, and you created a child resource pool with a reservation of 6GB, you cannot create a second child resource pool with a reservation of 6MB and **Type** set to **Fixed**. See [“Resource Pool Admission Control”](#) on page 47.

Understanding Expandable Reservations

When you power on a virtual machine or create a resource pool, the system checks whether the CPU and memory reservation is available for that action.

If **Expandable Reservation** is not selected, the system considers only the resources available in the selected resource pool.

If **Expandable Reservation** is selected (the default), the system considers both the direct parent resource pool and any ancestor resource pool when performing admission control. Ancestors include direct parents, parents of the parents, and so on. For each ancestor, resources can be used only if the ancestor pool is set to expandable for each ancestor and no **Limit** is set that would stop it from becoming larger by borrowing resources. 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.

NOTE Leave this option selected only if you trust the administrator of the child resource pool to not reserve more resources than appropriate.

Expandable Reservations Example

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 will 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 quite inflexible, especially in resource pool hierarchies, and can complicate tree operations in the resource pool hierarchy.

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

Viewing Resource Pool Information

When you select a resource pool in the VI Client, the Summary tab displays information about that resource pool. The following section lists information about the “[Resource Pool Summary Tab](#)” on page 51 and “[Resource Pool Resource Allocation Tab](#)” on page 52.

NOTE All other tabs are discussed in detail in the online Help.

Resource Pool Summary Tab

The Resource Pool Summary tab displays high-level statistical information about the resource pool.

RP-QA

Summary Virtual Machines Resource Allocation Performance Tasks & Events Alarms Permissions Maps

General		Resources	
Number of Virtual Machines:	2	CPU usage:	0 MHz
Number of Running Virtual Machines:	1	Memory usage:	20 MB
Number of Child Resource Pools:	0		

CPU		Memory	
Shares:	Custom (3192)	Shares:	Custom (100)
Reservation:	0 MHz	Reservation:	0 MB
Type:	Expandable	Type:	Expandable
Limit:	Unlimited	Limit:	Unlimited
Unreserved:	5426 MHz	Unreserved:	7074.73 MB

Commands

- New Virtual Machine
- New Resource Pool
- Edit Settings

Table 3-2.

Section	Description
General	<p>The General panel displays statistical information for the resource pool.</p> <ul style="list-style-type: none"> ■ Number of Virtual Machines — Number of virtual machines in this resource pool. Does not include the number of virtual machines in child resource pools. ■ Number of Running Virtual Machines — Number of running virtual machines in this resource pool. Does not include the number of virtual machines running in child resource pools. ■ Number of Child Resource Pools — Number of direct child resource pools. Does not include all resource pools in the hierarchy but only direct children.
CPU	Displays the CPU Shares , Reservation , Reservation Type , and Limit that were specified for this resource pool. Also displays the amount of CPU currently unreserved.
Commands	<p>Allows you to invoke commonly used commands.</p> <ul style="list-style-type: none"> ■ New Virtual Machine — Starts the New Virtual Machine wizard to create a new virtual machine in this resource pool. See “Resource Pool Admission Control” on page 47 for information on how the system performs admission control. ■ New Resource Pool — Displays the Create Resource Pool dialog box, which allows you to create a child resource pool of the selected resource pool. See “Creating Resource Pools” on page 48. ■ Edit Settings — Allows you to change the CPU and memory attributes for the selected resource pool. See “Changing Resource Pool Attributes” on page 55.
Resources	Displays CPU Usage and Memory Usage for the virtual machines within the selected resource pool.
Memory	Displays the Shares , Reservation , Reservation Type , and Limit that were specified for this resource pool. Also displays the amount of memory currently unreserved.

Resource Pool Resource Allocation Tab

The resource pool's Resource Allocation tab gives detailed information about the resources currently reserved and available for the resource pool, and lists the user of the resources, as discussed in the tables below.

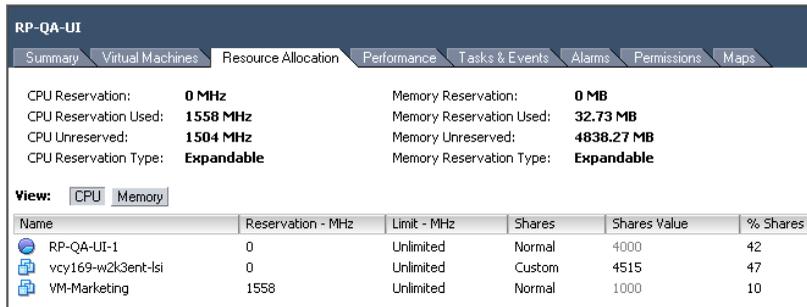


Figure 3-4. Resource Pool Resource Allocation Tab

The top portion of the display specifies the following information about the resource pool itself:

Table 3-3. Resource Allocation Tab Fields

Field	Description
CPU Reservation / Memory Reservation	Amount of CPU or memory specified in the reservation for this resource pool. Reservation can be specified during resource pool creation, or later by editing the resource pool.
CPU Reservation Used / Memory Reservation Used	CPU or memory reservation used. Reservations are used by running virtual machines or by child resource pools with reservations.
CPU Unreserved / Memory Unreserved	CPU or memory currently unreserved and available to be reserved by virtual machines and resource pools. Note: Look at this number when trying to determine whether you can create a child resource pool of a certain size, or whether you can power on a virtual machine with a certain reservation.
CPU Reservation Type / Memory Reservation Type	Expandable or Fixed . See “Understanding Expandable Reservations” on page 50.

Below the information specific to the resource pool is a list of the virtual machines and child resource pools of this resource pool.

NOTE This table does not list virtual machines assigned to child resource pools of this resource pool.

You can click the **CPU** or **Memory** tab to display the following information:

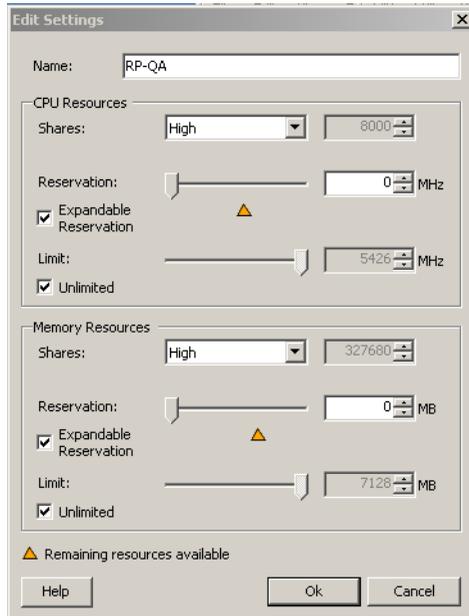
Table 3-4. Resource Allocation CPU and Memory information

Field	Description
Name	Name of the resource pool or virtual machine.
Reservation	Specified reservation for this virtual machine or resource pool. Default is 0, that is, the system reserves no resources for this resource pool.
Limit	Specified limit for this virtual machine or resource pool. Default is Unlimited , that is, the system allocates as many resources to this virtual machine as it can.
Shares	Specified shares for this virtual machine or resource pool. One of High , Normal , Low if one of the default settings has been selected. Custom if a custom setting has been selected.
Shares Value	Number of shares allocated to this virtual machine or resource pool. This number depends on the shares setting (High , Normal , Low , or Custom). See “Shares” on page 20.
%Shares	Shares value for this resource pool or virtual machine divided by the total number of shares allocated to all children of the parent resource pool. This value is unrelated to the parent resource pool’s local shares allocation.
Type	Reservation type. Either Fixed or Expandable . See “Understanding Expandable Reservation” on page 28.

Changing Resource Pool Attributes

To make changes to a resource pool

- 1 Select the resource pool in the VI Client's inventory panel.
- 2 In the Summary tab's Command panel, choose **Edit Settings**.



- 3 In the Edit Resources dialog box, you can change all attributes of the selected resource pool. The choices are discussed in [“Creating Resource Pools”](#) on page 48.

Monitoring Resource Pool Performance

Monitoring a resource pool's performance is useful if you want to understand the effectiveness of resource pool allocations.

To monitor a resource pool's performance

- 1 Select the resource pool in the inventory panel.
- 2 Click the **Performance** tab.

You see information about resource pool performance. Click **Change Chart Options** to customize the performance chart. See the online Help for a detailed discussion of performance charts and how to configure them.

Adding Virtual Machines to Resource Pools

When you create a new virtual machine, the Virtual Machine wizard allows you to add it to a resource pool as part of the creation process. You can also add an already existing virtual machine to a resource pool. This section discusses both tasks.

To create a virtual machine and add it to a resource pool

- 1 Select a host, then choose **File > New > Virtual Machine** (or press Ctrl-N).
- 2 Supply the information for the virtual machine, choosing a resource pool as the location when prompted by the wizard.

The wizard places the virtual machine into the resource pool you selected.

See the *Virtual Infrastructure User's Guide* for detailed information on creating virtual machines.

To add an existing virtual machine to a resource pool

- 1 Select the 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 desired resource pool object.

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 may 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 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 Reserved and unreserved CPU and memory change only if the virtual machine is powered on. If the virtual machine has been powered off or suspended, it can be moved but overall available resources for the resource pool are not affected.

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. See [“Resource Pool Admission Control”](#) on page 47.

Removing Virtual Machines from Resource Pools

You can remove a virtual machine from a resource pool in a number of ways, depending on your intention for the machine.

- Move the virtual machine to a different resource pool. See [“To add an existing virtual machine to a resource pool”](#) on page 56. You don't 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. (In economic terms, this is like deflation; when more shares are created, this is like inflation.) 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.

- Remove the virtual machine from the inventory or delete it from the disk using the virtual machine's right-button menu (or press the Delete key).

You need to power off the virtual machine before you can completely remove it. For more information, see the *Virtual Infrastructure User's Guide*.

Resource Pools and Clusters

When you add a host with an existing resource pool hierarchy to a cluster, what happens depends on the cluster. There are two options:

- [“Clusters Enabled for DRS”](#) on page 58
- [“Clusters Not Enabled for DRS”](#) on page 59

Clusters Enabled for DRS

If a cluster is enabled for DRS, and you move one or more hosts into the cluster, a wizard allows you to choose what happens to the host's resource pools:

- **Put this host's virtual machines in the cluster's root resources** — Collapses the host's resource pool hierarchy and makes all virtual machines direct children of the cluster. This is the same as the behavior shown for [“Clusters Not Enabled for DRS”](#) on page 59.

NOTE In that case, you may have to manually adjust the share values associated with individual virtual machines because the shares in the original host hierarchy are relative to the resource pools on the host.

- **Create a new resource pool for the host's virtual machines and resource pools** — Creates a resource pool corresponding to the host's root resource pool. By default, the resource pool is named **Grafted from <host_name>**, but you can choose a different name. The term grafted was chosen because the branches of the host's tree are added to the branches of the cluster's tree, just as fruit tree branches are grafted onto rootstock.

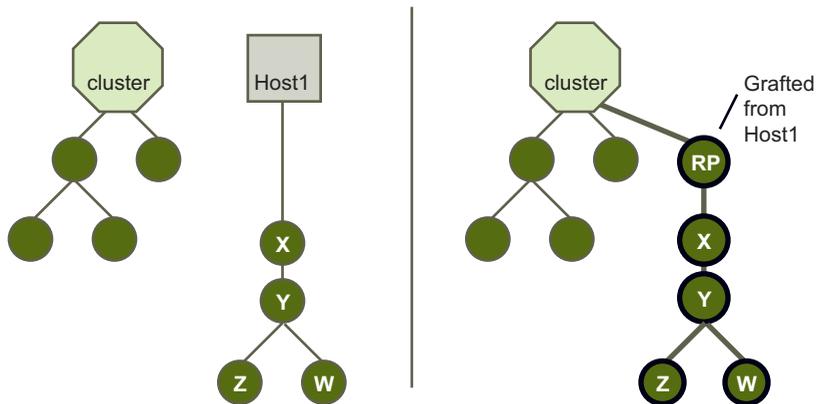


Figure 3-5. Resource Pool Hierarchy Grafted onto Cluster

In the example above, cluster **Cluster** and host **Host1** both have a hierarchy of resource pools. When you add the host to the cluster, the host's invisible top-level resource pool is grafted onto the cluster's resource pool hierarchy and is named **grafted from Host1** by default.

NOTE Shares remain allocated as they were before the host moved into the cluster. Percentages are adjusted as appropriate.

The resource pool hierarchy becomes completely independent of the host. If you later remove the host from the cluster, the cluster keeps the resource pool hierarchy and the host loses the resource pool hierarchy (though the virtual machines are removed along with the host). See [“Removing Hosts from Clusters”](#) on page 97.

NOTE The host must be in maintenance mode before you can remove it from the cluster. See [“Maintenance Mode”](#) on page 68.

Clusters Not Enabled for DRS

If the cluster is enabled for HA only (or neither HA nor DRS), and you move one or more hosts into the cluster, the cluster takes ownership of the resources. The hosts and virtual machines become associated with the cluster. The resource pool hierarchy is flattened.

NOTE In a non-DRS cluster there is no cluster-wide resource management based on shares. Virtual machine shares remain relative to each host.

In the illustration below, host H1 and host H2 both have a hierarchy of resource pools and virtual machines. When the two hosts are added to cluster C, the resource pool hierarchy is flattened and all virtual machines, as well as the hosts, become direct children of the cluster.

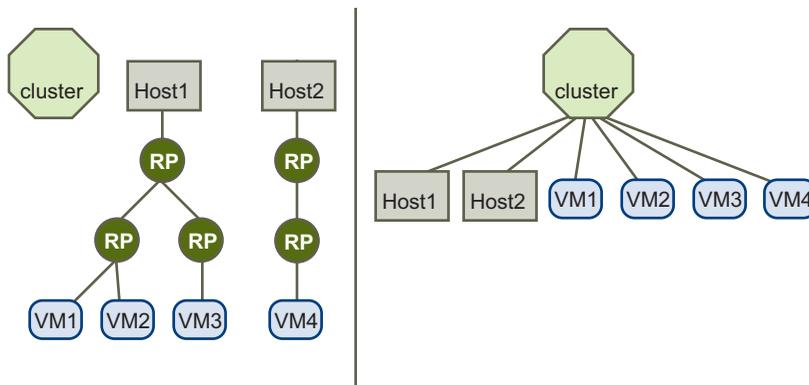


Figure 3-6. Flattened Resource Pool Hierarchy

4

Understanding Clusters

This chapter presents a conceptual introduction to clusters and to the DRS and HA features in the following sections:

- [“Introduction to Clusters”](#) on page 61
- [“Understanding VMware DRS”](#) on page 65
- [“Understanding VMware HA”](#) on page 69
- [“Using HA and DRS Together”](#) on page 76
- [“Valid, Yellow, and Red Clusters”](#) on page 76

NOTE All tasks described assume you have permission to perform them. See the online Help for information on permissions and how to set them.

Introduction to Clusters

A cluster is a collection of ESX Server hosts and associated virtual machines with shared resources and a shared management interface. When you add a host to a cluster, the host’s resources become part of the cluster’s resources. When you create a cluster, you can choose to enable it for DRS, HA, or both.

See [“Cluster Prerequisites”](#) on page 83 for information on the virtual machines in clusters and on how to configure them.

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

VMware DRS

The DRS feature improves resource allocation across all hosts and resource pools. DRS collects resource usage information for all hosts and virtual machines in the cluster and generates recommendations for virtual machine placement. These recommendations can be applied automatically. Depending on the configured DRS automation level, DRS displays or applies the following recommendations:

- **Initial placement** — When you first power on a virtual machine in the cluster, DRS either places the virtual machine or makes a recommendation.
- **Load balancing** — At runtime, DRS tries to improve resource utilization across the cluster either by performing automatic migrations of virtual machines (VMotion), or by providing recommendations for virtual machine migrations. Consider the simple example shown in [Figure 4-1, “VMware DRS,”](#) on page 63.

Assume Host 1 and Host 2 have identical capacity, and all virtual machines have the same configuration and load. However, because Host 1 has six virtual machines, its resources are overused while ample resources are available on Host 2 and Host 3. DRS migrates (or offers to migrate) virtual machines from Host 1 to Host 2 and Host 3.

See [“Understanding VMware DRS”](#) on page 65.

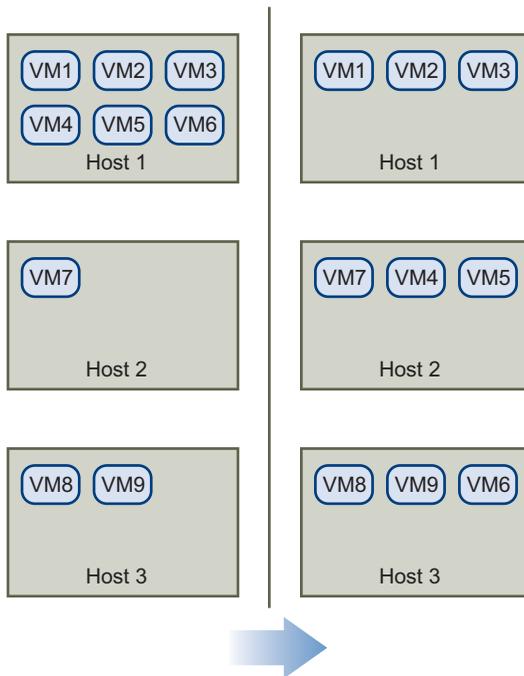


Figure 4-1. VMware DRS

VMware HA

A cluster enabled for HA monitors for host failure. If a host goes down, all virtual machines that were on the host are promptly restarted on different hosts.

When you enable a cluster for HA, you are prompted for the number of host failures allowed. If you specify the number of host failures as **1**, HA maintains enough capacity across the cluster to tolerate the failure of one host, so that all running virtual machines on that host can be restarted on remaining hosts. By default, you cannot power on a virtual machine if doing so violates required failover capacity (strict admission control). See [“Understanding VMware HA”](#) on page 69 for more information.

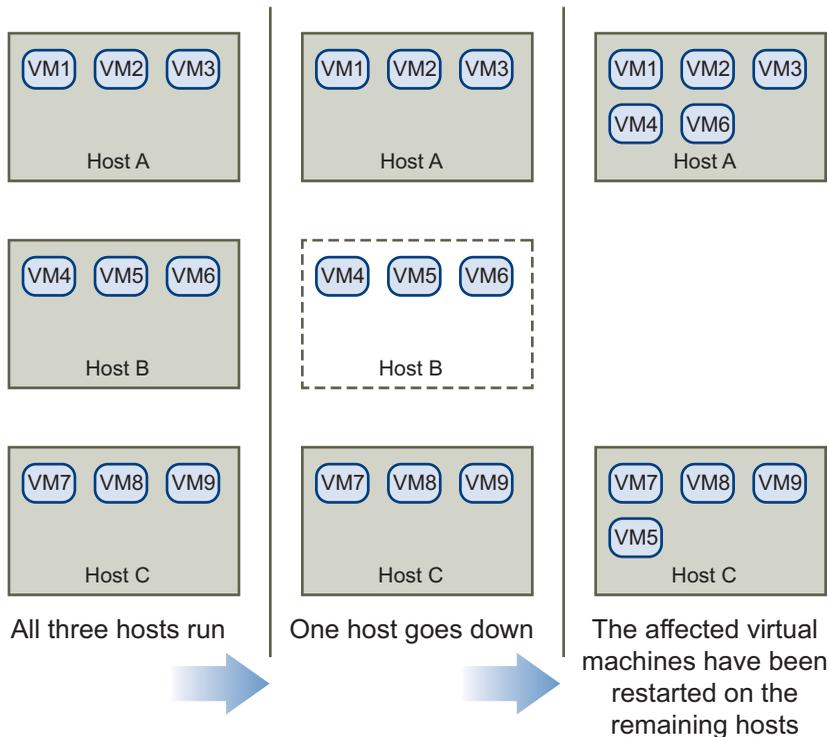


Figure 4-2. VMware HA

In [Figure 4-2](#), three hosts have three virtual machines each, and the corresponding HA cluster is configured for failover of one host. When Host B goes down, HA migrates the virtual machines from Host B to Host A and Host C.

Clusters and VirtualCenter Failure

You create and manage clusters using VirtualCenter, as discussed in [Chapter 5, “Creating a VMware Cluster,”](#) on page 83.

The VirtualCenter Server places an agent on each host in the system. If the VirtualCenter host goes down, HA and DRS functionality changes as follows:

- HA**—HA clusters continue to work even if the VirtualCenter Server goes down, and can still restart virtual machines on other hosts in case of failover; however, the information about virtual machine specific cluster properties (such as priority and isolation response) is based on the state of the cluster before the VirtualCenter Server went down.

- **DRS**—The hosts in DRS clusters continue running using available resources; however, there are no recommendations for resource optimization.

If you make changes to the hosts or virtual machines using a VI Client connected to an ESX Server host while the VirtualCenter Server is unavailable, those changes do take effect. When VirtualCenter becomes available again, you may find that clusters have turned red or yellow because cluster requirements are no longer met. See [“Valid, Yellow, and Red Clusters”](#) on page 76.

Understanding VMware DRS

When you enable a cluster for DRS, VirtualCenter continuously monitors the distribution of CPU and memory resources for all hosts and virtual machines in the cluster. DRS compares these metrics to what resource utilization ideally should be given the attributes of the resource pools and virtual machines in the cluster, and the current load.

NOTE DRS always evaluates both the specified configuration settings and the current load.

When you add a host to a DRS cluster, that host’s resources become associated with the cluster. The system prompts you whether you want to associate any existing virtual machines and resource pools with the cluster’s root resource pool or graft the resource pool hierarchy. See [“Resource Pools and Clusters”](#) on page 57. VirtualCenter can then perform the following actions:

- Assign virtual machines to the appropriate host during power on ([“Initial Placement”](#) on page 66), as in [Figure 4-3](#).

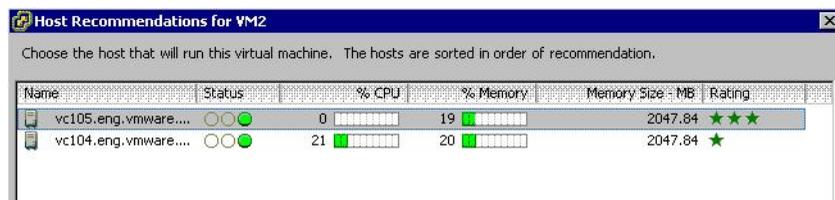


Figure 4-3. DRS Initial Placement Recommendation

- Migrate virtual machines to different hosts for improved load balancing ([“Virtual Machine Migration”](#) on page 66), as in the following screen:

Priority	Virtual Machine	Reason	Source Host	Target Host	CPU Load
★★★	vcy169:w2kas...	"Balance avera...	vcy169.eng.v...	vcy174.eng.v...	54

Figure 4-4. DRS Load Balancing Recommendation

Initial Placement

When you power on a virtual machine, VirtualCenter first checks that there are enough resources in that cluster to support that virtual machine.

VirtualCenter then needs to find a host on which the virtual machine can run.

- If the cluster is manual, VirtualCenter displays a list of recommended hosts, with more suitable hosts higher on the list. You may choose any host.
- If the cluster is partially automatic or automatic, VirtualCenter places the virtual machine on the appropriate host.

Virtual Machine Migration

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

- If the cluster is manual or partially automated, VirtualCenter does not take automatic actions to balance resources. Instead, the Summary page indicates that migration recommendations are available and the Migration page displays recommendations for changes that make the most efficient use of resources across the cluster.
- If the cluster is fully automated, VirtualCenter places virtual machines that join the cluster on appropriate hosts and migrates running virtual machines between hosts as needed to ensure efficient use of cluster resources. VirtualCenter displays a history of migrations in the VI Client's Migration tab.

NOTE Even in an automatic migration setup, users can explicitly migrate individual virtual machines as they want, but VirtualCenter 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. See [“Customizing DRS for Virtual Machines”](#) on page 114.

Migration Threshold

The migration threshold allows you to specify which recommendations are automatically applied when the cluster is in fully automated mode. You can move the slider to use one of five levels. Level 5 is the most aggressive, while Level 1 makes the smallest number of migrations.

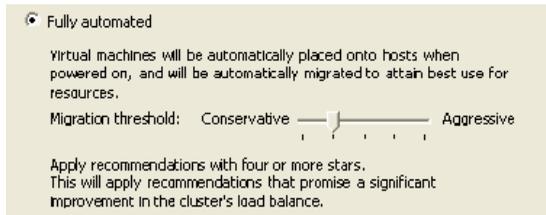


Figure 4-5. Migration Threshold Choices

- **Level 1** — Apply only five-star recommendations. Applies recommendations that must be followed to satisfy constraints such as affinity rules and host maintenance (see [“Maintenance Mode”](#) on page 68).
- **Level 2** — Apply recommendations with four or more stars. Includes Level 1 plus recommendations that promise a significant improvement in the cluster's load balance.
- **Level 3** — Apply recommendations with three or more stars. Includes Level 1 and 2 plus recommendations that promise a good improvement in the cluster's load balance.
- **Level 4** — Apply recommendations with two or more stars. Includes Level 1-3 plus recommendations that promise a moderate improvement in the cluster's load balance.
- **Level 5** — Apply all recommendations. Includes Level 1-4 plus recommendations that promise a slight improvement in the cluster's load balance.

Migration Recommendations

If you create a cluster in manual or partially automated mode, VirtualCenter displays migration recommendations on the Migrations page.

The system supplies as many recommendations as necessary to properly balance the resources of the cluster. Each recommendation includes a priority, the virtual machine to be moved, current (source) host and target host, CPU load and memory load, and a reason for the recommendation. The reason can be one of the following:

- Balance average CPU loads.

- Balance average memory loads.
- Satisfy affinity rule. See [“Using DRS Affinity Rules”](#) on page 101.
- Satisfy anti-affinity rule. See [“Using DRS Affinity Rules”](#) on page 101.
- Host is entering maintenance. See [“Maintenance Mode”](#) on page 68.

NOTE The system does not allow you to apply more than one recommendation that involves VMotion at one time; you must apply recommendations concurrently.

DRS Clusters, Resource Pools, and ESX Server

For clusters enabled for DRS, the resources of all hosts are assigned to the cluster.

DRS internally uses the per-host resource pool hierarchies to implement the cluster-wide resource pool hierarchy. When you view the cluster using a VI Client connected to a VirtualCenter Server, you see the resource pool hierarchy implemented by DRS.

When you view individual hosts using a VI Client connected to an ESX Server host, the underlying hierarchy of resource pools is presented. Because DRS implements the most balanced resource pool hierarchy it can, you should not modify the hierarchy visible on the individual ESX Server host. If you do, DRS will undo your changes immediately for better balance if you are in automatic mode. If you are in partially automatic or manual mode, DRS will make migration recommendations.

Maintenance Mode

Both standalone hosts and hosts within a cluster support a maintenance mode, which restricts the virtual machine operations on the host to allow you to conveniently shut down running virtual machines in preparation for host shut down.

While in maintenance mode, the host does not allow you to deploy or power on a virtual machine. Virtual machines that are running on the maintenance mode host continue to run normally. You can either migrate them to another host, or shut them down.

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 a host is under maintenance, you cannot perform operations such as powering on virtual machines, and the command selection changes accordingly.

DRS Clusters and Maintenance Mode

For DRS clusters, the automation mode determines cluster behavior when you enter maintenance mode.

- A fully automated DRS cluster automatically migrates running virtual machines to different hosts as soon as maintenance mode is enabled.

NOTE If no appropriate host is available, DRS displays information on the Tasks & Events tab.

- A partially automated or manual DRS cluster displays recommendations for migration of running virtual machines, with the recommendation type **Host Maintenance** displayed in the recommendation list.

Understanding VMware HA

The HA cluster feature allows the virtual machines running on ESX Server systems to automatically recover from host failures. When a host goes down, all associated virtual machines are immediately restarted on other hosts in the system. This section first considers differences between VMware HA clusters and traditional clustering solutions, then presents HA cluster concepts.

Traditional and HA Failover Solutions

Both VMware HA and traditional clustering solutions support automatic recovery from host failures. They are complementary because they differ in these areas:

- hardware and software requirements
- time to recovery
- degree of application dependency.

Traditional Clustering Solutions

A traditional clustering solution such as Microsoft Cluster Service (MSCS) or Veritas Clustering Service aims to provide immediate recovery with minimal downtime for applications in case of host or virtual machine failure. To achieve this, the IT infrastructure must be set up as follows:

- Each machine (or virtual machine) must have a mirror virtual machine (potentially on a different host).
- The machine (or the virtual machine and its host) are set up to mirror each other using the clustering software. Generally, the primary virtual machine sends heartbeats to the mirror. In case of failure, the mirror takes over seamlessly.

The following illustration shows different options for the setup of a traditional cluster for virtual machines.

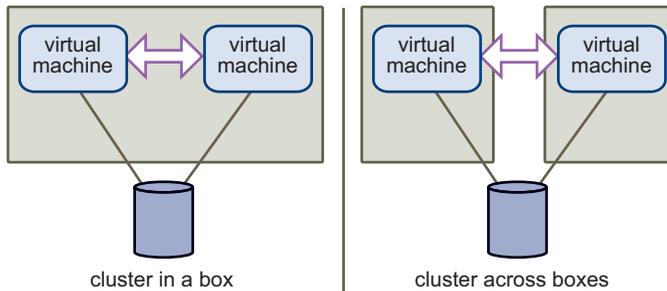


Figure 4-6. VMware Clustering Setup

Setting up and maintaining such a clustering solution is resource intensive. Each time you add a new virtual machine, you need corresponding failover virtual machines and possibly additional hosts. You have to set up, connect, and configure all new machines and update the clustering application's configuration.

The traditional solution guarantees fast recovery, but is resource- and labor-intensive. See the VMware document *Setup for Microsoft Cluster Service* for additional information on different cluster types and how to configure them.

VMware HA Solution

In a VMware HA solution, a set of ESX Server hosts is combined into a cluster with a shared pool of resources. VirtualCenter monitors all hosts in the cluster. If one of the hosts fails, VirtualCenter immediately responds by restarting each associated virtual machine on a different host.

Using HA has a number of advantages:

- **Minimal setup** — The New Cluster Wizard is used for initial setup. You can add hosts and new virtual machines using the VirtualCenter Client. All virtual machines in the cluster automatically get failover support without additional configuration.
- **Reduced hardware cost and setup** — In a traditional clustering solution, duplicate hardware and software must be connected and configured properly. The virtual machine acts as a portable container for the applications, and can be moved around. Duplicate configurations on multiple machines can be avoided. When you use VMware HA, you must have sufficient resources to fail over the number of hosts you want to guarantee. However, the VirtualCenter Server takes care of the resource management and cluster configuration.

- **Increased application availability** — Any application running inside a virtual machine has access to increased availability. Because the virtual machine can recover from hardware failure, all applications that are set up to start during the boot cycle have increased availability at no extra cost even if the application is not itself a clustered application.

Potential limitations of using HA clusters include loss of run-time state and a longer application downtime than in a traditional clustering environment with hot standby. If those limitations become issues, consider using the two approaches together.

VMware HA Features

A cluster enabled for HA:

- Supports easy-to-use configuration using the VirtualCenter client.
- Provides automatic failover on hardware failure for all running virtual machines within the bounds of failover capacity (see [“Failover Capacity,”](#) below).
- Works with traditional application-level failover and enhances it.
- Is fully integrated with DRS. If a host has failed and virtual machines have been restarted on other hosts, DRS can provide migration recommendations or migrate virtual machines for balanced resource allocation. If one or both of the source and target hosts of a migration fail, HA can help recover from that failure.

Failover Capacity

When you enable a cluster for HA, the New Cluster wizard prompts you for the number of hosts for which you want failover capacity. This number is shown as the **Configured Failover Capacity** in the VI Client. HA uses this number to determine if there are enough resources to power on virtual machines in the cluster.

You only need to specify the number of hosts for which you want failover capacity. HA computes the resources it requires to fail over virtual machines for that many hosts, using a conservative estimate, and disallows powering on virtual machines if failover capacity can no longer be guaranteed.

NOTE You can choose to allow the cluster to power on virtual machines even when they violate availability constraints. If you do that, the result is a red cluster, which means that failover guarantees might no longer be valid. See [“Valid, Yellow, and Red Clusters”](#) on page 76.

After you have created a cluster, you can add hosts to it. When you add a host to an HA cluster that is not enabled for DRS, all resource pools are immediately removed from the host, and all virtual machines become directly associated with the cluster.

NOTE If the cluster is also enabled for DRS, you can choose to keep the resource pool hierarchy. See [“Resource Pools and Clusters”](#) on page 57.

Planning for HA Clusters

When planning an HA cluster, consider the following:

- Each host has some memory and CPU to power on virtual machines.
- Each virtual machine must be guaranteed its CPU and memory reservation requirements.

In general, using a fairly uniform setup is highly recommended. HA always plans for a worst-case failure scenario. When computing required failover capacity, HA considers the host with the largest capacity running virtual machines with the highest resource requirements. HA might therefore be too conservative if the virtual machines or hosts in your cluster differ significantly.

During planning, you need to decide on the number of hosts for which you want to guarantee failover. HA tries to reserve resources for at least as many host failures by limiting the number of virtual machines that are powered on, which will consume these resources.

If you left the **Allow virtual machine to be started even if they violate availability constraints** option unselected (strict admission control), VMware HA does not allow you to power on virtual machines if they would cause the current failover level to exceed the configured failover level. It also disallows the following operations:

- Reverting a powered off virtual machine to a powered on snapshot.
- Migrating a virtual machine into the cluster.
- Reconfiguring a virtual machine to increase its CPU or memory reservation.

If you have selected the **Allow virtual machine to be started even if they violate availability constraints** option when you enable HA, you can power on more virtual machines than HA would advise. Because you configured the system to permit this, the cluster does not turn red. In this case, the current (available) failover level can also fall below the configured failover level if the number of hosts that have failed exceeds the configured number. For example, if you have configured the cluster for a one host failure, and two hosts fail, the cluster turns red.

A cluster that is beneath the configured failover level can still perform virtual machine failover in case of host failure, using virtual machine priority to determine which virtual machines to power on first. See [“Customizing HA for Virtual Machines”](#) on page 115.



CAUTION It is not recommended that you work with red clusters. If you do, failover is not guaranteed at the specified level.

VMware HA and Special Situations

VMware HA understands how to work with special situations to preserve your data:

- **Power off host**—If you power off a host, HA restarts any virtual machines running on that host on a different host.
- **Migrate virtual machine with VMotion** — If you are in the process of migrating a virtual machine to another host using VMotion, and the source or target host goes down, the virtual machine could be left in a failed (powered off) state depending on the stage in the migration. HA handles this failure and powers on the virtual machine on an appropriate host:
 - If the source host goes down, HA powers on the virtual machine on the target host.
 - If the target host goes down, HA powers on the virtual machine on the source host.
 - If both hosts go down, HA powers on the virtual machine on a third host in the cluster.
- **Current failover capacity does not match configured failover capacity**— A cluster turns red if current failover capacity is smaller than configured failover capacity. This can happen because more hosts failed than you configured the cluster to tolerate. If you turned off strict admission control, the cluster will not turn red, even if you power on more virtual machines than can be accommodated.

In case of a red cluster, HA fails over virtual machines with higher priorities first, then attempts to fail over other virtual machines. In such a situation, consider giving high priorities to certain virtual machines that are most critical to your environment for recovery. See [“Customizing HA for Virtual Machines”](#) on page 115.

- **Host network isolation** — A host in an HA cluster might lose its console network connectivity. Such a host is isolated from other hosts in the cluster. Other hosts in the cluster consider that host failed, and attempt to fail over its running virtual machines. If a virtual machine continues to run on the isolated host, VMFS disk locking prevents it from being powered on elsewhere. If virtual machines share the same network adapter, they don't have access to the network. It might be advisable to start the virtual machine on another host.

By default, virtual machines are shut down on the isolated host in case of a host isolation incident. You can change that behavior for each virtual machine. See [“Customizing HA for Virtual Machines”](#) on page 115.

Primary and Secondary Hosts

When you add a host to an HA cluster, that host has to communicate with a primary host in the same cluster to complete its configuration (unless it's the first host in the cluster, which makes it the primary host). The first hosts in the cluster become primary hosts, all others are secondary hosts. When a primary host goes down or is removed, HA automatically promotes another host to primary status. Primary hosts help to provide redundancy and are used to initiate failover actions.

If all the hosts in the cluster are not responding, and you add a new host to the cluster, HA configuration fails because the new host cannot communicate with any of the primary hosts. In this situation, you must disconnect all the hosts that are not responding before you can add the new host. The new host becomes the first primary host. When the other hosts become available again, their HA service is reconfigured.

HA Clusters and Maintenance Mode

When you put a host in maintenance mode, you are preparing to shut it down or do maintenance on it. You cannot power on a virtual machine on a host that is in maintenance mode. In the event of a host failure, HA therefore does not fail over any virtual machines to a host that is in maintenance mode, and such a host is not considered when HA computes the current failover level.

When a host exits maintenance mode, the HA service is re-enabled on that host, so it becomes available for failover again.

HA Clusters and Disconnected Hosts

When a host becomes disconnected, it exists in the VirtualCenter inventory, but VirtualCenter does not get any updates from that host, does not monitor it, and therefore has no knowledge of the health of that host. Because the status of the host is not known, and because VirtualCenter is not communicating with that host, HA cannot use it as a guaranteed failover target. HA does not consider disconnected hosts when computing the current failover level.

When the host becomes reconnected, the host becomes available for failover again.

Note that there is a difference between a disconnected host and a host that is not responding.

- A disconnected host has been explicitly disconnected by the user. As part of disconnecting a host, VirtualCenter disables HA on that host. The virtual machines on that host are not failed over, and not considered when VirtualCenter computes the current failover level.
- If a host is not responding, the VirtualCenter Server no longer receives heartbeats from it. This could be, for example, because of a network problem, because the host crashed, or because the VirtualCenter agent crashed.

VirtualCenter takes a conservative approach when considering such hosts. It does not include them when computing the current failover level, but assumes that any virtual machines running on a disconnected host will be failed over if the host fails. The virtual machines on a host that is not responding affect the admission control check.

HA Clusters and Host Isolation Timing Issue

Host failure detection occurs 15 seconds after the HA service on a host has stopped sending heartbeats to the other hosts in the cluster. A host stops sending heartbeats if it is isolated from the network. At that time, other hosts in the cluster treat this host as failed, while this host declares itself as isolated from the network.

By default, the isolated host powers off its virtual machines. These virtual machines can then successfully fail over to other hosts in the cluster. If the isolated host has SAN access, it retains the disk lock on the virtual machine files, and attempts to fail over the virtual machine to another host fails. The virtual machine continues to run on the isolated host. VMFS disk locking prevents simultaneous write operations to the virtual machine disk files and potential corruption.

If the network connection is restored before 12 seconds have elapsed, other hosts in the cluster will not treat this as a host failure. In addition, the host with the transient network connection problem does not declare itself isolated from the network and continues running.

In the window between 12 and 14 seconds, the clustering service on the isolated host declares itself as isolated and starts powering off virtual machines with default isolation response settings. If the network connection is restored during that time, the virtual machine that had been powered off is not restarted on other hosts because the HA services on the other hosts do not consider this host as failed yet.

As a result, if the network connection is restored in this window between 12 and 14 seconds after the host has lost connectivity, the virtual machines are powered off but not failed over.

Using HA and DRS Together

When HA performs failover and restarts virtual machines on different hosts, its first priority is the immediate availability of all virtual machines. After the virtual machines have been restarted, those hosts on which they were powered on might be heavily loaded, while other hosts are comparatively lightly loaded. HA uses the CPU and memory reservation to decide failover, while the actual usage might be higher. You can also set up affinity and anti-affinity rules in DRS to distribute virtual machines to help availability of critical resources. For example, you can use an anti-affinity rule to make sure two virtual machines running a critical application never run on the same host.

Using HA and DRS together combines automatic failover with load balancing. This combination can result in a fast rebalancing of virtual machines after HA has moved virtual machines to different hosts. You can set up affinity and anti-affinity rules to start two or more virtual machines preferentially on the same host (affinity) or on different hosts (anti-affinity). See [“Using DRS Affinity Rules”](#) on page 101.

Valid, Yellow, and Red Clusters

The VI Client indicates whether a cluster is valid, overcommitted (yellow), or invalid (red). Clusters can become overcommitted because of a DRS violation. Clusters can become invalid because of a DRS violation or an HA violation. A message displayed in the Summary page indicates the issue.

Valid Cluster

A cluster is valid unless something happens that makes it overcommitted or invalid.

- A DRS cluster can become overcommitted if a host fails.
- A DRS cluster can become invalid if VirtualCenter becomes unavailable and you power on virtual machines using a VI Client connected directly to an ESX Server host.
- An HA cluster becomes invalid if the current failover capacity is lower than the configured failover capacity or if all the primary hosts in the cluster are not responding. See [“Primary and Secondary Hosts”](#) on page 74.
- A DRS or HA cluster can become invalid if the user reduces the reservation on a parent resource pool while a virtual machine is in the process of failing over.

In a valid cluster, there are enough resources to meet all reservations and to support all running virtual machines. In addition, at least one host has enough resources for each virtual machine. If you use a particularly large virtual machine (for example, with an 8GB reservation), you must have at least one host with that much memory (or CPU). It's not enough if two hosts together fulfill the requirement.

Example 1: Valid Cluster, All Resource Pools of Type Fixed

The example below shows a valid cluster and how its CPU resources are computed. The cluster has the following characteristics:

- A cluster with total resources of 12GHz.
- Three resource pools, each of type **Fixed** (that is, **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 Reservation Used 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 therefore 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 (Reservation). One virtual machine with 3GHz is powered on. No resources for powering on additional virtual machines are available.

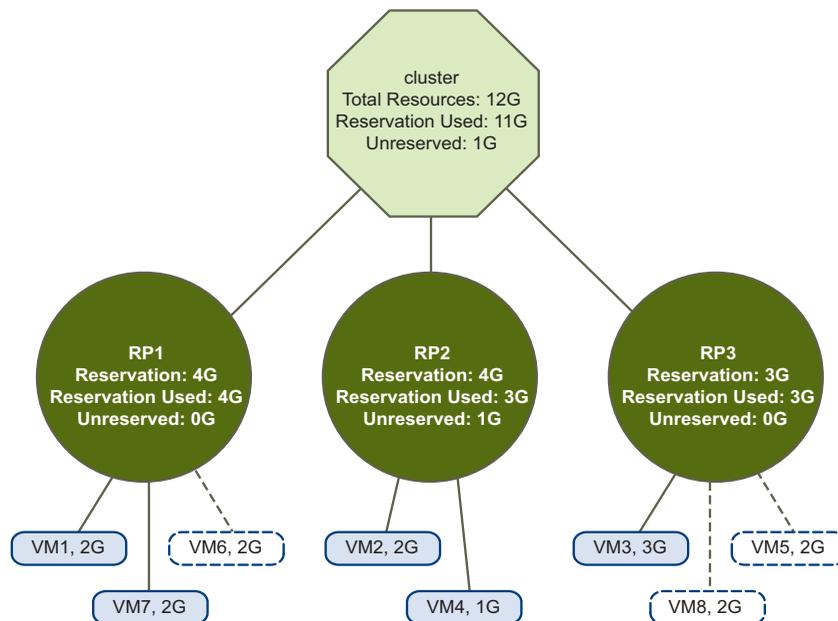


Figure 4-7. Valid Cluster (Fixed Resource Pools)

Example 2: Valid Cluster, Some Resource Pools of Type Expandable

Example 2 uses similar settings to Example 1; however, RP1 and RP3 use reservation type **Expandable**. A valid cluster could 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 of the three resource pools combined is 14 GHz (6GHz for RP1, 3 GHz for RP2, and 5GHz for RP3). 14GHz shows up as the **Reservation Used** 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 (e.g. VM1 and VM7) can use local 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). 1GHz remains unreserved.
- RP3 was created with a reservation of 4GHz. One virtual machine of 3GHz is powered on. Even though this resource pool is of type **Expandable**, no additional 2GB virtual machine can be powered on because the parent's extra resources are already used by RP1.

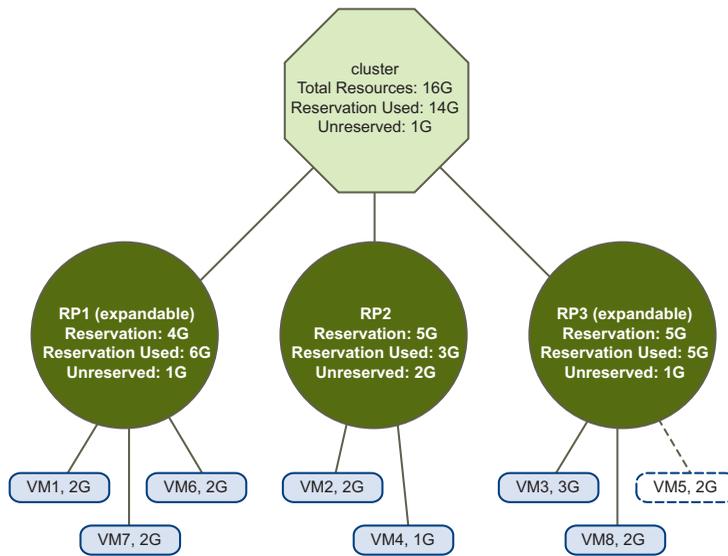


Figure 4-8. Valid Cluster (Expandable Resource Pools)

Yellow Cluster

A cluster becomes yellow when the tree of resource pools and virtual machines is internally consistent but there isn't enough capacity in the cluster to support all resources reserved by the child resource pools. Note also 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 goes down. It is recommended that you leave adequate additional resources in the cluster to avoid having your cluster turn yellow.

Consider the following 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 10GHz (3+3+4 GHz). That shows up as the **Reservation Used** in the cluster.
- One of the 4GHz hosts goes down, so the total resources are reduced to 8GHz.
- At the same time, VM4 (1GHz), VM8 (2GHz), 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 8 GHz. 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.

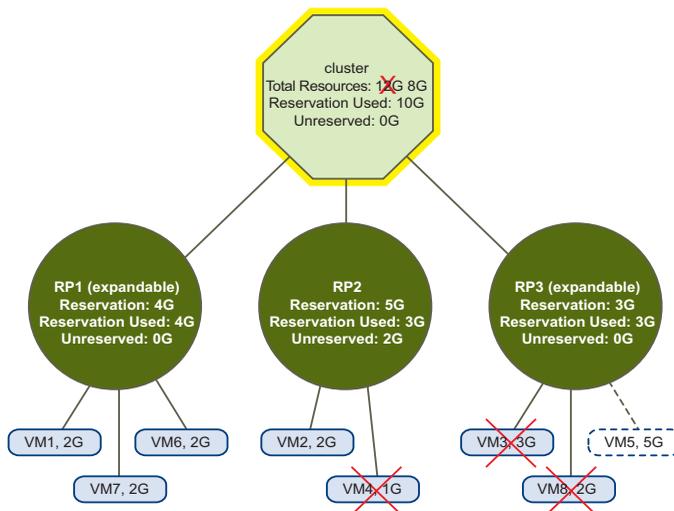


Figure 4-9. Yellow Cluster

Red Cluster

A cluster can become red because of a DRS violation or an HA violation.

The behavior of the cluster depends on the type of violation, as discussed in this section.

Red DRS Cluster

A cluster enabled for DRS becomes red when the tree is no longer internally consistent and does not have enough resources available. The total resources in the cluster have nothing to do with whether the cluster is yellow or red. It is possible for the cluster to be DRS red even if there are enough resources at the root level, if there is an inconsistency at a child level. For example, a DRS cluster turns red if the virtual machines in a fixed resource pool use more resources than the Reservation of that resource pool allows.

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're in the yellow state, not in the red state.

A cluster can also turn red if you reconfigure a resource pool while a virtual machine is in the process of failing over. A virtual machine that is in the process of failing over is disconnected and does not count toward the reservation used by the parent resource pool. So it is possible that you reduce the reservation of the parent resource pool before the failover completes. Once 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.

Consider the example in [Figure 4-10](#).

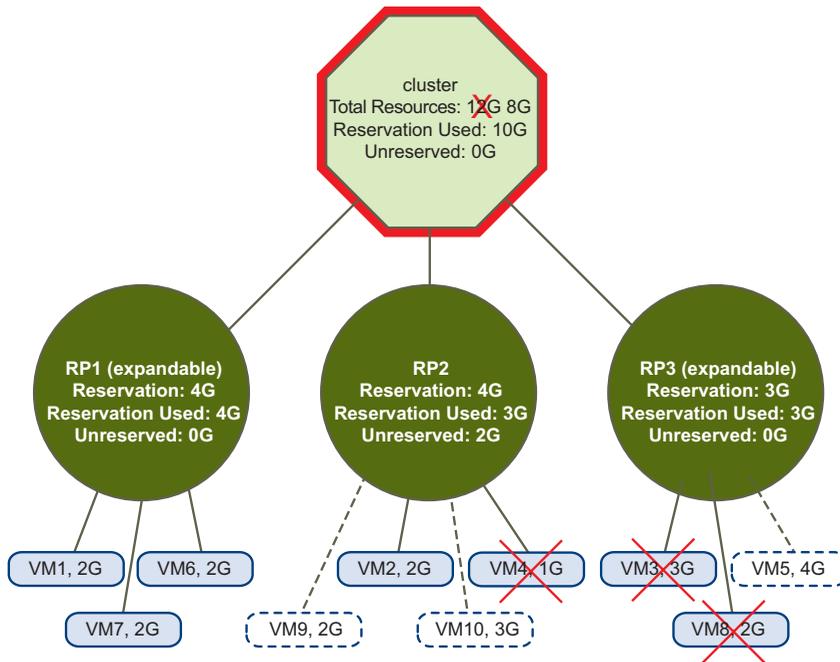


Figure 4-10. Red Cluster

In the example above, assume the administrator connects to the remaining hosts using the VI Client and powers on two virtual machines with reservations of 3GHz and 4GHz. (there are enough resources on the hosts to do so). When the VirtualCenter Server is restored, it detects that the reservation for RP2 has been exceeded, because virtual machines using a total of 5GHz are running. The cluster therefore turns red until you correct the problem.

NOTE HA behavior is not affected if a cluster is red because of a DRS issue.

Red HA Cluster

A cluster enabled for HA becomes red when the number of virtual machines powered on exceeds the failover requirements, that is, the current failover capacity is smaller than configured failover capacity. If strict admission control is disabled, clusters do not become red, regardless of whether or not the hosts can guarantee failover.

Inadequate failover capacity can happen, for example, if you power on so many virtual machines that the cluster no longer has sufficient resources to guarantee failover for the specified number of hosts.

It can also happen if HA is set up for two-host failure in a four-host cluster and one host fails. The remaining three hosts might no longer be able to satisfy a two-host failure.

If a cluster enabled for HA becomes red, it can no longer guarantee failover for the specified number of hosts but does continue performing failover. In case of host failure, HA first fails over the virtual machines of one host in order of priority, then the virtual machines of the second host in order of priority, and so on. See [“Customizing HA for Virtual Machines”](#) on page 115.

The Summary page displays a list of configuration issues for red and yellow clusters. The list explains what causes the cluster to become overcommitted or invalid.

NOTE DRS behavior is not affected if a cluster is red because of an HA issue.

5

Creating a VMware Cluster

This chapter describes the steps for creating a cluster. It discusses these topics:

- [“Cluster Prerequisites”](#) on page 83
- [“Cluster Creation Overview”](#) on page 86
- [“Creating a Cluster”](#) on page 87
- [“Viewing Cluster Information”](#) on page 89

Adding virtual machines to the cluster is discussed in [Chapter 8, “Clusters and Virtual Machines,”](#) on page 111.

NOTE All tasks assume you have permission to perform them. See the online Help for information on permissions and how to set them.

Cluster Prerequisites

Your system must meet certain prerequisites to use VMware cluster features successfully.

- In general, DRS and HA work best if the virtual machines meet VMotion requirements, as discussed in the next section.
- If you want to use DRS for load balancing, 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.

Clusters Enabled for HA

- For clusters enabled for HA, all virtual machines and their configuration files must reside on shared storage (typically a SAN), because you must be able to power on the virtual machine on any host in the cluster. This also means that the hosts must be configured to have access to the same virtual machine network and to other resources.
- Each host in an HA cluster must be able to resolve the host name and IP address of all other hosts in the cluster. To achieve this, you can either set up DNS on each host (preferred) or fill in the `/etc/hosts` entries manually (error prone and discouraged).

NOTE All hosts in an HA cluster must have DNS configured so that the short hostname (without the domain suffix) of any host in the cluster can be resolved to the appropriate IP address from any other host in the cluster. Otherwise, the *Configuring HA* task fails.

- For VMware HA, a redundancy in console networking is highly recommended (though not required). When a host's network connection fails, the second connection can broadcast heartbeats to other hosts.

To set up redundancy, you need two physical network adapters on each host. You then connect them to the corresponding service console, either using two service console interfaces or using a single interface using NIC teaming.

VirtualCenter VMotion Requirements

To be configured for VMotion, each host in the cluster must meet the following requirements.

Shared Storage

Ensure that the managed hosts use shared storage. Shared storage is typically on a storage area network (SAN). See the *VMware SAN Configuration Guide* for additional information on SAN and the *Server Configuration Guide* for information on 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 both source and target 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 both source and destination hosts (just like .vmdk virtual disk files). Swap files are placed on a VMFS by default, but administrators might override the file location using advanced virtual machine configuration options.

Processor Compatibility

Ensure that the source and destination hosts have a compatible set of processors.

VMotion transfers the running architectural state of a virtual machine between underlying VMware ESX Server systems. VMotion compatibility therefore means that the processors of the target 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 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.

- In most cases, different processor versions within the same family are similar enough to maintain compatibility.
- In some cases, 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.

NOTE VMware is working on maintaining VMotion compatibility across the widest range of processors through partnerships with processor and hardware vendors. For current information, contact your VMware representative.

Other Requirements

- For ESX Server 3.0, the virtual machine configuration file for ESX Server hosts must reside on a VMFS.
- VMotion does not currently support raw or undoable virtual disks or migration of applications clustered using Microsoft Cluster Service (MSCS).

- VMotion requires a Gigabit Ethernet network between hosts.
- 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.

Cluster Creation Overview

Creating clusters is a simple process.

- First, make sure that your system meets cluster prerequisites. See [“Cluster Prerequisites”](#) on page 83.
- Then, invoke the New Cluster wizard.

To invoke the cluster wizard

- 1 Right-click the datacenter or folder and choose **New Cluster** (Ctrl-L is the keyboard shortcut).
- 2 Choose cluster settings as prompted by the wizard and explained in this chapter.

The rest of this chapter discusses the pages of the cluster wizard in the following sections:

- [“Choosing Cluster Features”](#) on page 87
- [“Selecting Automation Level”](#) on page 87
- [“Selecting High Availability Options \(HA\)”](#) on page 88
- [“Finishing Cluster Creation”](#) on page 88

In the first panel, you choose whether to create a cluster that supports VMware DRS, VMware HA, or both. That choice affects the pages displayed subsequently, and implicitly determines the list of tasks displayed in the left panel of the wizard. If you select both DRS and HA, you are prompted for configuration information for both options.

NOTE When you create a cluster, it initially does not include any hosts or virtual machines.

Adding hosts is discussed in [“Adding Hosts to a DRS Cluster”](#) on page 96 and [“Adding Hosts to an HA Cluster”](#) on page 106.

Adding virtual machines is discussed in [Chapter 8, “Clusters and Virtual Machines,”](#) on page 111.

Creating a Cluster

This section discusses each of the pages in the New Cluster wizard.

- [“Choosing Cluster Features”](#) on page 87
- [“Selecting Automation Level”](#) on page 87
- [“Selecting High Availability Options \(HA\)”](#) on page 88
- [“Finishing Cluster Creation”](#) on page 88

Choosing Cluster Features

The first panel in the New Cluster wizard allows you to specify the following information:

- **Name** — This field specifies the name of the cluster. This name appears in the VI Client’s inventory panel. You must specify a name to continue with cluster creation.
- **Enable VMware HA** — If this box is selected, VirtualCenter automatically restarts running virtual machines on a different host when the source host fails. See [“Understanding VMware HA”](#) on page 69.
- **Enable VMware DRS** — If this box is selected, DRS uses load distribution information for initial placement and load balancing recommendations, or to place and migrate virtual machines automatically. See [“Understanding VMware DRS”](#) on page 65.

Specify the name and choose one or both cluster features, and click **Next** to continue.

NOTE You can later change the selected cluster features. See [Chapter 6, “Managing VMware DRS,”](#) on page 95 and [Chapter 7, “Managing VMware HA,”](#) on page 105.

Selecting Automation Level

If you have chosen the **Enable VMware DRS** option in the second panel of the wizard, the VMware DRS panel allows you to select the level of automation. See [“Understanding VMware DRS”](#) on page 65 for a detailed discussion of the different choices.

NOTE You can later change the level of automation for the whole cluster or for individual virtual machines. See [“Reconfiguring DRS”](#) on page 100 and [“Customizing DRS for Virtual Machines”](#) on page 114.

The following table summarizes the choices offered by the wizard.

Table 5-1.

	Initial Placement	Load Balancing
Manual	Display of recommended host(s).	Migration recommendation is displayed.
Partially Automatic	Automatic placement.	Migration recommendation is displayed.
Fully Automatic	Automatic placement.	Migration recommendations are executed automatically.

Selecting High Availability Options (HA)

If you have enabled HA, the New Cluster wizard allows you to set the following options. See [“Working with VMware HA”](#) on page 108 for detailed information.

Table 5-2.

Option	Description
Host Failures	Specifies the number of host failures for which you want to guarantee failover.
Admission Control	<p>Offers two choices:</p> <ul style="list-style-type: none"> ■ Do not power on virtual machines if they violate availability constraints enforces the failover capacity set above. ■ Allow virtual machines to be powered on even if they violate availability constraints allows you to power on virtual machines even if failover of the number of specified hosts can no longer be guaranteed. <p>If that option is not selected (the default), the following operations are also not allowed:</p> <ul style="list-style-type: none"> ■ Reverting a powered off virtual machine to a powered on snapshot ■ Migrating a virtual machine into the cluster ■ Reconfiguring a virtual machine to increase its CPU or memory reservation

Finishing Cluster Creation

After you’ve completed all selections for your cluster, the wizard presents a Summary page that lists the options you selected. Click **Finish** to complete cluster creation, or **Back** to go back and make modifications to the cluster setup.

You can now view the cluster information (see [“Viewing Cluster Information”](#) on page 89) or add hosts and virtual machines to the cluster (see [“Adding Hosts to a DRS Cluster”](#) on page 96 and [“Adding Hosts to an HA Cluster”](#) on page 106).

You can also customize cluster options, as discussed in [Chapter 6, “Managing VMware DRS,”](#) on page 95 and [Chapter 7, “Managing VMware HA,”](#) on page 105.

Viewing Cluster Information

This section discusses a few pages of particular interest when you select a cluster in the inventory panel.

NOTE For information about all other pages, see the online Help.

Summary Page

The Summary page displays summary information for the cluster.

Summary	Virtual Machines	Hosts	Migrations	Resource Allocation	Performance	Tasks & Events	Alarms	Permissions	Maps
General VMware DRS: Enabled VMware HA: Enabled Total CPU Resources: 12 GHz Total Memory: 16 GB Number of Hosts: 2 Total Processors: 4 Number of Virtual Machines: 6 Total Migrations: 0					VMware HA Admission Control: Do not allow constraint violations Current Failover Capacity: 2 hosts Configured Failover Capacity: 1 host				
Commands New Virtual Machine Add Host New Resource Pool Edit Settings					VMware DRS Automation Level: Manual Migration Recommendations: 0 Migration Threshold: Apply recommendations with four or more stars.				
					DRS Resource Distribution 				

Table 5-3. Cluster Summary Information

Panel	Description
General	<p>Includes information about the cluster:</p> <p>VMware DRS — Enabled or Disabled.</p> <p>VMware HA — Enabled or Disabled.</p> <p>Total CPU Resources — Total CPU resources available for the cluster. The sum of all resources available from the hosts.</p> <p>Total Memory — Total memory for the cluster. The sum of all resources available from the hosts.</p> <p>Number of Hosts — Number of hosts in the cluster. This number can change if you add or remove hosts.</p> <p>Total Processors — Sum of all processors of all hosts.</p> <p>Number of Virtual Machines — Total of all virtual machines in the cluster or any of its child resource pools. Includes virtual machines that are not currently powered on.</p> <p>Total Migrations — Total migrations performed by DRS or by the user since the cluster was created.</p>
Commands	<p>Allows you to invoke commonly used commands for a cluster.</p> <p>New Virtual Machine — Brings up a New Virtual Machine wizard. The wizard prompts you to choose one of the hosts in the cluster.</p> <p>Add Host — Adds a host not currently managed by the same VirtualCenter Server. To add a host managed by the same VirtualCenter Server, drag and drop the host in the inventory panel.</p> <p>New Resource Pool — Creates a child resource pool of the cluster.</p> <p>Edit Settings — Brings up the cluster's Edit Settings dialog box.</p>
VMware HA	<p>Displays the admission control setting, current failover capacity, and configured failover capacity for clusters enabled for HA.</p> <p>The system updates the current failover capacity whenever a host has been added to or removed from the cluster or when virtual machines have been powered on or powered off.</p>
VMware DRS	<p>Displays the automation level, migration threshold, and outstanding migration recommendations for the cluster.</p> <p>Migration recommendations appear if you select the Migrations tab. See “Migration Recommendations” on page 67.</p> <p>Automation level and migration threshold are set during cluster creation. See “Migration Threshold” on page 67.</p>
DRS Resource Distribution	<p>Displays two real-time histograms, Utilization Percent and Percent of Entitled Resources Delivered. The charts illustrate how balanced a cluster is. See “DRS Resource Distribution Charts” on page 91.</p>

DRS Resource Distribution Charts

The two DRS Resource Distribution charts allow you to evaluate the health of your cluster.

- The top chart is a histogram that shows the number of hosts on the X axis and the utilization percentage on the Y axis. If the cluster is unbalanced, you see multiple bars, corresponding to different utilization levels. For example, you might have one host at 20 percent CPU utilization, another at 80 percent CPU utilization, each represented by a blue bar. In automatic clusters, DRS migrates virtual machines from the heavily loaded host to the host that's at 20 percent resource utilization. The result is a single blue bar in the 40-50 percent range for hosts of similar capacity.

For a balanced cluster, this chart shows two bars: one for CPU utilization and one for memory utilization. However, if the hosts in the cluster are lightly utilized, you may have multiple bars for both CPU and memory in a balanced cluster.

- The bottom chart is a histogram that shows the number of hosts on the Y-axis and the percentage of entitled resources delivered for each host on the X-axis. While the top chart reports raw resource utilization values, the bottom chart also incorporates information about resource settings for virtual machines and resource pools.

DRS computes a resource entitlement for each virtual machine, based on its configured shares, reservation, and limit settings, as well as current resource pool configuration and settings. DRS then computes a resource entitlement for each host by adding up the resource entitlements for all virtual machines running on that host. The percentage of entitled resources delivered is equal to the host's capacity divided by its entitlement.

For a balanced cluster, a host's capacity should be greater than or equal to its entitlement, so the chart should ideally have a single bar for each resource in the 90-100 percent histogram bucket. An unbalanced cluster has multiple bars. Bars with low X-axis values indicate that virtual machines on those hosts are not getting the resources to which they are entitled.

The charts update each time the Summary page appears, and update periodically as performance limitations permit.

Migration Page

The Migration page displays the current set of DRS migration recommendations. The system provides recommendations to optimize resource utilization in the cluster. VirtualCenter updates the list of recommendations periodically, based on the value set for the cluster.

By default, the recommendations are sorted from highest to lowest priority. Click any column title to re-sort the list.

If there are no current recommendations, the Migration Recommendations section displays **No migration recommendations at this time.**

Migration Recommendations

Priority	Virtual Machine	Reason	Source Host	Target Host	CPU Load	Memory Load
★★★★★	test06	Satisfy anti-affinity rule	vcuiqa010.en...	vcuiqa012.en...	source: 6019 MHz, target: 24 MHz	source: 2 GB, target: 511 MB
★★★★★	test01	Balance average CPU loads	vcuiqa010.en...	vcuiqa012.en...	source: 6019 MHz, target: 24 MHz	source: 2 GB, target: 511 MB
★★★★★	test02	Balance average CPU loads	vcuiqa010.en...	vcuiqa012.en...	source: 6019 MHz, target: 24 MHz	source: 2 GB, target: 511 MB
★★★	test08	Balance average CPU loads	vcuiqa010.en...	vcuiqa012.en...	source: 6019 MHz, target: 24 MHz	source: 2 GB, target: 511 MB

Apply Migration Recommendation

Figure 5-1. Migration Recommendations Example

Each recommendation represents an action to improve overall performance of virtual machines within the cluster. There are several types of migration recommendations:

- Balance average CPU loads.
- Balance average memory loads.
- Satisfy affinity rule. See [“Using DRS Affinity Rules”](#) on page 101.
- Satisfy anti-affinity rule. See [“Using DRS Affinity Rules”](#) on page 101.
- Host is entering maintenance. See [“Maintenance Mode”](#) on page 68.

The Migration Recommendations table displays information about each item in the following columns:

Table 5-4. Migration Recommendations Information

Column	Description
Priority	Priority for this recommendation, as a series of stars. Five stars, the maximum, indicate a mandatory move due to a host entering maintenance mode or affinity rule violations. See “Migration Recommendations” on page 67.
Virtual Machine	Name of the virtual machine to migrate.
Reason	Reason why a virtual machine is recommended for migration. Each recommendation is associated with one or more reasons for the recommendation, as listed above.
Source Host	Host on which the virtual machine is currently running.
Target Host	Host to which the virtual machine will be migrated.

Table 5-4. Migration Recommendations Information (Continued)

Column	Description
CPU Load	CPU cycles used by the virtual machine.
Memory Load	Active memory used by the virtual machine.

Migration History

The Migration History immediately below the Migration Recommendations displays the recommendations applied for this cluster over a period of time.

Priority	Virtual Machine	Reason	Source Host	Target Host	CPU Load	Memory Load
★★★★★	test06	Satisfy anti-affinity rule	vcuiqa010.en...	vcuiqa012.en...	source: 6019 MHz, target: 24 MHz	source: 2 GB, target: 511 MB
★★★★★	test01	Balance average CPU loads	vcuiqa010.en...	vcuiqa012.en...	source: 6019 MHz, target: 24 MHz	source: 2 GB, target: 511 MB
★★★★★	test02	Balance average CPU loads	vcuiqa010.en...	vcuiqa012.en...	source: 6019 MHz, target: 24 MHz	source: 2 GB, target: 511 MB
★★★	test08	Balance average CPU loads	vcuiqa010.en...	vcuiqa012.en...	source: 6019 MHz, target: 24 MHz	source: 2 GB, target: 511 MB

Apply Migration Recommendation

Virtual Machine	Time	Source Host	Target Host
test02	5/19/2006 2:44:28 PM	vcuiqa012.eng.vmw...	vcuiqa010.eng.vmw...
test03	5/19/2006 12:06:58 ...	vcuiqa012.eng.vmw...	vcuiqa010.eng.vmw...
test03	5/19/2006 11:43:36 ...	vcuiqa010.eng.vmw...	vcuiqa012.eng.vmw...
test02	5/19/2006 11:04:27 ...	vcuiqa010.eng.vmw...	vcuiqa012.eng.vmw...
test01	5/19/2006 11:03:35 ...	vcuiqa010.eng.vmw...	vcuiqa012.eng.vmw...

Managing VMware DRS

This chapter explains how to add hosts to a DRS cluster, how to remove them, and how to customize DRS. It contains the following sections:

- [“Introduction”](#) on page 95
- [“Adding Hosts to a DRS Cluster”](#) on page 96
- [“Removing Hosts from Clusters”](#) on page 97
- [“Applying DRS Migration Recommendations”](#) on page 99
- [“Reconfiguring DRS”](#) on page 100
- [“Using DRS Affinity Rules”](#) on page 101

Adding, removing, and customizing virtual machines is discussed in [Chapter 8, “Clusters and Virtual Machines,”](#) on page 111.

NOTE All tasks described assume you are licensed and you have permission to perform them. See the online Help for information on permissions and how to set them.

Introduction

After you have created a cluster, you can enable it for DRS, HA, or both. You can then proceed to add or remove hosts, and customize the cluster in other ways.

You can customize DRS as follows:

- Specify the automation level and migration threshold during cluster creation. See [“Selecting Automation Level”](#) on page 87.
- Add hosts to the cluster. See [“Adding Hosts to a DRS Cluster”](#) on page 96

- Change the automation level or migration threshold for existing clusters, as discussed in [“Reconfiguring DRS”](#) on page 100.
- Set custom automation levels for individual virtual machines in your cluster to override the cluster-wide settings. For example, you can set the cluster’s automation level to automatic but the level of some individual machines to manual. See [“Customizing DRS for Virtual Machines”](#) on page 114.
- Group virtual machines by using affinity rules. Affinity rules specify that selected virtual machine should always be placed on the same host. Anti-affinity rules specify that selected virtual machines should always be placed on different hosts. See [“Using DRS Affinity Rules”](#) on page 101.

You can perform the following task on all clusters:

- Remove hosts from clusters, as discussed in [“Removing Hosts from Clusters”](#) on page 97.

Adding Hosts to a DRS Cluster

The procedure for adding hosts to a cluster is different for hosts currently managed by the same VirtualCenter Server (managed host) than for hosts not currently managed by that server. This section discusses the following procedures:

- [“Adding Managed Hosts to a Cluster”](#) on page 96
- [“Adding Unmanaged Hosts to a Cluster”](#) on page 97

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

Adding Managed Hosts to a Cluster

The VirtualCenter inventory panel displays all clusters and all hosts managed by that VirtualCenter Server. For information on adding a host to a VirtualCenter Server, see the *Virtual Infrastructure User’s Guide*.

To add a managed host to a cluster

- 1 Select the host from either the inventory or list view.
- 2 Drag the host to the target cluster object.
- 3 The wizard asks what you want to do with the host’s virtual machines and resource pools.

- If you choose **Put this host's virtual machines in the cluster's root resource pool**, VirtualCenter makes any direct children of the host (virtual machines or resource pools) direct children of the cluster and discards the hierarchy. Any existing resource pools are removed.
- If you choose **Create a new resource pool for this host's virtual machines and resource pools**, VirtualCenter 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>**.

NOTE If the 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.

NOTE When you later remove the host from the cluster, the resource pool hierarchy remains part of the cluster. The host loses the resource pool hierarchy. This loss makes sense because one of the goals of resource pools is to support host-independent resource allocation. You can, for example, remove two hosts and replace them with a single host with similar capabilities without additional reconfiguration.

If you want to take advantage of automatic migration features, you must also set up the host's VMotion network.

Adding Unmanaged Hosts to a Cluster

You can add a host that is not currently managed by the same VirtualCenter Server as the cluster (and is therefore not visible in the VI Client).

To add an unmanaged host to a cluster

- 1 Select the cluster to which you want to add the host and choose **Add Host** from the right-button menu.
- 2 Supply the host name, user name, and password, and click **Next**.
- 3 View the summary information and click **Next**.
- 4 Answer the prompt about virtual machine and resource pool location discussed in [“Adding Managed Hosts to a Cluster”](#) on page 96, above.

Removing Hosts from Clusters

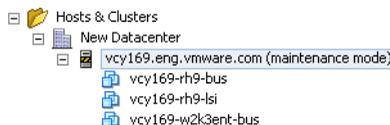
If you want to remove a host from a cluster, you must first place the host in maintenance mode. See [“Maintenance Mode”](#) on page 68 for background information.

To place a host in maintenance mode

- 1 Select the host and choose **Enter Maintenance Mode** from the right-button menu.

The host is in a state of **Entering Maintenance Mode** (as shown in the inventory panel) until you power down all running virtual machines or migrate them to different hosts. You cannot power on virtual machines or migrate virtual machines to a host entering maintenance mode.

When there are no powered on virtual machines, the host is in maintenance mode, as shown in the inventory panel.



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

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.

- 3 After you have moved the host, you can:
 - Remove the host from the VirtualCenter Server (Choose **Remove** from the right-button menu).
 - Run the host as a standalone host under VirtualCenter (Choose **Exit Maintenance Mode** from the right-button menu).
 - Move the host into another cluster.

Host Removal and Resource Pool Hierarchies

When you remove a host from a cluster, the host ends up with only the (invisible) 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 new, host-specific resource pool hierarchy.

Host Removal and Virtual Machines

Because the host must be in maintenance mode before you can remove it, all virtual machines must be powered off. When you then remove the host from the cluster, the

virtual machines that are currently associated with the host are also removed from the cluster.

NOTE Because DRS migrates virtual machines from one host to another, you might not have the same virtual machines on the host as when you originally added the host.

Host Removal and Invalid Clusters

If you remove a host from a cluster, the resources available for the cluster decrease.

If the cluster is enabled for DRS, removing a host can have the following results:

- If the cluster still 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.

If a cluster enabled for HA loses so many resources that it can no longer fulfill its failover requirements, a message appears and the cluster turns red. The cluster fails over virtual machines in case of host failure, but is not guaranteed to have enough resources available to fail over all virtual machines.

Applying DRS Migration Recommendations

If you create a cluster in manual or partially automated mode, VirtualCenter displays the number of outstanding migration recommendations on the Summary page and the recommendations themselves on the Migration page.

The system supplies as many recommendations as necessary to properly balance cluster resources and includes detailed information. See [“Migration Page”](#) on page 91.

The recommendations list shows all of the migrations that DRS suggests. Migrations that are all part of the same recommendation (for example, because of DRS affinity) are indicated by a link icon.

To apply migration recommendations

- 1 Select a recommendation, then click **Apply Recommendation**.

Migration Recommendations

Priority	Virtual Machine	Reason	Source Host	Target Host	CPU Load	Memory Load
★★★★★	test06	Satisfy anti-affinity rule	vcuiqa010.en...	vcuiqa012.en...	source: 6019 MHz, target: 24 MHz	source: 2 GB, target: 511 MB
★★★★★	test01	Balance average CPU loads	vcuiqa010.en...	vcuiqa012.en...	source: 6019 MHz, target: 24 MHz	source: 2 GB, target: 511 MB
★★★★★	test02	Balance average CPU loads	vcuiqa010.en...	vcuiqa012.en...	source: 6019 MHz, target: 24 MHz	source: 2 GB, target: 511 MB
★★★	test08	Balance average CPU loads	vcuiqa010.en...	vcuiqa012.en...	source: 6019 MHz, target: 24 MHz	source: 2 GB, target: 511 MB

Apply Migration Recommendation

At times, a virtual machine is part of a recommendation involving multiple virtual machines (for example, if the machine is part of an affinity or anti-affinity group). If you select such a virtual machine and the dependent virtual machines are not part of the selection, a warning dialog box asks whether you want to include the dependent virtual machines in this set of migrations.

- 2 Make one of the following choices:
 - Click **Yes** to add dependent virtual machines (if any), and click **OK** to apply the recommendations.
 - Click **No** to migrate only the initially selected virtual machines, and click **OK** to apply the recommendations.

Once a recommendation is applied, the number of migration recommendations decreases and the number of total migrations increases on the Summary page.

NOTE If the cluster is in Manual mode, and you select to apply several recommendations, making one change might affect the other recommendations (for example, because host resources are increased or decreased).

In that case, some of the recommendations might be displayed again after the recommendations are refreshed (which happens every five minutes by default). Select the recommendations again to have them take effect.

Reconfiguring DRS

You can turn off DRS for a cluster, or you can change the configuration options.

To turn off DRS

- 1 Select the cluster.
- 2 Choose **Edit Settings** from the right-button menu.

- 3 In the left panel, select **General**, and deselect the **VMware DRS** check box.
You are warned that turning off DRS destroys all resource pools in the cluster.
- 4 Click **OK** to turn off DRS and destroy all resource pools.



CAUTION The resource pools do not become reestablished when you turn DRS back on.

To reconfigure DRS

- 1 Select the cluster.
- 2 Choose **Edit Settings** from the right-button menu.
- 3 In the Cluster Settings dialog box, select **VMware DRS**.
- 4 You can now set the automation level:
 - Select one of the radio buttons to change automation level. See [“Selecting Automation Level”](#) on page 87.
 - If you’ve chosen **Fully automated**, you can move the **Migration Threshold** slider to change the migration threshold. See [“Migration Threshold”](#) on page 67.

NOTE The Advanced Options dialog box is helpful when you are working with VMware customer support on resolving an issue. Setting advanced options is not otherwise recommended.

Using DRS Affinity Rules

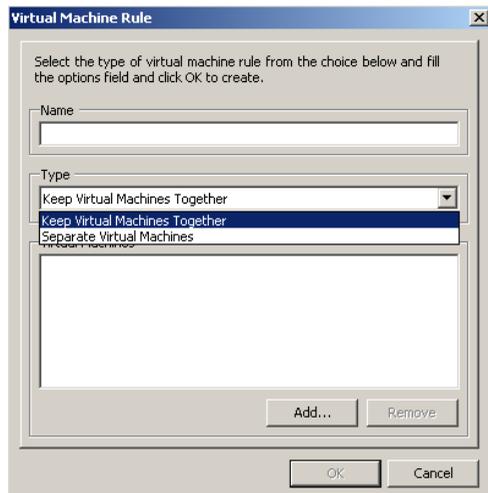
After you have created a DRS cluster, you can edit its properties to create rules that specify affinity. You can use these rules to determine that:

- DRS should try to keep certain virtual machines together on the same host (for example, for performance reasons).
- DRS should try to make sure that certain virtual machines are not together (for example, for high availability). You might want to guarantee certain virtual machines are always on different physical hosts. When there’s a problem with one host, you don’t lose both virtual machines.

NOTE These rules are completely different from an individual host’s CPU affinity rules. CPU affinity rules are discussed in [“Using CPU Affinity to Assign Virtual Machines to Specific Processors”](#) on page 120.

To create a DRS rule

- 1 Select the cluster and choose **Edit Settings** from the right-button menu.
- 2 In the Cluster Settings dialog box, choose **Rules**.



- 3 In the Virtual Machine Rule dialog box, name the rule so you can later find and edit it.
- 4 Choose one of the options from the pop-up menu:
 - **Keep Virtual Machines Together**
 - **Separate Virtual Machines**
- 5 Click **Add** to add virtual machines, and click **OK** when you're done.

After you've added the rule, you can edit it, look for conflicting rules, or delete it.

To edit an existing rule

- 1 Select the cluster and choose **Edit Settings** from the right-button menu.
- 2 In the left panel, select **Rules** (under **VMware DRS**).
- 3 Click **Details** for additional information on topics such as conflicting rules.
- 4 Make the desired changes in the dialog box, and click **OK** when you're done.

NOTE If you have two rules that 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.

Understanding Rule Results

When you add or edit a rule, and the cluster is immediately in violation of that rule, the system continues to operate and tries to correct the violation.

For manual and partially automated DRS clusters, migration recommendations are based on both rule fulfillment and load balancing. You are not required to fulfill the rules, but the corresponding recommendations remain until the rules are fulfilled.

Disabling or Deleting Rules

You can disable a rule or remove it completely.

To disable a rule

- 1 Select the cluster and choose **Edit Settings** from the right-button menu.
- 2 In the left panel, select **Rules** (under **VMware DRS**).
- 3 Deselect the check box to the left of the rule and click **OK**.

You can later enable the rule again by reselecting the check box.

To delete a rule

- 1 Select the cluster and choose **Edit Settings** from the right-button menu.
- 2 In the left panel, select **Rules** (under **VMware DRS**).
- 3 Select the rule you want to remove and click **Remove**.

The rule is deleted.

Managing VMware HA

This chapter explains how to add hosts to an HA cluster, how to remove them, and how to customize HA clusters. It contains the following sections:

- [“Introduction”](#) on page 105
- [“Adding Hosts to an HA Cluster”](#) on page 106
- [“Working with VMware HA”](#) on page 108

Adding, removing, and customizing virtual machines is discussed in [Chapter 8, “Clusters and Virtual Machines,”](#) on page 111.

NOTE All tasks described assume you are licensed and you have permission to perform them. See the online Help for information on permissions and how to set them.

Introduction

After you have created a cluster, you can enable it for DRS, HA, or both. You can then add or remove hosts, and customize the cluster in other ways.

You can customize HA as follows:

- During cluster creation, choose the number of host failures for the cluster and indicate whether you want to enforce strict admission control. See [“Selecting High Availability Options \(HA\)”](#) on page 88.
- Add hosts, as discussed in [“Adding Hosts to an HA Cluster”](#) on page 106
- Change the number of host failures or the admission control for existing clusters, as discussed in [“Working with VMware HA”](#) on page 108.

- Set a priority for individual virtual machines. HA uses virtual machine priority to decide the order of restart so that virtual machines with higher priority from the same host get precedence in case of insufficient resources. See [“Customizing HA for Virtual Machines”](#) on page 115.
- Set an Isolation Response for individual virtual machines. By default, all virtual machines are powered off if a host becomes isolated from the network. See [“Customizing HA for Virtual Machines”](#) on page 115.

You can perform the following task on all clusters:

- Remove hosts from clusters, as discussed in [“Removing Hosts from Clusters”](#) on page 97 in the previous chapter.

Adding Hosts to an HA Cluster

The procedure for adding hosts to a cluster is different for hosts currently managed by the same VirtualCenter Server (managed host) than for hosts not currently managed by that server. This section discusses the two procedures the following sections:

- [“Adding Managed Hosts to a Cluster”](#) on page 106
- [“Adding Unmanaged Hosts to a Cluster”](#) on page 106

After the host has been added, the virtual machines deployed to the host become part of the cluster.

Adding Managed Hosts to a Cluster

The VirtualCenter inventory panel displays all clusters and all hosts managed by that VirtualCenter Server. For information on adding a host to a VirtualCenter Server, see the *Server Configuration Guide*.

To add a managed host to a cluster

- 1 Select the host from either the inventory or list view.
- 2 Drag the host to the target cluster object.

Adding the host to the cluster spawns a **Configuring HA** system task on the host. After this task has completed successfully, the host is included in the HA service.

Adding Unmanaged Hosts to a Cluster

You can add a host that is not currently managed by the same VirtualCenter Server as the cluster (and is therefore not visible).

To add an unmanaged host to a cluster

- 1 Select the cluster to which you want to add the host and choose **Add Host** from the right-button menu.
- 2 Supply the host name, user name, and password, and click **Next**.

The host is added to the cluster. Adding the host to the cluster spawns a system task **Configuring HA** on the host. After this task has completed successfully, the host is included in the HA service.

Results of Adding Hosts to a Cluster

When a host is added to an HA cluster:

- The resources for that host are immediately available to the cluster for use in the cluster's root resource pool.
- Unless the cluster is also enabled for DRS, all resource pools are collapsed into the cluster's top-level (invisible) resource pool.

NOTE The resource pool hierarchy is lost. It does not become available when you later remove the host from the cluster.

- Any capacity on the host beyond what is required or guaranteed for each running virtual machine becomes available as spare capacity in the cluster pool. This spare capacity can be used for starting virtual machines on other hosts in case of a host failure.
- If you add a host with several running virtual machines, and the cluster no longer fulfills its failover requirements because of that addition, a warning appears and the cluster is marked red.
- By default, all virtual machines on the host that was added are given a restart priority of **Medium** and an isolation response of **Shutdown**. See [“Customizing HA for Virtual Machines”](#) on page 115 for additional information on those options and on how to configure them.
- The system also monitors the status of the HA service on each host and displays information about configuration issues on the Summary page.
- When a host is removed from the cluster (or disconnected or put in maintenance mode), the HA service is unconfigured. You may see a system-spawned **Unconfiguring HA** system task on the host, which has to complete.

Configuring and Unconfiguring HA on a Host

When you add a host to an HA cluster, a system task **Configuring HA** is spawned. This task has to complete successfully before the host is ready for HA. The host state is yellow while it is being configured or unconfigured for HA, and the Summary page shows the operation that may be pending.

A host is configured for HA if you:

- Enable HA for a cluster
- Connect to a host in an HA cluster
- Exit maintenance mode on the host

A host is unconfigured for HA if you:

- Disable HA on the cluster
- Disconnect the host
- Enter maintenance mode on the host



CAUTION When you disconnect a host from an HA cluster, you reduce the available resources for failover operations. If a cluster's failover capacity is less than or equal to the configured failover capacity and you begin the process of disconnecting a host, you receive a cluster failover warning. If you complete the disconnect, the cluster may be unable to maintain its configured failover level.

A system task **Unconfiguring HA** may get spawned. In case of disconnect or entering maintenance mode, the unconfiguration is done as part of the respective tasks, and no separate system task is spawned. The HA service is also monitored on each host, and if there is an error, the host's Summary page indicates that. The host is marked red.

When a configuration or unconfiguration task fails, you can get additional information in the related events for the task. You might also need to check the logs on the host. If you fix the error, the host as a **Reconfigured HA** task to reconfigure HA on a host where the host failed.

Working with VMware HA

Reconfiguring HA can mean turning it off or reconfiguring its options.

To turn off HA

- 1 Select the cluster.
- 2 Choose **Edit Settings** from the right-button menu.
- 3 In the left panel, select **General**, and deselect the **Distributed Availability Services** check box.

To reconfigure HA

- 1 Select the cluster.
- 2 Choose **Edit Settings** from the right-button menu.
- 3 In the Cluster Settings dialog box, select **Distributed Availability Services**.
- 4 Make changes to the number of host failovers or the admission control behavior. See [“Selecting High Availability Options \(HA\)”](#) on page 88.

Clusters and Virtual Machines

8

This chapter explains how to add, remove, and customize virtual machines. It contains the following sections:

- [“Adding Virtual Machines to a Cluster”](#) on page 111
- [“Powering On Virtual Machines in a Cluster”](#) on page 112
- [“Removing Virtual Machines from a Cluster”](#) on page 113
- [“Customizing DRS for Virtual Machines”](#) on page 114
- [“Customizing HA for Virtual Machines”](#) on page 115

NOTE All tasks assume you have permission to perform them. See the online Help for information on permissions and how to set them.

Adding Virtual Machines to a Cluster

You can add virtual machines to a cluster in one of three ways.

- [“Adding a Virtual Machine During Creation”](#) on page 111
- [“Migrating a Virtual Machine to a Cluster”](#) on page 112
- [“Adding a Host with Virtual Machines to a Cluster”](#) on page 112

Adding a Virtual Machine During Creation

When you create a virtual machine, you can add it to a cluster as part of the virtual machine creation process. When the New Virtual Machine wizard prompts you for the location of the virtual machine, you can choose a standalone host or a cluster, and can choose any resource pool inside the host or cluster.

See the *Virtual Machine Management Guide* for more information on deploying virtual machines.

Migrating a Virtual Machine to a Cluster

You can migrate an existing virtual machine from a standalone host to a cluster or from a cluster to another cluster. The virtual machine can be powered on or off. To move the virtual machine using VirtualCenter, you have two choices:

- Drag the virtual machine object on top of the cluster object.
- Right-click the virtual machine name and choose **Migrate**.

For DRS clusters, users are prompted for the following information:

- The location, which could be the cluster itself or a resource pool inside the cluster.
- A host on which to power on and run the virtual machine if the cluster is in manual mode. If the cluster is in fully automatic or partially automatic mode, DRS selects the host.

NOTE You can drag a virtual machine directly to a resource pool within a cluster. In this case, the Migration wizard is invoked 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.

Adding a Host with Virtual Machines to a Cluster

When you add a host to a cluster, all virtual machines on that host are added to the cluster. See [“Adding Hosts to a DRS Cluster”](#) on page 96 and [“Adding Hosts to an HA Cluster”](#) on page 106.

Powering On Virtual Machines in a Cluster

When you power on a virtual machine on a host that is part of a cluster, the resulting VirtualCenter behavior depends on the type of cluster.

DRS Enabled

If you power on a virtual machine and DRS is enabled, VirtualCenter first performs admission control; that is, it checks whether the cluster or resource pool has enough resources for the virtual machine. Virtual Center then checks whether any host in the cluster has enough resources for powering on the virtual machine. This must be a single host. It's not enough if two hosts jointly have sufficient resources.

If the cluster does not have sufficient resources, or if there is no single host with sufficient resources, a message appears. Otherwise, VirtualCenter proceeds as follows:

- If DRS is in manual mode, VirtualCenter displays a list of recommended hosts, ordered from best to worst. You can choose one of the hosts.
- If DRS is in partially automatic or automatic mode, VirtualCenter places the virtual machine on the most suitable host.

HA Enabled

If you power on a virtual machine and HA is enabled, VirtualCenter performs HA admission control, that is, checks whether enough resources exist to allow for the specified number of host failovers if you power on the virtual machine.

- If enough resources exist, the virtual machine is powered on.
- If not enough resources exist, and if strict admission control is being used (the default), a message informs you that the virtual machine cannot be powered on. If you are not using strict admission control, the virtual machine is powered on without warnings.

Removing Virtual Machines from a Cluster

You can remove virtual machines from a cluster as follows:

- [“Migrating Virtual Machines out of a Cluster”](#) on page 113
- [“Removing a Host with Virtual Machines from a Cluster”](#) on page 114

Migrating Virtual Machines out of a Cluster

You can migrate a virtual machine from a cluster to a standalone host or from a cluster to another cluster in one of two ways:

- Use the standard drag-and-drop method.
- Select **Migrate** from the virtual machine’s right-button menu or the VirtualCenter menu bar.

If the virtual machine is a member of a DRS cluster affinity rules group (see [“Using DRS Affinity Rules”](#) on page 101), VirtualCenter displays a warning before it allows the migration to proceed. The warning indicates that dependent virtual machines are not migrated automatically. You have to acknowledge the warning before migration can proceed.

Removing a Host with Virtual Machines from a Cluster

When you remove a host with virtual machines from a cluster, all its virtual machines are removed as well. You can remove a host only if it is in maintenance mode or disconnected. See [“Removing Hosts from Clusters”](#) on page 97.

NOTE If you remove a host from an HA cluster, the cluster can become red because it no longer has enough resources for failover. If you remove a host from a DRS cluster, the cluster can become yellow because it is overcommitted. See [“Valid, Yellow, and Red Clusters”](#) on page 76.

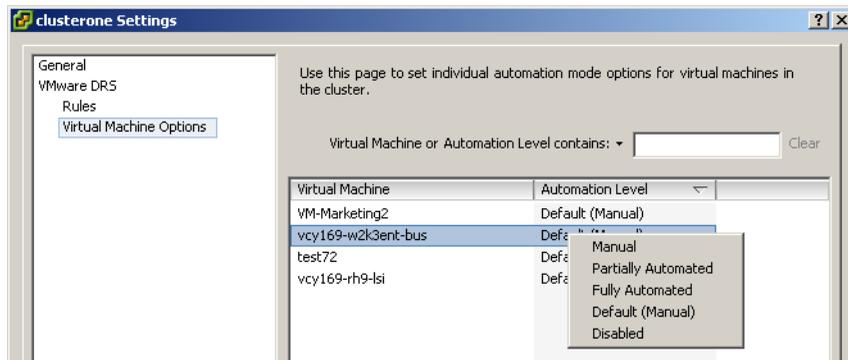
Customizing DRS for Virtual Machines

You can customize the automation mode for individual virtual machines in a DRS cluster to override the cluster’s automation mode. This allows you to fine tune automation to suit your needs. For example you could select **Manual** for specific virtual machines in a cluster with full automation, or **Partially Automated** for specific virtual machines in a manual cluster.

If a virtual machine is set to **Disabled**, VirtualCenter does not migrate that virtual machine or provide migration recommendations for it.

To set a custom automation mode for one or more virtual machines

- 1 Select the cluster and choose **Edit Settings** from the right-button menu.
- 2 In the Cluster Settings dialog box, select **Virtual Machine Options** in the left column.



- 3 Select an individual virtual machine, or Shift-select or Control-select multiple virtual machines.

- 4 From the right-button menu, choose the automation mode you want to use, then click OK.

Customizing HA for Virtual Machines

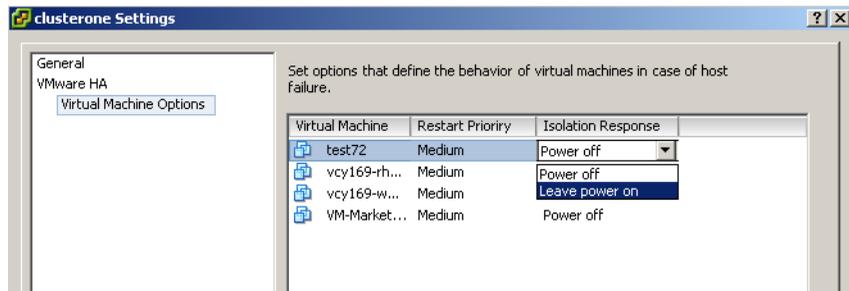
You can customize HA for restart priority and isolation response:

- **Restart priority** — Determines the order in which virtual machines are restarted upon host failure. Restart priority is always considered, but is especially important in the following cases:
 - If you've set host failure to a certain number of hosts (for example, three) and more hosts fail (for example, four).
 - If you've turned off strict admission control and have started more virtual machines than HA has been set up to support.
- **Isolation response** — Determines what happens when a host in an HA cluster loses its console network connection but continues running. The other hosts in the cluster no longer get heartbeats from this host, declare it dead, and try to restart its virtual machines. Disk locking prevents two instances of a virtual machine from running on two different host. By default, virtual machines are powered off on the isolated host in case of a host isolation incident, so that they can be restarted on a different host. You can change that behavior for individual virtual machines.

In the case of NAS or iSCSI storage, when a host loses console networking, the virtual machine might also lose access to its disk. In this case, the disk lock can be broken and the virtual machine may successfully be powered on on a different host. The virtual machine on the isolated host continues to run, but cannot access its disk any more (even if it regains network connectivity) because it has lost its disk lock. That virtual machine may be creating and consuming network I/O. It is therefore highly recommended that you keep the **Isolation Response** as **Power off** (the default) for virtual machines located on NAS or iSCSI storage.

To customize HA behavior for individual virtual machines

- 1 Select the cluster and choose **Edit Settings** from the right-button menu, then choose **Virtual Machine Options** under **VMware HA**.



- 2 For each virtual machine, you can select from the **Restart Priority** or **Isolation Response** pop-up menu to customize its settings.
 - **Restart Priority** — Indicates relative priority for restarting the virtual machine in case of host failure. Higher priority virtual machines are started first.

NOTE This priority applies only on a per-host basis. If multiple hosts fail, VirtualCenter first migrates all virtual machines from the first host in order of priority, then all virtual machines from the second host in order of priority, and so on.

- **Isolation Response** — Specifies what the ESX Server host that has lost connection with its cluster should do with running virtual machines. By default, each virtual machine is set to be shut down in the event of a host isolation incident.

You can choose **Leave running** to indicate that the virtual machines on isolated hosts should continue to run even if the host can no longer communicate with other hosts in the cluster. You might do this, for example, if the virtual machine network is on a different network that is robust and redundant, or if you want to keep the virtual machines running.

NOTE When you add a host to a cluster, all virtual machines in the cluster default to a restart priority of **Medium** and an isolation response of **Shutdown**

Advanced Resource Management

9

This chapter discusses some advanced resource management topics. It includes both conceptual information and discussion of advanced parameters you can set.

NOTE In most situations, you don't need the information presented in this chapter, or to use the special advanced settings that are discussed. In fact, using the advanced settings might be detrimental to your system's performance. However, under certain circumstances experienced administrators might find these advanced configuration options helpful.

This chapter includes the following sections:

- [“CPU Virtualization”](#) on page 118
- [“Using CPU Affinity to Assign Virtual Machines to Specific Processors”](#) on page 120
- [“Hyperthreading”](#) on page 122
- [“Virtual Memory in Virtual Machines”](#) on page 125
- [“Understanding Memory Overhead”](#) on page 128
- [“Memory Allocation and Idle Memory Tax”](#) on page 130
- [“How ESX Server Hosts Reclaim Memory”](#) on page 132
- [“Sharing Memory Across Virtual Machines”](#) on page 135
- [“Advanced Attributes and What They Do”](#) on page 136

NOTE Licenses for DRS and HA are not required for any of the topics and tasks discussed in this chapter.

CPU Virtualization

To understand CPU related issues, it's useful to consider the difference between emulation and virtualization.

- With emulation, all operations are executed 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.
- With virtualization, the underlying physical resources are used whenever possible and the virtualization layer executes instructions only as needed to make the virtual machines operate as if they were running directly on a physical machine.

Emulation provides portability and is often used to run software designed for one platform across several different platforms. For example, a number of Atari 2600 emulators can run Atari 2600 games on x86-based PCs. In contrast, virtualization emphasizes performance and runs directly on the processor whenever possible.

Virtualization Modes and Virtualization Overhead

The virtual machine can run in two different modes:

- **Direct execution** — Under certain conditions, the ESX Server Virtual Machine Monitor (VMM) can run the virtual machine directly on the underlying processor. This mode is called direct execution, and it provides near-native performance in the execution of the virtual machine's CPU instructions.
- **Virtualization mode** — If direct execution is not possible, the virtual machine CPU's instructions must be virtualized. This process adds a varying amount of virtualization overhead depending on the operation being performed.

In most cases:

- Unprivileged, user-level application code runs in direct execution mode because, in most operating systems, user-level code does not access privileged state data.
- Operating system code does modify privileged state data and thus requires virtualization mode.

As a result, a micro-benchmark that makes only system calls runs significantly more slowly in a virtual machine than on a native system. However, code that runs in direct

execution mode incurs little extra performance costs and runs at near-native CPU instruction speed.

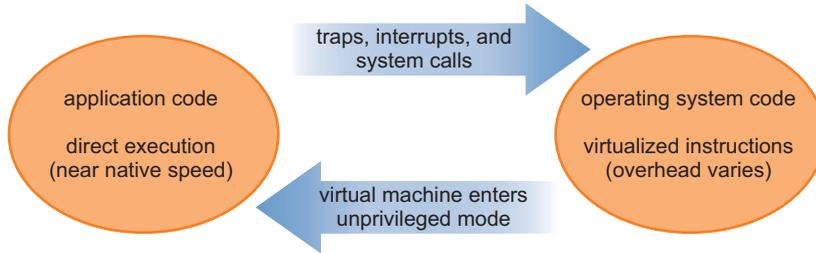


Figure 9-1. Execution States of the VMware VMM

Figure 9-1 gives a simplified view of the execution states of the VMware VMM. The virtual machine runs in direct-execution mode whenever possible. Traps, interrupts, system calls, and other events cause the virtual machine to go to the guest operating system, where instructions require virtualization by the VMM. When the guest operating system is no longer needed, the virtual machine returns to direct execution.

Performance Implications

CPU virtualization overhead depends on workload, including:

- How much time is spent in direct execution
- The cost of virtualizing the remaining instructions

Because of this, system-call only workloads have a higher virtualization overhead.

Virtualization and Processor-Specific Behavior

(SEE UPDATE) Because VMware software virtualizes the CPU, the virtual machine is aware of the specific model of the processor on which it is running. Some operating systems install different kernel versions tuned for specific processor models, and these kernels are installed in virtual machines as well. Because of the different kernel versions, it isn't possible to migrate virtual machines installed on a system running one processor model (for example, AMD) to a system running on a different processor (for example, Intel).

Performance Implications

CPU virtualization adds varying amounts of overhead depending on how much of the virtual machine's workload can run in direct execution mode, and on the costs of virtualizing the remaining instructions that can't be executed directly. See

[“Virtualization Modes and Virtualization Overhead”](#) on page 118 for background information.

An application is CPU-bound if most of the application's time is spent executing instructions rather than waiting for external events such as user interaction, device input, or data retrieval. For such applications, the CPU virtualization overhead requires additional instructions to be executed, which takes CPU processing time that could otherwise be used by the application itself. 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 utilization. If CPU is available to absorb the overhead, it can still deliver comparable performance in terms of overall throughput.

ESX Server 3 supports up to four virtual processors (CPUs) for each virtual machine.

NOTE Deploy single-threaded applications on uniprocessor virtual machines (instead of SMP virtual machines) for best performance and resource utilization.

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 could otherwise be used by other virtual machines.

Using CPU Affinity to Assign Virtual Machines to Specific Processors

Affinity means that you can restrict the assignment of virtual machines to a subset of the available processors in multiprocessor systems. You do so by specifying an affinity setting for each virtual machine.

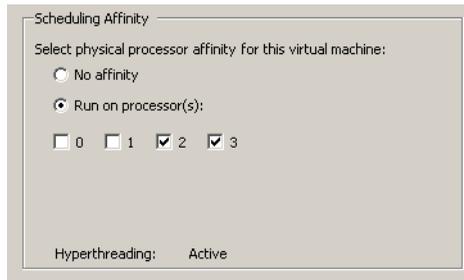


CAUTION Using affinity might not be appropriate for a variety of reasons. See [“Potential Issues with Affinity”](#) on page 121.

NOTE CPU affinity is completely different from DRS affinity, discussed in [“Customizing DRS for Virtual Machines”](#) on page 114.

To assign a virtual machine to a specific processor

- 1 In the VI Client inventory panel, select a virtual machine and choose **Edit Settings**.
- 2 Select the **Resources** tab and choose **CPU**, and then click the **Run on processor(s)** button.



- 3 Select the processors on which you want the virtual machine to run, and then click **OK**.

Potential Issues with Affinity

Virtual machine affinity assigns each virtual machine to processors in the specified affinity set. Before using affinity, consider the following potential issues:

- For multiprocessor systems, ESX Server systems perform automatic load balancing. Avoiding manual specification of virtual machine affinity improves the scheduler's ability to balance load across processors.
- Affinity can interfere with the ESX Server 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's already assigned to certain processors using affinity. See [Appendix A, "Using NUMA Systems with ESX Server,"](#) on page 147 for additional information on using NUMA with ESX Server hosts.

Hyperthreading

Intel Corporation developed hyperthreading technology to enhance the performance of its Pentium IV and Xeon processor lines. The technology allows a single processor to execute two independent threads simultaneously. While this feature does not provide the performance of a true dual-processor system, it can improve utilization of on-chip resources, leading to greater throughput for certain important workload types.

See <http://www.intel.com/technology/hyperthread> for an in-depth discussion of hyperthreading technology.

For additional information, see the white paper “Hyper-Threading Support in ESX Server 2” at http://www.vmware.com/vmtn/resources/esx_resources.html.

Introduction

Hyperthreading technology allows a single physical processor 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 often refers to a physical processor as a **core**, and this discussion uses that terminology as well.

While hyperthreading does not double the performance of a system, it can increase performance by better utilizing idle resources. 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-hyper-threaded processor. However, hyperthreading performance improvements are highly application-dependent, and some applications might actually see performance degradation with hyperthreading because many processor resources (such as the cache) are shared between both logical processors.

Enabling Hyperthreading

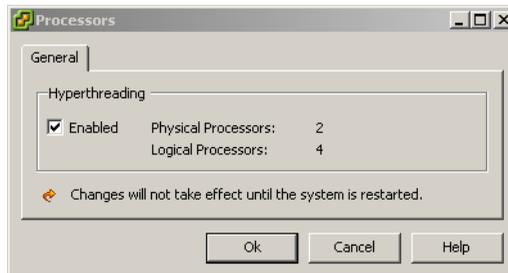
By default, hyperthreading is enabled. If it is disabled, you can enable it.

To enable hyperthreading

- 1 Ensure that your system supports hyperthreading technology.

All Intel Xeon MP processors and all Intel Xeon DP processors with 512 L2 cache support hyperthreading; however, not every Intel Xeon system ships with a BIOS that supports hyperthreading. Consult your system documentation to see if the BIOS includes support for hyperthreading. VMware ESX Server cannot enable hyperthreading on a system with more than 16 physical CPUs, because ESX Server has a logical limit of 32 CPUs.

- 2 Enable hyperthreading in the system BIOS. Some manufacturers label this option **Logical Processor** while others call it **Enable Hyperthreading**.
- 3 Make sure hyperthreading for your ESX Server host is turned on, as follows:
 - a In the VI Client, select the host and click the **Configuration** tab.
 - b Select **Processors** and click **Properties**.
 - c In the dialog box that appears, you can view hyperthreading status and turn hyperthreading off or on (it is on by default).



Hyperthreading and ESX Server

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

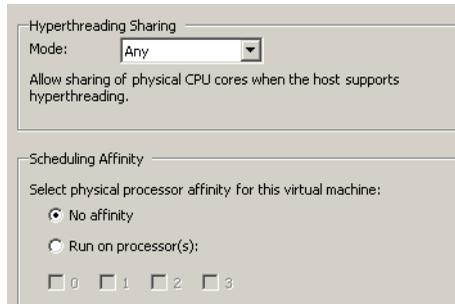
VMware ESX Server systems manage processor time intelligently to guarantee that load is spread smoothly across all physical cores in the system. If there is no work for a logical processor, it is put into a special **halted** state, which frees its execution resources and allows the virtual machine running 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, so that a virtual machine running with the full resources of a core is charged 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 Server resource allocation rules.

Advanced Server Configuration for Hyperthreading

You can specify how the virtual CPUs of a virtual machine can share physical cores on a hyper-threaded system. Two virtual CPUs share a core if they are both running on

logical CPUs of the core at the same time. You can set this for individual virtual machines, as follows:

- 1 In the VI Client's inventory panel, right-click the virtual machine and choose **Edit Settings**.
- 2 Click the **Resources** tab, and click **Advanced CPU**.
- 3 Choose from the pull-down menu to specify hyperthreading for this virtual machine.



You have the following choices

Table 9-1. Hyperthreading Sharing Choices

Option	Description
Any	This is the default for all virtual machines on a hyper-threaded 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 machines 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 is similar to none . Virtual CPUs from this virtual machine are not allowed to share cores with virtual CPUs from other virtual machines. Other virtual CPUs from the same virtual machine are allowed to share cores. This option is permitted 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 could 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 don't 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, both virtual CPUs could have been scheduled 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, it is possible that 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 Server system might detect that an application is interacting badly with hyperthreading technology. Certain types of self-modifying code, for instance, can disrupt the normal behavior of the Pentium IV trace cache and lead to substantial slowdowns (up to 90 percent) for an application sharing a core with the problem code. In those cases, the ESX Server host quarantines the virtual CPU running this code and places the virtual machine in the **none** or **internal** mode, as appropriate. Quarantining is necessary only rarely, and is transparent to the user.

Set the **Cpu.MachineClearThreshold** advanced setting for the host to **0** to disable quarantining. See [“Setting Advanced Host Attributes”](#) on page 136.

Caveats: Hyperthreading and CPU Affinity

Consider your situation carefully before you set CPU affinity on systems using hyperthreading. For example, if a high priority virtual machine is bound to CPU 0 and another high priority virtual machine is bound 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. You must make sure any custom affinity settings make sense for a hyper-threaded system. In this example, you can do so by binding the virtual machines to CPU 0 and CPU 2. Ideally, you should not use affinity settings at all. See [“Using CPU Affinity to Assign Virtual Machines to Specific Processors”](#) on page 120 for more information.

Virtual Memory in Virtual Machines

All modern operating systems provide support for virtual memory, allowing software to use more memory than the machine physically has. 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.

Virtual to Physical Memory Mapping

On a physical machine, page tables translate virtual memory addresses into physical memory addresses. Within a virtual machine, the guest operating system's page tables maintain the mapping from guest virtual pages to guest physical pages.

ESX Server 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 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 Server 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, see above).
- The shadow page tables are then 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, once 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.

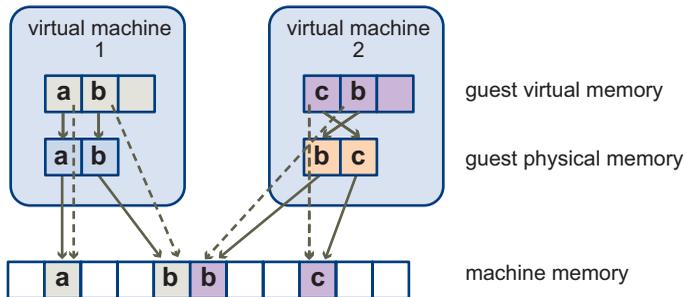


Figure 9-2. ESX Server Memory Mapping

The diagram above illustrates the ESX Server implementation of memory virtualization.

- The boxes shown in the figure 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 in [Figure 9-2](#) above.)
- The arrows from guest physical memory to machine memory show the mapping maintained by the VMM.
- The dashed arrows in [Figure 9-2](#) 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 Server can efficiently manage memory across all virtual machines. Some of the physical memory of a virtual machine might in fact be mapped to shared pages or to pages that are unmapped, or swapped out.

An ESX Server 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 Implications

The use of two page-coordinated 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 could 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.

Understanding Memory Overhead

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

ESX Server memory virtualization adds little time overhead to memory accesses. Because the processor's paging hardware uses the shadow page tables directly, most memory accesses in the virtual machine can execute without address translation overhead.

For example, if a page fault occurs in the virtual machine, control switches to the VMM so that the VMM can update its data structures.

The memory space overhead has two components:

- A fixed system-wide overhead for the service console and the VMkernel.
- Additional overhead for each virtual machine.

For ESX Server 3.0, 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. See [“Viewing Host Resource Information”](#) on page 13 for information on how to determine the available memory for a host.

Overhead memory includes space reserved for the virtual machine frame buffer and various virtualization data structures. Overhead memory depends on the number of virtual CPUs, the configured memory for the guest operating system, and on whether you are using a 32-bit or 64-bit guest operating system. The following table lists the overhead for each case.

Table 9-2. Overhead Memory on Virtual Machines

Virtual CPUs	Memory (MB)	Overhead for 32-bit virtual machine (MB)	Overhead for 64-bit virtual machine (MB)
1	256	79	174
1	512	79	176
1	1024	84	180
1	2048	91	188
1	4096	107	204
1	8192	139	236
1	16384	203	300
2	256	97	288
2	512	101	292
2	1024	101	300
2	2048	125	316
2	4096	157	349
2	8192	221	413
2	16384	349	541
4	256	129	511
4	512	133	515
4	1024	141	523
4	2048	157	540
4	4096	189	572
4	8192	222	605
4	16384	350	734

ESX Server also provides optimizations such as memory sharing (see [“Sharing Memory Across Virtual Machines”](#) on page 135) 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.

Memory Allocation and Idle Memory Tax

This section discusses in some detail how an ESX Server host allocates memory, and how you can use the **Mem.IdleTax** configuration parameter to change how an ESX Server host reclaims idle memory.

How ESX Server Hosts Allocate Memory

An ESX Server host allocates the memory specified by the **Limit** parameter to each virtual machine unless memory is overcommitted. An ESX Server host never allocates more memory to a virtual machine than its specified physical memory size. For example, a 1GB virtual machine might have the default limit (unlimited) or a user-specified limit (for example 2GB). In both cases, the ESX Server host never allocates more than 1GB, the physical memory size that was specified for it.

When memory is overcommitted, each virtual machine is allocated an amount of memory somewhere between what's specified by **Reservation** and what's specified by **Limit** (see [“Memory Overcommitment”](#) on page 41). The amount of memory granted to a virtual machine above its reservation usually varies with the current memory load.

An ESX Server 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 Server hosts use a modified proportional-share memory allocation policy. Memory shares entitle a virtual machine to a fraction of available physical memory. See [“Shares”](#) on page 20.
- **Working set size** — ESX Server hosts estimate the working set for a virtual machine by monitoring memory activity over successive periods of virtual machine execution time. Estimates are smoothed over several time periods using techniques that respond rapidly to increases in working set size and more slowly to decreases in working set size.

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

You can modify the default monitoring period of 60 seconds by adjusting the **Mem.SamplePeriod** advanced setting. **Mem.SamplePeriod** specifies the periodic time interval, measured in seconds of virtual machine execution time, over which memory activity is monitored to estimate working set sizes. See [“Setting Advanced Host Attributes”](#) on page 136.

How Host Memory Is Used

You can use the VI Client to see how host memory is used.

To view information about physical memory usage

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

The following information appears, as discussed in [Table 9-3](#).

Memory		Properties...
Physical		
Total	8.00 GB	
System	774.75 MB	
Virtual Machines	6.98 GB	
Service Console	272.00 MB	

Table 9-3. Host Memory Information

Field	Description
Total	Total physical memory for this host.
System	<p>Memory used by the ESX Server system.</p> <p>ESX Server 3.0 uses at least 50MB of system memory for the VMkernel, plus additional memory for device drivers. This memory is allocated automatically when the ESX Server is loaded and is not configurable.</p> <p>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.</p> <p>The ESX Server host also attempts to keep some memory free at all times to handle dynamic allocation requests efficiently. ESX Server sets this level at approximately six percent of the memory available for running virtual machines.</p>
Virtual Machines	<p>Memory used by virtual machines running on the selected host.</p> <p>Most of the host's memory is used for running virtual machines. An ESX Server host manages the allocation of this memory to virtual machines based on administrative parameters and system load.</p>
Service Console	<p>Memory reserved for the service console.</p> <p>Click Properties to change how much memory is available for the service console.</p>

Memory Tax for Idle Virtual Machines

If a virtual machine is not actively using its currently allocated memory, the ESX Server charges more for idle memory than for memory that's in use. (ESX Server never alters user-specified share allocations, but memory tax has a similar effect.)

Memory tax helps prevent virtual machines from hoarding idle memory. The default tax rate is 75 percent, that is, an idle page costs as much as four active pages.

The **Mem.IdleTax** advanced setting allows you to explicitly control the policy for reclaiming idle memory. You can use this option, together with the **Mem.SamplePeriod** advanced attribute, to control how the system reclaims memory. See [“Setting Advanced Host Attributes”](#) on page 136.

NOTE In most cases, changes to **Mem.IdleTax** are not necessary or even appropriate.

How ESX Server Hosts Reclaim Memory

This section gives background information on how ESX Server hosts reclaim memory from virtual machines. The hosts employ two distinct techniques for dynamically expanding or contracting the amount of memory allocated to virtual machines:

- ESX Server systems use a memory balloon driver (`vmmemctl`), loaded into the guest operating system running in a virtual machine. See [“Memory Balloon \(`vmmemctl`\) Driver.”](#)
- ESX Server 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. See [“Swapping”](#) on page 134.

Memory Balloon (`vmmemctl`) Driver

The `vmmemctl` driver 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 which closely matches the behavior of a native system under similar memory constraints. This technique effectively increases or decreases memory pressure on the guest operating system, causing the guest to invoke its own native memory management algorithms. When memory is tight, the guest operating system decides which particular pages to reclaim and, if necessary, swaps them to its own virtual disk.

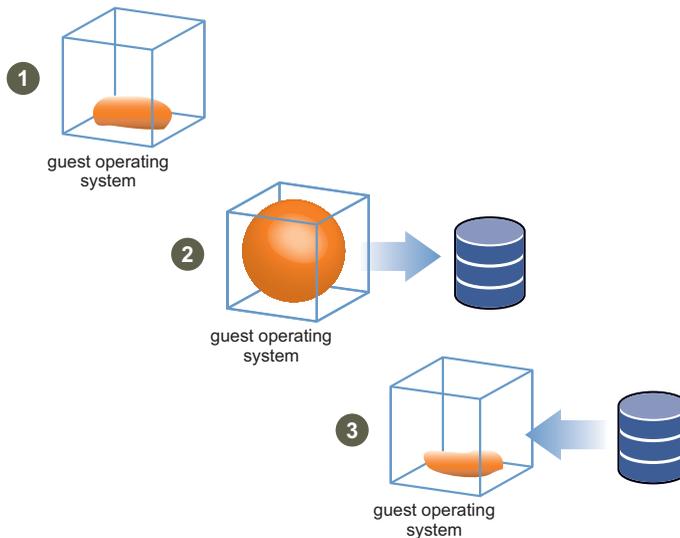


Figure 9-3. Memory Ballooning

NOTE You must configure the guest operating system with sufficient swap space. Some guest operating systems have additional limitations. See [“Swap Space and Guest Operating Systems”](#) on page 133.

If necessary, you can limit the amount of memory `vmxmemctl` 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 [“Setting Advanced Virtual Machine Attributes”](#) on page 140.

Swap Space and Guest Operating Systems

If you choose to overcommit memory with ESX Server, you need to be sure your guest operating systems have sufficient swap space. This swap space must be greater than or equal to the difference between the virtual machine’s **configured memory size** and its **Reservation**.



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 automatically try to increase the size of paging files, if there is sufficient free disk space.

For more information, refer to 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, refer to 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.

Swapping

To power on a virtual machine, a swap file is created. If this file cannot be created, the virtual machine cannot power on. By default, the swap file is created in the same location as the `.vmdk` and `-flat.vmdk` files, but you can use per-virtual machine configuration options to change this location. For more information on these options, see [“Setting Advanced Virtual Machine Attributes”](#) on page 140.

ESX Server hosts use swapping to forcibly reclaim memory from a virtual machine when no `vmmemctl` driver is available because the `vmmemctl` driver:

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

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

NOTE For optimum performance, ESX Server hosts use the ballooning approach (implemented by the `vmmemctl` driver) whenever possible. Swapping is a reliable mechanism of last resort that a host uses only when necessary to reclaim memory.

Swap Space and Memory Overcommitment

Swap space must be reserved on disk for any unreserved virtual machine memory, which is the difference between the reservation and the configured memory size. This swap reservation is required to ensure the system is able to preserve virtual machine memory under any circumstances. In practice, only a small fraction of the swap space may actually be used.

Similarly, while memory reservations are used for admission control, actual memory allocations vary dynamically, and unused reservations are not wasted.

Swap Files and ESX Server Failure

If an ESX Server system fails, and that system had running virtual machines that were using swap files, those swap files continue to exist and take up disk space even after the ESX Server system restarts.

To delete swap files

- 1 Start the virtual machine again.
- 2 Stop the virtual machine explicitly

NOTE These swap files can consume many gigabytes of disk space so it's important that you delete them properly that way.

Sharing Memory Across Virtual Machines

Many ESX Server 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, an ESX Server 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.

You can use the **Mem.ShareScanVM** and **Mem.ShareScanTotal** advanced settings to control the rate at which the system scans memory to identify opportunities for sharing memory. See [“Setting Advanced Host Attributes”](#) on page 136.

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 [“Setting Advanced Virtual Machine Attributes”](#) on page 140.

ESX Server 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 `esxtop` to observe the actual savings. You can find the information in the PSHARE field of the interactive mode in the Memory page. See [“Using the esxtop Utility in Interactive Mode”](#) on page 160.

Advanced Attributes and What They Do

This section lists the advanced attributes available for customizing memory management.



CAUTION Using these advanced attributes is appropriate only under special circumstances. In most cases, changing the basic settings (**Reservation**, **Limit**, **Shares**) or using the default settings results in an appropriate allocation.

Setting Advanced Host Attributes

(SEE UPDATE) This section guides you through setting advanced attributes for a host, and then lists a few attributes you might want to set under certain circumstances.

To set advanced attributes for a host

- 1 In the VI Client's inventory panel, select the virtual machine you want to customize.
- 2 Choose **Edit Settings** in the Commands panel and select the **Options** tab.
- 3 Select **Advanced**, and click the **Configuration Parameters** button.
- 4 Click **Advanced Settings**.

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

Cpu.VMAdmitCheckPerVcpuMin	<input type="text" value="1"/>
Perform additional admission control check that per VCPU Virtual Machine minimum does not e...	
Min:	0
Max:	1
Cpu.VMotionMinAllocPct	<input type="text" value="30"/>
Per Virtual Machine minimum CPU allocations (in %) for VMotion requirements	
Min:	0
Max:	200
Cpu.IntraCoreMigrate	<input type="text" value="0"/>
When to allow intra-core migrations [0:when inter-core migration allowed, 1:always]	
Min:	0
Max:	1
Cpu.IdlePackageRebalancePeriod	<input type="text" value="541"/>
Usec between chances to rebalance idle packages (0 to disable)	
Min:	0
Max:	100000

The following tables list the advanced resource management attributes discussed in this document.



CAUTION Setting these attributes is recommended only for advanced users with considerable experience using ESX Server hosts. In most cases, the default settings produce the optimum result.

Table 9-4. Advanced CPU Attributes

Attribute	Description
CPU.MachineClearThreshold	If you are using a host enabled for hyperthreading and set this attribute to 0, quarantining is disabled. See “Quarantining” on page 125.

Table 9-5. Advanced Memory Attributes

Attribute	Description	Default
Mem.CtlMaxPercent	Limits the maximum amount of memory that can be reclaimed from any virtual machine using <code>vmmemctl</code> , based on a percentage of its configured memory size. Specifying <code>0</code> disables reclamation via <code>vmmemctl</code> for all virtual machines. See “Memory Balloon (vmmemctl) Driver” on page 132.	65
Mem.ShareScanTotal	Specifies the total system-wide rate at which memory should be scanned for transparent page sharing opportunities. The rate is specified as the number of pages to scan per second. Defaults to 200 pages/sec.	200
Mem.ShareScanVM	Controls the rate at which the system scans memory to identify opportunities for sharing memory. Units are pages per second.	50
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. See “Memory Allocation and Idle Memory Tax” on page 130.	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. See “How ESX Server Hosts Allocate Memory” on page 130.	60
Mem.BalancePeriod	Specifies the periodic time interval, in seconds, for automatic memory reallocations. Reallocations are also triggered by significant changes in the amount of free memory.	15

Table 9-6. 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 this option is set 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 this option is set to 0 , the system does not automatically set memory affinity for virtual machines with CPU affinity set. See “VMware NUMA Optimization Algorithms” on page 150.	1
Numa.MigImbalanceThreshold	The NUMA rebalancer computes the CPU imbalance between nodes, taking into account 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. However, 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 you want to disable 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 NUMA rebalancing is enabled.	2

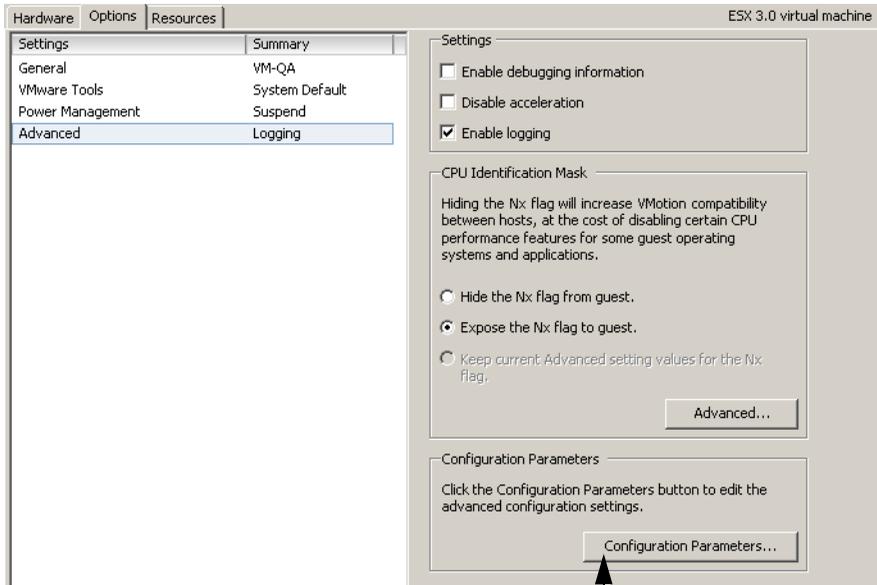
See [Appendix A, “Using NUMA Systems with ESX Server,”](#) on page 147 for additional information.

Setting Advanced Virtual Machine Attributes

This section first steps you through setting advanced attributes for a virtual machine, and then lists a small number of attributes you might want to set under certain circumstances.

To set advanced attributes for a virtual machine

- 1 Select the virtual machine in the VI Client’s inventory panel, and choose **Edit Settings** from the right-button menu.
- 2 Click **Options** and click **Advanced**.
- 3 Click the **Configuration Parameters** button.



Configuration Parameters

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

You can set the following advanced attributes for virtual machines:

Table 9-7. Advanced Virtual Machine Attributes

Attribute	Description
sched.mem.maxmemctl	Maximum amount of memory that can be reclaimed from the selected virtual machine by ballooning, in megabytes (MB). If the ESX Server host needs to reclaim additional memory, it is forced to swap. Swapping is less desirable than ballooning. See “Memory Balloon (vmmemctl) Driver” on page 132.
sched.mem.pshare.enable	Enables memory sharing for a selected virtual machine. See “Sharing Memory Across Virtual Machines” on page 135. This boolean value defaults to True . If you set it to False for a virtual machine, memory sharing is turned off.
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 where the virtual machine’s swap file is located. Defaults to the virtual machine’s working directory, that is, the VMFS directory that contains its configuration file.
sched.swap.file	Filename for the virtual machine’s swap file. By default, the system generates a unique name when it creates the swap file.
das.isolationaddress (SEE UPDATE)	Sets the address to ping to determine if a host is isolated from the network. If this option is not specified, the default gateway of the console network is used. This default gateway has to be some reliable address that is known to be available, so that the host can determine if it is isolated from the network.
das.defaultfailoverhost (SEE UPDATE)	If this is set, VMware HA will first try to fail over hosts to the host specified by this option. This is useful if you want to utilize one host as a spare failover host, but is not usually recommended, because VMware HA tries to utilize all available spare capacity among all hosts in the cluster. If the specified host does not have enough spare capacity, VMware HA tries to fail over the virtual machine to any other host in the cluster that has enough capacity.

Best Practices

This chapter discusses some best practices for users of ESX Server 3 and VirtualCenter 2. It discusses the following topics:

- [“Resource Management Best Practices”](#) on page 143
- [“Creating and Deploying Virtual Machines”](#) on page 144

Resource Management Best Practices

The following guidelines can help you achieve optimal performance for your virtual machines:

- If you expect frequent changes to the total available resources, use **Shares**, not **Reservation**, to allocate resources fairly across virtual machines. If you use **Shares**, and you upgrade the 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 or CPU.
- Use **Reservation** to specify the minimum acceptable amount of CPU or memory, not the amount you would like to have available. The host assigns additional resources as available based on the number of shares and the limit for your virtual machine.
- Use **Reservations** to specify the minimum reservation for each 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.
- Don't set **Reservation** too high. A reservation that's too high can limit the number of virtual machines in a resource pool.

- When specifying the reservations for virtual machines, always leave some room: Do not commit all resources. 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.
- Use resource pools for delegated resource management. To fully isolate a resource pool, make the resource pool type **Fixed** and use **Reservation** and **Limit**.
- Group virtual machines for a multi-tier service in a resource pool. Resource pools allow the ESX Server host to assign resources for the service as a whole.
- Enable PortFast on PortFast-capable switches to ensure communication between HA clusters and other cluster nodes.

Creating and Deploying Virtual Machines

This section gives some best practices information for planning and creating virtual machines.

Planning

Before you deploy a virtual machine, you need to:

- Plan your load mix.
- Understand goals and expectations.
- Understand the requirements, and what it means to be successful.
- Avoid mixing virtual machines that have competing resource requirements.
- Test before you deploy if you have specific performance expectations,.

Virtualization allows a number of virtual machines to share the host's resources. It does not create new resources. Virtualization can result in overheads.

Creating Virtual Machines

When you create virtual machines, be sure to size them according to your actual needs, just like physical machines. Overconfigured virtual machines waste shareable resources.

To optimize performance, disable unused virtual devices such as COM ports, LPT ports, floppy drives, CD-ROMs, USB adapters, and so on. Those devices are

periodically polled by the guest operating system even if they are not in use. This unproductive polling wastes shareable resources.

Install VMware Tools. VMware Tools helps you achieve higher performance, can result in more efficient CPU utilization, and includes disk, network, and memory reclamation drivers.

Deploying the Guest Operating System

Tune and size the virtual machine operating system just as you tune the operating system of a physical machine with registry, swap space, and so on. Disable unnecessary programs and services such as screen savers. Unnecessary programs and services waste shareable resources.

Keep the guest operating system up-to-date with the latest patches. If you are using Microsoft Windows as the guest operating system, check for any known operating system issues in Microsoft knowledge base articles.

Deploying Guest Applications

Tune and size applications on your virtual machines just like you tune and size applications on your physical machine.

Don't run single-threaded applications in an SMP virtual machine. Single-threaded workloads cannot take advantage of additional virtual CPUs, and unused virtual CPUs waste shareable resources. However, a workload consisting of several single-threaded applications running concurrently might be able to take advantage of additional virtual CPUs.

Configuring VMkernel Memory

VMkernel reclaims memory by ballooning and swapping. See [Chapter 9, "Advanced Resource Management,"](#) on page 117 for additional information. To use memory resources optimally, avoid high reclamation activity by correctly sizing virtual machines and by avoiding high memory overcommitment. See ["Memory Overcommitment"](#) on page 41.

VMkernel implements a NUMA scheduler, which supports both IBM and AMD NUMA architectures. The scheduler locates virtual machine memory and virtual CPUs on the same NUMA node. This prevents possible performance degradation due to remote memory accesses. The host hardware should be configured so that physical host memory is evenly balanced across NUMA nodes. See [Appendix A, "Using NUMA Systems with ESX Server,"](#) on page 147.

Using NUMA Systems with ESX Server



ESX Server supports memory access optimization for both Intel and AMD Opteron processors in server architectures that support NUMA (non-uniform memory access). This appendix gives background information on NUMA technologies and describes optimizations available with ESX Server. It discusses the following topics:

- [“Introduction”](#) on page 147
- [“ESX Server NUMA Scheduling”](#) on page 149
- [“VMware NUMA Optimization Algorithms”](#) on page 150
- [“Manual NUMA Controls”](#) on page 152
- [“IBM Enterprise X-Architecture Overview”](#) on page 153
- [“AMD Opteron-Based Systems Overview”](#) on page 153
- [“Retrieving NUMA Configuration Information”](#) on page 154
- [“CPU Affinity for Associating Virtual Machines with a Single NUMA Node”](#) on page 155
- [“Memory Affinity for Associating Memory Allocations with a NUMA Node”](#) on page 156

Introduction

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. The following systems offer a NUMA platform to support industry-standard operating systems, including Windows and Linux:

- IBM Enterprise X-Architecture, available in the IBM eServer x440, eServer x445, and eServer x460
- Fujitsu-Siemens Primergy T850
- HP ProLiant DL 385 and ProLiant DL 585
- Sun Microsystems Sun Fire V202 and V402

What Is NUMA?

For the past decade, processor clock speed has increased dramatically. A multi-gigahertz CPU, however, needs to be supplied with an 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.

NUMA is an alternative approach that links several small, cost-effective nodes via a high-performance connection. Each node contains both 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. Thus, memory access times are non-uniform depending on the location of the memory, as the technology's name implies.

This problem is amplified on symmetric multiprocessing (SMP) systems, where many processors must compete for bandwidth on the same system bus. High-end RISC UNIX 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 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.

However, 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 may 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 may be desirable for certain applications that demand 100 percent of their memory to be local, it creates an administrative burden and can lead to imbalance between various 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, if the system is intended to support uptimes of months or years.

ESX Server NUMA Scheduling

ESX Server uses a sophisticated NUMA scheduler to dynamically balance processor load and memory locality or processor load balance, as follows:

- 1 Each virtual machine managed by the NUMA scheduler is assigned a home node — one of the system's NUMA nodes containing both processors and local memory, as indicated by the System Resource Allocation Table (SRAT).
- 2 When memory is allocated to a virtual machine, the ESX Server 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 may 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 may migrate the virtual machine's memory dynamically to its new home node to improve memory locality. The NUMA scheduler may also swap virtual machines between nodes when this improves overall memory locality.

Some virtual machines are not managed by the ESX Server 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 have more virtual processors than the number of physical processor cores available on a single hardware node cannot be managed automatically. Virtual machines that are not

managed by the NUMA scheduler still run correctly. However, they don't benefit from ESX Server's NUMA optimizations.

The NUMA scheduling and memory placement policies in VMware ESX Server can manage all virtual machines transparently, so that administrators do not need to deal with the complexity of balancing virtual machines between nodes explicitly.

See also [“CPU Affinity for Associating Virtual Machines with a Single NUMA Node”](#) on page 155 and [“Memory Affinity for Associating Memory Allocations with a NUMA Node”](#) on page 156.

The optimizations work seamlessly regardless of the type of guest operating system. ESX Server 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.

VMware NUMA Optimization Algorithms

This section describes the algorithms used by VMware ESX Server to maximize application performance while still maintaining resource guarantees.

Home Nodes and Initial Placement

When a virtual machine is powered on, ESX Server 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. Thus, 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 2003 Server, provide this level of NUMA support, which is known as initial placement. It may 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 with an uptime measured in months or years.

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

To overcome these weaknesses, ESX Server 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 may 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 Server 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 (4 KB 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 Server host immediately begins to migrate its memory in this fashion, usually at a rate of approximately 100 KB (25 pages) per second. It manages this 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 Server 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 Server workloads present opportunities for sharing memory across virtual machines. For example, several virtual machines may be running instances of the same guest operating system, have the same applications or components loaded, or contain common data. In such cases, ESX Server 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 Server 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.

Manual 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 can be 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 Server host's automatic NUMA optimizations, as described in [“VMware NUMA Optimization Algorithms”](#) on page 150, result in good performance.

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

The VI Client allows you to specify that:

- A virtual machine should use only the processors on a given node (through the **CPU Affinity** option). See “[CPU Affinity for Associating Virtual Machines with a Single NUMA Node](#)” on page 155.
- The server should allocate memory only on the desired node (through the **Memory Affinity** option). See “[Memory Affinity for Associating Memory Allocations with a NUMA Node](#)” on page 156.

If both options are set before a virtual machine starts, the virtual machine runs only on the desired 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 should also be set manually, so that memory from the virtual machine’s previous node can be moved to its new node.

Manual NUMA placement may interfere with the ESX Server resource management algorithms, which attempt 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. You must consider these issues when placing NUMA manually.

IBM Enterprise X-Architecture Overview

The IBM Enterprise X-Architecture, which first appeared on the IBM eServer x440 and Fujitsu Siemens Primergy T850 in 2002, supports servers with up to four nodes (also called CECs or SMP Expansion Complexes in IBM terminology). Each node may 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. See the IBM Redbook *IBM eServer xSeries 440 Planning and Installation Guide* for a more detailed description of the Enterprise X Architecture. Go to www.redbooks.ibm.com, and search from there.

AMD Opteron-Based Systems Overview

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. For more information, see the HP ProLiant DL585 Server Technology technology brief. See also the *HP ROM-Based Setup Utility User Guide* at docs.hp.com/en/347569-003/347569-003.pdf.

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 Server 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 Server host does not detect the system as NUMA.

Currently shipping Opteron processors have either one or two 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. Thus, the single-core Opteron systems have a single processor per NUMA node and the dual-core Opteron systems have two processors per 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 Server 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 Server NUMA optimizations. Uniprocessor virtual machines (with a single virtual processor) can reside within a single NUMA node and are managed automatically by the ESX Server NUMA scheduler.

NOTE For small Opteron systems, NUMA rebalancing is now disabled by default to ensure scheduling fairness. You can use the **Numa.RebalanceCoresTotal** and **Numa.RebalanceCoresNode** options to change this behavior. See “[Setting Advanced Virtual Machine Attributes](#)” on page 140.

Retrieving NUMA Configuration Information

This section describes how to obtain statistics about your NUMA system.

Obtaining NUMA Statistics

To obtain NUMA statistics, issue the following command:

```
cat /proc/vmware/NUMA/hardware
```

Information about the following attributes is displayed:

- **Node** — Node number.
- **ID** — Hardware ID number of the NUMA node.
- **MachineMem** — Amount of physical memory located on this NUMA node.

- **ManagedMem** — Amount of physical memory located on this NUMA node, excluding memory used by the ESX Server virtualization layer.
- **CPUs** — A space-separated list of the physical processors in this node.
- **Total memory** — Amount of memory physically installed on each NUMA node. Not all of this memory can be managed by the VMkernel, because some of the memory is used by the service console.

Determining the Amount of Memory for Each NUMA Node

To determine the amount of memory for each node, type the following into the service console:

```
cat /proc/vmware/mem
```

Determining the Amount of Memory for a Virtual Machine

To determine the amount of memory for a virtual machine on a NUMA node, type the following into the service console:

```
cat /proc/vmware/vm/<id>/mem/numa
```

Here's an example of what you might see:

Node#	Pages/MB
0	13250/51
1	0/0

The preceding output indicates that the virtual machine, with the specified ID, occupies 51MB of memory on node 0, and no memory on node 1.

NOTE In the preceding example, the memory affinity is set so that only pages associated with node 0 are allocated for this virtual machine. If memory affinity had not been set, then typically the output would have shown a more even distribution of memory between nodes 0 and 1. For more information, see [“Memory Affinity for Associating Memory Allocations with a NUMA Node”](#) on page 156.

CPU Affinity for Associating Virtual Machines with a Single NUMA Node

In some cases, you may 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). See [“Retrieving NUMA Configuration Information”](#) on page 154 for information on obtaining these CPU numbers.

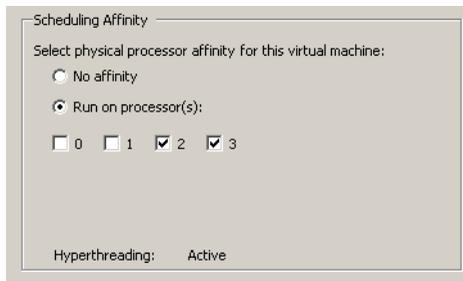


CAUTION There are a number of potential issues if you use CPU affinity. See [“Potential Issues with Affinity”](#) on page 121.

To set CPU affinity for a single NUMA node

- 1 Using a VI Client, right-click a virtual machine and choose **Edit Settings**.
- 2 In the Virtual Machine Properties dialog box, select the **Resources** tab and choose **CPU**.
- 3 In the Scheduling Affinity panel, you can 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.



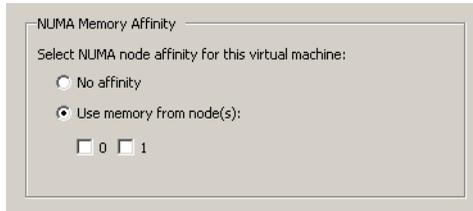
Memory Affinity for Associating Memory Allocations with a NUMA Node

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. (See [“Obtaining NUMA Statistics”](#) on page 154 to determine the NUMA node number.)

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.

To associate memory allocations with a NUMA node

- 1 Using a VI Client, right-click a virtual machine and choose **Edit Settings**.
- 2 In the Virtual Machine Properties dialog box, select the **Resources** tab, and choose **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 four CPUs to a single NUMA node for a virtual machine on an eight-way server. You want this virtual machine to run only on node 1.

The CPUs — for example, 4, 5, 6, and 7 — are the physical CPU numbers.

To bind a two-way virtual machine to use only the last four physical CPUs of an eight-processor machine

- 1 In the VI Client's inventory panel, select the virtual machine and choose **Edit Settings**. Select **Options**, click **Advanced**, and click the **Configuration Parameters** button.
- 2 In the VI Client, turn on CPU affinity for processors 4, 5, and 6. See [“To set CPU affinity for a single NUMA node”](#) on page 156.

To set the virtual machine's memory to specify that all of the virtual machine's memory should be allocated on node 1

- 1 In the VI Client's inventory panel, select the virtual machine and choose **Edit Settings**.
- 2 Select **Options**, click **Advanced**, and click the **Configuration Parameters** button.
- 3 In the VI Client, set memory affinity for the NUMA node to 1. See [“To associate memory allocations with a NUMA node”](#) on page 157.

Completing these two tasks ensures that the virtual machine runs only on NUMA node 1 and, when possible, allocates memory from the same node.

B

Using the esxtop Utility

This appendix explains how to use the `esxtop` performance monitoring tool. It discusses the following topics:

- [“Using the esxtop Utility for Performance Monitoring”](#) on page 159
- [“Using the esxtop Utility in Interactive Mode”](#) on page 160
- [“Using the esxtop Utility in Batch Mode”](#) on page 174
- [“Using the esxtop Utility in Replay Mode”](#) on page 175

Using the esxtop Utility for Performance Monitoring

The `esxtop` command-line tool provides a fine-grained look at how ESX Server uses resources in real time. The tool runs on the ESX Server host’s service console.

To invoke `esxtop`

- 1 Make sure you have root user privileges.
- 2 Type the command, using the desired options:

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

You can invoke `esxtop` in interactive (default), batch, or replay mode. The following sections discuss each mode and give references to available commands and display statistics:

- [“Using the esxtop Utility in Interactive Mode”](#) on page 160
- [“Using the esxtop Utility in Batch Mode”](#) on page 174
- [“Using the esxtop Utility in Replay Mode”](#) on page 175

Configuration File

The `esxtop` utility reads its default configuration from `~/ .esxtop3rc`. This configuration file consists of five lines.

The first four lines contain lower- and upper-case letters to specify which fields appear in which order on the CPU, memory, storage, and network panel. The letters correspond to the letters in the Fields or Order panels for the respective `esxtop` panel.

The fifth line contains information on the other options. Most important, if you have saved a configuration in secure mode, you do not get an insecure `esxtop` without removing the `s` from the fifth line of your `~/ .esxtop3rc` file. A number specifies the delay time between updates. As in interactive mode, typing `c`, `m`, `d`, or `n` determines the panel with which `esxtop` starts.

NOTE Editing this file is not recommended. 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 `esxtop` Utility in Interactive Mode

By default, `esxtop` runs in interactive mode. Interactive mode displays statistics in different panels, discussed below:

- [“CPU Panel”](#) on page 163
- [“Memory Panel”](#) on page 166
- [“Storage Panel”](#) on page 170
- [“Network Panel”](#) on page 173

A help menu is available for each panel.

Interactive Mode Command-Line Options

The following command-line options are available in interactive mode.

Table B-1. Interactive Mode Command-Line Options

Option	Description
<code>h</code>	Prints help for <code>esxtop</code> command-line options.
<code>v</code>	Prints <code>esxtop</code> version number.
<code>s</code>	Invokes <code>esxtop</code> in secure mode. In secure mode, the <code>-d</code> command, which specifies delay between updates, is disabled.

Table B-1. Interactive Mode Command-Line Options

Option	Description
d	Specifies the delay between updates. The default is five seconds. The minimum is two seconds. You can 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.

Common Statistics Description

Several statistics appear on the different panels while **esxtop** is running in interactive mode. The following statistics are common across all four panels.

The **Uptime** line, found at the top of each of the four **esxtop** panels, displays the current time, time since last reboot, number of currently running worlds and load averages. A world is an ESX Server 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 all the physical CPUs are fully utilized. A load average of 2.00 means that the ESX Server system may 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 Server system are half utilized.

Interactive Mode Single-Key Commands

When running in interactive mode, **esxtop** recognizes several single-key commands. Commands listed in this section are recognized in all four panels. The command to specify the delay between updates is disabled if the **s** option has been given on the command line (see [“Interactive Mode Command-Line Options”](#) on page 160). All sorting interactive commands sort in descending order.

Table B-2. esxtop 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. See “Statistics Columns and Order Pages” on page 162 for more information.

Table B-2. esxtop Interactive Mode Single-Key Commands (Continued)

Key	Description
o or O	Displays a panel for changing the order of statistics columns on the current panel. See “Statistics Columns and Order Pages” on page 162 for more information.
#	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 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	Writes the current setup to <code>~/ .esxtop3rc</code> . This is the recommended way to make changes to an esxtop configuration file. See “Configuration File” on page 160.
q	Quits interactive mode.
c	Switches to the CPU resource utilization panel.
m	Switches to the memory resource utilization panel.
d	Switches to the storage (disk) resource utilization panel.
n	Switches to the network resource utilization panel.

Statistics Columns and Order Pages

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 upper case, the field is displayed. An asterisk in front of the field description indicates if 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 upper case letter
- Move a field to the right by pressing the corresponding lower case letter

The illustration below shows a field order change.

```

root@danaki01:~#
Current Field order: ABCDEFGH

* A: ID = Id
* B: GID = Group Id
* C: NAME = Name
* D: NMEM = Num Members
* E: %STATE TIMES = CPU State Times
  F: EVENT COUNTS/s = CPU Event Counts
  G: CPU ALLOC = CPU Allocations
  H: SUMMARY STATS = CPU Summary Stats

Use a-h to change order.
Uppercase moves a field left, lowercase moves a field right.
Use any other key to return: █

```

Figure B-1. Field Order Change

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

```

root@danaki01:~#
7:21:04pm up 40 min, 39 worlds: CPU load average: 0,08, 0,09, 0,08
PCPU(%): 31,47, 41,81, 2,09, 2,86, 2,09, 2,27, 6,02, 5,14 ; used total: 11,72
CCPU(%): 0 us, 1 sy, 98 id, 1 wa ; cs/sec: 54

```

ID	GID	NAME	NMEM	%USED	%SYS	%ORLP	%RUN	%WAIT	%BWAIT	%TWAIT	%GRUN	%CSTP	%IDLE	%DOW	%EXTG	%LIMIT
1	1	idle	3	706,36	0,00	20,47	593,36	0,00	0,00	0,00	0,00	0,00	0,00	200,03	0,00	0,00
2	2	system	6	0,03	0,00	0,00	0,03	599,96	0,00	599,96	0,00	0,00	0,00	0,00	0,00	0,00
6	6	console	1	1,25	0,01	0,01	1,43	62,05	36,45	98,50	0,00	0,00	0,00	98,50	0,07	0,00
7	7	helper	11	0,00	0,00	0,00	0,00	1099,99	0,00	1099,99	0,00	0,00	0,00	0,00	0,00	0,00
8	8	drivers	7	0,01	0,00	0,00	0,01	700,00	0,00	700,00	0,00	0,00	0,00	0,00	0,00	0,00
12	12	vmware-vmkauthd	1	0,00	0,00	0,00	0,00	100,00	0,00	100,00	0,00	0,00	0,00	0,00	0,00	0,00
13	13	RHEL4AS	5	89,21	17,45	0,08	71,78	337,39	90,32	427,72	0,00	0,00	0,00	26,66	0,50	0,00

Figure B-2. CPU Panel

You can change the display using single-key commands. Statistics and single-key commands are discussed in the following tables.

- “CPU Panel Statistics” on page 164
- “CPU Panel Single-Key Commands” on page 166

Table B-3. CPU Panel Statistics

Line	Description
PCPU(%)	Percentage of CPU utilization per physical CPU and total average physical CPU utilization.
LCPU(%)	Percentage of CPU utilization per logical CPU. The percentages for the logical CPUs belonging to a package add up to 100 percent. This line appears only if hyper-threading is present and enabled. See “Hyperthreading and ESX Server” on page 123 for information on hyperthreading.
CCPU(%)	Percentages of total CPU time as reported by the ESX Server service console. <ul style="list-style-type: none"> ■ us — percentage user time. ■ sy — percentage system time. ■ id — percentage idle time. ■ wa — percentage wait time. ■ cs/sec — the context switches per second recorded by the service console.
ID	Resource pool ID or virtual machine ID of the running world’s resource pool or virtual machine, or world ID of running world.
GID	Resource pool ID of the running world’s resource pool or virtual machine.
NAME	Name of running world’s resource pool or virtual machine, or name of running world.
NMEM	Number of members in running world’s resource pool or virtual machine. If a Group is expanded using the interactive command e (see interactive commands below) then NMEM 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.
%USED	Percentage physical CPU used by the resource pool, virtual machine, or world.
%SYS	Percentage of time spent in the ESX Server 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, above.
%TWAIT	Percentage of time the resource pool, virtual machine, or world spent in the wait state. This percentage includes the percentage of time the resource pool, virtual machine, or world was idle.
%IDLE	Percentage of time the resource pool, virtual machine, or world was idle. Subtract this percentage from %TWAIT, above to see the percentage of time the resource pool, virtual machine, or world was waiting for some event.
%RDY	Percentage of time the resource pool, virtual machine, or world was ready to run.

Table B-3. CPU Panel Statistics (Continued)

Line	Description
%MLMTD	Percentage of time the ESX Server VMKernel deliberately didn't run the resource pool, virtual machine, or world because doing so would violate the resource pool, virtual machine, or world's limit setting. Even though the resource pool, virtual machine, or world is ready to run when it is prevented from running in this way, the %MLMTD time is not included in %RDY 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 . See "Creating and Customizing Resource Pools" on page 24.
AMAX	Resource pool, virtual machine, or world attribute Limit . A value of -1 means unlimited. See "Creating and Customizing Resource Pools" on page 24.
ASHRS	Resource pool, virtual machine, or world attribute Shares . See "Creating and Customizing Resource Pools" on page 24.
SUMMARY STATS	Set of CPU statistics made up of the following CPU configuration parameters and statistics. These statistics are applicable only to worlds and not to virtual machines or resource pools.
AFFINITY BIT MASK	Bit mask showing the current scheduling affinity for the world. See "Using CPU Affinity to Assign Virtual Machines to Specific Processors" on page 120.
HTSHARING	Current hyperthreading configuration. See "Advanced Server Configuration for Hyperthreading" on page 123.
CPU	The physical or logical processor on which the world was running when esxtop obtained this information.
HTQ	Indicates whether the world is currently quarantined or not. N means no and Y means yes. See "Quarantining" on page 125.
TIMER/s	Timer rate for this world.

Table B-3. CPU Panel Statistics (Continued)

Line	Description
%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 Server VMKernel, then the time spent doing so 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.

Table B-4. 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 example: <ul style="list-style-type: none"> ■ If the %Used by a resource pool is 30% on a 2-way server, then the resource pool is utilizing 30 percent of 2 physical CPUs. ■ If the %Used by a world belonging to a resource pool is 30 percent on a two-way server, then that world is utilizing 30% of 1 physical CPU.
U	Sort resource pools, virtual machines, and worlds by the resource pool's or virtual machine's %Used column.
R	Sort resource pools, virtual machines, and worlds by the resource pool's or virtual machine's %RDY column.
N	Sort resource pools, virtual machines, and worlds by the GID column. This is the default sort order.

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. For distinctions between machine memory and physical memory see [“Virtual Memory in Virtual Machines”](#) on page 125.

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 overcommit of 100 percent. See “[Memory Overcommitment](#)” on page 41.

```

root@danakil01:~
7:25:13pm up 44 min, 39 worlds; MEM overcommit average: 0,00, 0,00, 0,00
PMEM (MB): 16215 total: 272 cos, 202 vmk, 278 other, 15463 free
VMKMEM (MB): 15650 managed: 939 minfree, 1016 rsvd, 14547 ursvd, high state
COSMEM (MB): 72 free: 2047 swap_t, 2047 swap_f, 0,00 r/s, 0,00 w/s
NUMA (MB): 3615 ( 3265), 4096 ( 4090), 4096 ( 4090), 4021 ( 4015)
PSHARE (MB): 20 shared, 8 common; 12 saving
SWAP (MB): 0 curr, 0 target; 0,00 r/s, 0,00 w/s
MEMCTL (MB): 0 curr, 0 target, 0 max

  ID  GID  NAME          NMEM  MEMSZ  SZTGT  TCHD  %ACTV  %ACTVS  %ACTVF  %ACTVN  OVHDUN  OVHD  OVHDMAX
  ---  ---  ---          ---   ---   ---   ---   ---   ---   ---   ---   ---   ---
  12  12  vmware-vmkauthd  1    2,20  2,20  0,34  0     0     0     0     0,00  0,00  0,00
  13  13  RHEL4AS        5   256,00  317,87  17,92  5     2     4     7    16,91  28,95  72,98

```

Figure B-3. esxtop Memory Panel

Table B-5. esxtop Memory Panel

Field	Description
PMEM (MB)	<p>Displays the machine memory statistics for the server. All numbers are in megabytes.</p> <ul style="list-style-type: none"> ■ total — Total amount of machine memory in the server. ■ cos — Amount of machine memory allocated to the ESX Server service console. ■ vmk — Amount of machine memory being used by the ESX Server VMKernel. ■ other — Amount of machine memory being used by everything other than the ESX service console and ESX Server VMKernel. ■ free — Amount of machine memory that is free.
VMKMEM (MB)	<p>Displays the machine memory statistics for the ESX Server VMKernel. All numbers are in megabytes.</p> <ul style="list-style-type: none"> ■ managed — Total amount of machine memory managed by the ESX Server VMKernel. ■ min free — Minimum amount of machine memory that the ESX Server 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.

Table B-5. esxtop Memory Panel (Continued)

Field	Description
COSMEM (MB)	<p>Displays the memory statistics as reported by the ESX Server service console. All numbers are in megabytes.</p> <ul style="list-style-type: none"> ■ 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)	<p>Displays the ESX Server NUMA statistics. This line appears only if the ESX Server host is running on a NUMA server. All numbers are in megabytes. For each NUMA node in the server, two statistics are displayed:</p> <ul style="list-style-type: none"> ■ The total amount of machine memory in the NUMA node that is managed by the ESX Server. ■ The amount of machine memory in the node that is currently free (in parentheses).
PSHARE (MB)	<p>Displays the ESX Server page-sharing statistics. All numbers are in megabytes.</p> <ul style="list-style-type: none"> ■ 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 due to page sharing.
SWAP (MB)	<p>Displays the ESX Server swap usage statistics. All numbers are in megabytes.</p> <ul style="list-style-type: none"> ■ curr — Current swap usage ■ target — Where the ESX Server system expects the swap usage to be. ■ r/s — Rate at which memory is swapped in by the ESX Server system from disk. ■ w/s — Rate at which memory is swapped to disk by the ESX Server system. <p>See “Swapping” on page 134 for background information.</p>
MEMCTL (MB)	<p>Displays the memory balloon statistics. All numbers are in megabytes.</p> <ul style="list-style-type: none"> ■ curr — Total amount of physical memory reclaimed using the <code>vmmemctl</code> module. ■ target — Total amount of physical memory the ESX Server host attempts to reclaim using the <code>vmmemctl</code> module. ■ max — Maximum amount of physical memory the ESX Server host can reclaim using the <code>vmmemctl</code> module. <p>See “Memory Balloon (vmmemctl) Driver” on page 132 for background information on memory ballooning.</p>
AMIN	<p>Memory reservation for this resource pool or virtual machine. See “Reservation” on page 21.</p>

Table B-5. esxtop Memory Panel (Continued)

Field	Description
AMAX	Memory limit for this resource pool or virtual machine. A value of -1 means Unlimited. See “Limit” on page 21.
ASHRS	Memory shares for this resource pool or virtual machine. See “Shares” on page 20.
NHN	Current home node for the resource pool or virtual machine. This statistic is only applicable on NUMA systems. If the virtual machine has no home node, a dash (-) is displayed. See ESX Server NUMA scheduling on page 154.
NRMEM (MB)	Current amount of remote memory allocated to the virtual machine or resource pool. This statistic is applicable only on NUMA systems. See “VMware NUMA Optimization Algorithms” on page 150.
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.
SZTGT (MB)	Amount of machine memory the ESX Server VMKernel wants to allocate to a resource pool or virtual machine.
TCHD (MB)	Working set estimate for the resource pool or virtual machine. See “Memory Allocation and Idle Memory Tax” on page 130.
%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.
MCTL?	Memory balloon driver is installed or not. N means no, Y means yes. See “Memory Balloon (vmmemctl) Driver” on page 132.
MCTLSZ (MB)	Amount of physical memory reclaimed from the resource pool by way of ballooning. See “Memory Balloon (vmmemctl) Driver” on page 132.
MCTLTGT (MB)	Amount of physical memory the ESX Server system can reclaim from the resource pool or virtual machine by way of ballooning. See “Memory Balloon (vmmemctl) Driver” on page 132.
MCTLMAX (MB)	Maximum amount of physical memory the ESX Server 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 Server host expects the swap usage by the resource pool or virtual machine to be.

Table B-5. esxtop Memory Panel (Continued)

Field	Description
SWR/s (MB)	Rate at which the ESX Server host swaps in memory from disk for the resource pool or virtual machine.
SWW/s (MB)	Rate at which the ESX Server 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 due to resource pool or virtual machine shared pages.
OVHD (MB)	Current space overhead for resource pool. See “Understanding Memory Overhead” on page 128.
OVHDMAX (MB)	Maximum space overhead that may be incurred by resource pool or virtual machine. See “Understanding Memory Overhead” on page 128.

Memory Panel Interactive Commands

Table B-6. Memory Panel Interactive Commands

Command	Description
M	Sort resource pools or virtual machines by Group Mapped column.
B	Sort resource pools or virtual machines by Group Memctl column.
N	Sort resource pools or virtual machines by GID column. This is the default sort order.

Storage Panel

This panel displays server-wide storage utilization statistics. Statistics are aggregated per storage adapter by default. Statistics can also be viewed per storage channel, target, LUN, or world using a LUN.

```

root@danakil01:~
7:26:32pm up 45 min, 39 worlds; CPU load average: 0,11, 0,09, 0,09
ADAPTR CID TID LID WID NCHNS NTGTS NLUNS NVMS AQLEN LQLEN WQLEN ACTV QUED %USB LOAD CHDS/s READS/s WRITES/s MBREAD/s MBWRIT/s
vmhba0 - - - 1 1 1 2 128 0 0 0 0 0 0,00 0,00 0,00 0,00 0,00 0,00
vmhba1 0 0 0 - 1 1 1 1 2038 30 0 0 0 0 0,00 0,00 0,00 0,00 0,00 0,00
vmhba1 0 0 6 - 1 1 1 1 2038 30 0 0 0 0 0,00 0,00 0,00 0,00 0,00 0,00
vmhba1 0 0 7 - 1 1 1 1 2038 30 0 0 0 0 0,00 0,00 0,00 0,00 0,00 0,00
vmhba1 0 0 8 - 1 1 1 1 2038 30 0 0 0 0 0,00 0,00 0,00 0,00 0,00 0,00
vmhba1 0 0 9 - 1 1 1 1 2038 30 0 0 0 0 0,00 0,00 0,00 0,00 0,00 0,00
vmhba1 0 0 12 - 1 1 1 1 2038 30 0 0 0 0 0,00 0,00 0,00 0,00 0,00 0,00

```

Figure B-4. esxtop Storage Panel

Storage Panel Interactive Commands

Table B-7. Storage Panel Statistics

Column	Description
ADAPTR	Name of the storage adapter.
CID	Storage adapter channel ID. This ID is visible only if the corresponding adapter is expanded. See the interactive command e below.
TID	Storage adapter channel target ID. This ID is visible only if the corresponding adapter and channel are expanded. See the interactive commands e and a below.
LID	Storage adapter channel target LUN ID. This ID is visible only if the corresponding adapter, channel and target are expanded. See the interactive commands e , a , and t below.
WID	Storage adapter channel target LUN world ID. This ID is visible only if the corresponding adapter, channel, target and LUN are expanded. See interactive commands e , a , t , and l below.
NCHNS	Number of channels.
NTGTS	Number of targets.
NLUNS	Number of LUNs.
NVMS	Number of worlds.
SHARES	Number of shares.
BLKS	Block size in bytes. This statistic is applicable only to LUNs.
AQLEN	Storage adapter queue depth. Maximum number of ESX Server VMKernel active commands that the adapter driver is configured to support.
LQLEN	LUN queue depth. Maximum number of ESX Server VMKernel active commands that the LUN is allowed to have.
WQLEN	World queue depth. Maximum number of ESX Server VMKernel active commands that the world is allowed to have. This is a per LUN maximum for the world.

Table B-7. Storage Panel Statistics (Continued)

Column	Description
%USD	Percentage of queue depth (adapter, LUN or world) used by ESX Server VMKernel active commands.
LOAD	Ratio of ESX Server VMKernel active commands plus ESX Server VMKernel queued commands to queue depth (adapter, LUN or world).
ACTV	Number of commands in the ESX Server VMKernel that are currently active.
QUED	Number of commands in the ESX Server VMKernel that are currently queued.
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.
DAVG/cmd	Average device latency per command, in milliseconds.
KAVG/cmd	Average ESX Server VMKernel latency per command, in milliseconds.
GAVG/cmd	Average virtual machine operating system latency per command, in milliseconds.
ABRTS/s	Number of commands aborted per second.
RESETS/s	Number of commands reset per second.

Table B-8. Storage Panel Interactive Commands

Command	Description
e	Toggles whether storage adapter statistics are displayed expanded or unexpanded. Allows viewing storage resource utilization statistics broken down by individual channels belonging to an expanded storage adapter. You are prompted for the adapter name.
a	Toggles whether storage channel statistics are displayed expanded or unexpanded. Allows viewing storage resource utilization statistics broken down by individual targets belonging to an expanded storage channel. You are prompted for the adapter name first and then the channel ID. The channel adapter needs to be expanded before the channel itself can be expanded.
t	Toggles whether storage target statistics are displayed in expanded or unexpanded mode. Allows viewing storage resource utilization statistics broken down by individual LUNs belonging to an expanded storage target. You are first prompted for the adapter name, then for the channel ID and finally for the target ID. The target channel and adapter need to be expanded before the target itself can be expanded.

Table B-8. Storage Panel Interactive Commands

Command	Description
l	Toggles whether LUN are displayed in expanded or unexpanded mode. Allows viewing storage resource utilization statistics broken down by individual worlds utilizing an expanded storage LUN. You are prompted for the adapter name first, then the channel ID, then the target ID, and finally the LUN ID. The LUN target, channel, and adapter must be expanded before the LUN itself can be expanded.
r	Sorts by Reads column.
w	Sorts by Writes column.
R	Sorts by MB read column.
T	Sorts by MB written column.
N	Sorts first by ADAPTR column, then by CID column within each ADAPTR , then by TID column within each CID , then by LID column within each TID , and finally by WID column within each LID . This is the default sort order.

Network Panel

This panel displays server-wide network utilization statistics. Statistics are arranged per port per virtual network device configured. For physical network adapter statistics, see the row corresponding 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.

```

root@danakil01:~
7:26:49pm up 45 min, 39 worlds; CPU load average: 0,11, 0,09, 0,09
PORT ID UPLINK USED BY DTYP INAME PKT/s Mb/s PKTR/s MB/s %DEPTX %DEPRX
3 N 0:NCP H pps 0,00 0,00 0,00 0,00 0,00 0,00
16777217 Y 0:NCP S vSwitch0 4,16 0,01 2,97 0,05 0,00 0,00
16777218 N 0:NCP S vSwitch0 0,00 0,00 0,00 0,00 0,00 0,00
16777219 N 0:vswifi0 S vSwitch0 4,16 0,01 0,59 0,00 0,00 0,00
33554433 Y 0:NCP S vSwitch1 10270,79 5,02 31526,82 341,39 0,00 0,00
33554434 N 0:NCP S vSwitch1 0,00 0,00 0,00 0,00 0,00 0,00
33554435 N 1059:RHEL4RS S vSwitch1 10270,59 5,01 31526,23 342,35 100,00 0,00
50331649 Y 0:NCP S vSwitch2 0,00 0,00 0,59 0,00 0,00 0,00
50331650 N 0:NCP S vSwitch2 0,00 0,00 0,00 0,00 0,00 0,00

```

Figure B-5. esxtop Network Panel

Network Panel Statistics

Table B-9. Network Panel Statistics

Column	Description
PORT	Virtual network device port ID.
UPLINK	Y means the corresponding port is an uplink. N means it is not.

Table B-9. Network Panel Statistics

Column	Description
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	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.

Network Panel Interactive Commands

Table B-10. 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.

Using the esxtop Utility in Batch Mode

Batch mode allows you to collect and save resource utilization statistics in a file. If you want to run in batch mode, you must first prepare for batch mode.

To prepare for running esxtop in batch mode

- 1 Run `esxtop` in interactive mode.
- 2 In each of the four panels, select the columns in which you are interested.
- 3 Save this configuration in the `~/ .esxtop3rc` file using the `W` interactive command.

To run esxtop in batch mode

- 1 Invoke `esxtop` to redirect the output to a file. For example:

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

The file name needs to have a `.csv` extension. The `esxtop` utility does not enforce this, but the post-processing tools require it.

- 2 You can process statistics collected in batch mode using tools such as Microsoft Excel and Perfmon.

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

Batch Mode Command-Line Options

The following command-line options are available in batch mode.

Table B-11. Command-line Options in Batch Mode

Option	Description
<code>b</code>	Runs <code>esxtop</code> in batch mode.
<code>d</code>	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.
<code>n</code>	Number of iterations. <code>esxtop</code> collects and saves statistics this number of times, then exits.

Using the esxtop Utility in Replay Mode

In replay mode, `esxtop` replays resource utilization statistics collected using `vm-support`. See the `vm-support` man page for more information.

If you want to run in replay mode, you must first prepare for replay mode.

To prepare for running esxtop in replay mode

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

To run `esxtop` in replay mode

- Enter the following at the command-line prompt:

```
esxtop -R <vm-support_dir_path>
```

Additional command-line options are listed under “[Replay Mode Command-Line Options](#)” below.

You don’t have to run `esxtop` replay mode on the ESX Server service console.

Replay mode can be run to produce output in the same style as batch mode (see the command-line option `b`, below, for more information).

In replay mode, `esxtop` accepts the same set of interactive commands as in interactive mode. In replay mode, `esxtop` runs until there are no more snapshots collected by `vm-support` to be read or until `esxtop` completes the requested number of iterations (see the command-line option `n` below for more details).

Replay Mode Command-Line Options

A number of command-line options are available for `esxtop` replay mode.

Table B-12. Command-Line Options in Replay Mode

Option	Description
R	Path to the <code>vm-support</code> collected snapshot’s directory.
b	Runs <code>esxtop</code> in Batch mode.
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. <code>esxtop</code> updates the display this number of times and then exits.

Index

A

- admission control **22**
 - HA **88**
 - resource pools **47**
- advanced attributes **136**
 - hosts **136**
 - virtual machines **140**
- advanced resource management **117**
- advanced virtual machine attributes **140**
- affinity
 - CPU **120**
 - DRS **101**
 - potential issues **121**
- algorithms, NUMA **150**
- AMD Opteron-based systems **153**
- Any hyperthreading mode **124**
- applications
 - CPU-bound **120**
 - deploying **145**
 - single-threaded **120**
- architecture **36**
- automation level **87**
- automation modes, virtual machines **114**
- available memory **16**

B

- ballooning **132**
- best practices **143**

C

- caveats **125**

- cluster creation overview **86**
- cluster features, choosing **87**
- cluster resource pools **46**
- clusters
 - adding hosts **31, 112**
 - adding managed hosts **96, 106**
 - adding unmanaged hosts **97, 106**
 - adding virtual machines **111, 112**
 - and resource pools **57**
 - and virtual machines **111**
 - and VirtualCenter failure **64**
 - creating **29, 30, 83, 87, 88**
 - customizing **29, 30**
 - DRS **58**
 - DRS, adding hosts **96**
 - HA **84**
 - HA, adding hosts **106**
 - information **89**
 - introduction **61**
 - invalid **99**
 - powering on virtual machines **112**
 - prerequisites **83**
 - processor compatibility **85**
 - removing hosts **114**
 - removing virtual machines **113**
 - shared storage **84**
 - shared VMFS volume **84**
 - summary page **89**
- CPU
 - managing allocation **38**

- overcommitment **19**
 - virtual machines **19**
 - CPU affinity **120, 121**
 - and hyperthreading **125**
 - NUMA **156**
 - NUMA nodes **155**
 - potential issues **121**
 - CPU panel, esxtop **163**
 - CPU Reservation **17**
 - CPU Reservation Used **17**
 - CPU Unreserved **17**
 - CPU virtualization **39, 118**
 - direct execution **118**
 - virtualization modes **118**
 - CPU.Machine.ClearThreshold **137**
 - CPU.MachineClearThreshold **125**
 - CPU-bound applications **120**
 - custom automation mode **114**
- D**
- delegation of control through resource pools **45**
 - device drivers **37**
 - direct execution mode **118**
 - disabled virtual machine **114**
 - disk resources **35**
 - DNS **84**
 - short name **84**
 - DRS **95**
 - adding managed hosts **96**
 - adding unmanaged hosts **97**
 - automation level **87**
 - custom automation mode **114**
 - customizing virtual machines **114**
 - disabled virtual machine **114**
 - fully automatic **88**
 - host removal and virtual machines **98**
 - initial placement **62, 66**
 - introduction **65, 95**
 - maintenance mode **69**
 - manual **88**
 - migration history **93**
 - migration recommendations **67**
 - overview **62**
 - partially automatic **88**
 - reconfiguring **100**
 - red clusters **80**
 - rules **103**
 - turning off **100**
 - using together with HA **76**
 - virtual machine migration **66**
 - VMotion network **83**
 - DRS affinity rules **101**
 - DRS clusters **68**
 - adding hosts **96**
 - DRS load balancing **62**
 - DRS migration recommendations **99**
 - DRS Resource Distribution histograms **91**
 - DRS rules **102**
 - dual-processor virtual machine **19**
 - dynamic load balancing, NUMA **151**
- E**
- emulation **118**
 - Entering Maintenance Mode **98**
 - ESX Server
 - architecture **36**
 - memory allocation **130**
 - memory reclamation **132**
 - resource management **35**
 - esxtop
 - batch mode **174**
 - batch mode command-line

- options **175**
 - command-line options **159**
 - common statistics description **161**
 - configuration file **160**
 - CPU panel **163**
 - CPU panel single-key commands **166**
 - CPU panel statistics **164**
 - interactive mode **160**
 - interactive mode command-line options **160**
 - interactive mode single-key commands **161**
 - introduction **159**
 - invoking **159**
 - memory panel **166, 170**
 - network panel **173**
 - network panel interactive commands **174**
 - network panel statistics **173**
 - order pages **162**
 - performance monitoring **159**
 - replay mode **175**
 - replay mode command-line options **176**
 - statistics column **162**
 - storage panel **170**
 - storage panel interactive commands **171**
 - storage panel statistics **171**
 - examples
 - expandable reservations **50**
 - memory overhead **128**
 - NUMA **157**
 - red cluster **80**
 - reservation **21**
 - resource pools **25**
 - shares **21**
 - valid cluster **77, 78**
 - valid cluster using resource pools of type expandable **78**
 - valid cluster, all resource pools of type fixed **77**
 - yellow cluster **79**
 - expandable reservations **28, 50**
 - example **50**
- F**
- failover capacity **71**
 - fully automatic DRS **88**
- G**
- Grafted from **97**
- H**
- HA **105**
 - adding managed hosts **106**
 - adding unmanaged hosts **106**
 - admission control **88**
 - and host power off **73**
 - and traditional cluster solutions **69**
 - customizing virtual machines **115**
 - DNS connectivity **84**
 - failover capacity **71**
 - host network isolation **73**
 - introduction **69, 105**
 - migration with VMotion **73**
 - options **88**
 - red clusters **81**
 - redundant network paths **84**
 - shared storage **84**
 - turning off **109**
 - using together with DRS **76**
 - HA clusters
 - adding hosts **106**

- planning **72**
- high (shares) **20**
- high availability options **88**
- histograms, DRS Resource Distribution **91**
- home nodes, NUMA **150**
- host network isolation **73**
- host resource pools **46**
- hosts
 - adding to cluster **112**
 - adding to DRS clusters **96, 97**
 - adding to HA clusters **106**
 - entering maintenance mode **98**
 - losing resource pool hierarchy **97**
 - memory use **130**
 - removing and invalid clusters **99**
 - removing and resource pool hierarchies **98**
 - removing from clusters **114**
 - resource information **13**
 - Under Maintenance **98**
- hyperthreading **122, 123**
 - and CPU affinity **125**
 - CPU.MachineClearThreshold **125**
 - disabling **39**
 - disabling quarantining **137**
 - performance implications **122**
 - quarantining **125**
- hyperthreading modes
 - Any **124**
 - internal **124**
 - None **124**

I

- idle memory tax **130, 131**
- initial placement **62, 66**
 - NUMA **150**

- internal hyperthreading mode **124**
- invalid clusters, host removal **99**
- iSCSI storage
 - VMware HA **115**
- isolation response **115**
 - default **107**
- isolation through resource pools **45**

K

- knowledge base
 - accessing **12**

L

- limit
 - hyperthreading **124**
 - pros and cons **22**
 - resource pools **24**
- limit attribute **21**
- load balancing **62**
- logical processors **39**
- low (shares) **20**

M

- maintenance mode **68, 69, 98**
- manual DRS **88**
- Mem.BalancePeriod **138**
- Mem.CtlMaxPercent **138**
- Mem.IdleTax **132, 138**
- Mem.SamplePeriod **130, 138**
- Mem.ShareScanTotal **135, 138**
- Mem.ShareScanVM **135, 138**
- memory
 - available **16**
 - managing allocation **38**
 - NUMA **155**
 - overhead **41**
 - reclaiming unused **132**
 - service console **16**

- sharing across virtual machines **135**
 - virtual machines **18**
 - virtualization basics **40**
 - VMkernel memory **145**
 - memory affinity
 - NUMA **156, 157**
 - memory balloon driver **132**
 - memory idle tax **130, 131**
 - Mem.IdleTax **138**
 - memory mapping **126**
 - memory overcommitment **41, 135**
 - memory overhead **128**
 - examples **128**
 - Memory Reservation **17**
 - Memory Reservation Used **17**
 - memory sharing **41**
 - Memory Unreserved **17**
 - memory virtualization **39**
 - migration history **93**
 - migration page **91**
 - migration recommendations **67, 100**
 - applying **99**
 - migration threshold **67**
 - migration with VMotion, failure, and HA **73**
- N**
- NAS storage
 - VMware HA **115**
 - network resources **35**
 - None (hyperthreading mode) **124**
 - normal (shares) **20**
 - NUMA
 - AMD Opteron-based systems **153**
 - CPU affinity **155, 156**
 - CPU assignment **157**
 - determining memory for NUMA node **155**
 - determining memory for virtual machine **155**
 - dynamic load balancing **151**
 - example **157**
 - home nodes and initial placement **150**
 - introduction **147, 148**
 - manual controls **152**
 - memory affinity **156, 157**
 - optimization algorithms **150**
 - page migration **151**
 - statistics **154**
 - transparent page sharing **152**
 - using with ESX server **147**
 - NUMA configuration information **154**
 - NUMA scheduling **149**
 - Numa.AutoMemAffinity **139**
 - Numa.MigImbalanceThreshold **139**
 - Numa.PageMigEnable **139**
 - Numa.RebalanceCoresNode **139**
 - Numa.RebalanceCoresTotal **139**
 - Numa.RebalanceEnable **139**
 - Numa.RebalancePeriod **139**
- O**
- Opteron **153**
 - overcommitted cluster **79**
 - overcommitment **41, 135**
 - overhead **128**
 - examples **128**
 - overhead memory **41**
- P**
- page migration, NUMA **151**
 - partially automatic DRS **88**
 - performance **36**
 - CPU-bound applications **120**
 - monitoring **55**

- performance monitoring, esxstop **159**
- physical and logical processors **39**
- physical memory usage **131**
- physical processors **39**
- processors
 - logical **39**
 - physical **39**
- processor-specific behavior **119**

Q

- quarantining, hyperthreading **125**

R

- red clusters **80**
- red DRS cluster **80**
- red HA cluster **81**
- redundant network paths for HA **84**
- reservation **16**
 - example **21**
 - hyperthreading **124**
 - resource pools **24**
- reservation attribute **21**
- reservation type **24**
- resource management
 - best practices **143**
 - concepts **33**
- resource pool attributes, changing **55**
- resource pool hierarchies, host removal **98**
- resource pools **46**
 - adding virtual machines **56**
 - admission control **47**
 - and clusters **57**
 - creating **24, 25, 48**
 - customizing **24, 25**
 - delegation of control **45**
 - DRS clusters **68**
 - example **25**

- information **51**
- introduction **43, 44**
- isolation **45**
- performance **55**
- removing virtual machines **57**
- reservation type **24**
- resource allocation tab **52**
- root resource pool **44**
 - siblings **44**
 - summary tab **51**
- resources, reserving **20**
- restart priority **115**
 - default **107**
- root resource pool **44**
- rules **102**
 - deleting **103**
 - disabling **103**
 - DRS **103**
 - editing **102**
 - results **103**

S

- SAN and HA **84**
- sched.mem.maxmemctl **133, 141**
- sched.mem.pshare.enable **141**
- sched.swap.dir **141**
- sched.swap.file **141**
- sched.swap.persist **141**
- server configuration for
 - hyperthreading **123**
- service console **37**
 - memory use **16**
- shares **20**
 - example **21**
 - high **20**
 - high (number) **20**
 - low **20**

- low (number) **20**
 - normal **20**
 - normal (number) **20**
 - ratio **20**
 - resource pools **24**
 - sharing memory **41**
 - siblings **44**
 - single-processor virtual machine **19**
 - single-threaded applications **120**
 - SMP virtual machines **120**
 - stars, migration threshold **67**
 - statistics, esxtop **161**
 - swap space **133, 135**
 - Linux systems **134**
 - Windows systems **134**
 - swapping **134**
- T**
- threshold **67**
 - Total Migrations display **100**
 - traditional clustering solutions **69**
- U**
- Under Maintenance **98**
 - user groups
 - accessing **12**
- V**
- valid clusters **76**
 - example **77, 78**
 - Virtual Infrastructure SDK **38**
 - virtual machine attributes
 - changing **22**
 - shares, reservation, and limit **20**
 - Virtual Machine File System **37**
 - virtual machine migration **66**
 - virtual machine monitor **126**
 - virtual machines
 - adding during cluster creation **111**
 - adding to cluster **111, 112**
 - adding to resource pools **56**
 - assigning to a specific processor **121**
 - changing resource allocation **22**
 - configuration file **85**
 - CPU **19**
 - creating (best practice) **144**
 - customizing for DRS **114**
 - customizing for HA **115**
 - deploying (best practice) **144**
 - deploying applications **145**
 - deploying operating system **145**
 - disabled (DRS) **114**
 - dual-processor **19**
 - host removal **98**
 - memory **18, 40**
 - memory overhead **128**
 - number of virtual processors **120**
 - removing from cluster **113**
 - removing from resource pools **57**
 - resource allocation **18**
 - single-processor **19**
 - virtual memory **125**
 - virtual memory in virtual machines **125**
 - virtual processors per virtual machine **120**
 - virtual to physical memory mapping **126**
 - virtualization modes **118**
 - virtualization overhead **118**
 - VMFS **37**
 - VMkernel **36**
 - VMkernel hardware interface layer **37**
 - VMkernel memory **145**
 - VMkernel resource manager **37**
 - VMM **37, 126**

- vmmemctl **132**
 - Mem.CtlMaxPercent **138**
 - sched.mem.maxmemctl **141**
- VMotion requirements **84**
- VMware community forums
 - accessing **12**
- VMware DRS See DRS **95**
- VMware HA **108**
 - iSCSI storage **115**
 - NAS storage **115**
- VMware HA See HA **105**

W

- working set size **130**

Y

- yellow cluster **79**

Updates for the Resource Management Guide

ESX Server 3.0.1 and VirtualCenter 2.0.1

Last Updated: January 16, 2009

This document provides you with updates to the ESX Server 3.0.1 and VirtualCenter 2.0.1 version of the *Resource Management Guide*. Updated descriptions, procedures, and graphics are organized by page number so you can easily locate the areas of the guide that have changes. If the change spans multiple sequential pages, this document provides the starting page number only.

The following is a list of *Resource Management Guide* page updates in this document:

- [“Updates for Table 1-1 on Page 17”](#)
- [“Updates for the Virtualization and Processor-Specific Behavior Section on Page 119”](#)
- [“Updates for the Setting Advanced Host Attributes Section on Page 136”](#)
- [“Updates for Table 9-7 on Page 141”](#)

Updates for Table 1-1 on [Page 17](#)

More detailed information was added to the “Memory Reservation” and “Memory Reservation Used” rows of [Table 1-1](#).

Table 1-1. Host Resource Properties

Field	Description
CPU Reservation	Total CPU resources available for this host.
CPU Reservation Used	Total CPU resources of this host that are reserved by running virtual machines. Note: Virtual machines that aren't powered on don't consume CPU resources. For powered-on virtual machines, the system reserves CPU resources according to each virtual machine's Reservation setting.

Table 1-1. Host Resource Properties (Continued)

Field	Description
CPU Reservation Unused	<p>Total CPU resources of this host that are not currently reserved. Consider a virtual machine with reservation=2GHz that is totally idle. It has 2GHz reserved, but it is not actually using any of its reservation.</p> <p>Other virtual machines <i>cannot reserve</i> these 2GHz.</p> <p>Other virtual machines <i>can use</i> these 2GHz, that is, idle CPU reservations are not wasted.</p>
Memory Reservation	<p>Total memory resources available for this host.</p> <p>If a virtual machine has a memory reservation but has not yet accessed its full reservation, the unused memory can be reallocated to other virtual machines.</p>
Memory Reservation Used	<p>Total memory resources of this host that are reserved by a running virtual machine and virtualization overhead.</p> <p>Note: Virtual machines that aren't powered on don't consume memory resources. For powered-on virtual machines, the system reserves memory resources according to each virtual machine's Reservation setting and overhead.</p> <p>Once a virtual machine has accessed its full reservation, the vmkernel allows the virtual machine to retain this much memory, and will not reclaim it, even if the virtual machine becomes idle and stops accessing memory.</p>
Memory Reservation Unused	<p>Total memory resources of this host that are not currently reserved.</p>

Updates for the Virtualization and Processor-Specific Behavior Section on [Page 119](#)

The [Virtualization and Processor-Specific Behavior](#) section states: "Because of the different kernel versions, it is not possible to migrate virtual machines installed on a system running one processor model (for example, AMD) to a system running on a different processor (for example, Intel)." This sentence should be modified to say "Because of the different kernel versions, it is not possible to use VMotion to migrate virtual machines installed on a system running one processor model (for example, AMD) to a system running on a different processor (for example, Intel)."

Updates for the Setting Advanced Host Attributes Section on [Page 136](#)

The procedure [To set advanced attributes for a host](#) is incorrect and should be replaced by the following procedure:

To set advanced attributes for a host

- 1 In the VI Client's 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**), scroll in the right panel to find and change the attribute.

Updates for Table 9-7 on [Page 141](#)

The `das.isolationaddress` and `das.defaultfailoverhost` advanced attributes listed in [Table 9-7](#) must be set using the following cluster-level procedure.

To set advanced options for VMware HA

- 1 In the cluster's Settings dialog box, select **VMware HA**.
- 2 Click the **Advanced Options** button to open the Advanced Options (HA) dialog box.
- 3 Enter each advanced attribute you want to change in the Option column and enter the value that the attribute should be set to in the Value column.
- 4 Click **OK**.

