IBM eserver

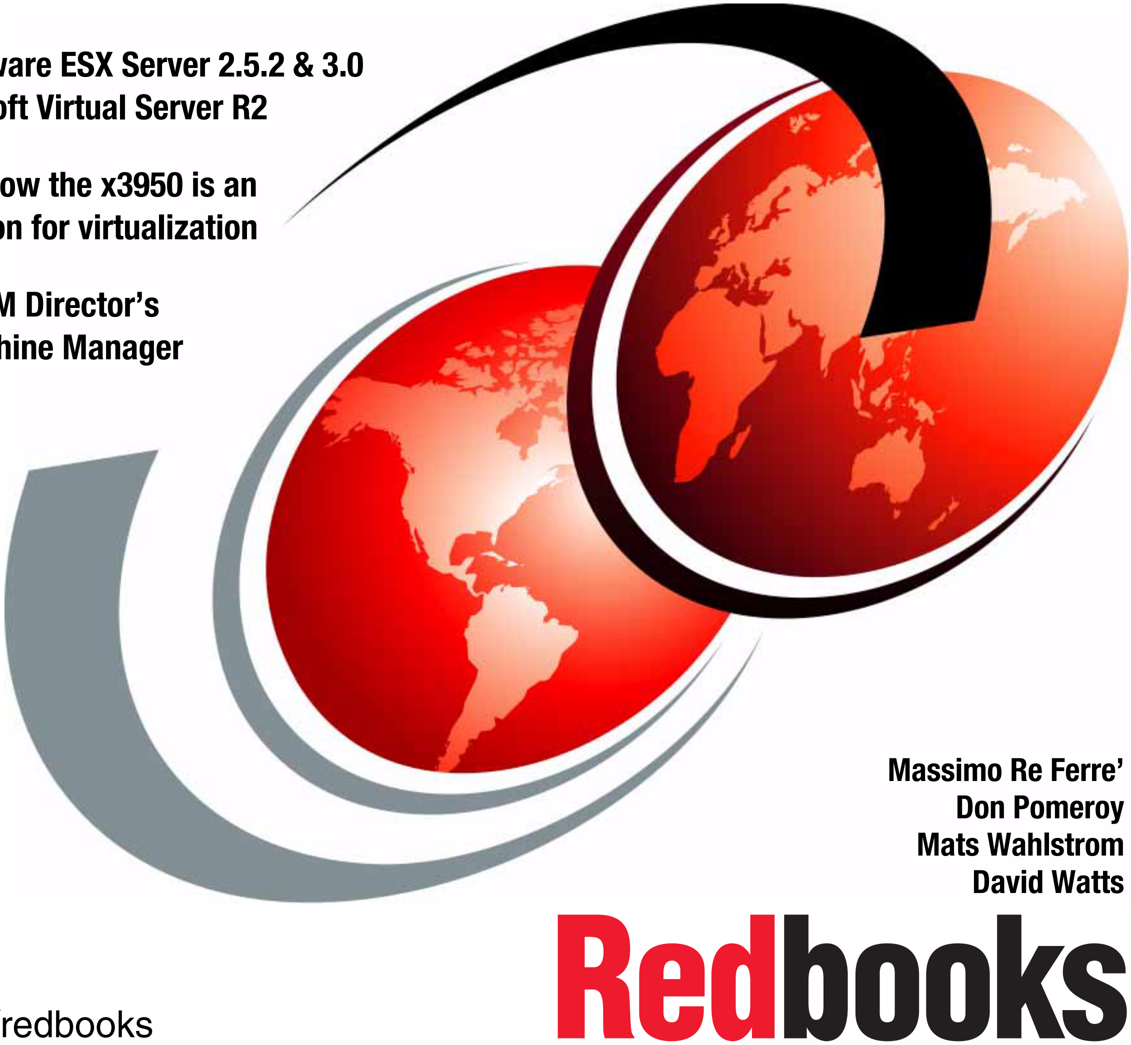
IBM

Virtualization on the IBM System x3950 Server

Covers VMware ESX Server 2.5.2 & 3.0 and Microsoft Virtual Server R2

Describes how the x3950 is an ideal solution for virtualization

Includes IBM Director's Virtual Machine Manager



Massimo Re Ferre'
Don Pomeroy
Mats Wahlstrom
David Watts

Redbooks

ibm.com/redbooks

International Technical Support Organization

Virtualization on the IBM System x3950 Server

June 2006

SG24-7190-00

Note: Before using this information and the product it supports, read the information in “Notices” on page vii.

First Edition (June 2006)

This edition applies to the IBM System x3950 and the IBM @server xSeries 460, VMware ESX Server 2.5.2 and 3.0, Microsoft Virtual Server R2, IBM Director 5.10, and IBM Virtual Machine Manager 2.1.

Note: This book is based on a pre-GA version of a product and may not apply when the product becomes generally available. We recommend that you consult the product documentation or follow-on versions of this redbook for more current information.

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

- eserver®
- eserver®
- Redbooks (logo) ™
- iSeries™
- i5/OS®
- xSeries®
- AIX 5L™
- AIX®
- BladeCenter®
- Chipkill™
- DB2 Universal Database™
- DB2®
- IBM®
- Predictive Failure Analysis®
- POWER™
- Redbooks™
- RETAIN®
- ServerProven®
- ServeRAID™
- System p5™
- System x™
- Tivoli Enterprise™
- Tivoli Enterprise Console®
- Tivoli®
- TotalStorage®
- X-Architecture™

The following terms are trademarks of other companies:

IPX, Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Visual Basic, Windows server, Windows NT, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Itanium, Pentium, Xeon, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Preface

Virtualization is becoming more and more a key technology enabler to streamline and better operate data centers. In its simplest form, *virtualization* refers to the capability of being able to run multiple operating system instances, such as Linux® and Microsoft Windows®, on a physical server.

Usually the concept of virtualization is usually associated with high-end servers, such as the IBM® System x3950 (and it's predecessor the IBM @server® xSeries® 460) that are able being able to support and consolidate multiple heterogeneous software environments. The System x3950 is a highly scalable x86 platform capable of supporting up to 32 processors and 512 GB of memory and is aimed at customers that wish to consolidate data centers.

Between the server hardware and the operating systems that will run the applications is a virtualization layer of software that manages the entire system. The two main products in this field are VMware ESX Server and Microsoft Virtual Server.

This IBM Redbook discusses the technology behind virtualization, the x3950 technology, and the two virtualization software products. We also discuss how to properly manage the solution as though they all were a pool of resources with Virtual Machine Manager, a unique consistent management interface.

This Redbook does not make any comparison between ESX Server and Virtual Server. Instead, we assume that you have already decided one over the other and are interested in learning more about the planning and implementation of each particular product.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Raleigh Center.

Massimo Re Ferre' is a Certified IT Architect within the IBM Systems Group in EMEA. For more than 10 years, he has worked for IBM on Intel/AMD solutions, architectures and related hardware and software platforms. In the past few years, he as worked on a number of server consolidation, rationalization and virtualization projects with key customers in the Europe-Middle East-Africa (EMEA) geography. He has also participated as a speaker in several IBM international events on the matter above. Massimo is a member of the Technical

Expert Council (TEC), which is the Italian affiliate of the IBM Academy of Technology.

Don Pomeroy is a Senior Systems Administrator for Avnet Inc, one of the world's largest B2B distributors of semiconductors, interconnect, passive and electromechanical components, enterprise network and computer equipment from leading manufacturers, and an IBM Business Partner and customer. His primary focus for the last two years has been server consolidation and virtualization with VMware products. He has the following certifications; A+, Network+, Security+, IBM Certified Specialist for xSeries, MCSA, and VCP.

Mats Whalstrom is a Senior IT Specialist within the IBM Systems & Technology Group in Raleigh, with over 10 years of experience within the Intel/AMD/PowerPC and storage arena. He is a senior member of the IBM IT Specialist accreditation board and has previously worked for IBM in Greenock, Scotland and Hursley, England. His current areas of expertise include BladeCenter®, storage hardware and software solutions. Previous to working on this redbook, he is a coauthor of the first and third editions of the redbook *IBM TotalStorage SAN File System*.

David Watts is a Consulting IT Specialist at the IBM ITSO Center in Raleigh. He manages residencies and produces Redbooks™ on hardware and software topics related to IBM System x™ systems and associated client platforms. He has authored over 40 Redbooks and Redpapers. He holds a Bachelor of Engineering degree from the University of Queensland (Australia) and has worked for IBM for over 15 years. He is an IBM Certified IT Specialist.

The team (l-r): David, Mats, Don, Massimo

Thanks to the following people for their contributions to this project:

From the ITSO:

Tamikia Barrow
Byron Braswell
Rufus Credle
Linda Robinson
Denice Sharpe
Jeanne Tucker

From IBM:

Jay Bretzmann
Marco Ferretti
Susan Goodwin
Roberta Marchini
Allen Parsons
Deva Walksfar
Bob Zuber

From VMware:

Richard Allen
Peter Escue
John Hawkins

From Microsoft®:

Jim Ni
Mike Sterling
Jeff Woolsey

Others

Eran Yona, Israeli Ministry of Defense
Jennifer McConnell, Avnet
Members of the VMTN Forum

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HZ8 Building 662
  P.O. Box 12195
  Research Triangle Park, NC 27709-2195

1

Introduction to virtualization

Virtualization is the concept of decoupling the hardware from the operating systems and applications. This can be implemented a number of ways as we will discuss, but fundamentally it is achieved by inserting a layer between the server hardware and software components and to provide either the necessary interfaces, or a simulation of one component to the other.

In this chapter, we cover the following topics:

1.1 Why switch to virtualization?

Using virtualization can reduce costs in a variety of ways, such as hardware, software and management costs. There are a number of areas in which virtualization can be used, including:

- Server consolidation and server containment

  This is one of the most important benefits and is one of the key reasons to use software virtualization technologies on high-end xSeries servers.

  Server containment is the next phase after server consolidation. After you have consolidated your existing production servers, you do not want requests for new Windows and Linux systems deployments compromising the streamlining you have achieved in your data center.

  Read about server containment in 1.5, “Server consolidation” on page 20.

- Flexible development and test environments

  The ease and flexibility of creating and reconfiguring guest operating systems (*virtual machines* or VMs) means that development and test environments get significant benefit from virtualization. In fact, this is where most of these x86 virtualization technologies were used when the technology was first available.

- Disaster recovery made much easier

  Disaster recovery (DR) is another key reason to implement virtualization technologies. Because the whole virtual machine running on any virtualization technology is typically self-contained in a single file, it becomes very easy to manipulate a Windows machine.

  While it is theoretically possible to achieve advanced DR scenarios using traditional Intel®-based systems and *boot from SAN* technologies, it introduces two key challenges:

  – Your primary site and your backup site need to be upgraded in lock-step with your production environment. If you install a new system in the production site (because of a new application) you must install its counterpart in the backup site SAN, otherwise it will not be possible to reactivate your application from the back up.

  – Not only must the backup site be aligned to the production site in terms of number of servers but also in terms of server configurations. The idea is that the SAN at the two locations mirror each other and allow for a system to boot with the same OS image (on the SAN) in either one site or the other. Your server hardware must be identical for this to happen.

  Virtualization technologies solve most of these issues because they are able to decouple the bindings between the operating system (with the application and data) and the underlying hardware. For example, you can have a different hardware topology in the recovery site, both in terms of numbers of servers and configuration of those, and still be able to boot all your guests on the two different data centers.

- Faster application deployment

  Products such as VMware ESX Server and Microsoft Virtual Server store the entire virtual machines in a single large file. As a result, operating system and application provisioning becomes very easy and can be achieved by simply copying a standard template virtual machine file to a new file, which becomes your new virtual machine. As a result, you can activate your new VM in a matter of hours (if not minutes), rather than days as you would if you were to purchase a traditional, physical server and install an OS and application on it.

  This advantage assumes that you still have systems resources available on your virtualized physical server (memory and CPU primarily) with which you can activate your newly created virtual machine.

  These concepts are very similar and complementary to the on-demand business model where IT is aligned, as fast as it can possibly be, to the requirements of the business. Faster application deployments and flexibility should not be seen as a technical advantage, but rather as a business advantage where for example being first to market with a new service can be key to success.

- Windows and Linux standardization and reliability

  Based on the fact that you can now create your virtual machines by simply copying a disk from a given template into a production disk, reliability and ease of maintenance of that guest operating system increases.

  In the past, administrators were used to installing operating systems on different hardware platforms with different disk controllers, different network cards, and so forth, thus increasing the complexity of software stack and drivers being installed. because one of the key advantages of virtualization is that of shielding the real hardware being used, it provides a consistent set of virtual hardware interfaces that get exposed to the guest operating system. Potentially every guest can have an identical hardware stack as the virtualization layer pretends to have all guest running on the same hardware. This means:

  - Easier maintenance

    You no longer need to maintain different sets of drivers nor you need to keep track of your Windows and Linux hardware configurations, thus increasing the ease of management of the software stack.

- More reliability

  You no longer need to validate or certify any given type of hardware and device drivers for your Windows and Linux platforms. Your Linux and Windows hosts can potentially all have a single (virtual) standard SCSI adapter, Ethernet interface and so forth. Having a consistent and thoroughly tested low-level standardization of your Linux and Windows devices will likely improve reliability.

- Easier problem determination

  During difficult problem determination steps to debug your application you no longer need to take into account potential incompatibilities of hardware and device drivers. Physical deployments typically require a deep investigation of the actual hardware and software configuration to determine the root cause of the problem.

1.2 Virtualization concepts

There are a number of concepts that characterize virtualization software. These include:

- Emulation versus virtualization
- Hosted solutions versus hypervisor solutions
- OS virtualization
- Full virtualization versus paravirtualization
- 32-bit versus 64-bit support

We discuss each of these in the following sections.

1.2.1 Emulation versus virtualization

There is a lot of confusion regarding this topic. Although these names often get used interchangeably, they mean two completely different things. In order to fully understand this we need to review CPU architectures.

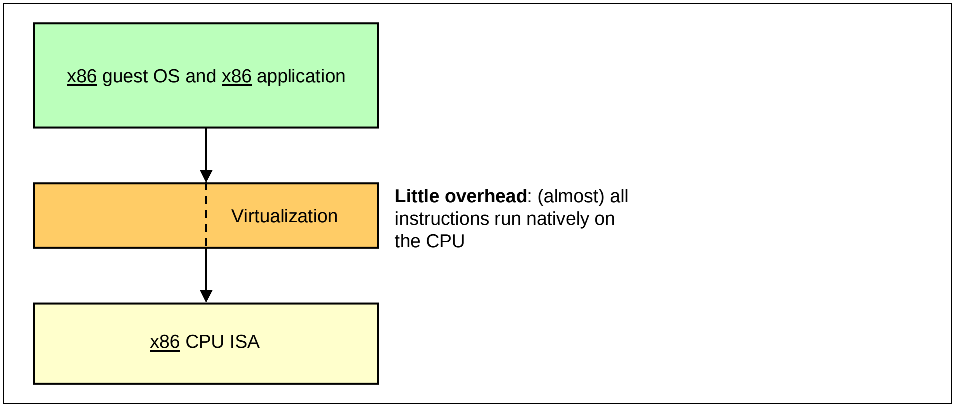
Each microprocessor has its own instruction set, the *instruction set architecture* (ISA). The instruction set is the language of the processors and different microprocessors implement different instruction sets. This is the reason for which an executable file or program can only run on a processor it has been compiled for. Table 1-1 shows some examples.

Table 1-1 Instruction set implementation examples

Processor	Architecture	Instruction set
Intel Pentium/Xeon®	Complex instruction set computing (CISC)	x86-64
AMD Opteron	CISC	x86-64
IBM POWER™	Reduced instruction set computing (RISC)	POWER RISC
Intel Itanium® 2	Explicitly parallel instruction computing (EPIC)	IA64
SUN Sparc	RISC	SUN RISC
MIPS	RISC	MIPS RISC

As you can see in this table, although all CPUs are 64-bit CPUs and some of them share the same architecture design, they are basically all different in terms of the language used internally. The only exceptions to this are the Intel Pentium/Xeon and the AMD Athlon/Opteron. These two are fully compatible because they both implement the very same exact instruction set, the industry standard x86 ISA.

Virtualization is where the instruction set that the guest operating systems expect and the instruction set that the hardware uses are identical. The advantage here is that instead of running a single operating system on top of the physical system, you can now run a number of operating sytems on a single physical system. The key, however, is that operating systems and applications still have to be designed and compiled for that ISA that the physical system uses.



Little overhead: (almost) all instructions run natively on the CPU

Figure 1-1 Virtualization

On the other hand, *emulation* is the mechanism where the instruction set of the guest OS is translated on-the-fly to the instruction set of the physical platform. This means that not only would you be able to run more than one operating system per physical system (as with virtualization), but you could also, through emulation, run different guest operating systems built for different CPU instruction sets.

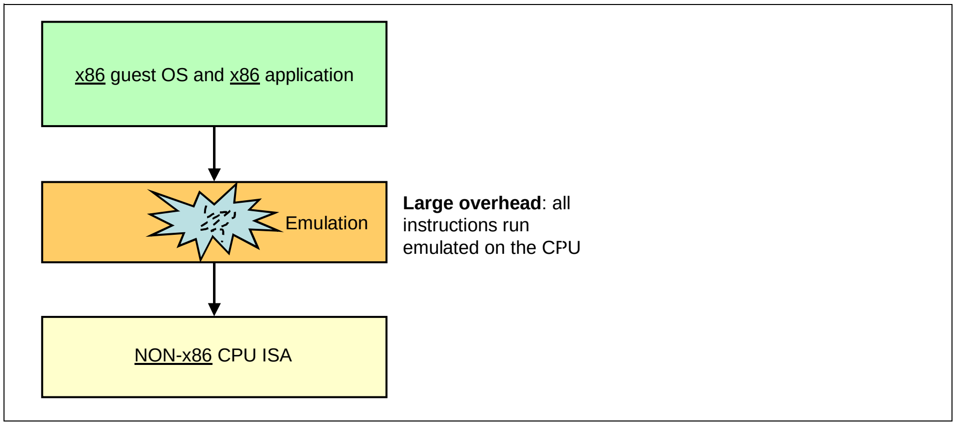

x86 guest OS and x86 application
Emulation
Large overhead: all instructions run emulated on the CPU
NON-x86 CPU ISA

Figure 1-2 Emulation

VMware ESX Server and Microsoft Virtual Server are examples of products that provide this type of virtualization mechanism.

Although emulation seems to be more appealing than virtualization, in reality that is not the case. Given the tremendous hit in performance when emulating a different CPU architecture, emulation has never taken off and most likely it will not in the future either.

As a result of this, for instance, you *cannot* do the following:

- run IBM AIX® on a VMware virtualization product
- run a VMware product on an Itanium based system

These are only examples. The key point is that the combination of CPU, virtualization software, and guest OS all need to be consistent with regards to the ISA.

1.2.2 Hosted solutions versus hypervisor solutions

Another source of misunderstanding is the terminology of hosted solution and hypervisor solution. In simple terms, *hosted solutions* are virtualization products that require a full operating system underneath the virtualization software.

Microsoft Virtual Server, illustrated n Figure 1-3 on page 7, is an example of a hosted solution.

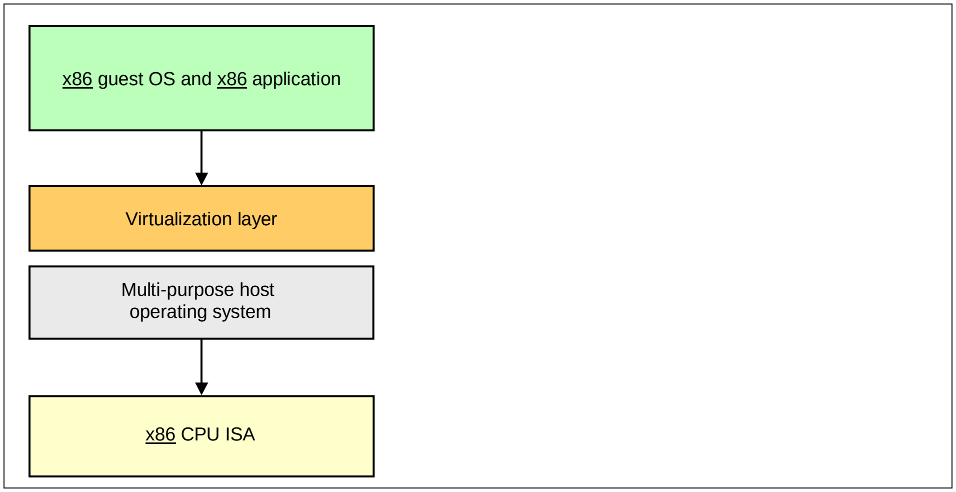

x86 guest OS and x86 application
Virtualization layer
Multi-purpose host operating system
x86 CPU ISA

Figure 1-3 Hosted virtualization solution (Microsoft Virtual Server)

On the other hand a *hypervisor solution* does not require any underlying host OS that supports the virtual machines. The hypervisor is a thin kernel that runs directly on the hardware and it is optimized to run specific tasks such as scheduling virtual machines.

Figure 1-4 shows the hypervisor architecture. The VMkernel that ships with VMware ESX Server is an example of hypervisor.

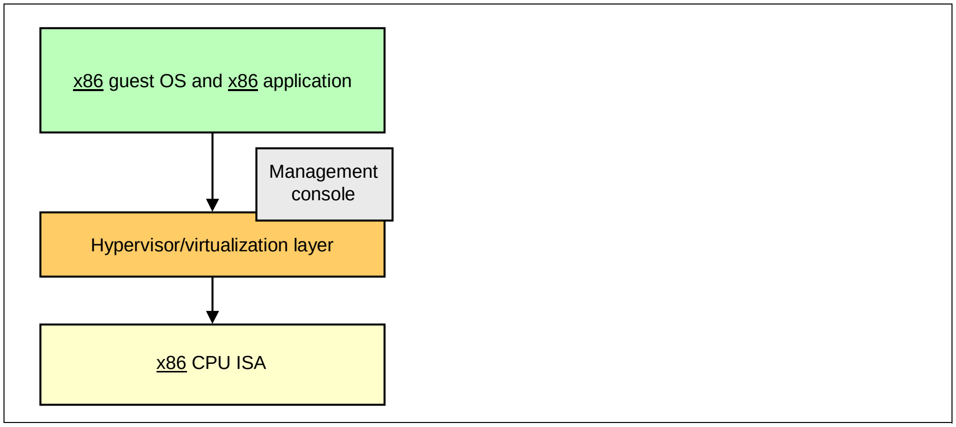

x86 guest OS and x86 application
Management console
Hypervisor/virtualization layer
x86 CPU ISA

Figure 1-4 Hypervisor virtualization solution (VMware ESX Server)

Note: Most if not all hypervisor solutions available today use standard multipurpose OS routines as a management console to install, boot and administer the hypervisor. For example, ESX Server uses a customized Red Hat version that they refer to as the Console OS.

As shown in Figure 1-4, this management console is not part of the hypervisor layer. During the boot process, the hypervisor takes complete control of the hardware and the management console is simply a more privileged virtual machine.

While a hypervisor solution is more challenging to develop and maintain, it provides much better performance and scalability compared to a hosted solution. This is not to diminish the challenges in developing a hosted solution, but there are certain things that the hypervisor has to take care of that a hosted virtualization solution does not. For example all the hardware support must be built into the low-level hypervisor because it runs directly on the hardware while the hosted virtualization technology simply inherit the hardware support of the hosting multipurpose OS.

1.2.3 OS virtualization

An alternative to hardware virtualization is OS virtualization. While hardware virtualization simulates an server hardware on which you can run various x86 guest operating systems, *OS virtualization* is a layer that simulates separate operating system instances to the applications you want to run. This approach is not as flexible as hardware virtualization, because all guest operating systems must be the same type as the master OS. There are advantages to OS virtualization including performance.

Examples of products implementing OS virtualization include:

- User Mode Linux (UML)
- SUN containers
- HP vPars

1.2.4 Full virtualization versus paravirtualization

Full virtualization is the concept of creating a virtual layer, typically a hypervisor, that fully simulates a standard x86 system. This means that the guest OS does not need to be modified to be aware of the virtualization layer and can run natively on the virtualization layer as thought it were on a standard x86 system.

Both VMware ESX and Microsoft Virtual Server both implement a full virtualization technology.

Figure 1-5 represents this concept.

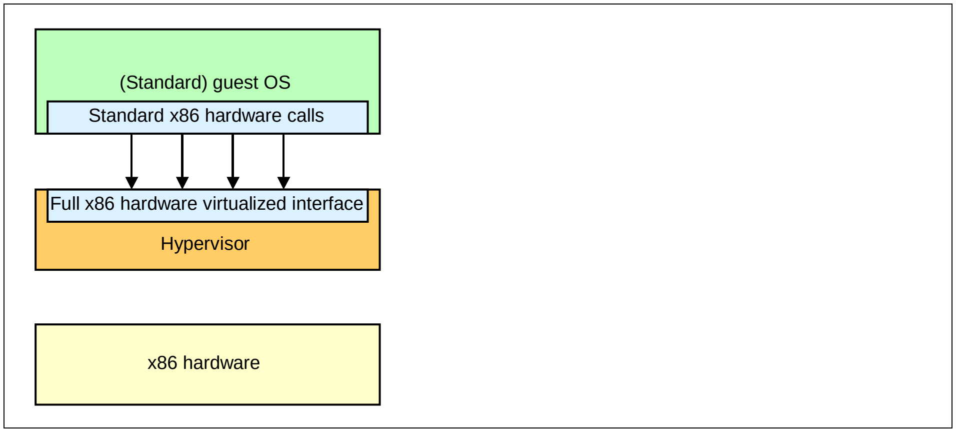

(Standard) guest OS
Standard x86 hardware calls
Full x86 hardware virtualized interface
Hypervisor
x86 hardware

Figure 1-5 Full virtualization architecture

With *Paravirtualization* (Figure 1-6), the guest OS is modified to be virtualization-aware so that the it can call the hypervisor directly to perform low-level functions. An example of virtualization product that does paravirtualization is XEN.

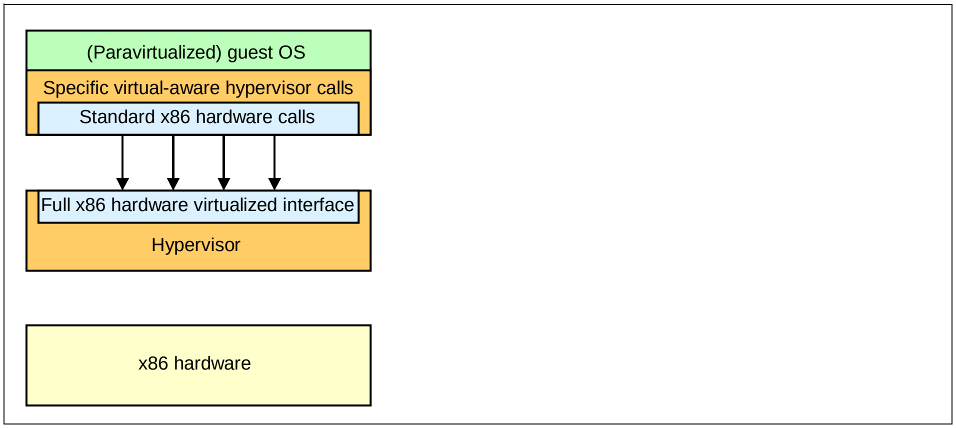

(Paravirtualized) guest OS
Specific virtual-aware hypervisor calls
Standard x86 hardware calls
Full x86 hardware virtualized interface
Hypervisor
x86 hardware

Figure 1-6 Paravirtualization architecture

There are at least two reasons for doing paravirtualization:

- Reduced complexity of the virtualization software layer

  Because the x86 ISA historically does not support virtualization, the full virtualization approach must implement mechanisms for which it traps certain privileged guest OS calls and they translate them into instructions that suit the virtualized environment. This is done using a technique called *binary translation*.

This means that the VMware and Microsoft products trap and translate certain instructions that are directed at the hardware, thereby allowing guest operating systems to operate as though they were in full control of the hardware.

This is where paravirtualization differs: instead of being the hypervisor trapping low-level instructions and transforming those, it is the virtual-aware guest OS that behaves differently and becomes aware that it is not the only operating system on the server.

This in turns means that the hypervisor does not need to provide the complexity of entirely simulating an x86 computer. So it can be streamlined to run more efficiently.

This does not mean, however, that using paravirtualization you simplify the entire complexity. It only means that you distribute the complexity across the hypervisor as well as the guest OS while in a fully virtualized environment it is only the virtualization layer to deal with the entire spectrum of the challenges.

- Performance

  The second reason to implement paravirtualization (and probably the more important) is performance. The Full Virtualization approach often suffers in terms of performance because there are many overheads in running a standard, unmodified guest OS on a virtualization layer.

  For example, a standard unmodified guest OS typically checks everything before passing the information to the hardware (CPU, memory and I/O), and so too does the virtualization layer (which it needs to do because it is closest to the hardware).

Note: Do not confuse the instruction translation occurring within the virtualization layer implementing the full virtualization approach with the emulation discussed in 1.2.1, “Emulation versus virtualization” on page 4. The overhead doing emulation is an order of magnitude larger than the slight performance penalties occurring using full virtualization technologies.

1.2.5 32-bit versus 64-bit support

IBM xSeries servers such as the x3950 have Intel EM64T-based processors which support 32-bit and 64-bit operating systems and applications. These processors are full 64-bit processors with 64-bit memory addressability, but they also have the advantage of native backward compatibility with 32-bit operating systems and applications.

Most virtualization software users have implemented a 32-bit-only environment similar to Figure 1-7.

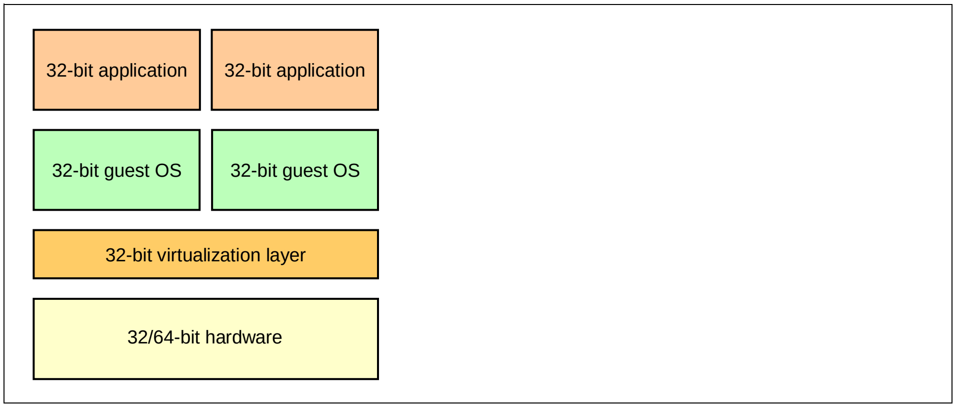


Figure 1-7 Current 32-bit virtualization layout

As more 64-bit applications become available in the future as well as key virtualization product being ported to 64-bit x86 architecture, we could achieve maximum flexibility as shown in Figure 1-8 on page 11:

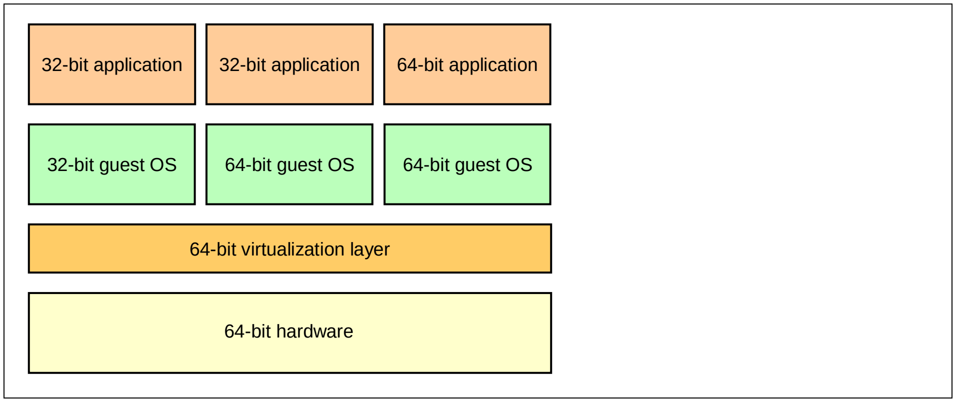


Figure 1-8 Future 64-bit virtualization layout

As you can see, with this configuration, both 32-bit as well as 64-bit guest operating systems can be run side-by-side.

Note: Running a 32-bit OS or application on 64-bit Intel EM64T or AMD AMD64 processors is not referred to as emulation.

Virtualization products such as VMware Workstation are already capable of running 64-bit guest operating systems and this capability is likely to extend to all x86-centric virtualization products.

1.3 Available virtualization products

The x86 virtualization space is one of the most active and fast-moving software (and hardware) ecosystems. Three or four years ago there were just a handful of independent software and hardware vendors proposing products and solutions in this area. Today there are dozens of software and hardware vendors encouraging the adoption of these technologies and this is becoming mainstream at all levels.

The pioneer of these software technologies, at least from a commercial perspective, is VMware. But other companies, including Microsoft, are increasingly keen to create their roadmaps based on the virtualization concepts and advantages we have discussed so far. Other than VMware and Microsoft, there are a number of other software vendors that are selling their own virtual machine technologies. We describe Xen here, although there are others on the market.

1.3.1 VMware

VMware Inc. is based in Palo Alto, California and is one the companies that pioneered x86 server virtualization technologies. They have grown their software offering in the last few years from a simple virtualization layer product set to a more complex product portfolio that includes management tools and add-on features. The virtualization products VMware have available at the moment are:

- VMware Workstation is a hosted virtualization solution for desktop PCs and mobile computers. Typical usage scenarios range from software development environments, demo and test, and similar end-users' requirements.
- VMware GSX Server is a hosted virtualization solution for departmental servers. Typical usage scenarios are consolidation of servers in remote branch offices and consolidation of islands of under-used servers in specific departments.
- VMware ESX Server is an enterprise product based on hypervisor technologies aimed at the datacenter virtualization. The server is capable of simultaneously hosting many different operating systems and applications. The Console OS is provided as an interface to support the VMware ESX Server kernel.

ESX Server is the VMware product is the natural complement to the System x3950 and we we focus on it throughout this redbook.

There are a number of other products and add-on features that VMware offers, including these:

- VMware VirtualCenter is a management software used to configure and monitor properly the VMware ESX and VMware GSX platforms.
- VMware VMotion is a feature that allows the administrator to move a running virtual machine from a physical host to another physical host without downtime.

1.3.2 Microsoft

Microsoft moved into the virtualization space when it acquired Connectix in 2003. Connectix was one of the first companies to ship virtualization software offerings. Since then Microsoft has incorporated products and assets from Connectix and built its own virtualization product portfolio:

- Microsoft Virtual PC is a hosted virtualization solution for desktop PCs and mobile computers. Typical usage scenarios range from software development environments, demo & test and similar end-users requirements.
- Microsoft Virtual Server is a hosted virtualization solution for departmental servers. Typical usage scenarios are consolidation of test and development servers, existing Windows NT® 4.0 servers, and infrastructure services.

We focus on Microsoft Virtual Server throughout this redbook.

1.3.3 Xen

Although VMware and Microsoft are currently the two biggest players in the x86 virtualization realm, at least from a commercial point of view, it is worth mentioning that they are not the only software vendors working in this field.

Specifically there is an open-source project developed at the University of Cambridge, called *Xen*, and it is gaining interest in the x86 community. Xen is similar in concept to VMware ESX Server in that it is a hypervisor. At the time of writing, the Xen project is still in the R&D stage and is not generally considered a ready-to-use commercial alternative to the VMware or Microsoft products.

One interesting thing to note is that while both VMware and Microsoft are implementing full virtualization and moving towards providing the choice to customers to run in either full virtualization or paravirtualization mode, Xen has used a different approach. Xen 3.x supports paravirtualized Linux guests (as did

Xen 2.x) but now also plans to support unmodified guest OSes such as Windows Server 2003 when run on a server with Intel Virtualization Technology (VT).

Note: While VMware and Microsoft will use hardware-assist virtualization technologies built into the CPUs to improve performance, Xen 3.0 will require *CPU hardware assist* in the processor to run in full virtualization mode. Refer to 1.7, "Virtualization futures" on page 30

1.3.4 Summary of supported technology

It is important to understand that this redbook does not cover the entire spectrum of technologies we have mentioned in 1.2, "Virtualization concepts" on page 4. Specifically all the topics we discuss in this redbook focus on:

- Virtualization (not emulation)
- The x86 instruction set

Table 1-2 on page 14 shows a summary of the virtualization technology supported by key products.

Table 1-2 Summary of most common virtualization technologies (parenthesis indicates future plans)

Product	**VMware Workstation**	**GSX Server**	**ESX Server**	**Microsoft Virtual PC**	**MS Virtual Server**	**XEN**
Audience	Personal	Workgroup	Enterprise	Personal	Workgroup & Enterprise	
Full Virtualization	Yes	Yes	Yes	Yes	Yes	No[1]
Paravirtualization	No[1]	No[1]	No[1]	No[1]	No[1]	Yes
Hosted or Hypervisor	Hosted	Hosted	Hypervisor	Hosted	Hosted[2]	Hypervisor
Virtualization or Emulation	All of these x86 products currently implement Virtualization					
64-bit guest OS support	Yes	No[1]	No[1]	No[1]	No[1]	
Notes: 1. The vendor has indicated that this is in plan for future releases 2. Microsoft has stated that MSVS will implement a hypervisor layer in a future release						

Note: Full Virtualization support and Paravirtualization support are not mutually exclusive. A virtualization layer can provide either one or both at the same time.

1.4 Scale-up versus scale-out

Historically, the concept of *scaling out* versus *scaling up* has been introduced to describe the use of many, small, low-cost servers versus fewer, big, expensive servers, with low-cost and expensive being the keywords. There are certainly technical differences between the two implementations, but generally hardware cost is one of the key reasons to choose a scale-out approach because, computing power being equal, a single big SMP server usually costs more than many small servers.

It is easy to agree that when using 2-socket blade servers, you are implementing a scale-out approach, and when using a single 16-socket x86 server you are implementing a scale-up approach. However, it is not easy to agree on which approach we are using if we compare 4-socket to 8-socket servers. We do not pretend to be able to give you this definition, because this is largely determined by the size of your whole infrastructure, your company strategies, and your own attitudes.

For this discussion, we always refer to the extremes: 2-socket blade servers for the scale-out approach and 16-socket x3950 for the scale-up approach. Although 4-way servers should fall into the scale-out category and 8-way configurations should fall into the scale-up approach, this varies depending on your company's characteristics and infrastructure size.

Having said this, we also want to remark that this market is being segmented by Intel in three big server realms:

- 1-way servers: Intel Pentium® 4
- 2-way servers: Intel Xeon
- 4-way+ servers: Intel Xeon MP

From this segmentation, it is easy to depict that 4-way, 8-way and 16-way systems might all fall into the premium price range as opposed to the 2-way platforms (we do not even consider 1-socket platforms for virtualization purposes). The point here is that there is much more difference as far as the price is concerned between two 2-way servers versus one 4-way server than there is between two 4-way servers versus one 8-way server. There is a common misunderstanding that servers capable of scaling at 8-socket and above are much more expensive than servers capable of scaling at 4-socket maximum. While this might be true for other industry implementations, the modularity of the System x3950 allows a pricing policy that is very attractive even for high-end configurations.

It is very difficult to state clearly whether an 4-socket based system or an 8-socket system fall into one category or into the other. The terms scale-up and scale-out should be used in relative terms rather than in absolute terms.

1.4.1 Scale-up (fewer, larger servers)

In this section, we discuss the characteristics (including advantages and disadvantages) of a scale-up implementation approach in order to create or *feed* the infrastructure.

The IBM building block for a scale-up approach is the System x3950, which is a modular, super-scalable server that can drive, potentially, as many as 32 CPUs in a single system image. In our case, *single system image* means a single instance of VMware ESX Server with a single Console OS (one ESX Server system).

Figure 1-9 illustrates a 16-socket configuration using System x3950 modules in a single system image.

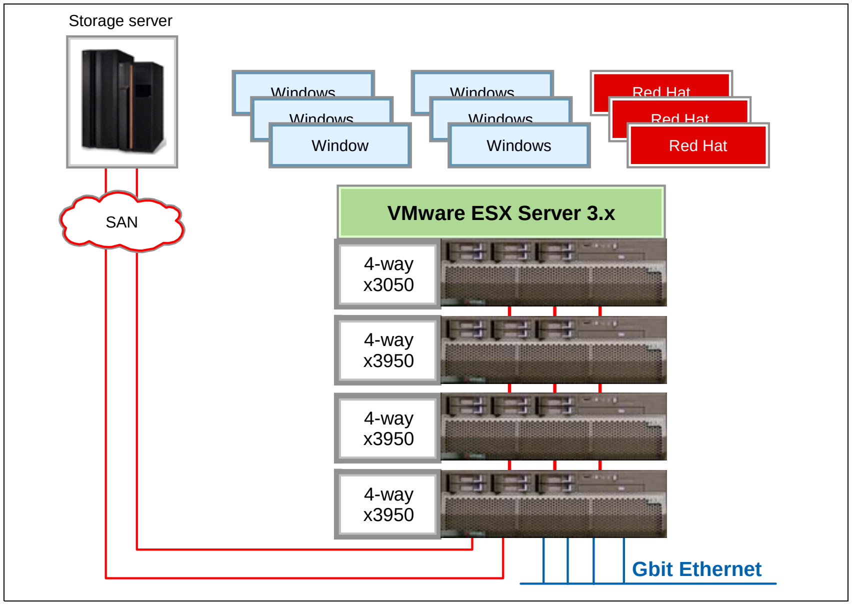
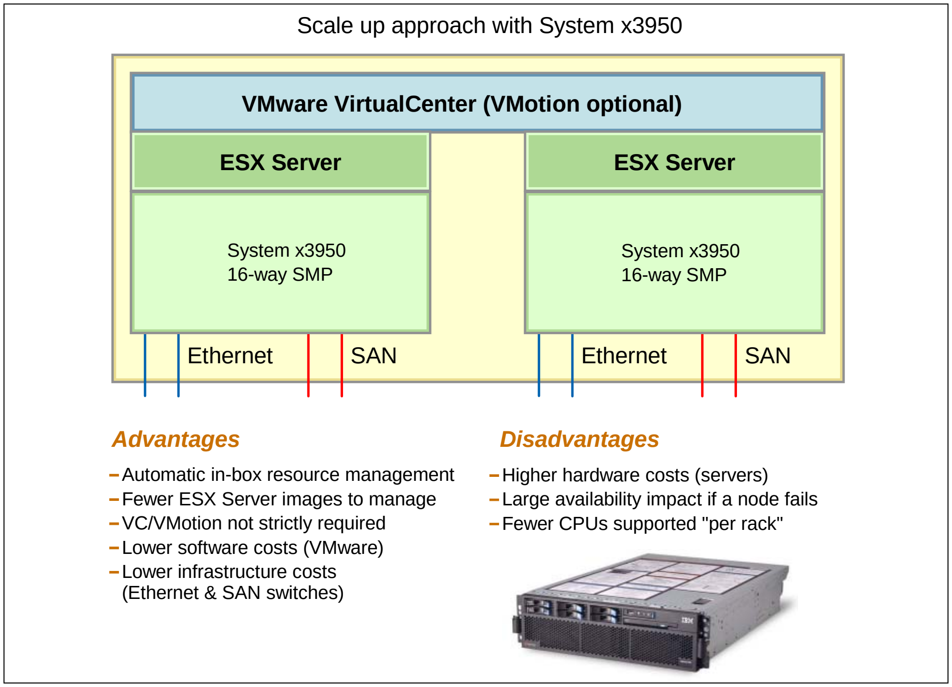


Figure 1-9 Scale-up implementation

In this scenario, all 16 sockets are configured so that ESX Server can be installed once and drive all of the available resources. (Note that the number of SAN/Ethernet connections in this picture is merely an example).

Figure 1-10 on page 17 shows how a (simplified) scale-up approach would be implemented.



Advantages

- Automatic in-box resource management
- Fewer ESX Server images to manage
- VC/VMotion not strictly required
- Lower software costs (VMware)
- Lower infrastructure costs (Ethernet & SAN switches)

Disadvantages

- Higher hardware costs (servers)
- Large availability impact if a node fails
- Fewer CPUs supported "per rack"

Figure 1-10 Scale-up characteristics

Figure 1-10 lists advantages and disadvantages upon which the industry generally agrees. Of course, one reader might rate fewer ESX Server images to manage as a key advantage while another reader might not be concerned about having to deal with 30 ESX Server system images. Even if we could try to give each of these a weight based on our experience, we realize that this will be different from customer to customer.

A typical concern regarding this kind of high-end configuration is performance and scalability. This is a real concern in most of the scenarios where a single application tries to benefit from all available resources. Usually there is a scaling penalty because some applications are written in a way that limits their ability to generate more workload to keep the server busy.

With ESX Server implementations, this is not usually a concern. In fact, we are not dealing here with a single application running on the high-end system but rather with multiple applications (and OS images) running on the same high-end systems. Because the virtualization layer is very efficient, if the server is not being fully utilized you can add more virtual machines to result in more workload for the server. This has proven to scale almost linearly.

The System x3950 leverages the NUMA architecture, which enables administrators to add CPU-memory bandwidth as they add CPUs to the configuration. ESX Server 3.x fully exploits the NUMA architecture of the System x3950.

1.4.2 Scale-out (more, smaller servers)

This section describes the advantages and disadvantages of implementing the VMware infrastructure with many small servers, each running ESX Server. Although many options exist for implementing a farm comprised of small low-end servers, we consider the use of the IBM BladeCenter as the most viable alternative when discussing this requirement. BladeCenter is the name of the powerful integrated chassis that, along with many other infrastructure components such Ethernet and SAN switches, contains the IBM HS20 and IBM HS40 blades (the 2-socket and 4-socket Intel-based blade servers) as well as the IBM LS20 blade (the 2-socket Opteron-based blade server).

All of our discussions about the IBM blades can apply to other 2-socket traditional servers, either rack or tower, but we think that the use of blades makes more sense in a scale-out implementation such as this.

Figure 1-11 shows a typical ESX Server deployment on an IBM BladeCenter.

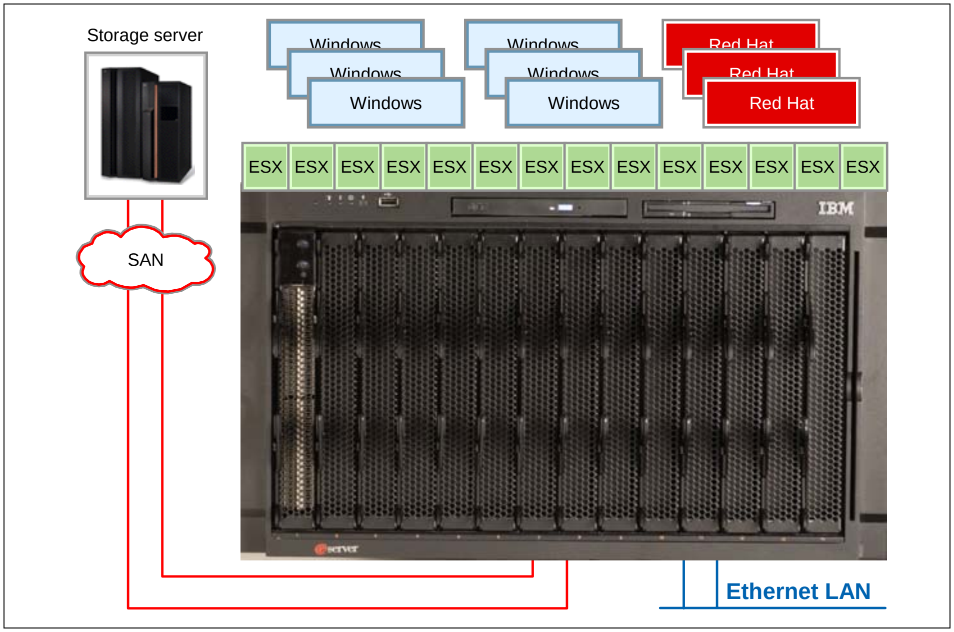


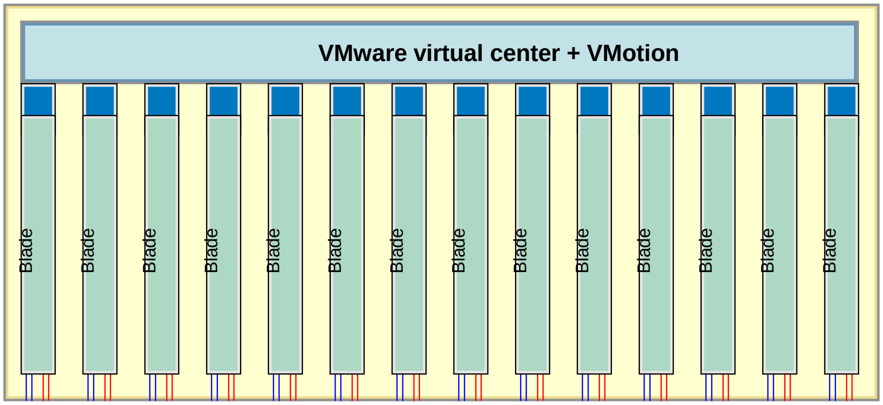
Figure 1-11 Scale-out implementation

Although from a graphical perspective the two solutions (System x3950 and BladeCenter) look similar, there is one key differentiated between the two. The x3950 is operated by a single ESX Server image. However in this scenario every blade must be driven by its own ESX Server image, resulting in 14 ESX Server installations to drive 14 independent HS20 blades.

As anticipated, this would not make much difference if we were to use 14 standard 2-socket servers instead of blades. As a result, we would have had 14 ESX Server installations to operate 14 independent tower or rack servers.

Figure 1-12 shows how a (simplified) scale-out approach would be implemented.

Scale out approach with IBM BladeCenter



Advantages

- Lower hardware costs (servers)
- Low H.A. impact in case of failure of a node
- More CPUs supported for each rack

Disadvantages

- Manual across-boxes resource management
- VC/VMotion mandatory for reasonable management
- Many hypervisor (ESX) images to maintain
- Higher software costs (VMware)
- Higher infrastructure costs (Ethernet/SAN switches)

Figure 1-12 Scale-out characteristics

This approach has a number of advantages, including the high availability and resiliency of the infrastructure. If a single module fails (be it blade hardware or the ESX Server image), you lose only a small part of your virtual infrastructure as opposed to what would happen if you lost an entire 16-CPU server (which is supposed to run eight times the number of virtual machines that the 2-socket blade supports). Another easy advantage to notice is server costs. Eight 2-socket blades usually cost less than a single 16-socket server. The reason is the premium price associated with 4-CPU and above systems.

One drawback to the proposed solution is the management of all of those Console OS images. For example, think of a minor or major upgrade to an ESX Server. You would have to perform that on every blade.

Another important drawback of this approach is that, although 2-socket server costs are much lower than those of high-end SMP servers, we need to look at the scenario as a whole. Every 2-socket server must have a (typically redundant) network connection and a (typically redundant) SAN connection. So, to the raw costs of the servers being used, you have to add the costs of:

- Pure costs and management of the cabling
- Greater power consumption and heat production
- Maintenance for the above infrastructure
- Ethernet ports on the departmental network switch
- Fibre Channel ports on the SAN switch
- The Ethernet and Fibre Channel adapters (host bus adapters or HBAs) per each server

We are not implying that a single 16-socket or 8-socket server can always be configured to use only a pair of redundant HBAs. We have customers running 16-socket xSeries servers with only two HBAs, but because typically the dozens of virtual machines running on those are CPU-bound and memory-bound and do not require specific disk bandwidth, we understand that for scalability reasons more HBAs might be required for those high-end configurations. However, when using low-end servers, you would have to configure eight 2-socket servers—so, for a total eight HBAs (16 if redundancy is a requirement), this also means eight (or 16) SAN switch ports that (likely) would be under-used. This waste of money and resources is exactly what we are trying to avoid with virtualization technologies, which are all about simplifying the whole infrastructure through better resource utilization, server consolidation and server containment.

Although the IBM BladeCenter offering is very appealing from a management perspective because all of the infrastructure modules are collapsed into the chassis, this does not mean that they do not exist. No matter how these switches are deployed (racked traditionally or into the BladeCenter chassis), they still have to be purchased.

1.5 Server consolidation

As we have mentioned, server consolidation is one of the many advantages of virtualization technologies. In the beginning, most organizations viewed virtualization as a means by which to reduce the complexity of their distributed IT environment, being able to run their Windows OS images on fewer physical servers.

In 1.4, “Scale-up versus scale-out” on page 15 we introduced this concept of implementing a virtual infrastructure using either many small servers or fewer bigger servers. While we fully understand that there is no definite answer and, depending on the situation being considered, either approach would work fine, we also believe there are some potential technical exposures in this discussion.

Specifically, if you look at the past records of trends and technologies for operating properly an IT environment, there are some interesting patterns and recurring trends that generally followed this order:

1. In the 70s and the first part of the 80s it was a common and best practice to use powerful mainframe computers to manage the IT operations properly of any given organization. This was the *centralized* era. Life was good from a management perspective. But most organizations felt that these systems had a high cost of acquisition and that they needed to be more flexible, whatever that meant.
2. In the second part of the 80s and in the 90s we saw a strong trend for which computing resources tended to be distributed across the business units inside the organization. A new application model emerged during those years, the *client/server* era. While the hardware and software acquisition costs dramatically decreased in this new model, the Total Cost of Ownership (TCO) increased drastically. Organizations were supposed to introduce more flexibility in their infrastructure but in turns they lost complete control of their IT and also they created application silos that could not talk to each other.
3. As a way to improve upon the client/server model, in the beginning of 2000 most of the IT analyst agreed that, because of the poor manageability of the distributed infrastructure and the high costs of ownership associated with that, a certain degree of consolidation was desirable for the sake of those organizations that embraced the client/server trend.

Keeping this pattern progression in mind, it is not by chance that mainframes, which were supposed in the late 80s to be phased out over time because of the client/server trend, regained a lot of interest and are now enjoining a sort of rejuvenation because of their openness to the Linux environment. We have been experiencing the same pattern for UNIX® configurations where most organizations are now implementing high-end partitionable IBM System p5™ servers to consolidate their workloads instead of using low-end under-utilized POWER-based machines.

We are not encouraging you here to endorse these specific technologies. However, the point we are trying to make is that, assuming that the backbone of your IT infrastructure is based on the x86 standard architecture, it would be advisable to follow this consolidation pattern as much as possible to avoid following in the same pitfalls that most organization fell into during the *client/server* time frame. While we appreciate that the single big server is not a viable solution in the x86 space, nor we would suggest that approach anyway, and we understand that these deployments are based on a modular concept, the point here is *how big is a module* supposed to be and *how many modules* you need to run your x86 IT infrastructure smoothly and efficiently. Answering these questions leads to the discussion of scaling-up versus scaling-out or consolidating versus distributing.

1.5.1 So should we scale up or scale out?

Unfortunately there is no generic answer for this complex matter. We have had instances of serious issues in finding a common agreement regarding the size of the servers that can be treated as modules or components of a scale-up or a scale-out approach, so how can we find agreement about when to use one approach and when to use the other?

For example, we have had customers implementing Oracle RAC in a scale-out approach using four 2-socket Intel-based systems instead of a single high-end (scale-up) 8-socket Intel-based system, and we have met other customers reporting implementing a scale-out approach using four 8-socket Intel-based systems instead of a single high-end (scale-up) proprietary UNIX platform. We must look at these cases in perspective and, as you can see, the scope of your project and the size of the infrastructure are key elements in the equation.

Size and scope of the project

Regarding the size and scope of the project, we tend to hear absolutes when talking to customers regarding this topic, such as “two-socket servers are always the best solution” or “four-socket servers are always better.” It would be better to think instead in terms of the percentage of computational power that each single server or node brings to a virtual infrastructure design.

To clarify this concept, say your company needs to consolidate 30 servers onto a VMware virtual infrastructure (Figure 1-13 on page 23). For the sake of the discussion, they would all run on eight physical CPU packages based on preliminary capacity planning. What would your options be? You could install a single 8-socket server but that probably would not be the proper solution, mainly for redundancy reasons.

You could install two 4-socket servers but this would cause a 50% reduction in computing power if a hardware or software failure occurs on either of the two servers. However, we know many customers who find themselves in that situation because they thought that the value of such aggressive consolidation is worth the risks associated with the failure of an entire system. Many other customers would think that a good trade-off is to install four 2-socket servers, as a node failure really provides a 25% deficiency of computing power in the infrastructure, which is more than acceptable in many circumstances.

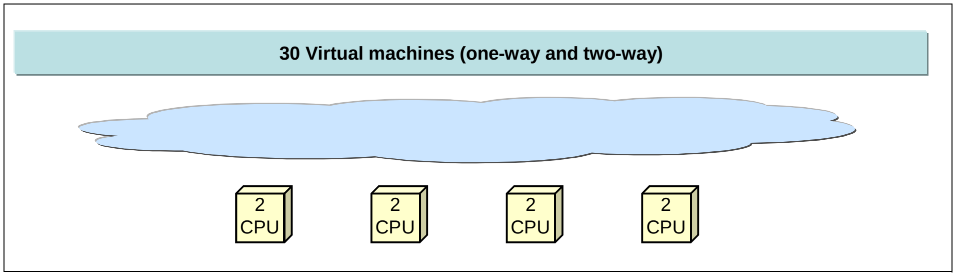


Figure 1-13 Project scope: 30 virtual machines (with two-socket systems)

Here is a second example: Say that your company needs to consolidate 300 servers. For the sake of the discussion, they would all run on 80 physical CPU packages based on a similar preliminary capacity planning. You can deduce from this second example that the absolute numbers that have been used in the first example might have a very different meaning here. In this exact situation, for example, the use of 10 8-socket ESX Server systems would cause a 10% reduction of computing power in case of a single node failure, which is usually acceptable given the RAS (reliability, availability, serviceability) features of such servers and the chance of failure. (Remember that in the first example the failure of a single 2-socket server brings a much larger deficiency of 25%).

Figure 1-14 illustrates the layout.

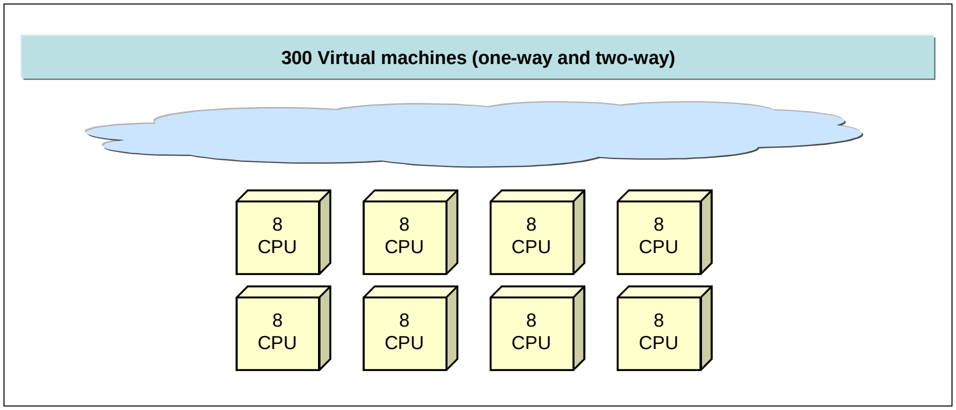


Figure 1-14 Project scope: 300 virtual machines (with 8-socket systems)

However, the use of 2-socket server blocks means that the administrator has to set up, administer, and update as many as 40 ESX Server systems, which could pose a problem for regular maintenance and associated infrastructure costs.

Figure 1-15 on page 24 shows how such an implementation would look.

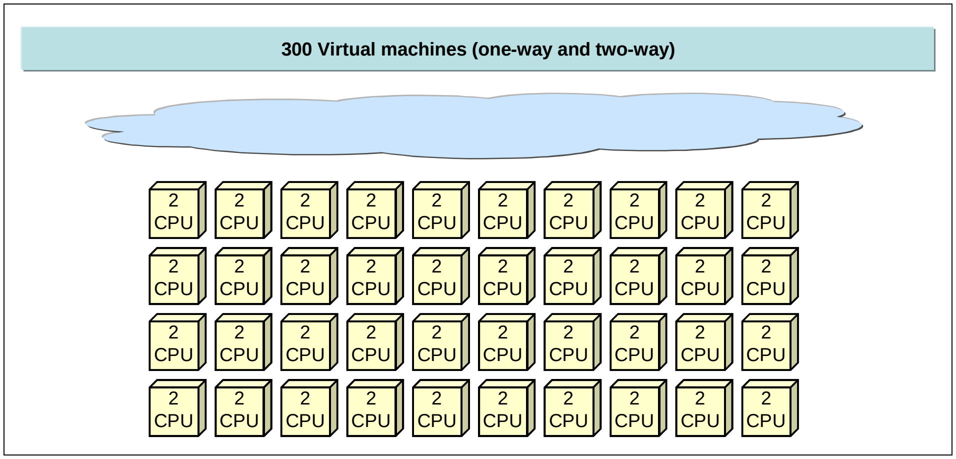


Figure 1-15 Project scope: 300 virtual machines (with two-socket systems)

Also take into account that VMware is introducing with ESX Server 3.0 4-way SMP-capable virtual machines. The bigger the VM is, the bigger your host is supposed to be. Although you could theoretically run a 4-way VM on a dual-core two-way server (4 cores), our experience with ESX Server 2.x running an n-way VM on an n-core system reveals this is not a good practice. ESX Server 3.0 is supposed to improve this situation, although it is too early to say for sure what the result will be.

There are certainly good practical reasons for using a host that can support many guests at the same time especially if the peaks of some of these virtual machines (possibly not all) overlap at some point during the day. For more information about the implications of using 4-way virtual machines on ESX Server 3.0 refer to 3.7, “Four-way virtual machines” on page 159.

Note: Do not think about absolute numbers and so called general sweet-spots regarding the configuration of the servers (that is, virtual infrastructure building blocks) but rather put them into your perspective.

1.5.2 Server containment

When streamlining an IT infrastructure and defining new operational standards, specific attention should be paid to fixing the management issues associated with the current environment with the proper technologies. However it is important to make sure that your infrastructure does not grow in complexity at the same pace as before as new projects and new line of business requirements imply the deployments of new IT services.

In fact it is not uncommon to find organizations that made a point in time analysis of their situation. They fixed that by rationalizing their current IT environment using proper consolidation technologies only to find themselves in a similar, unmanageable situation 1 or 2 years down the road. This is what happens when the requirements to install new services, and hence servers, can bring organizations back to a scenario that is as complex as the one they had to fix in the past.

While server containment is typically referred to as the advantage of deploying new services on spare resources available across the virtual infrastructure without the need to buy dedicated hardware for that application, we need to realize that the resources of the virtual infrastructure are not infinite. This simply means that, at some point, you will be required to add resources to the virtual infrastructure to respond efficiently to new demanding workloads. Usually when this happens, the answer is to *add new servers* to the virtual infrastructure.

While this might be considered a valid approach, one of the pitfalls of it is that you are in fact growing your number of physical servers, the number of hypervisors (or host OSs in hosted type of solutions) and so forth. So in few words you are adding to the infrastructure more objects that you need to manage, monitor, upgrade, and so on.

The x3950 might provide a different alternative to this approach. Instead of adding new servers, new hypervisors, and new points of management, you are rather adding more computational resources to the very same number of objects. If your infrastructure is comprised of 10 4-way x3950s and, just to make an easy example, you need to double the capacity to sustain the workload of the IT infrastructure. You can either double the number and get to 20 4-way x3950s or you can double the resources of the current x3950 getting to 10 8-way x3950s.

As you can easily imagine, the last scenario does not modify in any manner the way you manage your infrastructure, it just happens that the same number of server objects are now capable of supporting twice the workload for the very same complexity as before.

1.6 Dual core CPUs

Notice at this point that sometimes we refer to these x86 platforms as either 2-socket, 4 socket, 8-socket, or 16-socket configurations. Historically we have been used to referring to these systems as *n-way* systems. Due to the current trends in microprocessors the term n-way or n-CPU could become misleading if not used in the proper context.

The *Paxville* dual-core processors in the x3950 are the first Intel processor to offer multiple cores. Dual-core processors are a concept similar to a two-way system except that the two cores are integrated into one silicon die. This brings the benefits of two-way SMP with less power consumption and faster data throughput between the two cores. To keep power consumption down, the resulting core frequency is lower, but the additional processing capacity means an overall gain in performance.

Figure 1-16 compares the basic building blocks of the Xeon MP single-core processor (*Potomac*) and dual-core processor (*Paxville*).

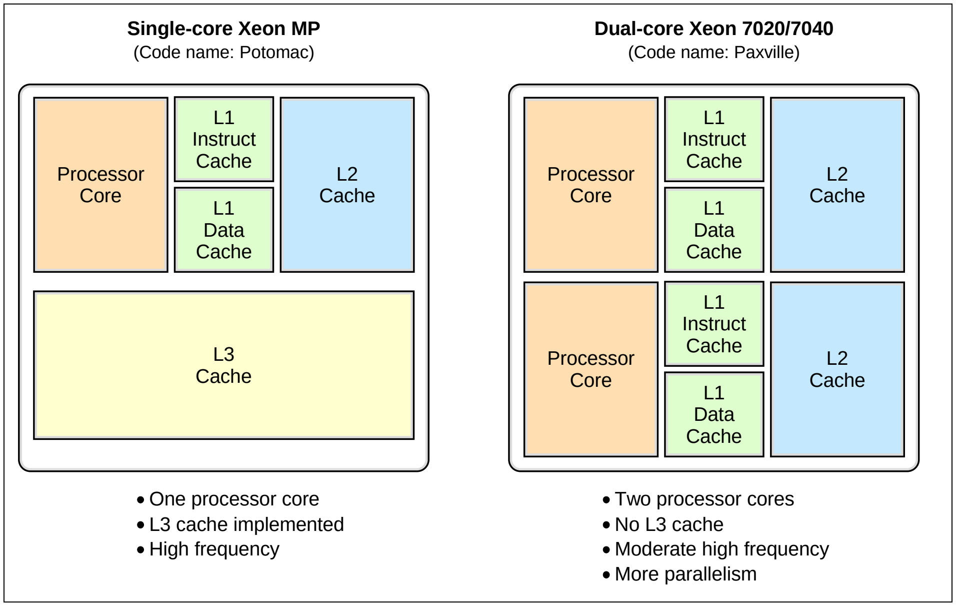


Figure 1-16 Features of single-core and dual-core processors

In addition to the two cores, the dual-core processor has separate L1 instruction and data caches for each core, as well as separate execution units (integer, floating point, and so on), registers, issue ports, and pipelines for each core. A dual-core processor achieves more parallelism than Hyper-Threading Technology, because these resources are not shared between the two cores. Estimates are that there is a 1.2 to 1.5 times improvement when comparing the dual-core Xeon MP with current single-core Xeon MP.

With double the number of cores for the same number of sockets, it is even more important that the memory subsystem is able to meet the demand for data throughput. The 21 GBps peak throughput of the X3 Architecture of the x3950 with four memory cards is well-suited to dual-core processors.

Two terms are relevant here:

- The term *core* refers to the portion of the processor chip that comprises of a processor’s execution engine, registers, segments, system bus interface, and internal cache.
- The term *socket* refers to the device on the system board of the server where the physical processor chip is inserted.

For example, an x3950 has four sockets, meaning that it can have up to four physical processors inserted. All of the x3950 models have dual-core processors, however the x460 which it replaced had models with either dual-core or single-core processors.

The use of the term *way* to describe the number of processors can be confusing because it can be interpreted as meaning cores or sockets. Throughout this redbook we are use the terms *n-CPU* system, *n-way* system or *n-socket* system to mean the same thing. That is, we consider the physical microprocessor package the CPU and not the cores available inside the package.

Another way to look at this is the number of threads that can simultaneously executed on a CPU. Original Intel CPUs were only able to execute a single thread at a time. When Intel introduced the NetBurst architecture they doubled some components of the CPU, thus making it appear as though the single CPU package were two different processors. This is Hyper-Threading. Specifically each logical CPU has its own architectural state, that is, its own data, segment and control registers and its own advanced programmable interrupt controller (APIC).

On the other hand the logical processors share the execution resources of the processor core, which include the execution engine, the caches, the system bus interface, and firmware. As a result of this the performance increase due to Hyper-Threading was not in the range of 2x (compared to that of using two different physical processors) but it is more like a maximum 1.3x (30%) improvement.

With dual-core CPUs, we are just making a step forward doubling all the core components on the die (that includes the execution resources). Again, there are a number of things that these cores share between each other. One is the path to memory. While two CPUs have their own system bus interface, dual-core CPUs share that connection, thus potentially limiting their bandwidth to memory and I/O devices.

Even more interesting is that while the current dual-core CPUs have dedicated on-board cache (that is, one for each core) future multi-core processors might end up sharing the cache. Table 1-3 on page 28 shows current and future CPU characteristics.

Table 1-3 Current and future CPU characteristics

	Architectural state	**Execution resources**	**Onboard caches**	**System bus i'face**	**Core frequency**	**Threads per CPU**
Legacy CPU	Exclusive	Exclusive	Exclusive	Exclusive	Highest	1
CPU with Hyper-Threading Technology	Exclusive	Shared	Shared	Shared	Highest	2
Dual-core CPU	Exclusive	Exclusive	Exclusive	Shared	High	4 (with HT)
Four-core CPU	Exclusive	Exclusive	Shared*	Shared	Medium-high*	8 (with HT)*
* because multi-core designs have not been officially announced these are speculations about their characteristics.						

As you can see from Figure 1-17, the notion that a core is a CPU is a bit misleading. The gain that you can achieve using a dual-core CPU is solely dependent on the workload being run:

- If the workload is CPU bound, then the dual-core will almost double the performance (despite the dual-core CPUs have slightly lower clock frequency).
- If the workload being run on these systems is memory-bound, then a dual-core CPU will not offer signifcant gains in performance because both cores share the same System Bus Interface.

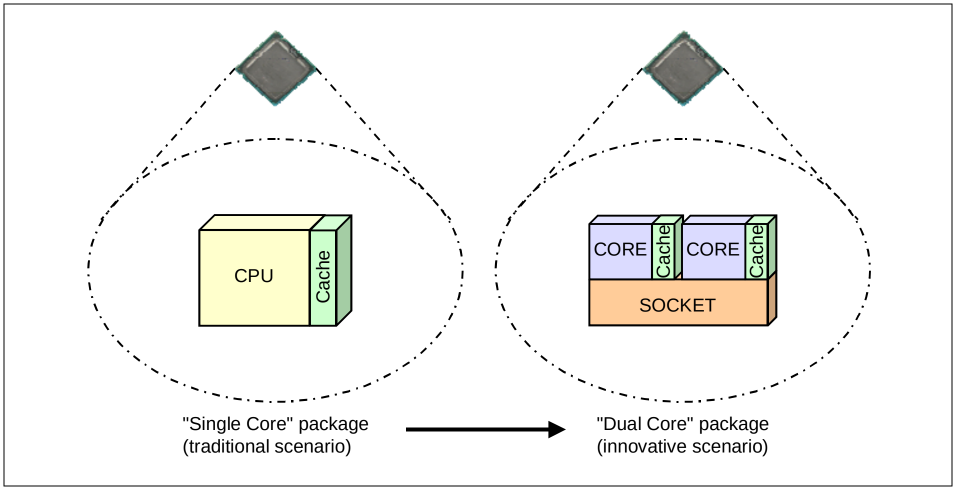


Figure 1-17 From single-core to dual-core CPUs

In reality the workloads are typically heterogeneous and account for a mix of CPU, Memory and I/O subsystem usage. In those real-life scenarios the gain that a dual-core CPU can achieve is on average about 50%.

And as we move forward towards multi-core CPUs we will likely see more shared components on the socket such as the various caches included in the processor package (Figure 1-18).

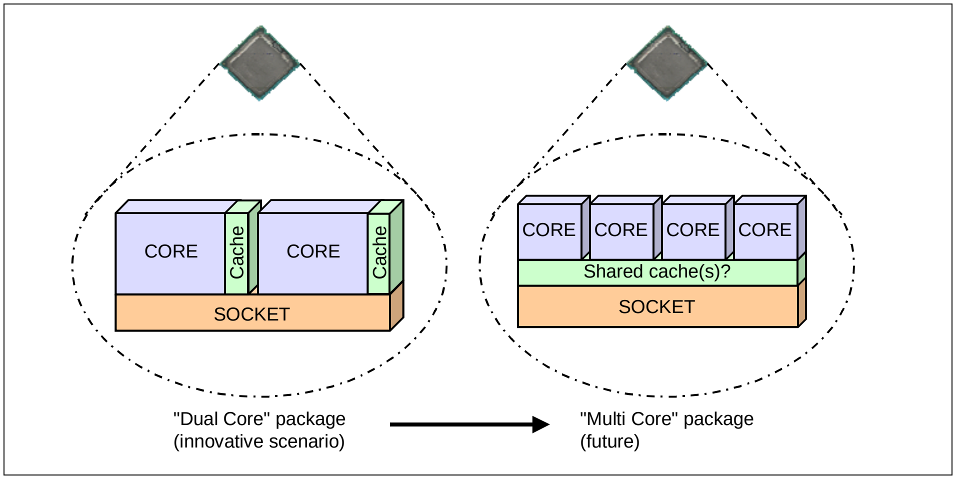


Figure 1-18 From dual-core CPUs to multi-core CPUs

Note: Multi-core CPU designs have not been formally announced.

With the move away from processor speed as a means to improve processor performance, CPU designers are looking to parallelism for innovation. The implementation of dual-core and quad-core processors means that more CPU operation can be performed per second. But to achieve maximum efficiency, this requires that the system software must also be redesigned to better support parallelism.

In the case of systems based on virtualization, it is the virtualization software that must be designed to take advantage of the multiple cores and the parallelism that this provides. Both ESX Server and Virtual Server are able to take advantage of the parallelism offered by dual-core processors.

- VMware ESX Server achieves parallelism by running multiple instances of the Virtual Machine Monitor (VMM), the VMkernel process that runs a VM. To run more virtual machines, additional VMM processes are launched.

- Microsoft Virtual Server is implemented as a single win32 service running on Windows Server 2003. Because there is only one process for all VMs, the service uses multi-threading to handle additional requests of the VMs.

Only applications that are designed to be multi-threaded are able to take advantage of multi-core processors. Those that have limited multi-threading capability will likely run faster on a single-core processor with a faster clock speed.

1.7 Virtualization futures

This section describes what support vendors are planning in the field of virtualization.

1.7.1 Intel servers and hardware virtualization support

Virtualization technologies have been around for years on mainframe systems. As virtualization is becoming more pervasive in the x86 space, vendors are enhancing their products so that they can better support the customer demand for virtualization.

Intel, for example, is working to create processors and supporting electronics that are virtualization-aware. Existing Intel CPUs work on the assumption that only one operating system is running on the system. Intel CPUs implement four ring levels, 0, 1, 2, and 3, where instructions are executed. A *ring* simply represents the privilege context in which the binary code is executed. Code that execute at lower rings is more privileged and is allowed to do more things than code that executes at higher ring levels.

For a number of reasons, only two levels are used in the x86 space. The operating system executes its core functions at ring 0 while applications (and higher level functions of the operating system) execute code at ring 3. The other levels are not used. This is shown in Figure 1-19 on page 31.

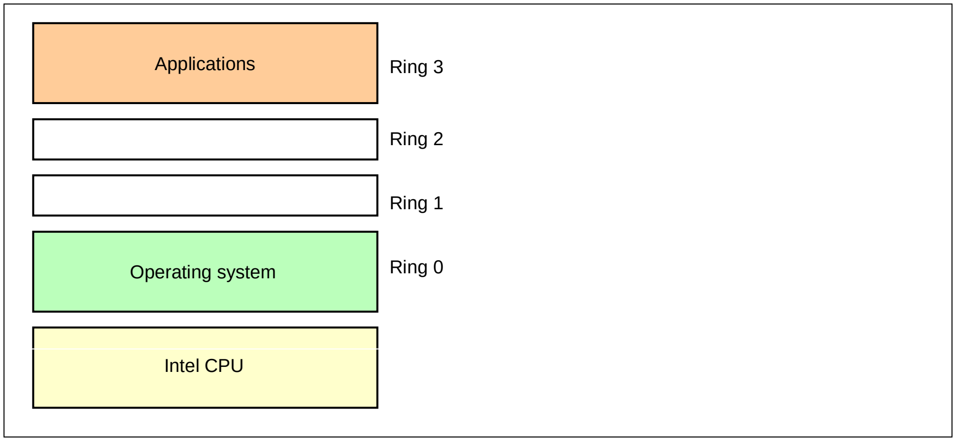


Figure 1-19 Traditional software stack without virtualization.

Because the operating system expects to run in Ring 0, virtualizing this stack is not easy because the virtualization layer must own the hardware, not the guest operating system. To accomplish this, *depriviledging techniques* are used to run the virtualization layer in the ring with the highest privilege, ring 0, lowering the privilege of the OS to either ring 1 or ring 3, depending on the deprivileging implementation chosen. This is shown in Figure 1-20.

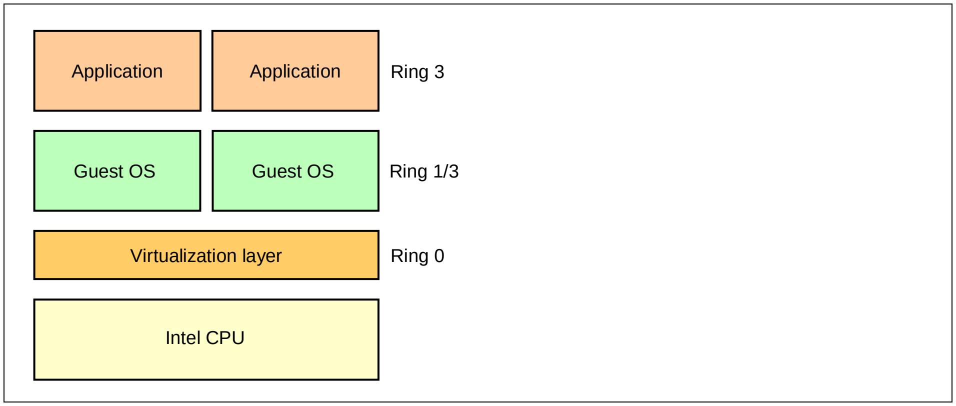


Figure 1-20 virtualization implementation on traditional x86 hardware

Note: To implement lowering the privilege level of the guest operating systems you can either use paravirtualization techniques or a full virtualization solution with deprivileging, or instruction translation occurring inside the virtualization layer. Refer to 1.2, “Virtualization concepts” on page 4 for more information.

Deprivileging techniques however have their own problems:

- With full virtualization the virtualization layer introduces some overhead for the translation.
- With paravirtualization you have to modify the guest operating systems and this introduces a number of concerns regarding compatibility and support.

Intel is addressing this problem introducing virtualization-aware processors that implement a new ring level of even higher privilege level. Virtualization software can then operate in this new level as shown in Figure 1-21. The matter is complex in all its details but the concept is that instead of deprivileging the guest OS to run in a different ring, you are super-privileging the virtualization software on a level below ring 0, in a ring that did not exist before.

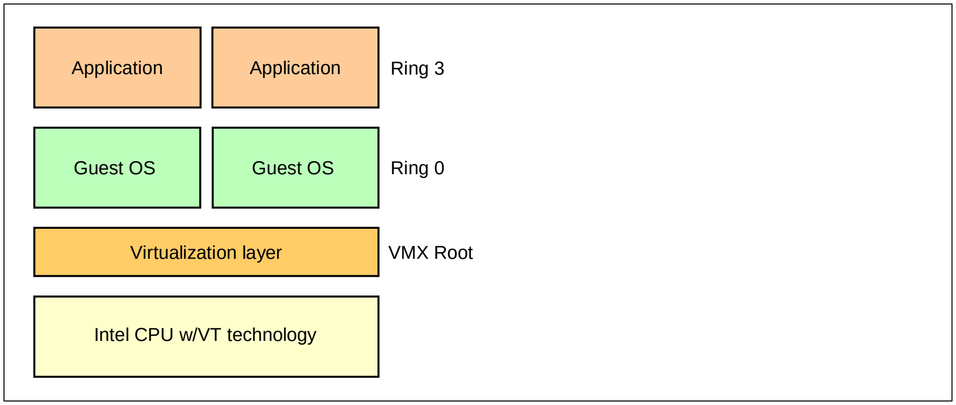


Figure 1-21 Virtualization with hardware assist

This new feature is called Intel Virtualization Technology (VT), earlier known by its codename *Vanderpool*. Intel VT is generally referred to as one of the *hardware assist* technologies that enable Intel based systems to become more virtualization-aware.

Note: The new announced Intel Xeon dual-core CPUs, code named *Paxville*, already ship with this feature but require a special BIOS release to enable it. The current plan is to release this BIOS update in the first half of 2006. Some server virtualization software products such as Xen 3.0 already supports this feature and both VMware and MSVS has announced they will introduce it soon.

It is important to understand that this is one of many steps to create systems that are fully virtualization aware. In this early stage, CPU hardware assists are more relevant but in the near future we will see this technology move to the memory and I/O subsystem as well.

1.7.2 What to expect from virtualization software in the future

Software vendors, specifically those developing the virtualization software layers, are working hard to improve their offering in terms of features and scenarios where their technology can be applied.

- Security

  One of the things that it would not be hard to imagine is that virtualization layers will build more advanced security features to protect the guest environments it runs. Features in future generations of virtualization products include intrusion-detection systems as well as a network interposition function to filter malicious code trying to traverse the virtualization layer to impact other virtual machines.

- Reliability

  On the front of high availability the potentials are even more surprising. Think about this: Because the virtualization layer is aware of every single instruction being executed on the hardware, you can simply instruct the hypervisor running one or more virtual machines to execute code in lock step on another physical systems.

  Should the primary system fail, the surviving system could continue to run the virtual machine without the already small downtime associated with common high-availability solutions such as VMware High Availability. This means you can potentially achieve true hardware fault tolerance.

As you can see today we are basically only scraping the surface of all the scenarios that are ahead of us running software in virtual environments.

2

The IBM System x3950 Server

The IBM System x3950 (formerly the xSeries 460) is based on IBM X3 Architecture, the third generation of IBM Enterprise X-Architecture™ technology. Delivering industry-leading x86 performance, XpandOnDemand scalability up to 32 processors, partitioning up to eight chassis, and the investment protection of 64-bit extensions and dual core CPU capability. X3 Architecture drives the x3950 to deliver the performance, availability, expandability, and manageability required for the next generation of industry-standard servers and an ideal platform for high performance and larger workload virtualization.

Topics in this chapter are:

2.1 Technical specifications

This section is a overview of the System x3950 server (Figure 2-1) specifications:

Figure 2-1 The IBM System x3950

The following are the key features of the x3950:

- Four-way capable server expandable up to 32-way by adding additional nodes
- IBM X3 Architecture featuring the XA-64e third-generation chipset
- Two standard Intel Xeon MP dual-core processors upgradable to four-way. These Processors support 64-bit addressing with the Intel Extended Memory 64 Technology (EM64T) architecture
- 2 GB memory standard expandable to 64 GB (using 4 GB DIMM slots), using high performance PC2-3200 ECC DDR2 DIMMs
- Active Memory with Memory ProteXion, memory mirroring, memory hot-swap and hot-add, and ChipKill
- Six full-length 64-bit 266 MHz PCI-X 2.0 Active PCI slots
- Additional nodes using with the x3950 E (or the x3950 E) modular expansion enclosure form a server with up to 32 processors, 512 GB of RAM, and 48 PCI-X slots
- Integrated Adaptec AIC-9410 serial-attached SCSI (SAS) controller

  Support for internal RAID arrays using an optional ServeRAID™-8i adapter. ServeRAID-6M is also supported for external SCSI storage with the EXP400 enclosure.
- Six internal hot-swap drive bays for internal storage

- Integrated Dual-port Broadcom 5704 PCI-X Gigabit Ethernet.
- Integrated Baseboard Management Controller and Remote Supervisor Adapter II SlimLine adapter are both standard.
- Support for the IBM Integrated xSeries Adapter for iSeries™ (IXA) for a direct high speed link to an iSeries server
- Three-year warranty on-site, nine hours per day, five days per week, with a next business day response.

The x3950 can form a multinode configuration by adding one or more x3950 E or MXE-460 modular expansion enclosures. The following configurations are possible:

- Eight-way configuration with two nodes, one x3950 and one x3950 E
- 16-way configuration with four nodes, one x3950 and three x3950 Es
- 32-way configuration with eight nodes, one x3950 and seven x3950 Es

Table 2-1 shows the major differences between the xSeries 460 and the x3950.

Table 2-1 Major differences between x460 and x3950

Feature	xSeries 460 server	System x3950 server
Dual-core processors	Single-core models are upgradable to support dual-core processors using the Dual Core X3 Upgrade Kit, 41Y5005. This kit does not include the dual-core CPUs	All models standard with dual-core processors
Processor/memory controller	Hurricane 2.1	Hurricane 3.0
Intel Virtualization Technology (VT)	Not supported	Supported
IPMI version supported in BMC	IPMI 1.5	IPMI 2.0

2.1.1 x3950 E

The x3950 E modular expansion enclosure (formerly the MXE-460) is a system used to extend an x3950 configuration. Like the x3950, the x3950 E contains microprocessors, memory, disks, and PCI-X adapters. However unlike the x3950, the x3950 E can only be used to *expand* an x3950 configuration.

The x3950 E is functionally identical to the x3950 and supports the same hardware options. The key differences between the x3950 and x3950 E are:

- The x3950 E is for expansion purposes and cannot be used as the primary node of a multi-node complex, nor can it be used as the primary node in a partition.

- The x3950 E does not have a DVD-ROM drive installed
- The x3950 E has no processors installed as standard
- The x3950 E has no memory installed (although two memory cards are installed)

Note: MXE-460 servers can be used interchangeably with the x3950 E provided all systems in the multi-node complex are at the latest firmware levels. A flyer that ships with the x3950 indicates the required minimum levels.

2.2 Dual-core processors and Hyper-Threading

The x3950 server has dual-core *Paxville* Xeon MP CPUs with Hyper-Threading support. Existing x460 server models have either dual-core or single-core processors standard.

2.2.1 Dual-core processors

The new *Paxville* dual-core processors are similar to a two-way system, except that the two processors, or *cores*, are integrated into one silicon die. This brings the benefits of two-way SMP with less power consumption and faster data throughput between the two cores. To keep power consumption down, the resulting core frequency is lower, but the additional processing capacity means an overall gain in performance.

With double the number of cores for the same number of sockets, it is even more important that the memory subsystem is able to meet the demand for data throughput. The 21 GBps peak throughput of the X3 Architecture of the x3950 with four memory cards is well suited to dual-core processors.

Figure 2-2 on page 39 compares the basic building blocks of the Xeon MP single-core processor (*Potomac*) and dual-core processor (*Paxville*).

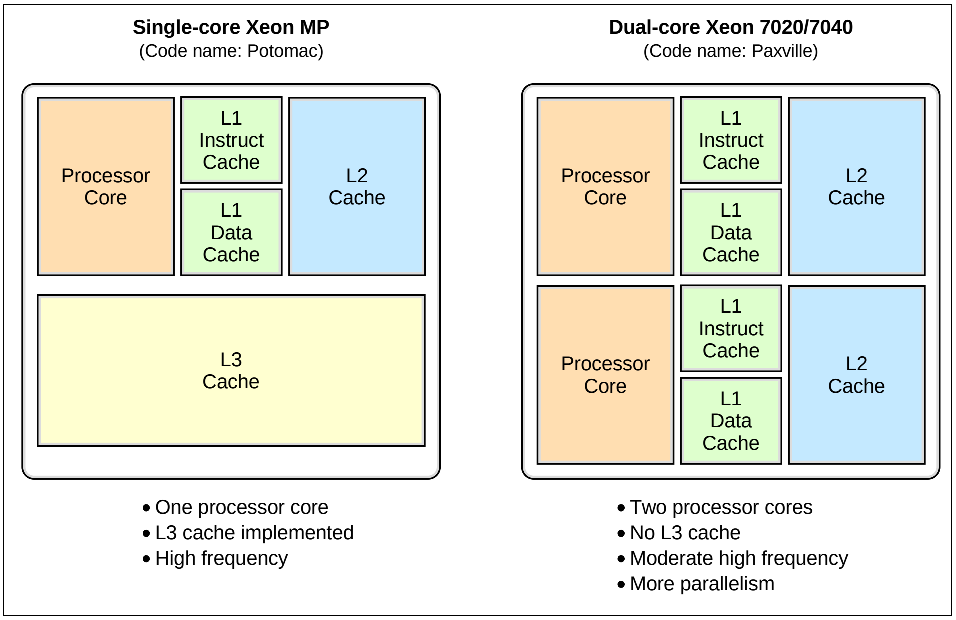


Figure 2-2 Feature of single core and dual core processors

2.2.2 Hyper-Threading

Hyper-Threading technology enables a single physical processor to execute two separate code streams, *threads*, concurrently. To the operating system, a processor with Hyper-Threading appears as two *logical* processors, each of which has its own architectural state: data, segment and control registers, and advanced programmable interrupt controller (APIC).

Each logical processor can be individually halted, interrupted, or directed to execute a specified thread, independently of the other logical processor on the chip. Unlike a traditional two-way SMP configuration that uses two separate physical processors, the logical processors share the execution resources of the processor core, which include the execution engine, the caches, the system bus interface, and the firmware.

The basic layout of a Hyper-Threading-enabled microprocessor is outlined in Figure 2-3 on page 40, where you can clearly see that only the components for the architectural state of the microprocessor have doubled.

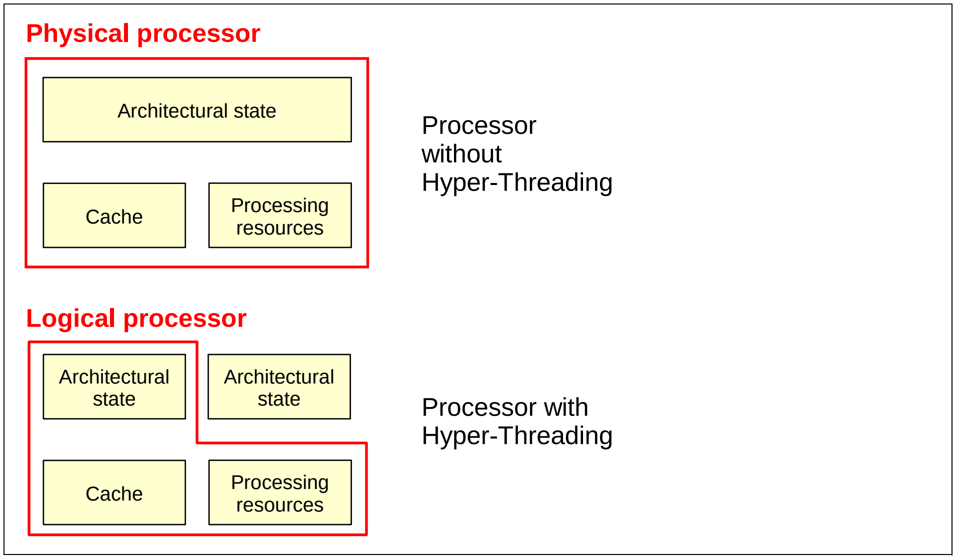


Figure 2-3 Architectural differences associated with Hyper-Threading

Hyper-Threading Technology is designed to improve server performance by exploiting the multithreading capability of operating systems, such as Microsoft Windows 2003, Linux, VMware ESX Server, and server applications, in such a way as to increase the use of the on-chip execution resources available on these processors.

Fewer or slower processors usually achieve the best gains from Hyper-Threading because there is a greater likelihood that the software can spawn sufficient numbers of threads to keep both paths busy. The following performance gains are likely:

- Two physical processors: up to about 25% performance gain
- Four physical processors: up to about 15% performance gain
- Eight physical processors: up to about 10% performance gain

2.3 X3 Architecture

The IBM X3 Architecture is the culmination of many years of research and development and has resulted in what is currently the fastest processor and memory controller in the Intel processor marketplace. With support for up to 32 Xeon MP processors and over 20 GBps of memory bandwidth per 64 GB of RAM up to a maximum of 512 GB, the xSeries servers that are based on the X3 Architecture offer maximum performance and broad scale-up capabilities.

For more detailed information about the X3 Architecture see the Redbook *Planning and Installing the IBM @server X3 Architecture Servers*, SG24-6797.

The x3950 use the third-generation IBM XA-64e chipset. The architecture consists of the following components:

- One to four Xeon MP processors
- One Hurricane Memory and I/O Controller (MIOC)
- Two Calgary PCI Bridges

Figure 2-4 shows a block diagram of the X3 Architecture

.

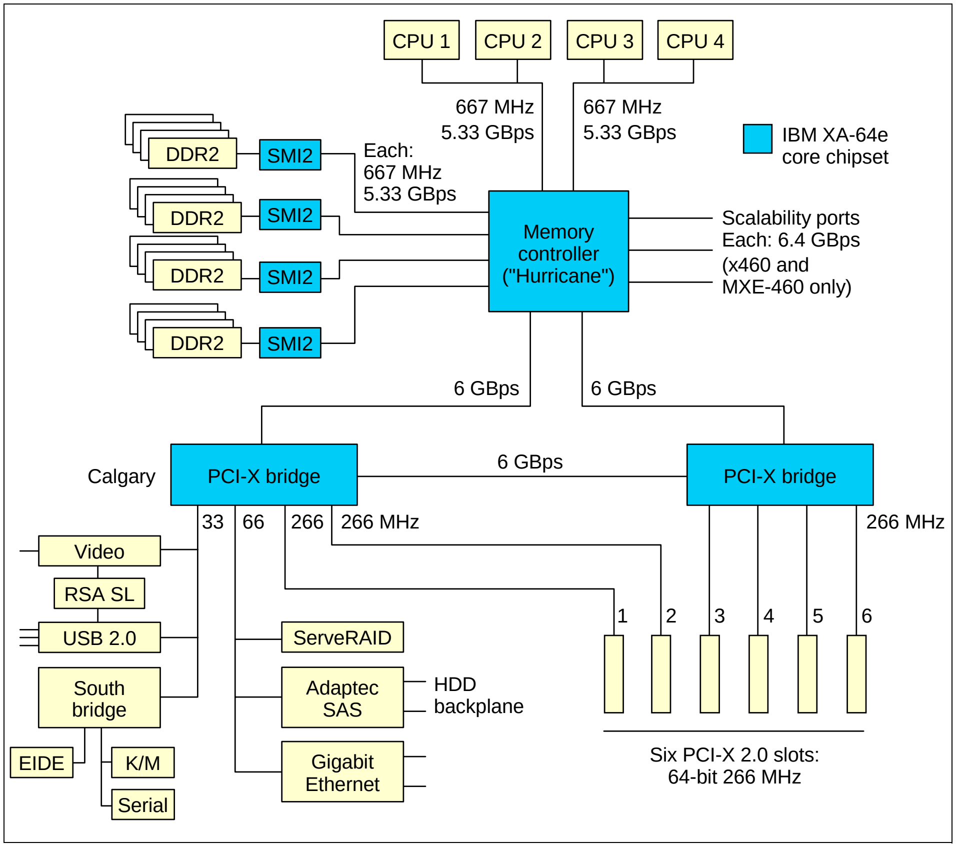


Figure 2-4 X3 Architecture system block diagram

Each memory port out of the memory controller has a peak throughput of 5.33 GBps. DIMMs are installed in matched pairs, two-way interleaving, to ensure the memory port is fully utilized. Peak throughput for each PC2-3200 DDR2 DIMM is 2.67 GBps. The DIMMs are run at 333 MHz to remain in sync with the throughput of the front-side bus.

Because there are four memory ports, spreading installed DIMMs across all four memory ports can improve performance. The four independent memory ports, or memory cards, provide simultaneous access to memory. With four memory cards installed, and DIMMs in each card, peak memory bandwidth is 21.33 GBps.

The memory controller routes all traffic from the four memory ports, two microprocessor ports, and the two PCI bridge ports. The memory controller also has embedded DRAM, which holds a snoop filter lookup table. This filter ensures that snoop requests for cache lines go to the appropriate microprocessor bus and not both of them, thereby improving performance.

One PCI bridge supplies four of the six 64-bit 266 MHz PCI-X slots on four independent PCI-X buses. The other PCI bridge supplies the other two PCI-X slots (also 64-bit, 266 MHz), plus all the onboard PCI devices, including the optional ServeRAID-8i and Remote Supervisor Adapter II SlimLine daughter cards.

2.3.1 X3 Architecture memory features

There are a number of advanced features implemented in the X3 Architecture memory subsystem, collectively known as *Active Memory*:

- Memory ProteXion

  The Memory ProteXion feature (also known as *redundant bit steering*) provides the equivalent of a hot-spare drive in a RAID array. It is based in the memory controller, and it enables the server to sense when a chip on a DIMM has failed and to route the data around the failed chip.

  Normally, 128 bits out of every 144 are used for data and the remaining 16 bits are used for ECC functions. However, X3 Architecture needs only 12 bits to perform the same ECC functions, thus leaving four bits free. These four bits are equivalent to an x4 memory chip on the DIMM that Memory ProteXion uses. In the event that a chip failure on the DIMM is detected by memory scrubbing, the memory controller can reroute data around that failed chip through these spare bits.

  It can do this automatically without issuing a Predictive Failure Analysis® (PFA) or light path diagnostics alert to the administrator, although an event is logged to the service processor log. After the second DIMM failure, PFA and light path diagnostics alerts occur on that DIMM as normal.

- Memory scrubbing

  Memory scrubbing is an automatic daily test of all the system memory that detects and reports memory errors that might be developing before they cause a server outage.

Memory scrubbing and Memory ProteXion work in conjunction with each other and do not require memory mirroring to be enabled to work properly.

When a bit error is detected, memory scrubbing determines if the error is recoverable or not. If it is recoverable, Memory ProteXion is enabled and the data that was stored in the damaged locations is rewritten to a new location. The error is then reported so that preventative maintenance can be performed. As long as there are enough good locations to allow the proper operation of the server, no further action is taken other than recording the error in the error logs.

If the error is not recoverable, then memory scrubbing sends an error message to the light path diagnostics, which then turns on the proper lights and LEDs to guide you to the damaged DIMM. If memory mirroring is enabled, then the mirrored copy of the data from the damaged DIMM is used until the system is powered down and the DIMM replaced.

- Memory mirroring

  Memory mirroring is roughly equivalent to RAID-1 in disk arrays, in that usable memory is halved and a second copy of data is written to the other half. If eight GB is installed, then the operating system sees four GB after memory mirroring is enabled. It is disabled in the BIOS by default. Because all mirroring activities are handled by the hardware, memory mirroring is operating system independent.

  When memory mirroring is enabled, certain restrictions exist with respect to placement and size of memory DIMMs and the placement and removal of memory cards.

- Chipkill™ memory

  Chipkill is integrated into the XA-64e chipset, so it does not require special Chipkill DIMMs and is transparent to the operating system. When combining Chipkill with Memory ProteXion and Active Memory, X3 Architecture provides very high reliability in the memory subsystem.

  When a memory chip failure occurs, Memory ProteXion transparently handles the rerouting of data around the failed component as described previously. However, if a further failure occurs, the Chipkill component in the memory controller reroutes data. The memory controller provides memory protection similar in concept to disk array striping with parity, writing the memory bits across multiple memory chips on the DIMM. The controller is able to reconstruct the missing bit from the failed chip and continue working as usual. One of these additional failures can be handled for each memory port for a total of four Chipkill recoveries.

- Hot-swap and hot-add memory

  The X3 Architecture servers support the replacing of failed DIMMs while the server is still running. This hot-swap support works in conjunction with memory mirroring. The server also supports adding additional memory while the server is running. Adding memory requires operating system support.

  These two features are mutually exclusive. Hot-add requires that memory mirroring be disabled and hot-swap requires that memory mirroring be enabled.

In addition, to maintain the highest levels of system availability, if a memory error is detected during POST or memory configuration, the server can disable the failing memory bank automatically and continue operating with reduced memory capacity. You can re-enable the memory bank manually after the problem is corrected by using the Setup menu in the BIOS.

Memory mirroring, Chipkill, and Memory ProteXion provide multiple levels of redundancy to the memory subsystem. Combining Chipkill with Memory ProteXion allows up to two memory chip failures for each memory port on the system, for a total of eight failures sustained.

1. The first failure detected by the Chipkill algorithm on each port does not generate a light path diagnostics error because Memory ProteXion recovers from the problem automatically.
2. Each memory port could then sustain a second chip failure without shutting down.
3. Provided that memory mirroring is enabled, the third chip failure on that port would send the alert and take the DIMM offline, but keep the system running out of the redundant memory bank.

2.3.2 XceL4v cache

The XceL4v dynamic server cache serves two purposes in the x3950:

- As a single 4-way server the XceL4v and its embedded DRAM (eDRAM) is used as a snoop filter to reduce traffic on the front side bus. It stores a directory of all processor cache lines to minimize snoop traffic on the dual front side buses and minimize cache misses.
- When the x3950 is configured as a multi-node server, this technology dynamically allocates 256 MB of main memory in each node for use as an L4 cache directory and scalability directory. In a 32-way configuration, this means there will be 2 GB of XceL4v cache.

With advances in chip design, IBM has now reduced the latency of main memory to below that of the XceL4 cache in the x445. The time it takes to access data directly from memory is almost as fast as accessing it from L3. As a result, on a four-way system there is little/no need for either a L3 cache or L4 cache.

The next section does into greater detail on the benefits of the XceL4v cache on the x3950 server.

2.4 NUMA Architecture

Nonuniform Memory Access (NUMA) is an architecture designed to improve performance and solve latency problems inherent in large SMP systems with more than four processors. The x3950 implements a NUMA-based architecture and can scale up to 32 CPUs using multiple x3950 or x3950 E nodes.

All SMP systems must perform processor-to-processor communication, known as *snooping*, to ensure all processors receive the most recent copy of requested data. Because any processor can store data in a local cache and modify that data at any time, all processor data requests must first be sent to every other process in the system so that each processor can determine if a more recent copy of the requested data is in that processor cache.

Snooping traffic is an important factor affecting performance and scaling for all SMP systems. The overhead of this communication becomes greater with an increase in the number of processors in a system. Also, faster processors result in a greater percentage of time spent performing snooping because the speed of the communications does not improve as the processor clock speed increases, because latency is largely determined by the speed of the front-side bus.

It is easy to see that increasing the number of processors and using faster processors results in greater communication overhead and memory controller bottlenecks. But unlike traditional SMP designs, which send every request from every processor to all other processors, greatly increasing snooping traffic, the x3950 has a more optimal design. The XceL4v cache in the x3950 improves performance because it filters most snooping operations.

The IBM XceL4v cache improves scalability with more than four processors because it also caches remote data addresses. Before any processor request is sent across the scalability bus to a remote processor, the memory controller and cache controller determine whether the request should be sent at all. To do this, the XceL4v dynamic server cache keeps a directory of all the addresses of all data stored in all remote processor caches. By checking this directory first, the XceL4v can determine if a data request must be sent to a remote processor and only send the request to that specific node where the processor caching the requested data is located.

The majority of data requests are not found in the XceL4 cache and can be sent directly to the memory controller to perform memory look-up without any remote processor-to-processor communication overhead. The directory-based coherency protocol used by the XceL4 cache greatly improves scalability of the x3950 because processor-to-processor traffic is greatly reduced.

The term NUMA is not completely correct because not only memory can be accessed in a non-uniform manner but also I/O resources. PCI-X and USB devices may also be associated with nodes. The exception to this are earlier I/O devices such as diskette and CD-ROM drives which are disabled because the classic PC architecture precludes multiple copies of these traditional items.

The key to this type of memory configuration is to limit the number of processors that directly access a piece memory, thereby improving performance because of the much shorter queue of requests. The objective of the operating system is to ensure that memory requests are fulfilled by local memory whenever possible.

However, an application running on CPUs in node 1 can still need to access memory physically located in node 2 (a remote access). This access incurs longer latency because the travel time to access remote memory on another expansion module is clearly greater. to reduce unnecessary remote access, the x3950 maintain a table of data in the firmware called the Static Resource Allocation Table (SRAT). The data in this table is accessible by operating systems such as Windows Server 2003 (Windows 2000 Server does not support it), ESX Server, and current Linux kernels.

These modern operating systems attempt to allocate resources that are local to the processors being used by each process. So when a process and its threads start on node 1, all execution and memory access will be local to node 1. As more processes are added to the system, the operating system will balance them across the nodes. In this case, most memory accesses will be evenly distributed across the multiple memory controllers, reducing remote access, greatly reducing queuing delays, and improving performance.

2.5 Multi-node configurations

IBM *XpandOnDemand* scalability features give the flexibility to expand the x3950 server's capacity in terms of number of CPUs, memory and I/O slots, as the demand grows.

Customers can expand the server:

- CPUs from two-way up to 32-way
- Memory from 2 GB to 512 GB
- PCI-X slots from 6 to 48

This scalability is achieved by connecting multiple x3950s or x3950 Es to the base x3950. These *nodes* then form a single complex. The supported expansion steps are listed in Table 2-2.

> **Note:** MXE-460 servers can be used interchangeably with the x3950 E provided all systems in the multi-node complex are at the latest firmware levels. A flyer that ships with the x3950 indicates the required minimum levels.

An x3950 server can be configured together with one, three or seven x3950 Es to form a single 8-way, 16-way or 32-way complex.

Table 2-2 x3950 scalability options

Nodes	CPUs	Maximum RAM	PCI slots	Number of x3950 Es*
1	2-way	64 GB	6	None
1	4-way	64 GB	6	None
2	8-way	128	12	1
4	16-way	256 GB	24	3
8	32-way	512 GB	48	7
* Additional nodes can be either x3950 E or x3950 servers				

You can also form multinode complexes using multiple x3950s or combinations of x3950s and x3950 Es. With these combinations, you can partition the complex as described in 2.5.1, “Partitioning” on page 50.

As shown in Figure 2-5 on page 48, a scalable system comprises an x3950 server and 1, 3 or 7 x3950 E systems.

A fully configured, eight-node, scalable system has 32 processors, 512 GB of memory (using 4 GB DIMMs), 48 PCI-X 2.0 adapters, 3.5 TB of disk space (non-RAID) and 16 Gigabit Ethernet connections.

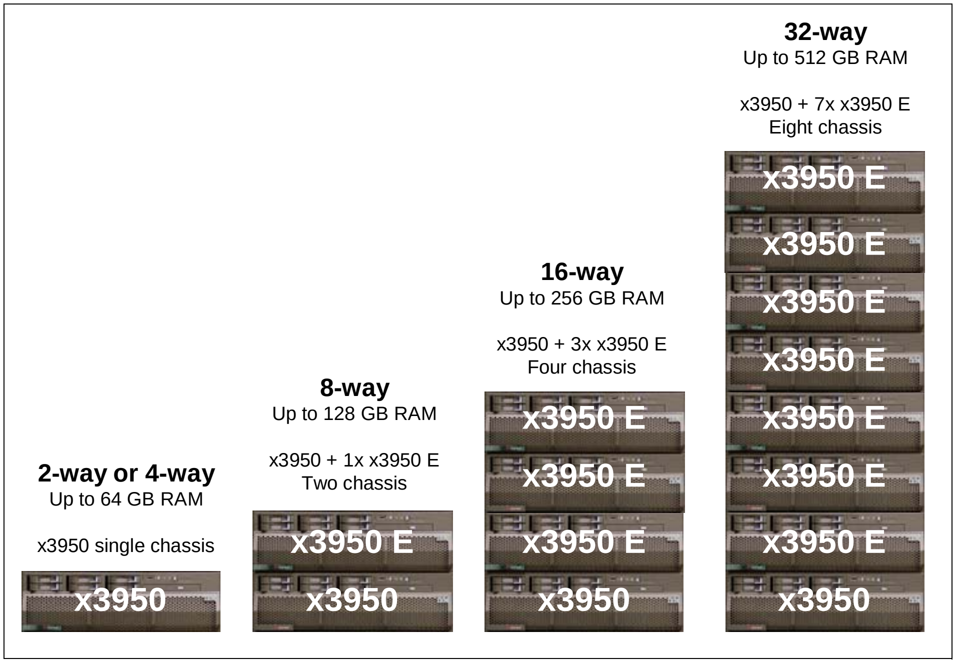


Figure 2-5 The four multi-node configurations supported

The following configuration rules apply:

- Multinode configurations

  The multi-node configurations can have more than one x3950. Only one is shown in Figure 2-5 for simplicity. In fact, if you wish to use partitioning as described in 2.5.1, “Partitioning” on page 50, you will need one x3950 for each partition to be the primary node.

  The first x3950 is known as the primary node; all other nodes in a complex are called secondary nodes. The primary node must always be an x3950 (x3950 Es cannot be primary nodes).

  Only 1, 2, 4, or 8 nodes are supported. Other combinations cannot be selected in the configuration utility.

- Combinations of x460 and x3950

  It is supported to have multi-node complexes that are comprised of x460 and MXE-460 servers along with x3950 and x3950 E servers. The only requirement is that all nodes must be at certain code levels (BIOS, RSA II firmware, BMC firmware, and so on). A Technical Update bulletin enclosed with the x3950 and x3950 E lists the required minimum levels.

- Processors

  The x3950 and all x3950 Es must each have four processors installed and all processors must be the same speed and cache size. The x3950 and x3950 E are technically capable of have less than four CPUs installed in a multi-node configuration, but this type of configuration requires SPORE approval before it can be supported.

- Memory

  For performance reasons, you should have the same amount of memory in each node. A minimum of 2 GB of RAM is required in each node.

  In a multi-node configuration, 256 MB of RAM per node is allocated to the XceL4v cache. In an 8-node 32-way complex, this means that 2 GB of RAM is allocated to XceL4v and is unavailable to the operating system.

- 32-core and 64-core configurations

  Due to export regulations, partitions with 32 single-core processors or 16 or 32 dual-core processors are prevented from being configured unless a special memory card is installed, part 41Y5004. See *Planning and Installing the IBM @server X3 Architecture Servers*, SG24-6797 for details of this card.

- Firmware

  All system firmware, including the system BIOS, diagnostics, BMC firmware and RSA II SlimLine firmware, must be at the same level across all systems.

  Updating the system BIOS in every node in a scalable system can be performed from the primary node. The BIOS code in all secondary nodes will automatically be updated as well. The server diagnostics, as well as the BMC and RSA II SlimLine firmware must be individually updated on each node, but this can be performed remotely. The RSA II firmware can be updated using the RSA II Web interface or IBM Director. The BMC firmware can be updated with an RSA II remote console session using the remote diskette function.

- Disk drives installed in any of the x3950 Es are seen by the operating system as normal disk drives.

- If you install the optional ServeRAID-8i RAID adapter into each node, you will be able to form RAID arrays, but these arrays cannot span nodes. All drives in each array must be local to the RAID adapter.

- All PCI-X slots and onboard Gigabit Ethernet ports in the x3950 E are visible to the operating system as well.

- The concept of forming a single system image from multiple nodes is called *merging* the nodes. When a multi-node system is started and the nodes are merged, all existing devices on the secondary nodes (COM ports, keyboard, video, mouse and DVD-ROM drives) are disabled. PCI devices on all nodes (including the onboard Gigabit Ethernet and USB controllers) available to the operating system.

- Power control is as follows:
  - Pressing the power button of any node in a complex powers on or off the entire complex.
  - Pressing the reset button on any node in the complex restarts the entire complex.

2.5.1 Partitioning

As discussed in 2.5, “Multi-node configurations” on page 46, the complex can be configured as one scalable partition with two, four or eight nodes. Alternatively it is also possible to split this complex into multiple independent partitions. For example, an eight-node configuration can be split into two 4-node systems by changing the configuration without changes to the cabling.

The decision whether partitioning is required or not must be made during the planning stage of a multi-node system, because the primary node in a multi-node complex always must be an x3950. It is not supported to configure multiple partition on a complex that consists of one x3950 with one, three or seven x3950 Es attached. You must have one x3950 as the primary node for every partition you create.

See 2.6.3, “Partitioning example” on page 63 for an example of how to partition a 4-node complex into two separate 2-node partitions.

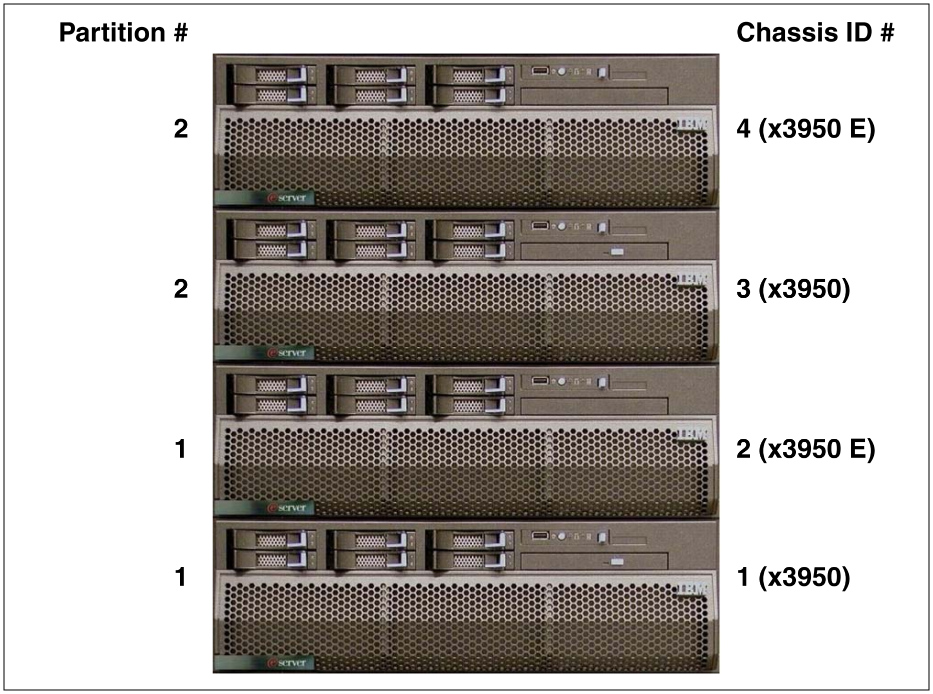


Figure 2-6 Four-node complex split into two partitions

2.5.2 Cabling for multi-node configurations

In order to create a multi-node complex, the servers and expansion modules are inter-connected using a high-speed data bus known as the *scalability bus*. The connection uses copper colored *scalability cables*.

> **Note:** These cables are not compatible with the x440 and x445 equivalent cables.

To build your multinode complex, you must order the appropriate number of each type of scalability cable for your specific configuration, as in Table 2-3. The x3950 and x3950 E do not come with scalability cables.

Table 2-3 Scalability cable configuration table

Chassis configuration	2.3m (short) cables needed	2.9m (long) cables needed
4-way: 1-chassis	None	None

Chassis configuration	2.3m (short) cables needed	2.9m (long) cables needed
8-way: 2 chassis	2	None
16-way: 4 chassis	6	None
32-way: 8 chassis	8	4

There are three different cabling schemes, depending on the number of nodes used in the complex. These are shown in the following diagrams. In a 2-node configuration, two scalability cables are used to connect both chassis. The second cable provides redundancy for the chassis interconnect as well as a slight performance benefit.

In any multi-node configuration any *one* scalability cable can fail without impact on the server's operation. In this situation, a warning LED on the light path diagnostic panel will be lit and an event will be logged into the RSA event log as shown in Figure 2-7.

```
SMI reporting Scalability Event:Double Wide Link Down.Chassis Number = 1. Port Number = 0.
SMI reporting Scalability Event:Link Down.Chassis Number = 1. Port Number = 1.
SMI reporting Scalability Event:Double Wide Link Down.Chassis Number = 2. Port Number = 0.
SMI reporting Scalability Event:Link Down.Chassis Number = 2. Port Number = 1.
```

Figure 2-7 Messages in the RSA II event log when a scalability cable fails

Figure 2-8 depicts the scalability cabling plan for a 2-node / 8-way configuration.

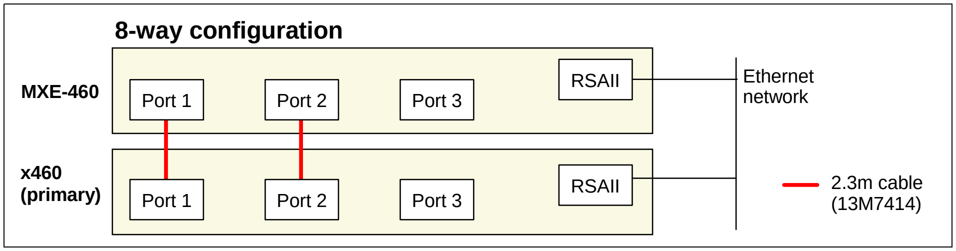


Figure 2-8 Cabling for a two-node configuration

Figure 2-9 on page 53 depicts the scalability cabling plan for a 4-node / 16-way configuration. This uses just the short (2.3 m) cables.

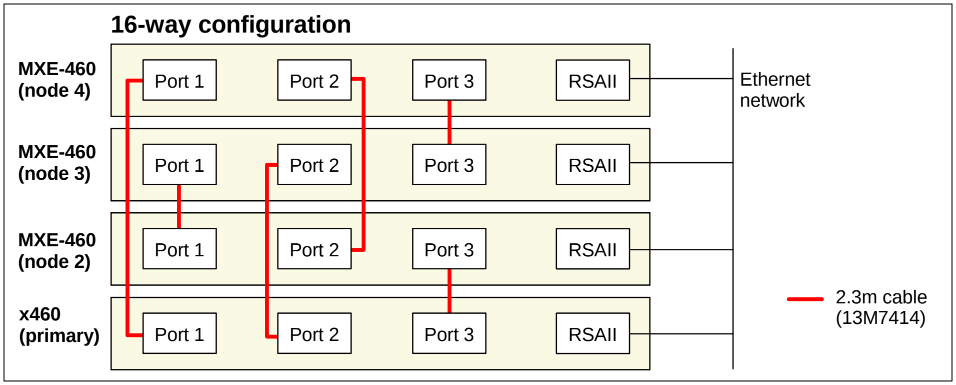


Figure 2-9 Cabling for a four-node configuration

Figure 2-10 depicts the scalability cabling plan for an 8-node / 32-way configuration. This one uses a combination of short (2.3 m) and long (2.9 m) cables.

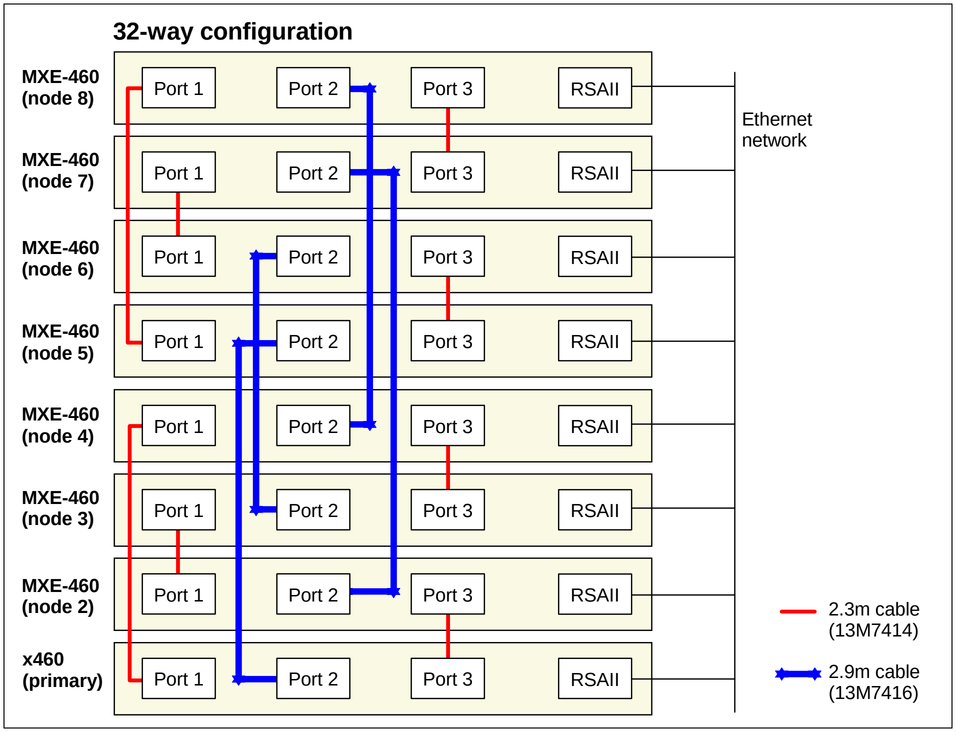


Figure 2-10 Cabling for a eight-node configuration

All Port 2 connectors use the longer 2.9m cable because they are further apart. One exception to this is the Port 2 cable between nodes 3 and 6. The recommendation is to use the longer cable so you can insert a KVM or other device in the middle of the complex. The choice of cable is purely a matter of physical connectivity. It does not affect signaling or performance.

2.6 Scalable system setup

The setup of a multi-node complex will be configured through the Web interface of the RSA II SlimLine adapter, which is standard with every x3950 and x3950 E.

The multi-node configuration data is stored on the RSA in all chassis that are members of the scalable partition. Figure 2-12 on page 57 depicts the RSA Web interface for the scalable partition configuration.

2.6.1 RSA II SlimLine setup

By default, the RSA II SlimLine is configured to obtain an IP address using DHCP and if that is not available, then to configure a static IP address of 192.168.70.125.

To configure the RSA II SlimLine to use another static address, you can use one of two methods:

- Access the adapter from a client computer, for example using a notebook computer connected to the RSA with a crossover Ethernet cable. Open a Web browser and point it to the adapter’s IP address (192.168.70.125).

  > **Tip:** The default logon credentials are `USERID` and `PASSW0RD`. All characters are uppercase, the 0 in `PASSW0RD` is a zero and not the letter O).

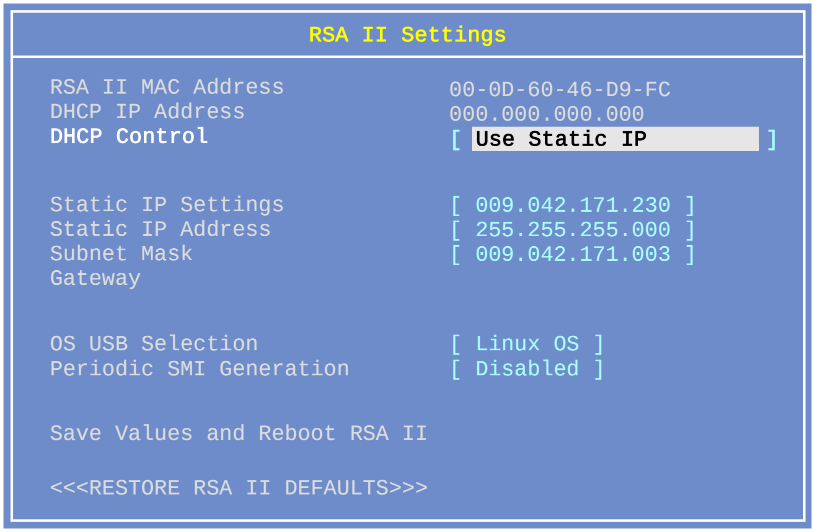
- Configure the adapter from the server’s BIOS Setup/Configuration:

  a. Press F1 during system startup when prompted to enter the Configuration/Setup utility.

  b. Choose **Advanced Setup** → **RSA II Settings.**

  c. Select **Use Static IP** under DHCP Control and enter a static IP address, subnet mask and gateway in the appropriate fields as required as shown in Figure 2-11 on page 55.

  d. Select **Save Values and Reboot RSA II** to make the configured network settings active.



```
                           RSA II Settings

  RSA II MAC Address                     00-0D-60-46-D9-FC
  DHCP IP Address                        000.000.000.000
  DHCP Control                           [ Use Static IP      ]

  Static IP Settings                     [ 009.042.171.230 ]
  Static IP Address                      [ 255.255.255.000 ]
  Subnet Mask                            [ 009.042.171.003 ]
  Gateway

  OS USB Selection                       [ Linux OS ]
  Periodic SMI Generation                [ Disabled ]

  Save Values and Reboot RSA II

  <<<RESTORE RSA II DEFAULTS>>>
```

Figure 2-11 RSA setup within Configuration/Setup utility

Alternatively, an IP address also can be assigned using DHCP by selecting **DHCP Enabled** in the field DHCP Control. If DHCP is used, we recommend you use an IP reservation. The IP address assigned by the DHCP server is also displayed in this window in the field **DHCP IP Address**.

The RSA’s MAC address is displayed as **RSA II MAC Address** which can be helpful for network switch configuration.

In the **OS USB Selection** field, select the appropriate value for your operating system. This setting determines how the RSA presents emulated keyboard and mouse in a remote control session to the operating system.

- For Microsoft Windows, select **Other OS.**
- For VMware ESX Server select **Linux OS.**

The **Periodic SMI Generation** setting is set to **Disabled** by default and should not be changed on the X3 Architecture servers. This feature was intended for support of older operating systems that did not include adequate checking of CPU states. Modern operating systems poll for CPU machine checks without this feature. No function is lost by disabling it.

The communication between the RSA adapters is handed through the systems management Ethernet connections. Therefore it is very important to ensure secure and reliable network connections. We recommend you connect all RSAs to a separate management network, which is exclusively used for management and not shared with the production or campus LAN.

2.6.2 Create a two node partition

For proper operation it is necessary to maintain continuous connectivity between the RSAs. As a result, we recommend you assign static IP addresses to the RSA II cards. If you are using DHCP, you should configure address reservations to ensure the addresses never change.

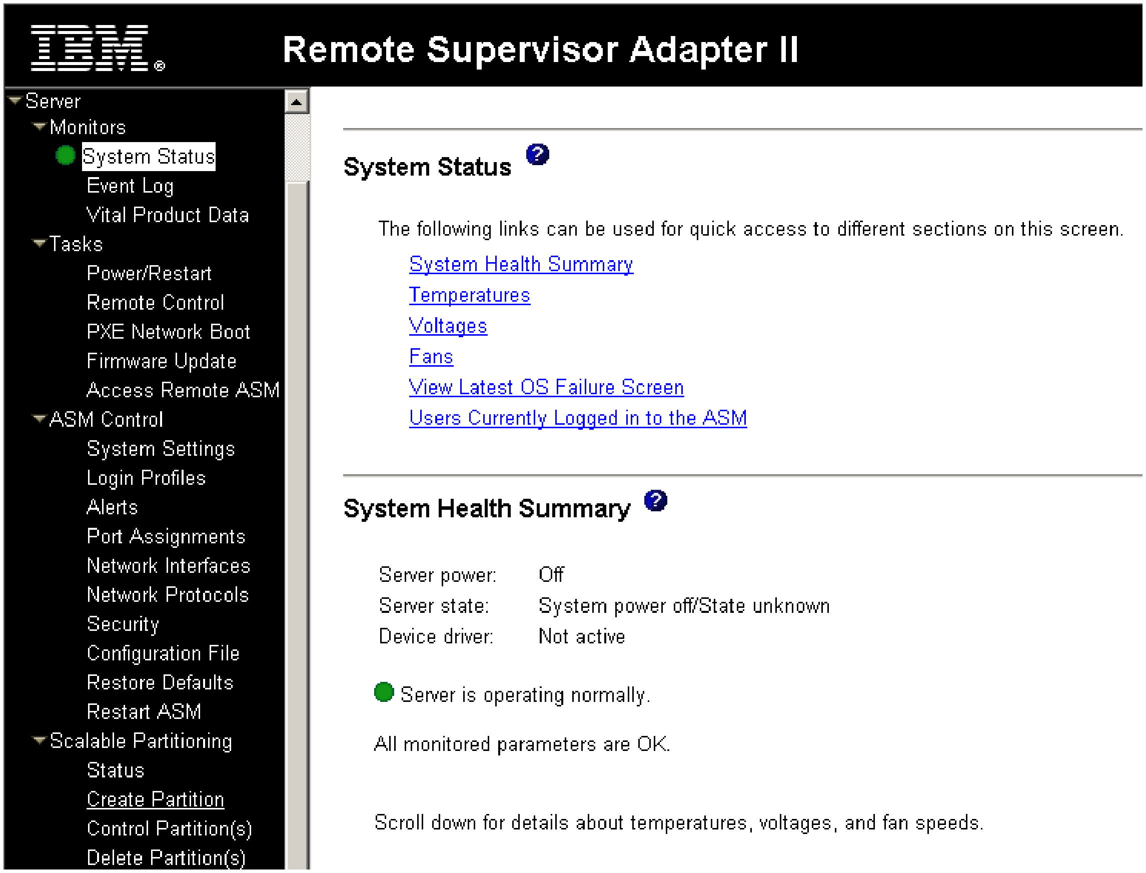
2.6.1, “RSA II SlimLine setup” on page 54 describes how to prepare the RSA II adapter.

To set up a scalable system, the following prerequisites must be fulfilled:

- The firmware on all nodes must be at the same level. This includes system BIOS, diagnostics, BMC firmware, and RSA firmware.
- The settings made in the Configuration/Setup utility for memory and CPUs must be identical.
- During the configuration with the RSA II Web interface, all participating nodes must be powered off, but connected to AC power.
- The operating system must reside on the primary node. This means either internal SAS hard disk drives or host bus adapters (HBAs) for SAN boot must be installed in the primary node.
- You can only create partitions out of set combinations of nodes. If you select any other combination, you get an error message when you try to create the configuration. Valid node combinations are:
  - Nodes 0 and 1
  - Nodes 2 and 3
  - Nodes 4 and 5
  - Nodes 6 and 7
  - Nodes 0, 1, 2, and 3
  - Nodes 4, 5, 6, and 7

To create a scalable partition and then activate it, do the following:

1. Ensure all nodes are powered off but that they are connected to AC power.
2. Login to the RSA of the server that will be the primary node in the complex. Open a Web browser and enter the Ethernet address of the primary node’s RSA II adapter. See Figure 2-12 on page 57.



System Status

The following links can be used for quick access to different sections on this screen.

System Health Summary
Temperatures
Voltages
Fans
View Latest OS Failure Screen
Users Currently Logged in to the ASM

System Health Summary

Server power:	Off
Server state:	System power off/State unknown
Device driver:	Not active

● Server is operating normally.

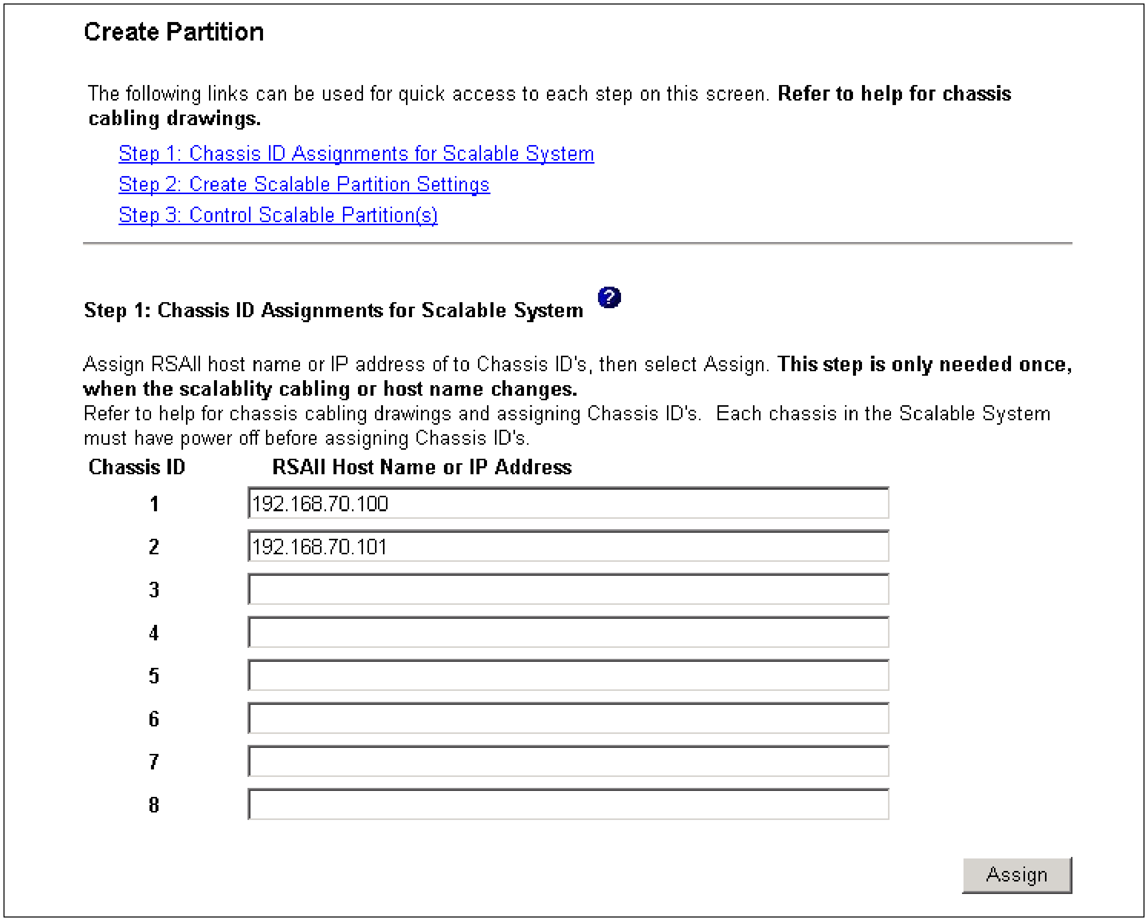
All monitored parameters are OK.

Scroll down for details about temperatures, voltages, and fan speeds.

Figure 2-12 RSA II Web browser

> **Note:** Scalable partitions can be created only using the primary node's RSA interface. It is not possible to create, delete, start or stop a scalable partition from a secondary node's RSA interface.

3. In the RSA main menu click **Scalable Partitioning** → **Create Partition**. Figure 2-13 on page 58 opens.



Create Partition

The following links can be used for quick access to each step on this screen. **Refer to help for chassis cabling drawings.**

Step 1: Chassis ID Assignments for Scalable System
Step 2: Create Scalable Partition Settings
Step 3: Control Scalable Partition(s)

Step 1: Chassis ID Assignments for Scalable System

Assign RSAII host name or IP address of to Chassis ID's, then select Assign. **This step is only needed once, when the scalablity cabling or host name changes.**
Refer to help for chassis cabling drawings and assigning Chassis ID's. Each chassis in the Scalable System must have power off before assigning Chassis ID's.

Chassis ID	RSAII Host Name or IP Address
1	192.168.70.100
2	192.168.70.101
3	
4	
5	
6	
7	
8	

Figure 2-13 Assign chassis IDs

4. Click **Step 1: Chassis ID Assignments for Scalable System** or scroll down to that section.

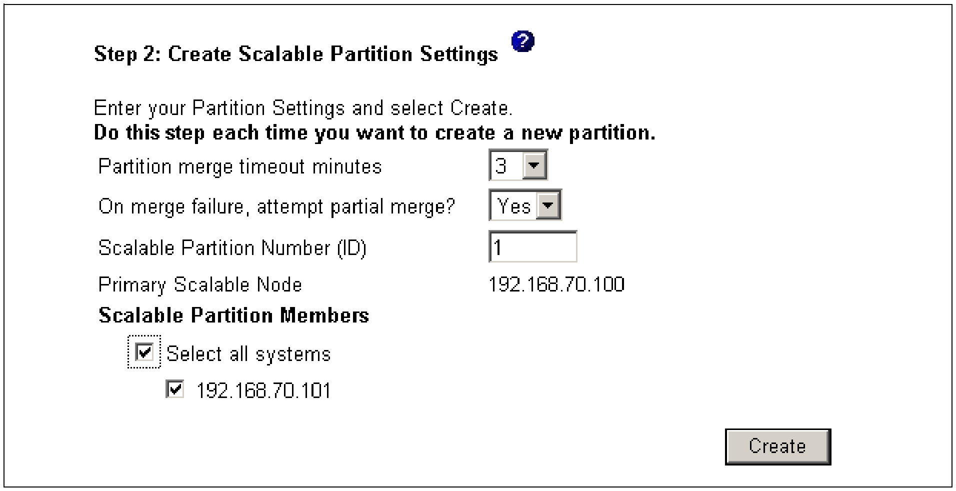
> **Note:** Step 4 (**Step 1: Chassis ID Assignments for Scalable System)** should only be performed when the cabling is first done or when it is changed. Do not click **Assign** if you are simply adding more partitions to an existing configuration.

5. Enter the IP address or host name for each node to be part of the Scalable System and click **Assign**. See Figure 2-13. If you intend to use host names instead of IP addresses for the RSA adapters, note that only fully qualified

domain names (FQDN) may be used, for example `mxe460rsa.itso.ral.ibm.com`.

Note: In order to assign the IDs, the systems all must be powered off. An error pop-up window opens if a chassis is not powered off. To delete a configuration, just assign new IDs and click the Assign button.

6. Click **Step 2: Create Scalable Partition Settings** section in the RSA menu. Figure 2-14 opens.



Step 2: Create Scalable Partition Settings

Enter your Partition Settings and select Create.
Do this step each time you want to create a new partition.

Partition merge timeout minutes: 3

On merge failure, attempt partial merge?: Yes

Scalable Partition Number (ID): 1

Primary Scalable Node: 192.168.70.100

Scalable Partition Members

☑ Select all systems

☑ 192.168.70.101

Figure 2-14 Create scalable partition settings

7. Enter the following parameters:

   – Partition Merge timeout minutes: Use this field to specify the minutes for BIOS/POST to wait for merging scalable nodes to complete. Allow at least 8 seconds for each GB of memory in the scalable partition. Merge status for POST/BIOS can be viewed on the monitor attached to the primary node.

     The default value is **5 minutes**, we do not recommend setting it lower than **3 minutes**.

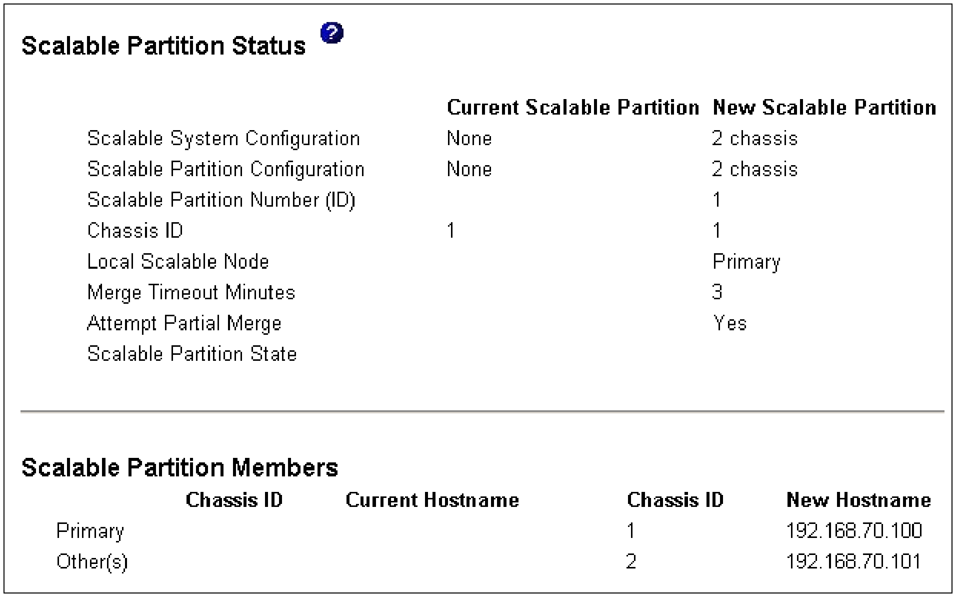
   – On merge failure, attempt partial merge?: Use this field to specify **Yes** or **No** if BIOS/POST should attempt a partial merge if an error is detected during full merge in a 4 or 8 node configuration. If your application is sensitive to load order or full scalable partition support, you might not want a partial merge. See “Node failover and recovery” on page 70 for more information about node failures.

- Scalable Partition Number (ID): Use this field to specify the scalable partition ID for this scalable partition, which must be unique within the scalable system. We recommend you use something simple such as **1**, **2**, **3**, and so forth.
- Primary Scalable Node: This is the primary node for the partition. because configuring the partition is always done on the primary node, the IP address listed is that of the node you are connected to and cannot be changed.

  Ensure you have a bootable hard drive attached to the primary node.
- Scalable Partition Members: This list is the scalable system members assigned in Step 1: Chassis ID Assignment. The primary node is always part of the partition and is not listed.

  Select the secondary nodes you want to use to form a partition with the primary node from which you are working from. Remember that partitions can be 2, 4 or 8 nodes only (in other words, you must select 1, 3 or 7 systems from the list under Scalable Partition Members, because the primary node is already preselected.

8. When you have selected your Scalable Partition Members, click **Create**.
9. Check the status by clicking **Scalable Partitioning** → **Status**. You can see a screen similar to Figure 2-15 on page 61.



Scalable Partition Status

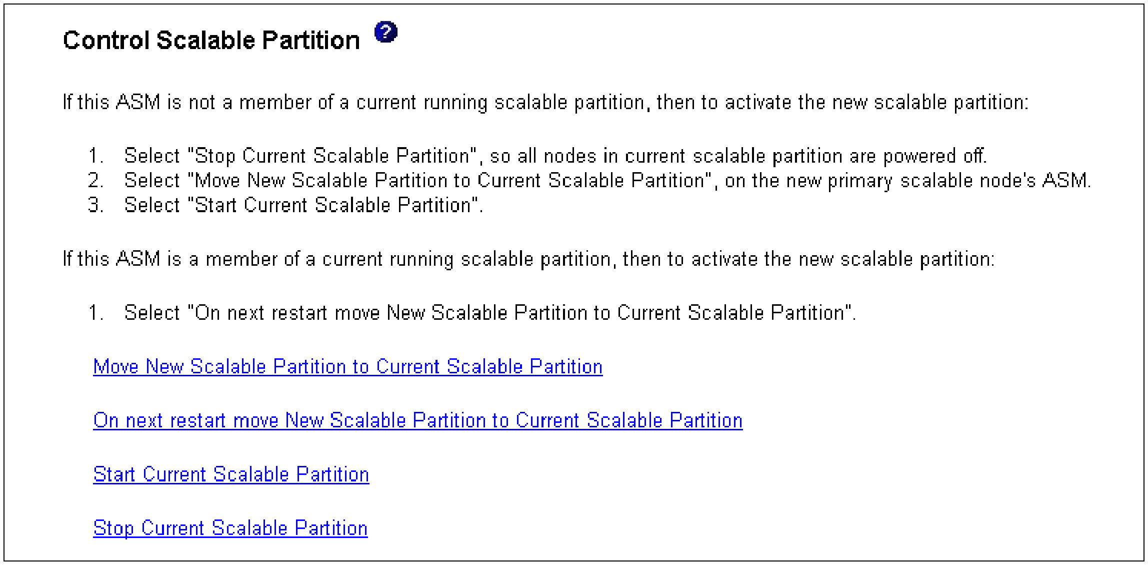
	Current Scalable Partition	New Scalable Partition
Scalable System Configuration	None	2 chassis
Scalable Partition Configuration	None	2 chassis
Scalable Partition Number (ID)		1
Chassis ID	1	1
Local Scalable Node		Primary
Merge Timeout Minutes		3
Attempt Partial Merge		Yes
Scalable Partition State		

Scalable Partition Members

	Chassis ID	Current Hostname	Chassis ID	New Hostname
Primary			1	192.168.70.100
Other(s)			2	192.168.70.101

Figure 2-15 Scalable partition status

10.Before using the partition, you must move the new partition to be the current partition. To move your new partition to the current partition click **Scalable Partitioning** → **Control Partition(s)** and then click **Move New Scalable Partition to Current Scalable Partition** as shown in Figure 2-16 on page 62.



Control Scalable Partition

If this ASM is not a member of a current running scalable partition, then to activate the new scalable partition:

1. Select "Stop Current Scalable Partition", so all nodes in current scalable partition are powered off.
2. Select "Move New Scalable Partition to Current Scalable Partition", on the new primary scalable node's ASM.
3. Select "Start Current Scalable Partition".

If this ASM is a member of a current running scalable partition, then to activate the new scalable partition:

1. Select "On next restart move New Scalable Partition to Current Scalable Partition".

Move New Scalable Partition to Current Scalable Partition

On next restart move New Scalable Partition to Current Scalable Partition

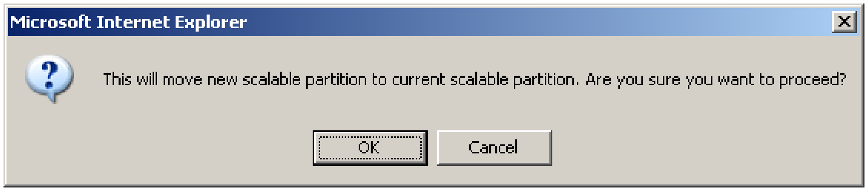
Start Current Scalable Partition

Stop Current Scalable Partition

Figure 2-16 Control Scalable Partition

> **Note:** If you have a current partition running and want to schedule the new partition to become active at a latter time you can select the **On Next Restart Move New Scalable Partition to Current Scalable Partition** and schedule a reboot to take place at the time you want the new partition to become the current partition.

11. Click **OK** the dialog box shown in Figure 2-17. This is your last chance to cancel. After you click **OK** any current running partition will be replaced with the new partition. Each partition can only have one current partition and one new partition.

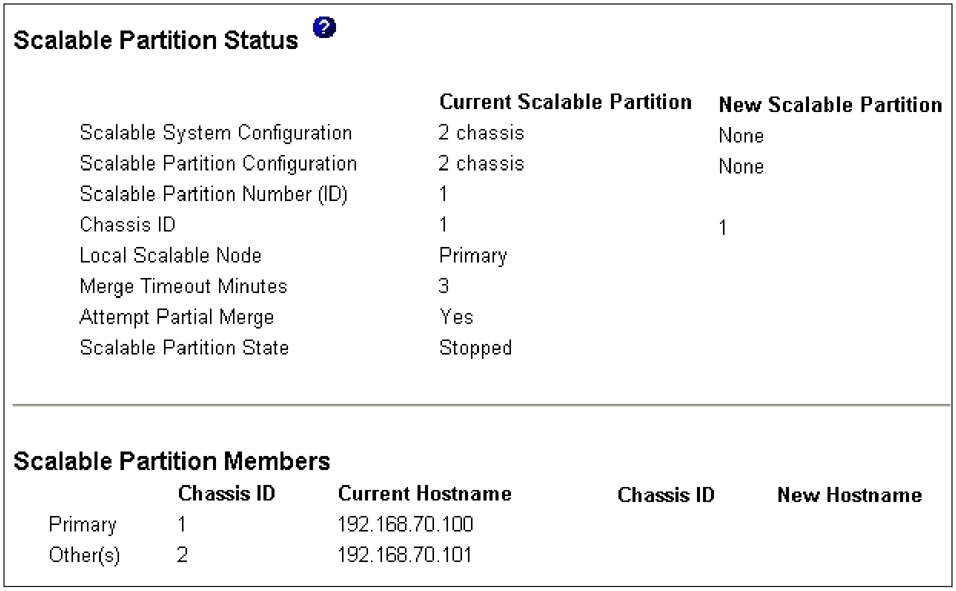


Microsoft Internet Explorer

This will move new scalable partition to current scalable partition. Are you sure you want to proceed?

Figure 2-17 Move scalable partition

12. Check your status again by clicking **Scalable Partitioning** → **Status**. You can see your partition has moved to Current Scalable Partition as shown in Figure 2-18 on page 63.



Scalable Partition Status

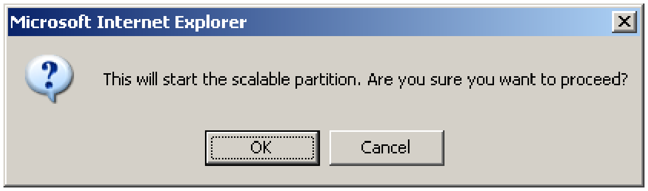
	Current Scalable Partition	New Scalable Partition
Scalable System Configuration	2 chassis	None
Scalable Partition Configuration	2 chassis	None
Scalable Partition Number (ID)	1	
Chassis ID	1	1
Local Scalable Node	Primary	
Merge Timeout Minutes	3	
Attempt Partial Merge	Yes	
Scalable Partition State	Stopped	

Scalable Partition Members

	Chassis ID	Current Hostname	Chassis ID	New Hostname
Primary	1	192.168.70.100		
Other(s)	2	192.168.70.101		

Figure 2-18 Current Scalable Partition Status

13. To start the current partition click **Scalable Partitioning** → **Control Partition(s)** → **Start Current Scalable Partition**. This powers on all servers in the partition. Click **OK** in the dialog box shown in Figure 2-19.



Microsoft Internet Explorer

This will start the scalable partition. Are you sure you want to proceed?

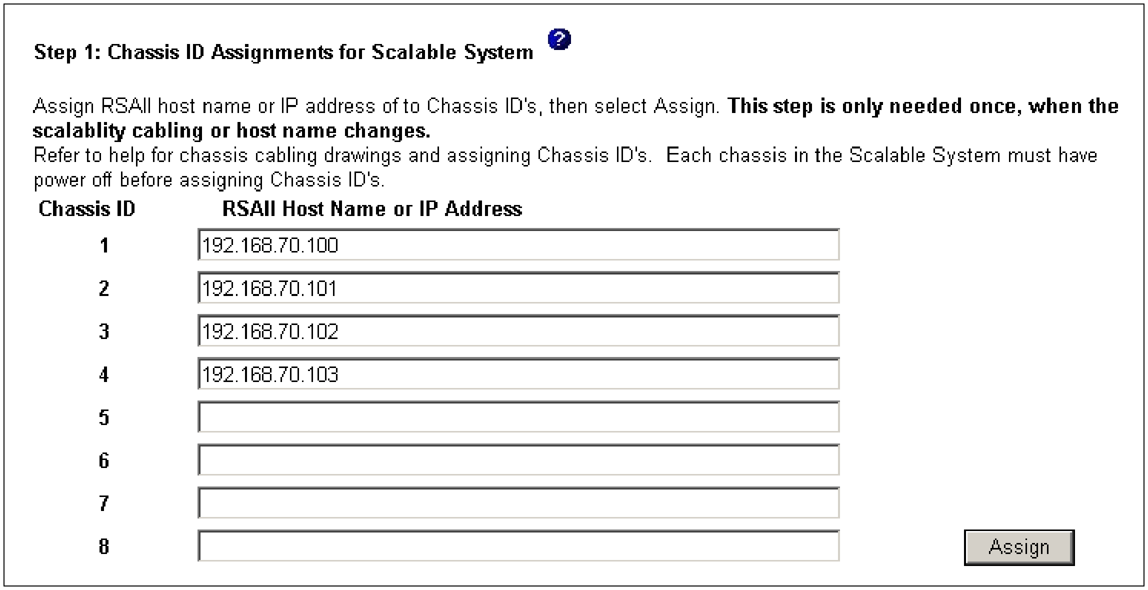
Figure 2-19 Start scalable partition

2.6.3 Partitioning example

The steps to configure multiple server partitions are described in 2.6, “Scalable system setup” on page 54. As an example, to configure two 2-node partitions in a four-node complex, do the following:

1. Logon to the RSA Web GUI of chassis 1.
2. In the RSA main menu click **Scalable Partitioning** → **Create Partition**.
3. Click **Step 1: Chassis ID Assignments for Scalable System** or scroll down to that section.

4. Enter the IP Addresses or FQDN of all four systems as shown in Figure 2-20 and click **Assign**. This is to create a complex of four nodes. Then we will create two separate two-node partitions.



Step 1: Chassis ID Assignments for Scalable System

Assign RSAII host name or IP address of to Chassis ID's, then select Assign. **This step is only needed once, when the scalablity cabling or host name changes.**
Refer to help for chassis cabling drawings and assigning Chassis ID's. Each chassis in the Scalable System must have power off before assigning Chassis ID's.

Chassis ID	RSAII Host Name or IP Address
1	192.168.70.100
2	192.168.70.101
3	192.168.70.102
4	192.168.70.103
5	
6	
7	
8	

Figure 2-20 Create scalable system with four chassis

5. Create a new partition with chassis 2 as secondary node.

   a. Click **Step 2: Create Scalable Partition Settings** section in the RSA menu.

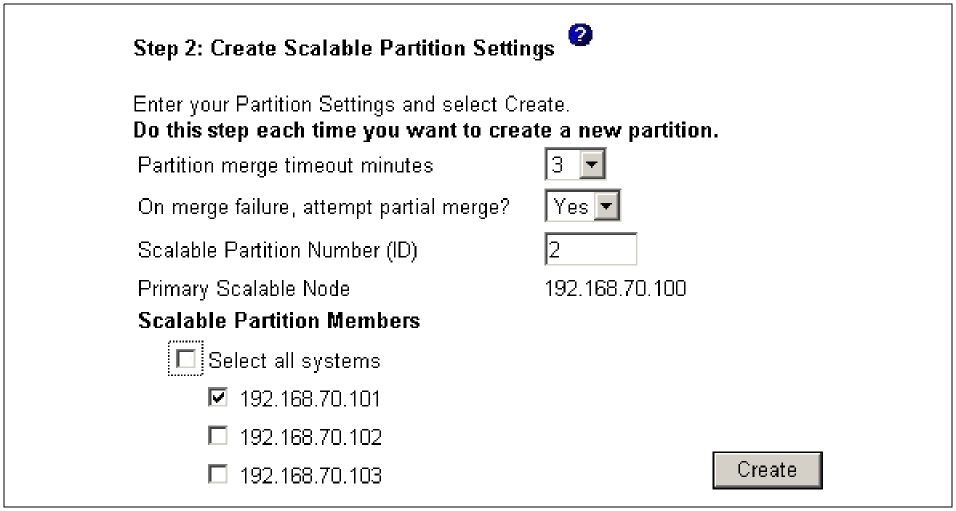
   b. Enter the following parameters:

      - Partition Merge timeout minutes: Use this field to specify the minutes for BIOS/POST to wait for merging scalable nodes to complete. Allow at least 8 seconds for each GB of memory in the scalable partition. Merge status for POST/BIOS can be viewed on the monitor attached to the primary node.
      - The default value is **5 minutes**. We do not recommend setting it lower than **3 minutes**.
      - On merge failure, attempt partial merge?: Use this field to specify **Yes** or **No** if BIOS/POST should attempt a partial merge if an error is detected during full merge in a 4 or 8 node configuration. If your application is sensitive to load order or full scalable partition support you may not want a partial merge. See “Node failover and recovery” on page 70 for more information about node failures.

- Scalable Partition Number (ID): Use this field to specify the scalable partition ID for this scalable partition, which must be unique within the scalable system. We recommend you use something simple like **1, 2, 3,** and so forth.
- Primary Scalable Node: This is the primary node for the partition. because configuring the partition is always done on the primary node, the IP address listed is that of the node you are connected to and cannot be changed.
- Scalable Partition Members: This list is the scalable system members assigned in Step 1: Chassis ID Assignment. The primary node is always part of the partition and is not listed.

Select the secondary node you wish to use to form a partition with the primary node you are working from. In our example here is it **192.168.70.101**.

6. When you have entered all the fields, click **Create.** See Figure 2-21.



Step 2: Create Scalable Partition Settings

Enter your Partition Settings and select Create.
Do this step each time you want to create a new partition.

Partition merge timeout minutes	3
On merge failure, attempt partial merge?	Yes
Scalable Partition Number (ID)	2
Primary Scalable Node	192.168.70.100

Scalable Partition Members

☐ Select all systems
☑ 192.168.70.101
☐ 192.168.70.102
☐ 192.168.70.103

Figure 2-21 Select partition members

7. Click **Scalable Partitioning** → **Control Partition(s)** and then click **Move the new Scalable partition configuration to Current Partition configuration**.
8. To start the current partition click **Scalable Partitioning** → **Control Partition(s**) then click **Start Current Scalable Partition**, see Figure 2-16 on page 62. This will power on both servers in the partition. Click **OK** in the dialog box shown in Figure 2-19 on page 63.
9. Now we create the second partition. Logon to the RSA Web GUI of chassis 3. In our example it is **192.168.70.102**.

10. Create a new partition with chassis four as secondary node. This is the same process as above, except with chassis 3 (**192.168.70.102**) as the primary node and chassis 4 (**192.168.70.103**) as the secondary node.
11. To move your new partition to the current partition click **Scalable Partitioning** → **Control Partition(s)** then click **Move New Scalable Partition to Current Scalable Partition.** See Figure 2-16 on page 62.
12. To start the current partition click **Scalable Partitioning** → **Control Partition(s**) then click **Start Current Scalable Partition**, see Figure 2-16 on page 62. This will power on both servers in the partition. Click **OK** in the dialog box shown in Figure 2-19 on page 63.

The two partitions now behave like two independent 8-way servers. Each partition can be controlled independently with the RSA in each primary node. This means, they can powered on and off without any impact on each other. For example, powering off partition 1 does not affect partiton 2. Though they are still wired as a 4-node complex, there will be no data transmitted between the partitions.

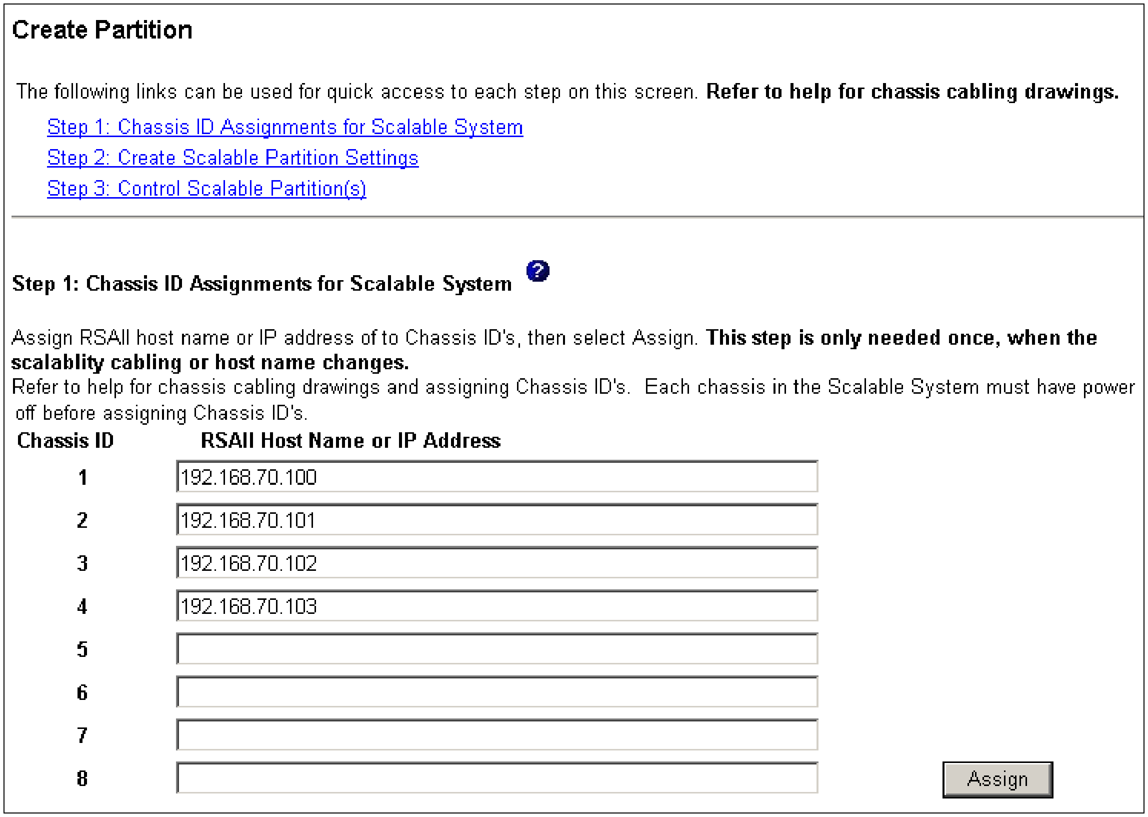
2.6.4 Adding additional nodes

If you already have a multi-node configuration and want to add additional nodes the process is similar to 2.6.2, “Create a two node partition” on page 56.

The following is an example of how to add two additional nodes to an existing two node system.

> **Tip:** The key concept is that you are not actually adding two nodes, but instead creating a new four node partition.

1. Shut down the existing system and cable all four nodes as illustrated in Figure 2-9 on page 53.
2. Logon to the RSA Web GUI of chassis 1
3. In the RSA main menu click **Scalable Partitioning** → **Create Partition.** Enter the IP address or host name for all four nodes to be part of the Scalable System and click **Assign**. See Figure 2-22 on page 67.



Create Partition

The following links can be used for quick access to each step on this screen. **Refer to help for chassis cabling drawings.**

Step 1: Chassis ID Assignments for Scalable System
Step 2: Create Scalable Partition Settings
Step 3: Control Scalable Partition(s)

Step 1: Chassis ID Assignments for Scalable System

Assign RSAII host name or IP address of to Chassis ID's, then select Assign. **This step is only needed once, when the scalablity cabling or host name changes.**
Refer to help for chassis cabling drawings and assigning Chassis ID's. Each chassis in the Scalable System must have power off before assigning Chassis ID's.

Chassis ID	RSAII Host Name or IP Address
1	192.168.70.100
2	192.168.70.101
3	192.168.70.102
4	192.168.70.103
5	
6	
7	
8	

Figure 2-22 Four node chassis ID assignments

4. Click **Step 2: Create Scalable Partition Settings** section in the RSA menu.
5. Enter the following parameters:
   - Partition Merge timeout minutes: Use this field to specify the minutes for BIOS/POST to wait for merging scalable nodes to complete. Allow at least 8 seconds for each GB of memory in the scalable partition. Merge status for POST/BIOS can be viewed on the monitor attached to the primary node.

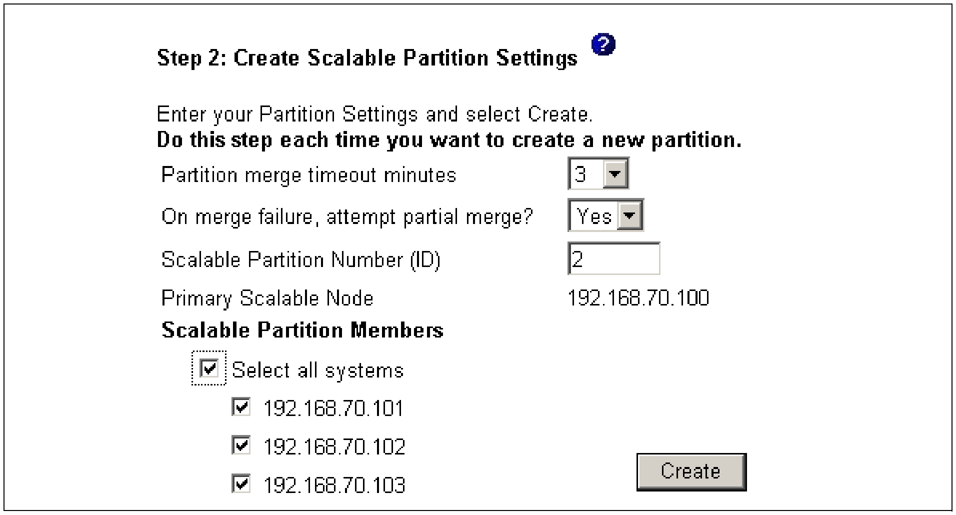
     The default value is **5 minutes**, we do not recommend setting it lower than **3 minutes**.
   - On merge failure, attempt partial merge?: Use this field to specify **Yes** or **No** if BIOS/POST should attempt a partial merge if an error is detected during full merge in a 4 or 8 node configuration. If your application is sensitive to load order or full scalable partition support, you might not want

a partial merge. See “Node failover and recovery” on page 70 for more information about node failures.

- Scalable Partition Number (ID): Use this field to specify the scalable partition ID for this scalable partition, which must be unique within the scalable system. We recommend you use something simple such as **1, 2, 3**, and so forth.
- Primary Scalable Node: This is the primary node for the partition. because configuring the partition is always done on the primary node, the IP address listed is that of the node you are connected to and cannot be changed.

  Ensure you have a bootable hard drive attached to the primary node.
- Scalable Partition Members: This list is the scalable system members assigned in Step 1: Chassis ID Assignment. The primary node is always part of the partition and is not listed.

  Select the secondary nodes you want to use to form a partition with the primary node from which you are working. Remember that partitions can be 2, 4 or 8 nodes only (in other words, you must select 1, 3 or 7 systems from the list under Scalable Partition Members, because the primary node is already preselected. In our example, click **select all systems** as show in Figure 2-23.



Step 2: Create Scalable Partition Settings

Enter your Partition Settings and select Create.
Do this step each time you want to create a new partition.

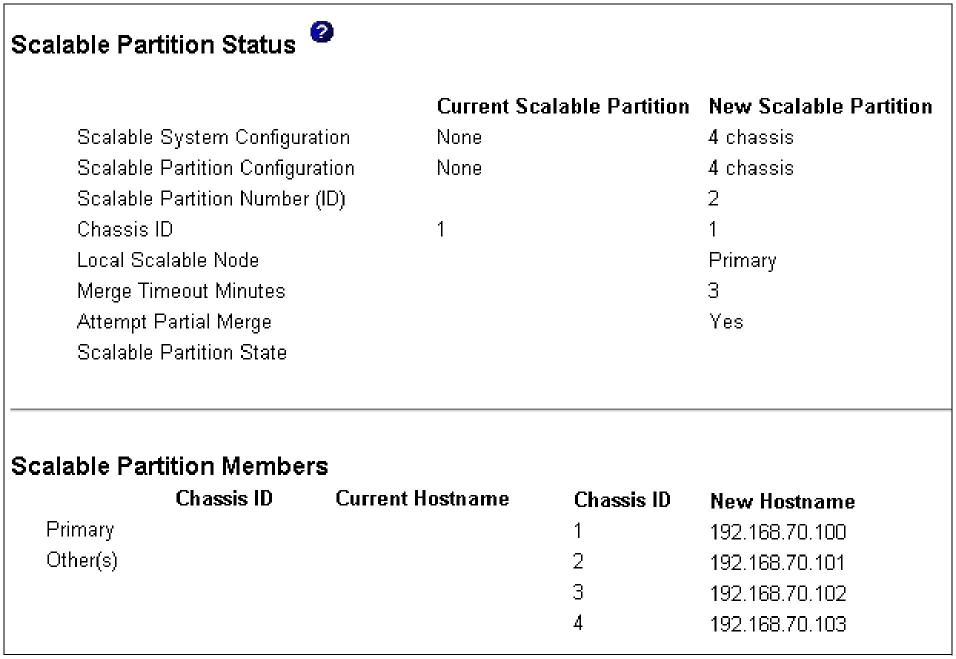
Partition merge timeout minutes	3
On merge failure, attempt partial merge?	Yes
Scalable Partition Number (ID)	2
Primary Scalable Node	192.168.70.100

Scalable Partition Members

☑ Select all systems
☑ 192.168.70.101
☑ 192.168.70.102
☑ 192.168.70.103

Figure 2-23 Select all systems

6. Check the status by clicking **Scalable Partitioning** → **Status**. You can see a screen similar to Figure 2-24 on page 69.



Scalable Partition Status

	Current Scalable Partition	New Scalable Partition
Scalable System Configuration	None	4 chassis
Scalable Partition Configuration	None	4 chassis
Scalable Partition Number (ID)		2
Chassis ID	1	1
Local Scalable Node		Primary
Merge Timeout Minutes		3
Attempt Partial Merge		Yes
Scalable Partition State		

Scalable Partition Members

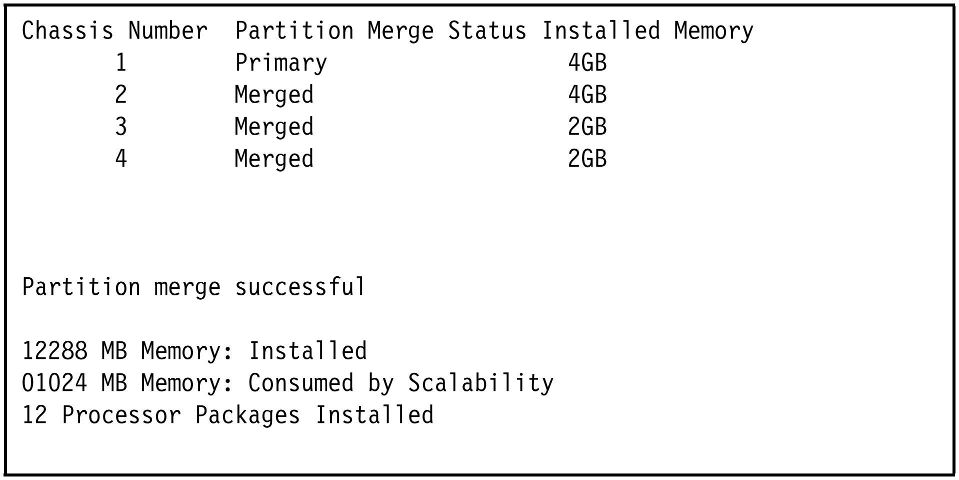
	Chassis ID	Current Hostname	Chassis ID	New Hostname
Primary			1	192.168.70.100
Other(s)			2	192.168.70.101
			3	192.168.70.102
			4	192.168.70.103

Figure 2-24 Four node new scalable partition

7. To make your new partition active, click **Scalable Partitioning** → **Control Partition(s)** then click **Move New Scalable Partition to Current Scalable Partition.**

8. To start the current partition click **Scalable Partitioning** → **Control Partition(s)** → **Start Current Scalable Partition**. This will power on all servers in the partition. Click **OK** in the dialog box shown in Figure 2-19 on page 63.

9. On the monitor of the primary partition, or through the RSA II Web interface Remote Control of the primary node, you can see on the screen if all partitions have merged successfully. See Figure 2-25 on page 70.



```
Chassis Number  Partition Merge Status Installed Memory
       1        Primary                   4GB
       2        Merged                    4GB
       3        Merged                    2GB
       4        Merged                    2GB



Partition merge successful

12288 MB Memory: Installed
01024 MB Memory: Consumed by Scalability
12 Processor Packages Installed
```

Figure 2-25 Successful partition merge

Note: In our lab environment we connected two 4-way x3950s along with two 2-way x3950s in order to demonstrate how to configure a four node system. That is why Figure 2-25 shows 12 installed processors.

This is not a supported configuration. For support, all four nodes must have identical speed, and number and type of processors installed. Also, for performance reasons, we recommend that all nodes have the same amount of memory installed.

2.7 Node failover and recovery

What happens if a node fails and how do you recover? In some senses a multi-node x3950 acts as one server for example, push the power button on the primary node and all nodes power on, or put in the UpdateXpress CD in the primary node and the BIOS will be upgraded on all nodes. In case of a node failure there are two basic scenarios to consider.

- Failure of a secondary node

  In the event of a secondary node failure, the system complex (all nodes) reboots and the remaining nodes merge and boot. This is the case if you have set On merge failure, attempt partial merge? to **Yes** in the RSA II Web interface Scalable Partitioning set up. See 2.6.2, “Create a two node partition” on page 56 for more information about creating a multi-node partition.

Both Windows Server 2003 and ESX Server 3 were able to boot with the surviving nodes remerged (minus the resources in the failed node) without a problem. If the system is heavily utilized, you might have a noticeable performance decrease, but in most cases this is more desirable than no performance from a system that is down completely.

If you have On merge failure, attempt partial merge? set to **No**, then the remaining node will all boot as independent systems. If only the primary node has a boot drive, then only that node will boot and you will probably see a significant performance impact. You may choose to have boot drives in the other nodes (strictly for use in failure scenarios such as this), but you will need to ensure that all independent nodes can boot and coexist on your network.

- Failure of a primary node

  Things are a little more complex if your primary node fails. If the primary node fails, the remaining nodes boot as stand-alone systems, because secondary nodes need to be able to contact the primary node in order to merge. Also, In most configurations the primary node boots the other drives, so if the primary node fails, none of the secondary nodes will have boot drives in order to start the operating system. To recover from the primary node failing, you have a couple of options:

  - One option would be to move the ServeRAID 8i controller and OS system hard drives to a secondary node and run that node in a stand-alone configuration until the primary node is back online, if you application can run on only one node.
  - Another option would be to create a new partition with some of the remaining nodes and move your OS systems drives into the new primary partition. One important consideration is that you cannot create a system complex with an odd number of nodes, so if you have a four node x3950 configuration and the primary node failed you could create a new two node x3950 configuration.

  Note that if you have created multiple partitions in your complex, then you have one primary node for each partition. If the primary node of one partition fails, then that partition will not remerge upon restart. However all other partitions in your complex will be unaffected by this failure.

Both VMware ESX Server 3 and Microsoft Virtual Server have features that can help minimize the impact of a node failure.

- ESX Server 3 when combined with VirtualCenter 2 with VMware High Availability (HA) offers the ability to restart virtual machines automatically on another host in the event of a node failure. More information about HA is available in 3.6, “Load balancing and fault tolerance considerations” on page 137.

- Microsoft Virtual Server can be clustered at the host OS level between multiple machines, using Microsoft Clustering Service. This would allow virtual machines to be moved to the other cluster node in the event of a server failure. See 4.7.2, “Clustered installation” on page 184 for more information about clustering with MSVS.

2.8 BIOS settings

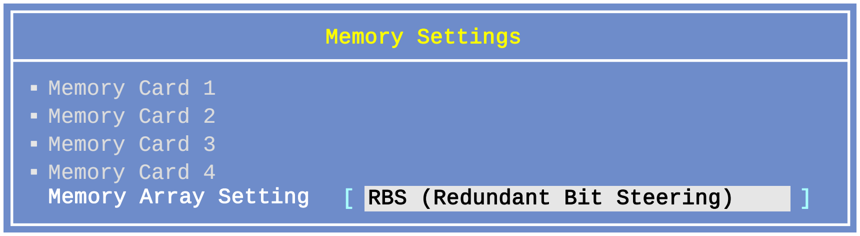
The following sections details some of the more important BIOS settings in regards to a multi-node x3950 configuration.

2.8.1 Memory configuration in BIOS

Depending on your needs, the system memory can be configured in four different ways:

- RBS - Redundant Bit Steering (default)
- FAMM - Full Array Memory Mirroring
- HAM - Hot Add Memory
- HPMA - High Performance Memory Array

You configure the memory subsystem in the server’s BIOS Setup menu by selecting **Advanced Settings** → **Memory** → **Memory Array Setting**, shown in Figure 2-26.



Memory Settings

- Memory Card 1
- Memory Card 2
- Memory Card 3
- Memory Card 4

Memory Array Setting [RBS (Redundant Bit Steering)]

Figure 2-26 Memory options in BIOS

The choices are shown in Table 2-4.

Table 2-4 Memory configuration modes in BIOS

Mode	Memory ProteXion	Memory-mirroring	Hot-swap memory	Hot-add memory
HPMA (high performance memory array)	Disabled	Disabled	Disabled	Disabled
RBS (redundant bit steering) (default)	Yes	Disabled	Disabled	Disabled

Mode	**Memory ProteXion**	**Memory-mirroring**	**Hot-swap memory**	**Hot-add memory**
FAMM (full array memory mirroring)	Yes	Yes	Yes	Disabled
HAM (hot-add memory)	Yes	Disabled	Disabled	Yes

The memory configuration mode you select depends on what memory features you want to use:

- Redundant Bit Steering (RBS):

  This option enables Memory ProteXion and is the default/standard setting. Select **RBS** if you are not using mirroring, hot-swap or hot-add. See 2.3.1, "X3 Architecture memory features" on page 42 for more information about how RBS works.

- Full Array Memory Mirroring (FAMM)

  Select **FAMM** to enable memory mirroring (and to enable hot-swap).

  FAMM reduces the amount of addressable memory by half on each chassis in the partition, but provides complete redundancy of all addressable memory. RBS is available in the mode Automatic Failover is available in this mode. See 2.3.1, "X3 Architecture memory features" on page 42 for more information.

- Hot-Add Memory (HAM)

  Select **HAM** to enable the use of hot-add in the future.

  HAM provides an array layout which supports runtime hot memory add within an OS that supports that feature. This setting has lower performance and may also restrict the amount of memory that can be installed in each chassis as addressable ranges must be reserved on each chassis for the hot add function. RBS is available in this mode. See 2.3.1, "X3 Architecture memory features" on page 42 for more information.

- High Performance Memory Array (HPMA)

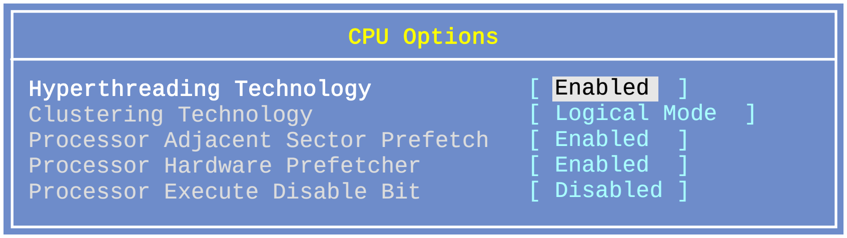
  HPMA optimizes the installed memory array on each chassis in the partition for maximum memory performance. Hardware correction (ECC) of a single correctable error per chip select group (CSG) is provided, but RBS is not available.

  We recommend that you *do not* select the HPMA setting in a production environment because this disables Memory ProteXion.

2.8.2 CPU Options

Hyper-Threading Technology is enabled by default on the x3950. To disable it, if necessary, do the following:

1. Press F1 during system startup to enter the System Configuration Utility.
2. From the main menu, select **Advanced Setup** → **CPU Options**. Figure 2-27 opens.



```
                          CPU Options

Hyperthreading Technology              [ Enabled  ]
Clustering Technology                  [ Logical Mode  ]
Processor Adjacent Sector Prefetch     [ Enabled  ]
Processor Hardware Prefetcher          [ Enabled  ]
Processor Execute Disable Bit          [ Disabled ]
```

Figure 2-27 Hyper-Threading setting

3. With Hyper-Threading Technology selected, press the right arrow key to change the value to **Disabled**.
4. Save changes and exit the System Configuration Utility.

On x3950 multi-node configurations, all the nodes in a partition must have the same Hyper-Threading setting. You must set this individually on each node.

Clustering Technology

For certain operating systems it is necessary to configure how the routing of processor interrupts in a multi-processor system is handled. It is a low-level setting that sets a multi-processor interrupt communication protocol (XAPIC). The settings are functional only, and do not affect performance.

Clustering here refers to the ability of the x3950 to have CPUs across multiple processor buses. The processors are clustered into pairs of processors, one pair for each bus. Each server has two processor buses, and each additional node in an x3950 multi-node complex has two extra processor buses.

> **Note:** The term *clustering* here does not refer to the cluster technology provided by services such as Microsoft Cluster Service.

The choices are **Logical Mode**, **Physical Mode** and **Special Mode**.

- The **Logical Mode** is the default mode for the system. It can be used by Windows, Linux, and ESX Server.

- **Physical Mode** is the required setting for Linux to boot systems when there are 16 or more processor cores in the complex (16-way or larger with the current processors, or 8-way or more when dual-core processors become available).
- The **Special Mode** is required for the 64-bit edition of Red Hat Enterprise Linux 3.0 and only when single-core processors are installed. This is the only operating system that needs it. With dual-core CPU systems, a BIOS update will remove Special Mode from the choices and if it was previously set, will force it to Logical Mode.

Processor Adjacent Sector Prefetch

When this setting is enabled, which is the default, the processor retrieves both sectors of a cache line when it requires data that is not currently in its cache. When it is disabled, the processor will only fetch the sector of the cache line that contains the data requested.

This setting may affect performance, depending on the application running on the server and memory bandwidth utilization. Typically it will affect certain benchmarks by a few percent, although in most real applications it will be negligible. This control is provided for benchmark users that wish to fine-tune configurations and settings.

Processor Hardware Prefetcher

By default, hardware prefetching is enabled which enables the processors to prefetch extra cache lines for every memory request. This optimizes the x3950 as a database server. Recent tests in the performance lab have shown that you will get the best performance for most commercial application types if you disable this feature. The performance gain can be as much as 20% depending on the application.

To disable prefetch, go to BIOS Setup (press F1 when prompted at boot) and select **Advanced Settings** → **CPU** and set **HW Prefetch** to **Disabled**. For high-performance computing (HPC) applications, we recommend you leave HW Prefetch enabled. Future releases of BIOS that ship to enable dual-core will have HW Prefetch disabled by default.

Processor Execute Disable Bit

Execute Disable Bit is a function of new Intel processors with lets you prevent the execution of data that is in memory as thought it were code. When enabled (the default is disable), this will prevent viruses or the like from gaining unauthorized access to applications by exploiting buffer overruns in those applications.

Note: This function is only used for 32-bit operating environments where the processor is in one of the following modes:

- Legacy protected mode, if Physical Address Extension (PAE) is enabled on a 32-bit operating system
- IA-32e mode, when EM64T is enabled on a 64-bit operating system

The operating system must also implement this function.

If this feature is enabled and provided the operating system has marked the memory segment as containing data, then the processor will not execute any code in the segment.

This parameter in BIOS is disabled in BIOS by default just in case the applications you run on your server are affected by it (at least one Java™ application is known to fail if this setting is enabled). For added protection, you may wish to enable it, but you should first test your applications to ensure they will still run as expected before enabling this in a production environment.

2.9 Performance tuning and optimization

The x3950 is designed to deliver superior performance. This section explains how the performance can be improved further by tuning certain settings in the Configuration/Setup Utility or choosing the optimal way to install hardware options.

2.9.1 Optimal memory module installation

Each x3950 supports a maximum number of four memory cards. Up to four memory DIMMs can be installed in each memory card. Each memory card is driven by its own memory interface with a maximum bandwidth of 5.3 GBps.

Aspects of the memory configuration that can affect performance include:

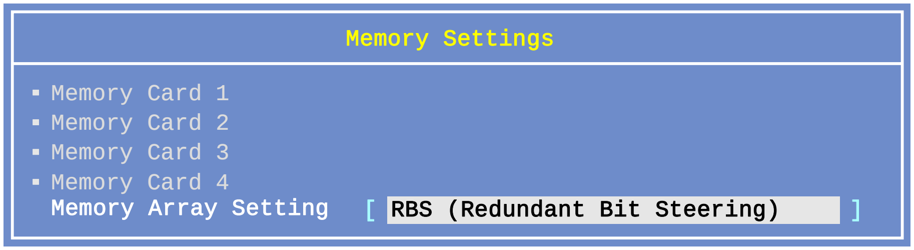
- Spread the installed DIMMs over as many memory cards as possible, preferably up to the maximum of four memory cards. Because each memory card has a dedicated link to the memory controller, the performance gain in doing this can be significant. For example, with four DIMMs in a single memory card, the x3950 is about 50% slower than with four DIMMs across two memory cards (two in each).

  From a performance point of view, we do not recommend you operate the server with only one memory card installed.

- Use as many DIMMs as possible. The optimal configuration is to have all 16 slots in the server populated with DIMMs. The gains are not as significant as with the use of memory cards, perhaps a 3% to 5% improvement, but this method can still make a difference if this memory configuration is suitable for your customer.
- In multi-node configurations, make sure each x3950 and x3950 E are configured with the same memory configuration (same amount, type, and placement of DIMMs).

2.9.2 Memory settings in BIOS

By default, the Memory Array Settings is set to **Redundant Bit Steering (RBS)** as shown in Figure 2-28, which provides better protection from multiple single-bit errors.



```
                         Memory Settings

 ▪ Memory Card 1
 ▪ Memory Card 2
 ▪ Memory Card 3
 ▪ Memory Card 4
   Memory Array Setting   [ RBS (Redundant Bit Steering)    ]
```

Figure 2-28 Memory settings

However, if maximum of performance is required, choose **High Performance Memory Array (HPMA)** in the menu **Advanced Setup** → **Memory Settings**. In this mode, only one single-bit correctable error can be recovered in a chip select group using ECC circuitry.

Note: While changing this setting can result in a small memory performance increase, you will lose the redundancy provided by RBS.

2.9.3 CPU settings in BIOS

By default, thex3950 server is optimized for database transaction processing. This is achieved by enabling hardware prefetching on the processors which forces the processors to prefetch extra cache lines for every request.

If you plan to run applications that do not take advantage of prefetch, such as Java, file/print, or a Web server, then you can gain 10% to 20% by disabling prefetch. To disable prefetch, go to **Advanced Setup** → **CPU Options** and set **Processor Hardware Prefetcher** to **Disabled**.

The default is **Enabled**. This setting affects all processors in the chassis.

2.9.4 PCI adapter placement

In x3950 multi-node configurations, performance can also be impacted by the installation of PCI cards such as network adapters, fibre channel HBAs and so on. To distribute the load equally, we recommend you spread the placement of the adapters across all the nodes. Adapter placement also has an impact on what happens in the even of a node failure, because you would not want all your network cards or HBAs in one node if that node failed.

3

VMware ESX Server

This chapter describes how to configure ESX Server 3.0 to take advantage of the processing power and scalability features of the x3950. We also describe design and implementation of ESX Server 2.5.x.

Topics are the following:

Note: This chapter is based on ESX Server 3.0 beta 1. You may find that the windows and steps described here may be slightly different in beta 2 and the final generally available versions.

3.1 What is new with ESX Server and VirtualCenter?

VMware ESX Server 3.0 and VirtualCenter 2.0 represent the next generation of virtual infrastructure products from VMware. Significant new features, including many often requested by customers such as iSCSI support, have been added. It is clear with the release of these products that VMware is looking for its customers to move beyond the consolidation of low utilization workloads to virtualizing their entire x86 based infrastructure. Larger workloads such as databases, messaging systems, CRM, EPR, etc. are no longer considered bad candidates for virtualization.

Features such as 4-way Virtual SMP, 16 GB of RAM available to virtual machines, as well as 32 logical CPUs with up to 128 virtual CPUs per ESX Server system will greatly increase the workloads that can be virtualized.

Definitions: The following terms definitions are used in this book.

A *logical CPU* is the compute elements of a physical processor. Processor cores and Hyper-Threading are both included in the calculation. For example, a single core processor with Hyper-Threading disabled counts as 1 logical CPU, and a dual-core processor with Hyper-Threading enabled counts as 4 logical CPUs. ESX Server 3.0 supports up to 32 logical CPUs.

A *virtual CPU* is the processor equivalent that a virtual machine is presented by ESX Server. For example a two-way SMP virtual machine is presented with two virtual CPUs. For an ESX Server configuration with three 2-way VMs and five 1-way VMs, there are a total of 11 virtual CPUs: 3x2 + 5x1.

New features such as VMware High Availability (VMware HA, formerly known as Distributed Availability Services or DAS), Distributed Resource Scheduler (DRS), and Consolidated Backup will provide higher availability, guaranteed service level agreements, and quicker recovery from failures than was ever possible before and coming close to the availability you get from more expensive and complicated alternatives such as physically clustered servers. The System x3950 server with its scale-up abilities is uniquely positioned to take advantage of the larger workloads now available to be virtualized.

The new features in ESX Server 3 and VirtualCenter 2 include the following:

- NAS and iSCSI support: The ability to store virtual machines on lower cost NAS and iSCSI storage should allow more companies to take full advantage of all the features that VMware's Virtual Infrastructure provide. Features such as VMotion and HA (see below) will be supported on NAS and iSCSI.

Note: At the time of this writing, the specific NAS and iSCSI hardware to be supported by VMware was not available.

- 4-Way Virtual SMP: Virtual SMP is an add-on that allows you to create virtual machines with more than one virtual CPU. With ESX Server 3.0, you will have the ability to configure up to four virtual CPUs per VM with the new 4-Way Virtual SMP option. This will allow the virtualization of larger work load applications such as database and messaging servers.
- 16 GB RAM for virtual machines: ESX Server 3 will allow you allocate up to 16 GB of memory per virtual machine. This combined with 4-Way Virtual SMP will allow for the virtualization of work loads and systems previously not allowed, and provide all the benefits of the Virtual Infrastructure environment to these systems.
- VMware High Availability. An optional component of VirtualCenter 2, VMware HA (formerly known as Distributed Availability Services) detects failed virtual machines and automatically restarts them on alternate ESX Server hosts. With the virtual machine's disks and configuration files residing on shared storage, fail over time can be quite short.

  HA will have added intelligence and rules that can be applied to restart VMs appropriately, for example, not restarting two load balanced virtual machines on the same ESX Server host. HA will provide higher availability with the added cost or complexity of alternatives such as clustering.
- Distributed Resource Scheduler: Another optional component to VirtualCenter 2, Distributed Resource Scheduler (DRS), will allow for automatic balancing of virtual machines across your ESX Server hosts. DRS uses VMotion to migrate virtual machines from one ESX Server host to another when it detects that not enough resources are available for a given virtual machine.

  DRS still provides the ability to move virtual machines manually, as well as to override, or decline suggested VMotion activities. You will have the ability to exclude certain virtual machines from DRS, so that they can only ever be moved manually. DRS should allow for higher ESX Server utilization because workloads can be automatically migrated for optimal performance.
- VMware Consolidated Backup: Consolidated backup is another optional component for ESX Server 3 and provides host-free, LAN-free, agentless backup of Windows virtual machines. This will provide an easy way to backup an entire running virtual machine while allowing file-level restores.

  Consolidated backup works by automatically acquiescing a virtual disk before creating an online snapshot with no virtual machine downtime required. A separate physical machine can mount the snapshots and use a standard backup agent to back up the data. This also means it might be possible to remove backup agents from Windows virtual machines.

- Simplified Service Console: ESX Server 3 has a new service console based on Red Hat Enterprise Linux 3. The new service console acts more like a standard virtual machine (with virtual I/O devices) consumes less resources and provides greater flexibility for installing third party applications with the service console. All storage and networking devices are now dedicated to the VMkernel meaning no need to divide devices between the service console and virtual machines. Service console resource needs are no longer dependant on the number of virtual machines running.
- VMFS3: With ESX Server 3 includes an updated file system that has a number of improvements including:
  - Improved disk locking to increase scaling for access by a larger number of ESX hosts to shared storage
  - Greater reliability due to distributed journaling
  - The flexibility to resize and add LUNs on the fly

  VMFS3 is no longer a flat file system so you will be able to create directories and subdirectories.
- Hot-add Virtual Disks: ESX Server 3 will provide the ability to add a virtual disk to a virtual machine while running.
- Multiple snapshots: ESX Server 3 will add a multiple snapshot feature similar to what is available in the VMware Workstation 5.x product.
- Large-scale management: VirtualCenter 2 can manage hundreds of ESX Server hosts and thousands of virtual machines. VirtualCenter 2 is designed from the ground up to handle the largest virtual infrastructure deployments.
- Unified user interface: ESX Server 3 and VirtualCenter 2 share a new VMware Virtual Infrastructure Client accessible from any Windows PC or browser. Remotely access and manage ESX Server hosts, virtual machines and VirtualCenter Management Servers from the new VC client. ESX Server 3 no longer contain the MUI for management of the ESX server, instead you will connect to the ESX host from the new VirtualCenter client, or better yet, do all your administration from directly with VirtualCenter.
- Improved Virtual Infrastructure management: VirtualCenter 2 centralizes storage of virtual machine configuration files and VMware licenses for greater deployment flexibility and ease of management. There is a new Licensing Server that can be installed on the VirtualCenter server, within a virtual machine, or on a physical server, to manage all the licenses. Licenses will be allocated out of a pool. All virtual machine configuration files (.vmx, nvram, etc.) are now stored on a shared VMFS volume instead of each individual ESX server. There is a new Web-based remote console to allow system

administrators to connect to virtual machines through a web browser instead of needing a remote console application installed. VirtualCenter 2 has enhanced performance graphs and counters.

- Improved security: Access controls within VirtualCenter have been greatly expanded. Custom roles and permissions are now available. You will have much more flexibility in deciding who can control and change resources. VirtualCenter auditing has been improved to provide an accurate audit trail of who made what changes.
- Expanded ESX Server hardware support: A broader range of hardware will be added to the hardware compatibility list (HCL) for ESX Server 3 allowing customers greater flexibility in their choice of hardware. New servers including more dual-core CPU models, more SAN products, NAS, iSCSI, as well as a wider range of I/O cards will be supported. NIC support increased to 20 Gigabit Ethernet adapters and 26 10/100 Ethernet adapters. Maximum logical CPUs supported has doubled from 16 to 32, helping to enable scale-up implementations such as a multi-node x3950 configuration.
- Improved Networking: In addition to support for more physical NICs, virtual networking has bee improved by increasing the number of ports per virtual switch (vSwitch) to 1024 with up to 32 uplinks per vSwitch and a maximum of 1024 vSwitches per ESX Server system. Per-port NIC teaming, traffic shaping policies and new security policies greatly increase the flexibility of network configurations within ESX Server.
- Expanded ESX Server guest operating system support: New operating systems will be supported as guests in ESX Server 3. Most notably Red Hat Enterprise Linux 4, with others expected as well.
- Improved support for Citrix: Citrix was one of the high visibility applications that VMware targeted for improved performance.

  We were not able to do any Citrix testing for this Redbook, but have been told ESX Server 3 will improve Citrix performance by up to 50%.

Note: This Redbook was written with early beta versions of ESX Server 3 and VirtualCenter 2. Features and specifications are subject to change.

3.1.1 Integration of ESX and VirtualCenter

Although ESX and VirtualCenter are still two different products, the new products further expand ESX Server's dependency on VirtualCenter for key features such as VMotion, HA, and DRS. To take full advantage of VMware's Virtual Infrastructure vision you will need ESX Server and VirtualCenter, as well as the optional add-ons such as VMotion and Virtual SMP.

Given the expanded and enhanced management and security functionality of VirtualCenter 2, its hard to recommend any deployment of ESX Server 3 without VirtualCenter 2. Some basic management of ESX servers can be done without VirtualCenter by using the Virtual Machine Manager (VMM) plug in for IBM Director. See 5.2, “Virtual Machine Manager” on page 216 for information about how to manage ESX Server with IBM Director and VMM.

3.1.2 Virtual Infrastructure Overview

VMware Virtual Infrastructure is a collection of software products that allows for the consolidation and partitioning of industry standard x86 based servers. It also includes the software products to manage these products and the virtual machines running on them. ESX Server 3 and VirtualCenter 2 add some additional new components including:

- VMware High Availability (formerly Distributed Availability Services)
- Distributed Resource Scheduling
- Consolidated Backup
- Licensing Server
- New VirtualCenter Client
- WebCenter (Browser Client)

Figure 3-1 show the basic Virtual Infrastructure components.

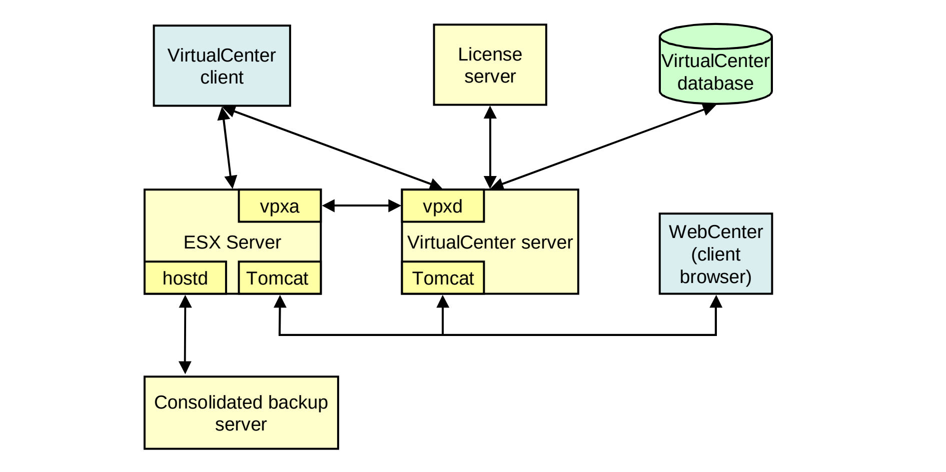


Figure 3-1 Virtual Infrastructure components

The only components that must be a physical server is the ESX Server and the Consolidated Backup Server. VirtualCenter as well as the Licensing Server can be installed on a physical server or a virtual machine. The Licensing Server is the central repository to store and manage all your licenses and can be installed on the same system as VirtualCenter.

The new VirtualCenter Client is used to connect to both VirtualCenter as well as ESX Servers, as the MUI Web interface from ESX Server 2.x has been removed. WebCenter is a Web interface to connect to virtual machines and can be used in place of the remote console application for end users whom you do not want to have access to VirtualCenter.

There is nothing to install on the client side as they connect to the virtual machine with a Web browser. HA and DRS are features that can be unlocked by entering the licensing information into the Licensing Server. There is no additional software associated with the components. For Consolidated Backup it is required that you have a physical Windows system to act as the backup proxy server for all your virtual machines. There is no additional software to install on the ESX Serve or VirtualCenter systems for Consolidated Backup.

See 3.4, “Architecture and design” on page 111 for a more detailed discussion.

Note: Microsoft Access is no longer supported as a database type for VirtualCenter. The MSDE (Microsoft Desktop Engine) version of SQL 2000 is now recommended for demonstration and evaluation purposes.

3.1.3 VirtualCenter objects and concepts

VirtualCenter 2 introduces a number of new objects and concepts that are different from what you might be familiar with in VirtualCenter 1.x versions. The hierarchy in VirtualCenter 2 has been greatly enhanced to provide more flexibility in how you logically design your virtual infrastructure. This is a brief overview only. Consult the official VMware documentation for detailed instructions on how to configure and use VirtualCenter.

- VirtualCenter 2 introduces an important new object called a *cluster*. A cluster is a collection of ESX Server systems joined together to for load balancing with DRS and fail over with HA. A cluster can be either a DRS cluster, HA cluster, or both. You can only create a cluster if you have DRS and HA.
- The resource pool is another important new object in VirtualCenter 2. Basically a *resource pool* is a collection of CPU and memory resources that can be allocated to virtual machines put into that resource pool. There are minimum and maximum amounts of CPU and RAM expressed in MHz and MB.

  Resource pools use a proportional share mechanism to allocate CPU and memory when there is contention for these resources between virtual machines. A good example of a resource pool would be if you had an ESX

Server host with a total of 12 GHz of CPU speed (such as four 3 GHz processors) and 32 GB of RAM, and you created a resource pool called "Web Dev" for your web development team to which you assigned 4 GHz of CPU and 8 GB of RAM. Within this resource pool the developers can divide these resources between as many virtual machines as they like. Resource pools can have child resource pools as well.

- The *data center* object was designed to provide a container for ESX Server hosts that will connect to the same networks and data stores. This is most likely a physical location, for example London and Denver. Other objects such as clusters, resource pools, and folders can appear under the data center. The data center is the VMotion boundary because you are combining systems that are connecting to the same shared storage (SAN, SCSI, or NAS) and have access to the same physical networks.
- The folder is a logical container to help you organize your objects within VirtualCenter. Folders can contain data centers, Clusters, ESX Server hosts, virtual machines, as well as other folders.

  There are a number of different ways folders can be used to organize all your objects within VirtualCenter. For example, you could organize data centers by physical location, a USA folder with New York, Phoenix, and Chicago data centers in that folder, and an EU folder with London, Paris and Milan data centers. You can organize your ESX Server hosts a number of different ways also, for example Intel systems in one folder, and AMD Opteron systems in another. Also to organize virtual machines by function such as development, productions, staging, and so on.

As you can see, folders and the other new objects give you much more flexibility in how you organize your virtual infrastructure. Figure 3-2 on page 87 shows a view of how some of the different objects relate.

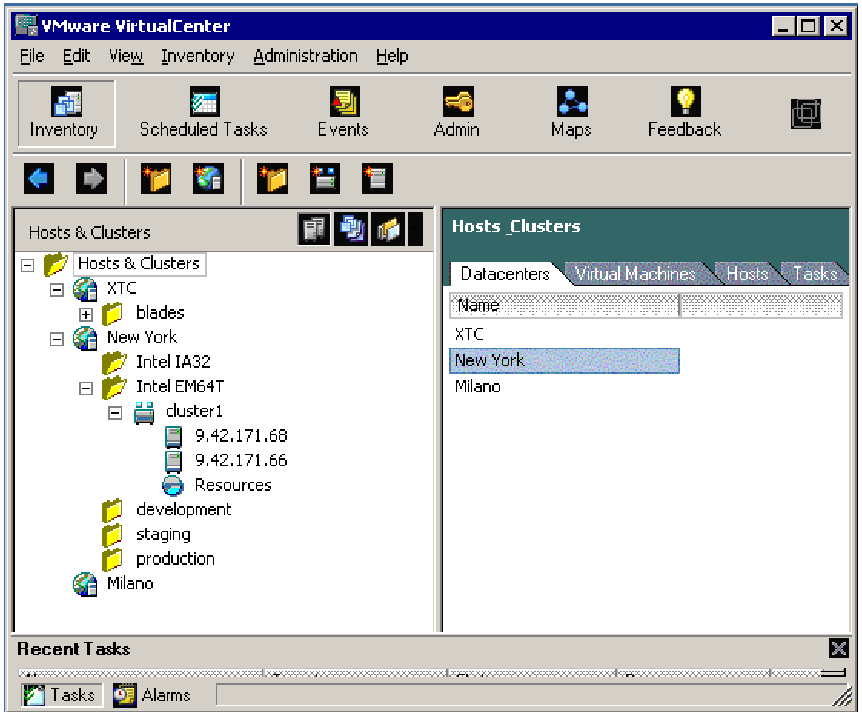


Figure 3-2 VirtualCenter objects

3.2 NUMA, dual-core, and Hyper-Threading

With VMware ESX Server 3.0, the maximum number of logical CPUs supported is doubled to 32. A logical CPU is the compute elements of a physical processor such that a core with Hyper-Threading disabled counts as 1 logical CPU Table 3-1 compares the maximum x3950 multi-node configurations available to use all logical CPUs on ESX Server 2.5.x and 3.0. Single and dual-core as well as Hyper-Threading (HT) configurations are shown.

Table 3-1 Maximum CPU configurations with ESX Server (nodes refers to the number of x3950 servers)

ESX Server Version	Single-core CPUs, HT disabled	Single-core CPUs, HT enabled	Dual-core CPUs, HT disabled	Dual-core CPUs, HT enabled
2.5.x	4 nodes, 16 CPUs	2 nodes, 8 CPUs	2 nodes, 8 CPUs	1 node, 4 CPUs
3.0	8 nodes, 32 CPUs	4 nodes, 16 CPUs	4 nodes, 16 CPUs	2 nodes, 8 CPUs

Note: Even though configuration that will allow you to make use of the maximum number of the 32 *logical processors* available in ESX 3.0 is a 2 node, 8 dual-core CPUs with Hyper-Threading enabled (2 node x 4 CPUs x 2 cores x 2 logical Hyper-Threading-enabled processors = 32), an 8 node, 32 single core CPUs configuration would still offer the highest performance.

There are three ways of describing the number of processors. They are based on:

- The socket (the physical connection the chip has to the system board)
- The core (the compute engine and associated circuitry in the chip)
- The logical processor using Hyper-Threading Technology

Figure 3-3 shows a graphical representation of how these are related.

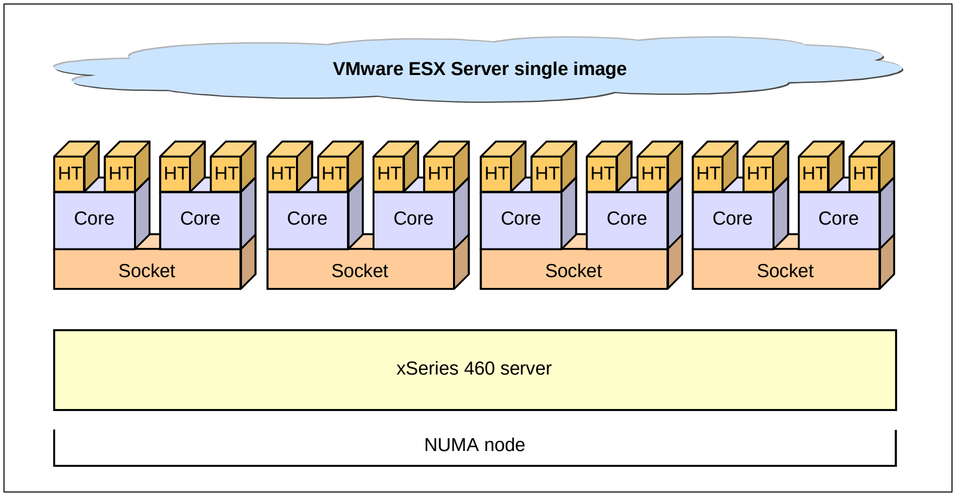


Figure 3-3 Single-node x3950 configuration and different levels of processor.

The hardware in Figure 3-3 is a single x3950 server and represents a single NUMA node. So in a multi-node ESX Server configuration, we have a number of NUMA nodes equal to the number of the x3950 chassis connected together. Figure 3-4 on page 89 shows a two-node configuration.

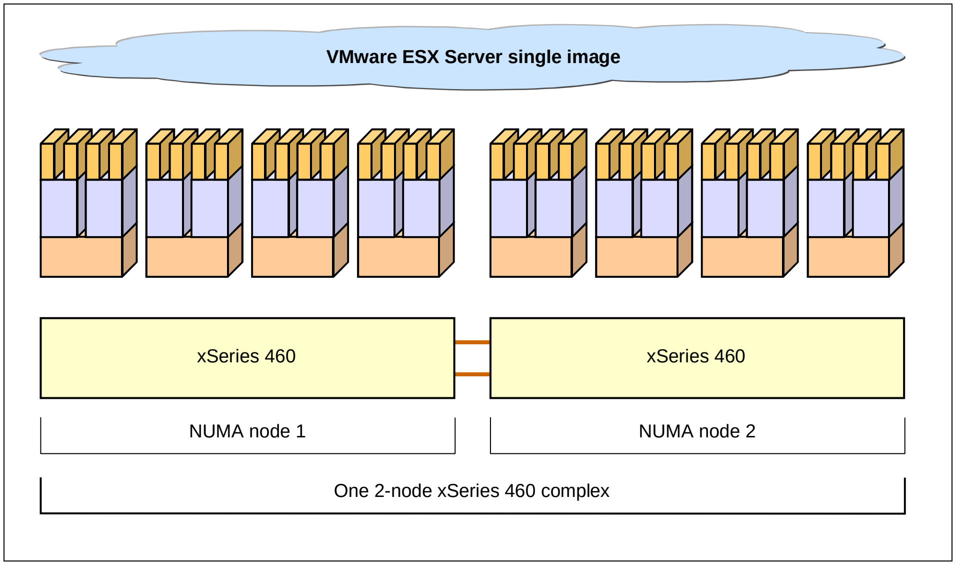

VMware ESX Server single image
xSeries 460
xSeries 460
NUMA node 1
NUMA node 2
One 2-node xSeries 460 complex

Figure 3-4 x3950 two-node configuration

ESX Server 3.0 supports up to 32 logical processors. It is important to understand that the number of sockets, cores and the state of Hyper-Threading all directly impact the number of logical processor that ESX Server detects. In the case of Figure 3-4, even though this is configuration with eight physical Intel processors (sockets), the combination of dual-core and Hyper-Threading enabled means that ESX Server detects 32 processors, its limit.

3.2.1 ESX Server and NUMA

The intelligent, adaptive NUMA scheduling and memory placement policies in ESX Server can manage all VMs transparently, so that administrators do not need to deal with the complexity of balancing VMs between nodes by hand. However, manual override controls are also available and administrators with advanced skills can still optimize their systems as they see fit.

These optimizations work seamlessly regardless of the type of guest operating systems running. ESX Server will provide transparent NUMA support even to guests that do not support NUMA hardware, such as Windows NT 4.0. This unique feature of ESX Server allows clients to take advantage of cutting-edge new hardware, even when tied to earlier operating systems.

Home nodes

ESX Server assigns each VM a *home node* when the VM begins running. A VM will only run on processors within its home node. Newly-allocated memory comes from the home node as well. Thus, if a VM's home node does not change, the VM uses only local memory, avoiding the performance penalties associated with remote memory accesses to other NUMA nodes. New VMs are assigned to home nodes in a round robin fashion. The first VM goes to the first node, the second VM to the second node, and so on. This policy ensures that memory will be evenly used throughout all nodes of the system.

Several commodity operating systems, such as Windows 2003 Server, provide this level of NUMA support, which is known as *initial placement*. It might be sufficient for systems that only run 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 VMs. The system places two of them on the first node and two on the second node. Now consider what happens if both VMs on the second node are stopped, or if they simply become idle. The system is then completely imbalanced, with the entire load placed on the first node. Even if the system allows one of the remaining VMs to run remotely on the second node, it will suffer a serious performance penalty because all of its memory will remain on its original node.

Dynamic load balancing and page migration

To overcome the weaknesses of initial-placement-only systems, as described in the previous section, 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 whether it should rebalance the load by moving a virtual machine from one node to another. This calculation takes into account the relative priority of each virtual machine to guarantee that performance is not compromised for the sake of fairness.

The rebalancer selects an appropriate VM and changes its home node to the least-loaded node. When possible, the rebalancer attempts to move a VM that already has some memory located on the destination node. From that point on, the VM allocates memory on its new home node, unless it is moved again. It only runs on processors within the new home node.

Rebalancing is an effective solution to maintain fairness and ensure that all nodes are fully utilized. However, the rebalancer might need to move a VM to a node on which it has allocated little or no memory. In this case, the VM will incur 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. 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. The system uses the VM monitor layer and the processor's memory management hardware to seamlessly remap the VM'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 VM moves to a new node, ESX Server immediately begins to migrate its memory in this fashion. It adaptively manages the migration rate to avoid overtaxing the system, particularly when the VM has very little remote memory remaining or when the destination node has little free memory available. The memory migration algorithm also ensures that it will not move memory needlessly if a VM is moved to a new node for only a short period of time.

When all these techniques of initial placement, dynamic rebalancing, and intelligent memory migration work in tandem, 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 VMs are started, the system takes time to readjust, migrating VMs and memory to new, optimal locations. After a short period of time, the system completes its readjustments and reaches a steady state.

Manual NUMA controls

Some administrators with advanced skills might prefer to control the memory placement and processor utilization by hand. This can be useful, for example, if a VM runs a memory-intensive workload, such as an in-memory database or a scientific computing application with a large dataset. Such an application can have performance improvements if 100% of its memory is allocated locally, while VMs managed by the automatic NUMA optimizations often have a small percentage (5-15%) of their memory located remotely. An administrator might also wish to optimize NUMA placements manually if the system workload is known to be simple and unchanging. For example, an eight-processor system running eight VMs with similar workloads would be easy to optimize by hand.

VMware ESX Server provides two sets of controls for NUMA placement, so that administrators can control both memory and processor placement of a VM. The ESX Server web-based Management User Interface (MUI) allows you to indicate that a VM should only use the processors on a given node through the **Only Use Processors** option and that it should only allocate memory on the desired node through the **Memory Affinity** option. If both of these are set before a VM starts, it only runs on the desired node and all of its memory is allocated locally. An administrator can also manually move a VM to another node after the VM has started running. In this case, the **Page Migration Rate** of the VM should also be set manually, so that memory from the VM's previous node can be moved to its new node. The ESX Server documentation contains a full description of how to set these options.

Note that manual NUMA placement may interfere with the ESX Server resource management algorithms, which attempt to give each VM a fair share of the system's processor resources. For example, if ten VMs with processor-intensive workloads are manually placed on one node, and only two VMs are manually placed on another node, then it is impossible for the system to give all twelve VMs equal shares of the system's resources. You should take these issues into account when using manual placement.

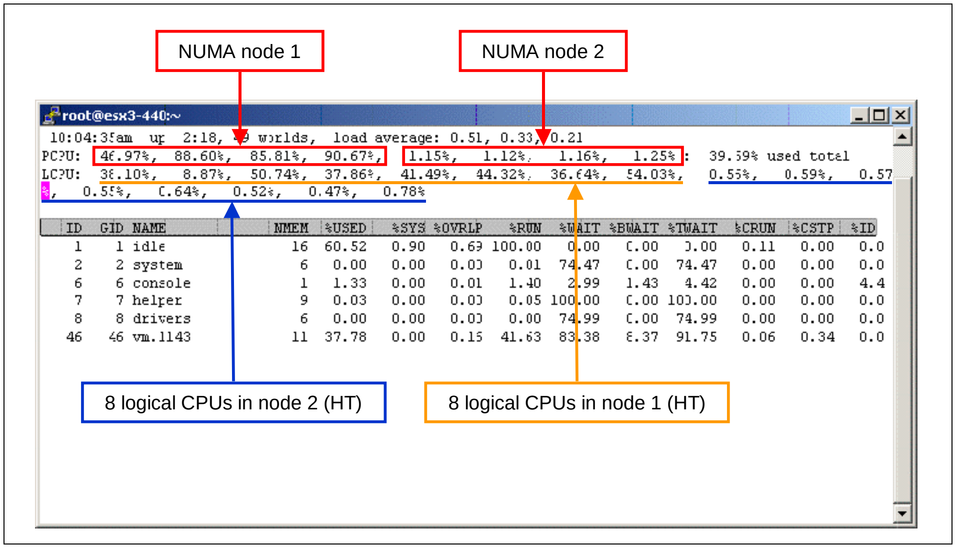
Conclusion

ESX Server's rich manual and automatic NUMA optimizations allow you to fully exploit the advanced scalability features of the Enterprise X-Architecture platform. By providing a dynamic, self-optimizing NUMA load balancer in conjunction with patented memory migration techniques, ESX Server can maintain excellent memory performance even in the face of changing workloads. If you need more information about the technical details of NUMA system configuration, consult the ESX Server documentation.

Real life example of NUMA algorithms

Figure 3-5 on page 93 shows a simple real life example of what happens when the algorithms described above are being leveraged by the VMkernel.

We have created a 4-way virtual machine on a 2-node 8-way system with single core CPUs and Hyper-Threading enabled. We have then stressed the 4-way virtual machine to simulate an heavy workload to push it towards full utilization of the resources associated to that. Figure 3-5 on page 93 shows the utility **`esxtop`**.



```
 10:04:35am  up  2:18, 49 worlds,  load average: 0.51, 0.33, 0.21
PCPU:  46.97%,  88.60%,  85.81%,  90.67%,  1.15%,   1.12%,   1.16%,   1.25% :  39.59% used total
LCPU:  38.10%,   8.87%,  50.74%,  37.86%,  41.49%,  44.32%,  36.64%,  54.03%,   0.55%,   0.59%,   0.57
%,   0.55%,   0.64%,   0.52%,   0.47%,   0.78%

  ID  GID NAME              NMEM  %USED   %SYS %OVRLP   %RUN  %WAIT %BWAIT %TWAIT  %CRUN  %CSTP  %ID
   1    1 idle                16  60.52   0.90   0.69 100.00   0.00   0.00   0.00   0.11   0.00  0.0
   2    2 system               6   0.00   0.00   0.00   0.01  74.47   0.00  74.47   0.00   0.00  0.0
   6    6 console              1   1.33   0.00   0.01   1.40   2.99   1.43   4.42   0.00   0.00  4.4
   7    7 helper               9   0.03   0.00   0.00   0.05 100.00   0.00 100.00   0.00   0.00  0.0
   8    8 drivers              6   0.00   0.00   0.00   0.00  74.99   0.00  74.99   0.00   0.00  0.0
  46   46 vm.1143             11  37.78   0.00   0.15  41.63  83.38   8.37  91.75   0.06   0.34  0.0
```

Figure 3-5 esxtop running on a NUMA system

As you can see from this basic example the workload being generated by the 4-way virtual machine is kept local in a single NUMA node (NUMA node 1 in this case).

3.2.2 ESX Server and Hyper-Threading

The concepts behind Hyper-Threading are discussed in 2.2.2, “Hyper-Threading” on page 39.

ESX Server has been Hyper-Threading aware since V2.1 and the VMkernel is capable of scheduling workloads on the system taking into account this feature and its limitations. ESX Server implements algorithms and options for proper handling of threads running on Hyper-Threading enabled processors. These algorithms can also be set at the VM level.

As a basic rule Hyper-Threading on an ESX Server system can be either enable or completely disabled. In order for Hyper-Threading to be enabled on an ESX Server system it must be enabled both in the system BIOS as well as on the ESX Server configuration.

For ESX Server 2.5.2 you can do this on the Web interface in the Startup Profile while on ESX Server 3.0 you have to do that in the processor configuration panel.

Assuming Hyper-Threading is enabled, ESX Server can handle it in three different way on a per-VM basis:

- **Any** (default)

  Any is the default behavior and allows the VMkernel maximum flexibility to schedule virtual machines. The vmkernel is aware that two logical processors are part of the same core and it will endeavor to not schedule both logical processors if possible. However, if it must allocate the second logical processor on the one core, it will do so. If the virtual machine configuration files do not get modified this is the default for all workloads.

- **Internal**

  Internal only applies to SMP-enabled virtual machines and instructs the VMkernel to schedule two of its virtual CPUs on the same physical CPU package. This is useful in those situations where the application being executed benefits the most by running its threads on a single physical package thereby sharing the one cache.

  The various cache levels on an Intel Xeon CPU are shared among the Hyper-Threading enabled Logical CPU and, in some circumstances, it might be better to localize workloads that could benefit from such a consistent cache content.

- **None**

  None is, on the other hand, a virtual machine configuration that forces the VMkernel to *halt* the partner logical CPU when the other is scheduled on the same CPU package. This is useful in those situation when it is known that the performance of a workload will suffer if both logical processors in a Hyper-Threading-enabled processor are in operation.

  because the virtual machine is scheduled on a logical CPU and it forces the VMkernel not to use its partner logical CPU, this virtual machine is charged twice in terms of systems resources being utilized.

For more information about how ESX Server 2.x works with with Hyper-Threading Technology, refer to this document:

http://www.vmware.com/pdf/esx21_hyperthreading.pdf

3.2.3 ESX Server and dual-core processors

The concepts behind dual-core processors are discussed in 2.2.1, “Dual-core processors” on page 38.

The ESX Server scheduling algorithms for processes on dual-core processors is very similar to that of single-core processors. The VMkernel scheduler treats each core as a physical processor. Figure 3-6 on page 95 shows this concept:

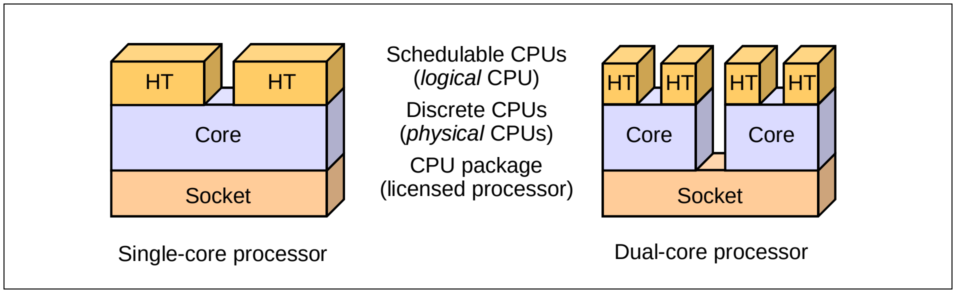
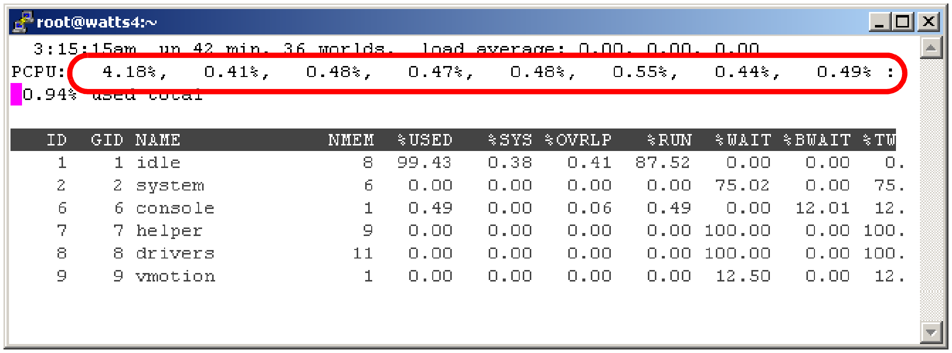


Figure 3-6 Single-core and dual-core comparison in terms of scheduling.

So on a 4-way x3950 with dual-core CPUs and Hyper-Threading turned off, the output from esxtop would be similar to Figure 3-7, showing eight physical CPUs (PCPU).



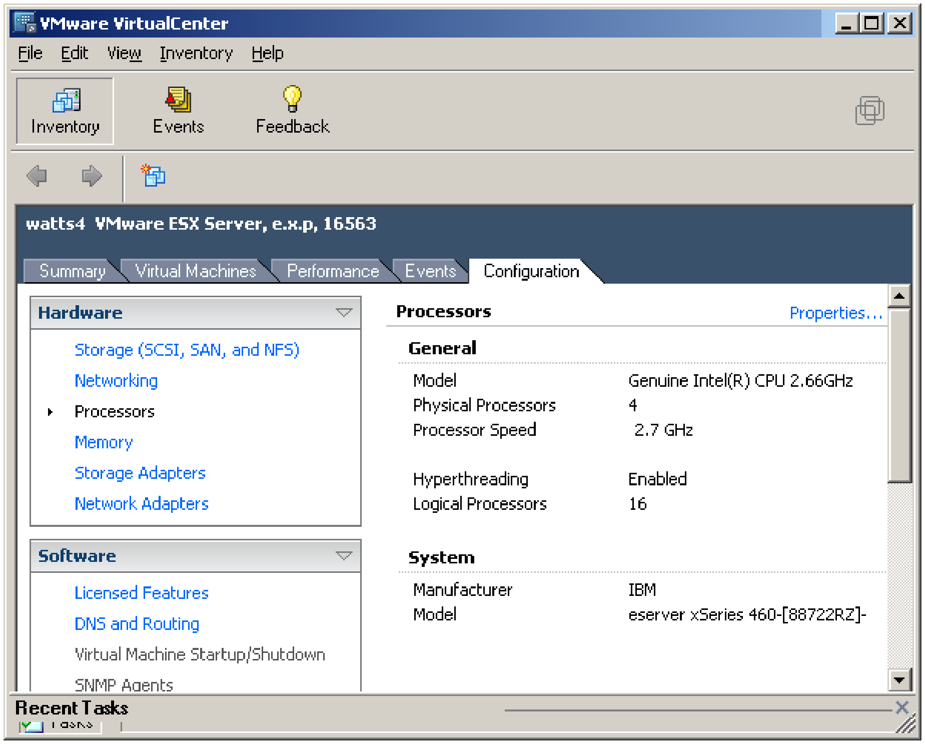
```
  3:15:15am  up 42 min, 36 worlds,  load average: 0.00, 0.00, 0.00
PCPU:    4.18%,    0.41%,    0.48%,    0.47%,    0.48%,    0.55%,    0.44%,    0.49% :
 0.94% used total

   ID   GID NAME              NMEM  %USED    %SYS %OVRLP   %RUN  %WAIT %BWAIT %TW
    1     1 idle                 8  99.43    0.38   0.41  87.52   0.00   0.00   0.
    2     2 system               6   0.00    0.00   0.00   0.00  75.02   0.00  75.
    6     6 console              1   0.49    0.00   0.06   0.49   0.00  12.01  12.
    7     7 helper               9   0.00    0.00   0.00   0.00 100.00   0.00 100.
    8     8 drivers             11   0.00    0.00   0.00   0.00 100.00   0.00 100.
    9     9 vmotion              1   0.00    0.00   0.00   0.00  12.50   0.00  12.
```

Figure 3-7 **`esxtop`** *on an x3950 4-way dual-core with Hyper-Threading turned off*

The graphical interface also provides similar information:

- The number of CPU sockets (physical processors)
- The number of executable processors (logical processors),

Figure 3-8 on page 96 shows a 4-way x3950 system with dual-core CPUs and Hyper-Threading turned on.



watts4 VMware ESX Server, e.x.p, 16563

Processors

General

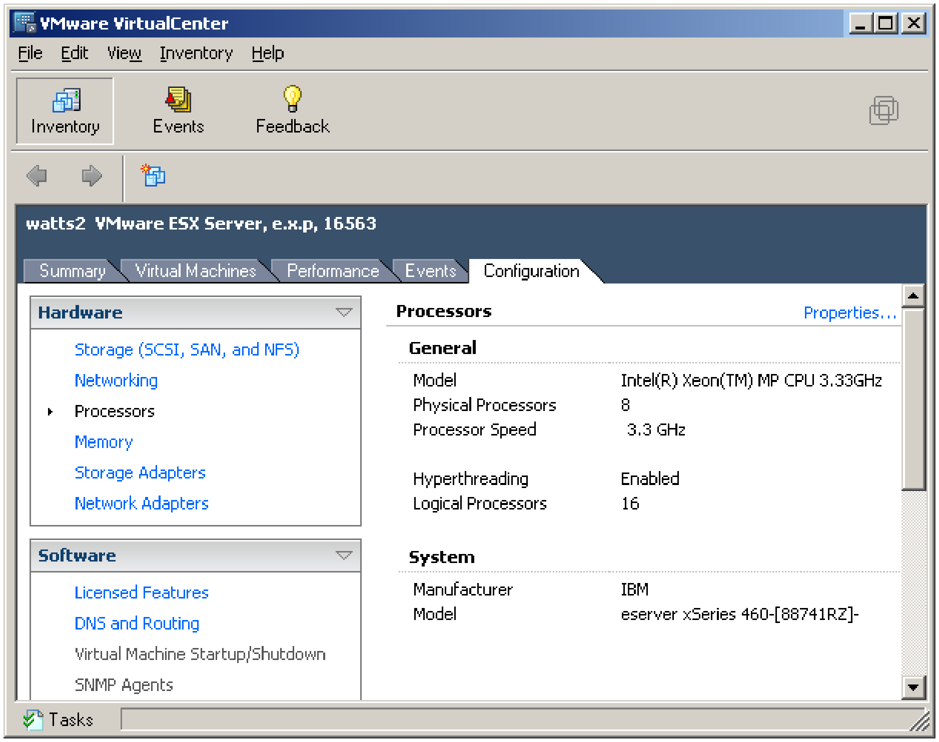
Model	Genuine Intel(R) CPU 2.66GHz
Physical Processors	4
Processor Speed	2.7 GHz
Hyperthreading	Enabled
Logical Processors	16

System

Manufacturer	IBM
Model	eserver xSeries 460-[88722RZ]-

Figure 3-8 **`esxtop`** *on an x3950 4-way dual-core with Hyper-Threading turned on*

At the time of writing, however, this display can cause confusion because, under certain circumstances, you cannot tell from the interface the exact configuration of your system. Figure 3-9 on page 97 is an example of a potential misleading screen.



watts2 VMware ESX Server, e.x.p, 16563

Processors

General

Model	Intel(R) Xeon(TM) MP CPU 3.33GHz
Physical Processors	8
Processor Speed	3.3 GHz
Hyperthreading	Enabled
Logical Processors	16

System

Manufacturer	IBM
Model	eserver xSeries 460-[88741RZ]-

Figure 3-9 8-way server with 16 logical CPUs

Looking at Figure 3-9 it would be difficult to understand whether the 16 logical processors are eight dual-core CPUs with Hyper-Threading turned off or eight single-core CPUs with Hyper-Threading turned on.

However using **esxtop**, you can easily determine whether those logical processors are actual cores or the result of Hyper-Threading. Cores will show up as PCPUs (physical CPUs) while Hyper-Threaded logical CPUs will show up as LCPUs (logical CPUs).

Note: the discussion in this section applies to both ESX 2.5.x as well as to ESX 3.0. There is no difference in how the two versions deal with these different processor layers.

3.3 Configuration best practices

As with section 3.4, “Architecture and design” on page 111, we are going to divide configuration best practices into two sections:

- 3.3.1, “ESX Server 2.5.x configuration best practices” on page 98
- 3.3.2, “ESX Server 3 configuration best practices” on page 104

These recommendations are being made specifically for the x3950 server in a multi-node configuration, although they may apply to other servers as well.

3.3.1 ESX Server 2.5.x configuration best practices

The following information applies to ESX Server 2.5.x although some information can apply to ESX Server 3 as well.

> **Note:** For multi-node x3950 configurations with single-core CPUs you must use ESX 2.5.1 upgrade 1 or higher For dual-core CPUs, you must use ESX 2.5.2 or higher.

Firmware and BIOS settings

We recommend you use the latest UpdateXpress CD to update your BIOS and firmware to the latest levels. Download the latest from:

http://www.pc.ibm.com/support?page=MIGR-53046

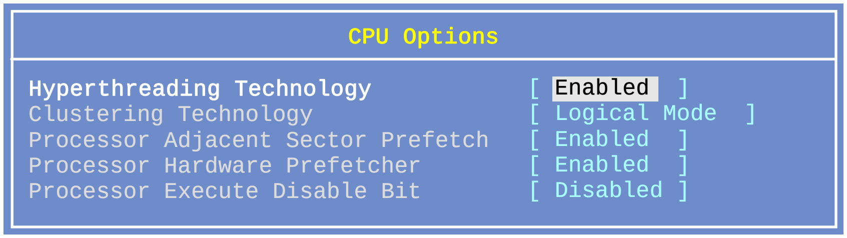
For ESX Server 2.5.x we recommend the following BIOS setting be changed.

- Disable Hardware Prefetch

  To disable prefetch, go to **Advanced Setup** → **CPU Options** and set Processor Hardware Prefetcher to **Disabled**.

- Disable Hyper-Threading

  ESX Sever 2.5.x supports up to 16 logical processors. Because each processor with Hyper-Threading enabled would appear as two logical processors to ESX you will need to disable Hyper-Threading on a 16-way, 4 node x3950, or a 8-way, 2 node x3950 using dual-core CPUs. From the main BIOS menu, select **Advanced Setup** → **CPU Options**. Figure 3-10 on page 99 appears. With Hyper-Threading Technology selected, press the right arrow key to change the value to **Disabled**.



```
                          CPU Options

Hyperthreading Technology                 [ Enabled  ]
Clustering Technology                     [ Logical Mode  ]
Processor Adjacent Sector Prefetch        [ Enabled  ]
Processor Hardware Prefetcher             [ Enabled  ]
Processor Execute Disable Bit             [ Disabled ]
```

Figure 3-10 Hyper-Threading setting

If you are under these thresholds then we recommend you leave Hyper-Threading enabled. In most cases it will provide a slight performance increase especially in cases where you are running multiple Virtual SMP virtual machines. VMware has a white paper on Hyper-Threading that you can look at for more information:

`http://www.vmware.com/pdf/esx21_Hyper-Threading.pdf`

- RSA II configuration

  Set the **OS USB Selection** to Linux OS. See 2.6.1, “RSA II SlimLine setup” on page 54 for instructions on how to configure the RSA II adapter.

Memory DIMM configuration

Follow the recommendation in 2.9.1, “Optimal memory module installation” on page 76 Remember ESX Server 2.5.x supports up to 64 GB of RAM per server. VMware recommends that you burn in your memory for 48 hours before putting a new server into production. A free tool such as Memtest86+ should be used.

CPU considerations

The x3950 and all x3950 Es must each have four processors installed and all processors must be the same speed, number of cores, and cache size.

PCI card placement

In x3950 multi-node configurations, performance can also be impacted by the installation of PCI cards such as network adapters, Fibre Channel HBAs and so on. To distribute the load equally, we recommend you spread the placement of the adapters across all the nodes. Spreading the adapters also helps in the event of a node failure. See 2.7, “Node failover and recovery” on page 70 for more information about recovering from node failures.

- Fiber channel adapters

  On a two-node x3950, for example, we recommended that you place one Fibre Channel HBA in node 1 and the other in node 2. We recommend a minimum of two HBAs to provide for redundancy in any ESX Server implementation. For QLogic based HBAs it is recommended that you also change the Port Down Retry value in the QLogic BIOS to 15.

- ServeRAID 8i RAID controller and SAS

  The ServeRAID 8i controller upgrades the on board disk controller of the x3950 to provide full RAID capabilities and is recommended for all ESX Server implementations. The ServeRAID 8i does not use a PCI slot because it has its own slot on the system board.

  We recommend you place the ServeRAID 8i and hard drives in the primary node and on all secondary nodes disable the onboard SAS controller in the BIOS.

- Network controllers

  On a two-node x3950, we recommend a minimum of four Gigabit network controllers:

  - One for the service console
  - One for VMotion
  - Two for a virtual switch for the virtual machines to use

  On a four-node x3950, we recommend a minimum of six network cards, although you would have eight onboard NICs and might want to use all of them:

  - One for the service console
  - One for VMotion
  - Four-six for virtual machines

  Table 3-2 shows how you would configure on the onboard NICs in a 2-way or 4-way x3950 with ESX Server 2.5.x.

  Remember, in ESX Server 2.5.x you are limited to 32 ports per virtual switch, so with a 4-node configuration, we would have to create a minimum of two virtual switches. Also, you are limited to 8 Gigabit NICs in ESX Server 2.5.x, so if you wish to use add-in PCI NICs, you must disable the onboard NICs in the BIOS.

  Note: This is to illustrate basic network configuration and does not take into account more advanced topics such as backup networks and DMZ networks.

Table 3-2 Network configuration in ESX Server 2.5.x

NIC	Node	Purpose	Label	vSwitch
NIC1	1	Service Console	eth0	none

NIC	Node	Purpose	Label	vSwitch
NIC2	1	VMs	vmnic0 Outbound Adapter 0	vSwitch1
NIC3	2	VMotion	vmnic1 Outbound Adapter 1	VMotion
NIC4	2	VMs	vmnic2 Outbound Adapter 2	vSwitch2
NIC5	3	VMs	vmnic3 Outbound Adapter 3	vSwitch1
NIC6	3	VMs	vmnic4 Outbound Adapter 4	vSwitch2
NIC7	4	VMs	vmnic5 Outbound Adapter 5	vSwitch1
NIC8	4	VMs	vmnic6 Outbound Adapter 6	vSwitch2

Hard drives

We recommend you install ESX Server on a RAID 1 array and add a hot spare drive for increased fault tolerance. The size of hard drives you need will depend on how much RAM you have in the server and how many virtual machines you plan to run.

We recommend you configure your server with three 72.3 GB hard drives, two for a RAID 1 array and one hot spare. This will provide enough space for a configuration up to a 4-node x3950 with 64 GB of RAM and 64 virtual machines running. This is assuming all the virtual machines are running on the SAN and not local disk.

Disk partitioning

Disk partition size depends on a number of factors including the number of virtual machines that will be running and the amount of RAM installed. Your swap should be 2x the amount of RAM, and the VMFS2 volume used for the VMkernel swap should be at least as large as the amount of physical RAM installed in the server.

Table 3-3 on page 102 shows an example of how to partition the disks for a two-node, 8-way x3950 with 32 GB of RAM designed to run 32 virtual machines, assuming a 72.3 GB local RAID-1 array and with virtual machines stored on a SAN.

Table 3-3 ESX Server 2.5.x disk partitioning

Partition	Size	Comment
/boot	50 MB	Service Console boot files. Should be created as a primary partition.
/	4 GB	Root partition. Numerous problems can develop if the root partition runs out of disk space. Should be created as a primary partition.
swap	1 GB	Swap file for service console, should be twice the amount of RAM assigned to the service console. Should be created as a primary partition.
/var	1 GB	Various ESX Server logs are stored in this partition. This size should be sufficient to not run out of space. This is also used if you plan to use the VMware method of a scripted install.
/home	512 MB	Virtual machine configuration files are stored here. They are small and this will be enough space regardless of how many virtual machines you have running.
/vmimages	10 GB	This partition is can be used to store ISO images of various OS and application CDs that can then be mounted by the virtual machines.
VMFS2	32 GB	This partition will be formatted as the VMFS2 file type to create the VMkernel swap. This should be equal to the size of the physical RAM in the server. It can be made larger to allow more over allocation of memory.
core dump	100 MB	In the event of an ESX server crash, a log is put in the coredump partition to send to VMware support.
/tmp	1 GB	Optional. Some people like to create a partition for temp files

Service console memory and CPU

For a two-node x3950, set the service console RAM to 512 MB. For a four-node x3950, set the service console RAM to 800 MB. If you plan on running additional applications within the service console we would recommend you increase the RAM to 800 MB (this is the maximum) for all x3950 multi-node configurations.

You should also take note of the minimum CPU value assigned to the service console. By default, when you install ESX Server 2.5.x, it will allocate 8% of CPU0 as the minimum for the service console. This is based on the assumption that no additional applications were to be installed in the service console.

Because we recommend that you install the IBM Director agent on your x3950 server, we also recommend that you increase the minimum CPU guaranteed to the service console. We recommend that you increase this amount to the following:

- Two-node configurations: 15% minimum CPU
- Four-node configurations: 30% minimum CPU

Remember, these minimum values are only enforced if the service console needs the additional CPU cycles, and there is contention for resources. Under most circumstances the service console will use less CPU than the minimum listed here, and the unused processor capacity is available to virtual machines.

Network configuration

We recommend you start with all the network controllers set to **auto negotiate** for their speed and duplex settings. Our experience is that this is the best setting for the onboard Broadcom NICs that are in the x3950.

If you experience network-related performance issues you can try changing the NIC settings to **1000/Full.** These settings are in the MUI under **Options** → **Network Connections**. See Table 3-2 on page 100 for our recommended configuration using the onboard NICs. This is a basic configuration. For more information about advanced networking topics, VMware has several white papers about networking, available from:

http://www.vmware.com/vmtn/resources/esx_resources.html

Time synchronization

It is important that ESX Server keep accurate time. To sync ESX Server with a NTP server follow the directions as outlined in VMware KB Answer ID# 1339:

http://www.vmware.com/support/kb/enduser/std_adp.php?p_faqid=1339

VMware also recommends that you sync your virtual machine’s time with the ESX Server’s time. This is a function of the VMware Tools that are installed in the virtual machines. For more detailed information about timekeeping, see the VMware white paper *Timekeeping in VMware Virtual Machines* available from:

http://www.vmware.com/pdf/vmware_timekeeping.pdf

Storage configuration

There are a couple of changes that can be made to optimize your storage I/O.

- Fiber channel queue depth on QLogic HBAs

  VMware recommends in high I/O environments that you increase the HBAs maximum queue depth. This can be done in the Service Console by editing the hwconfig file. Step by step instructions are provided in the KB Answer 1267.

`http://www.vmware.com/support/kb/enduser/std_adp.php?p_faqid=1267`

The recommended value to use is `64`. While it is possible to change this setting for Emulex based HBAs, VMware has no specific recommendation to due so, therefore we would recommend only changing this setting with QLogic HBAs.

- Outstanding disk requests per virtual machine

  This setting goes along with the queue depth setting. VMware recommends that you change both of them at the same time. This is done by changing the `Disk.SchedNumReqOutstanding` value in the **Options** → **Advanced Settings** in the MUI. Change this setting to match what was changed in the HBA max queue depth change (`64` is the recommended value). The VMware KB Answer 1268 has step by step instructions for changing this setting:

  `http://www.vmware.com/support/kb/enduser/std_adp.php?p_faqid=1268`

Additional best practices

For additional best practices on ESX 2.5.x, see the *Best Practices for VMware ESX Server 2* white paper:

`http://www.vmware.com/pdf/esx2_best_practices.pdf`

3.3.2 ESX Server 3 configuration best practices

Best practices are usually developed over a period of time by the user community and vendor working together. Given that the ESX Server 3 public beta was not yet available at the time of this writing, the recommendations we make here might change over time. However as a start, we highlight what we think will be important.

RSA II configuration

Set the OS USB Selection to **Linux OS**. See 2.6.1, “RSA II SlimLine setup” on page 54 for instructions on how to configure the RSA II adapter.

Memory configuration

Follow the recommendation in 2.9.1, “Optimal memory module installation” on page 76. Remember ESX Server 3.0 supports up to 64 GB of RAM per server. VMware recommends that you burn in your memory for 48 hours before putting a new server into production. They suggest a free tool such as Memtest86+ be used.

CPU considerations

The x3950 and all x3950 Es must each have four processors installed and all processors must be the same speed and cache size.

PCI card placement

In x3950 multi-node configurations, performance can be affected also by the installation of PCI cards such as network adapters, Fibre Channel HBAs and so on. To distribute the load equally, we recommend you spread the placement of the adapters across all the nodes. Spreading the adapters also helps in the event of a node failure. See 2.7, “Node failover and recovery” on page 70 for more information about recovering from node failures.

- Fibre Channel adapters

  On a two-node x3950, for example, we recommended that you place one Fibre Channel HBA in node 1 and the other in node 2. We recommend a minimum of two HBAs to provide for redundancy in any ESX Server implementation. For QLogic based HBAs, it is recommended that you also change the Port Down Retry value in the QLogic BIOS to 15.

- ServeRAID 8i RAID controller and SAS

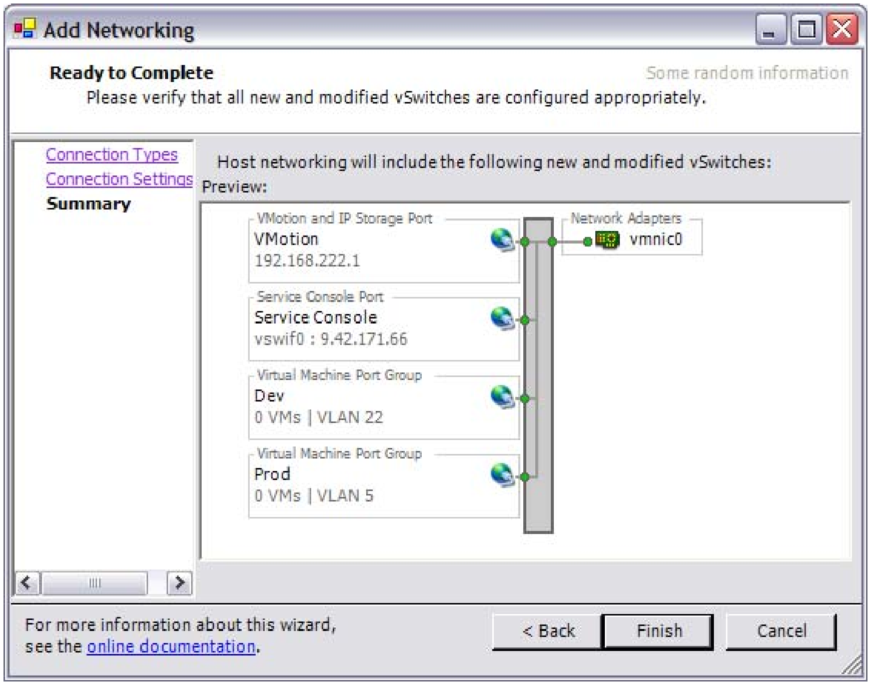
  The ServeRAID 8i controller upgrades the on board disk controller of the x3950 to provide full RAID capabilities and is recommended for all ESX Server implementations. The ServeRAID 8i does not use a PCI slot as it has its own slot on the system board.

  We recommend you place the ServeRAID 8i and hard drives in the primary node and on all secondary nodes disable the onboard SAS controller in the BIOS.

- Network card configuration

  ESX Server 3 can support up to 1024 ports for each virtual switch (ESX Server 2.5.x was limited to 32 ports per virtual switch). It is no longer necessary to separate the console OS, virtual machines, and VMotion NICs. You can simply assign all physical NICs to the switch created during install and create new port groups for each.

  See Figure 3-11 on page 106 for an example of a virtual switch with various ports. In our conversations with VMware, they think this is just as secure as having one NIC dedicated to the service console.



Add Networking

Ready to Complete

Please verify that all new and modified vSwitches are configured appropriately.

Host networking will include the following new and modified vSwitches:

Preview:

VMotion and IP Storage Port
VMotion
192.168.222.1

Service Console Port
Service Console
vswif0 : 9.42.171.66

Virtual Machine Port Group
Dev
0 VMs | VLAN 22

Virtual Machine Port Group
Prod
0 VMs | VLAN 5

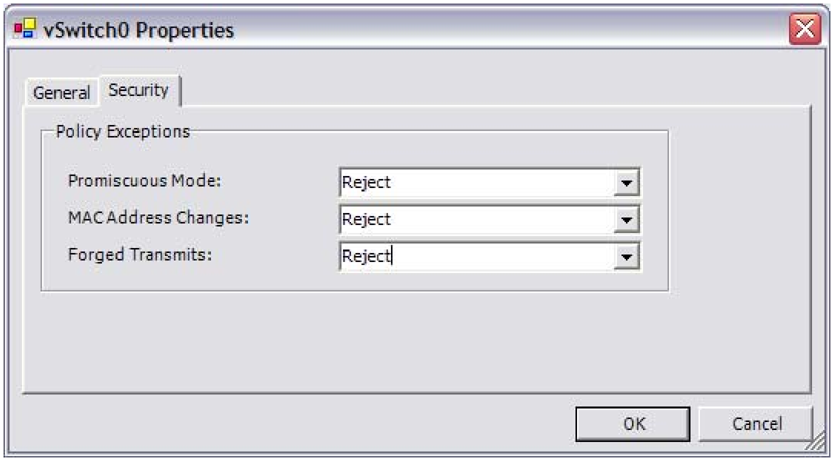
Network Adapters
vmnic0

For more information about this wizard, see the online documentation.

Figure 3-11 virtual switch connections

There can be valid reasons to create multiple virtual switches, such as to isolate one type of network traffic completely, NAS for example. ESX Server 3 introduces new security policies for networking. By default, they are all set to **Allow**.

You must test for your environment, but we suggest you change the settings to **Reject**, as shown in Figure 3-12 on page 107.



Policy Exceptions

Promiscuous Mode:	Reject
MAC Address Changes:	Reject
Forged Transmits:	Reject

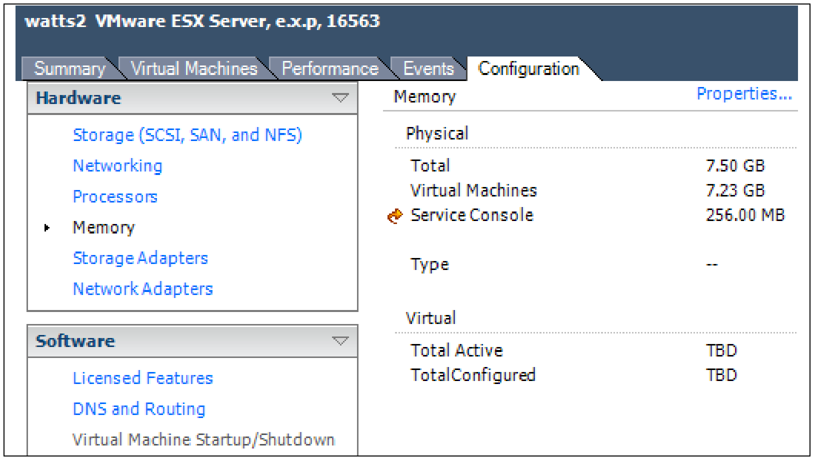
Figure 3-12 New virtual switch security policy

See 3.4.2, "ESX Server 3 design" on page 119 for more information.

We expect that after ESX Server 3 is released, VMware and its user community will develop a more comprehensive list of best practices for the extensive new networking features in ESX Server 3.

Service console memory and CPU resources

With ESX Server 3, the amount of RAM assigned to the service console is no longer tied to the number of virtual machines. The default amount is 256 MB, as you can see in Figure 3-13. This setting can be changed after, but not during, the installation.



Memory

Physical	
Total	7.50 GB
Virtual Machines	7.23 GB
Service Console	256.00 MB
Type	--
Virtual	
Total Active	TBD
TotalConfigured	TBD

Figure 3-13 Service console memory

As with ESX Server 2.5.x, you might want to increase this amount if you are running additional applications in the service console. We recommend you increase it to 512 MB if you will be running IBM Director Agent, backup clients, or other applications in the service console.

Regarding CPU resources, although the service console in ESX Server 3 is designed to consume less resources, at this time we feel the minimum CPU settings recommended for ESX 2.5.x can still be applied to ESX Server 3. We recommend you increase this amount to the following:

- Two-node configurations: 15% minimum CPU
- Four-node configurations: 30% minimum CPU
- Eight-node configurations: 50% minimum CPU

Disk drives and partitioning

Because ESX Server 3 no longer requires a local VMFS partition for the VMkernel swap file, it is now possible to use smaller hard drives of the same size, no matter how many virtual machines you are running, or how much RAM is installed in the physical server.

With ESX Server 3, disk partitioning is no longer dictated by how much RAM is installed in the physical server and how many virtual machines you have running. Table 3-4 shows the recommended partitioning scheme for ESX Server 3 on a multi-node x3950.

Table 3-4 ESX Server 3 partitioning

Partition	Size (MB)	Comment
/boot	100	Boot files. Should be primary partition.
/	4096	Root partition. Should be primary partition.
swap	1024	service console swap. Should be primary partition.
/var/log	1024	VMware log files
coredump	100	VMkernel coredump partition
/tmp	1024	Optional temp partition
VMFS3	Rest of disk	Could be used for virtual machines.

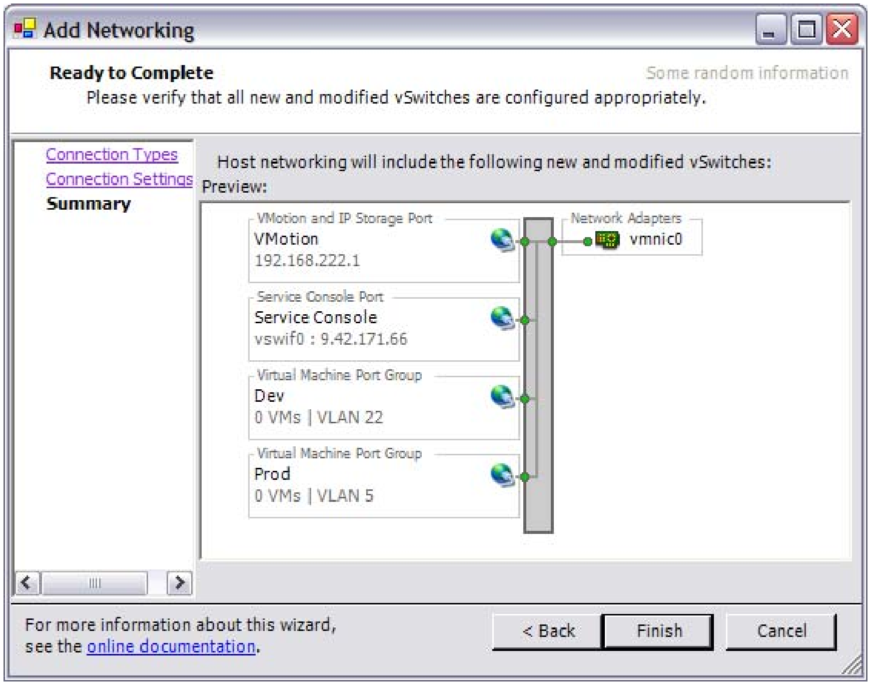
It is still recommended that the swap partition be twice the amount of RAM allocated to the service console. The other partitions will not need to change size based upon the number of virtual machines, amount of RAM, and so forth.

As you can see, even on a multi-node x3950 with 32 GB or more of RAM, we could fit everything on a 18 GB disk as opposed to ESX Server 2.5.x where we needed 72.3 GB disks. As you can see in Table 3-4 on page 108, we no longer need the /home or /vmimages partitions from ESX Server 2.5.x because all the virtual machine configuration files as well as all your ISO files are now stored on shared storage, SAN, iSCSI, or NAS.

Network configuration

As we discuss in 3.4.2, “ESX Server 3 design” on page 119, there have been a number of changes to networking in ESX Server 3. It will take time for VMware and the user community to develop best practices regarding all the new network features. Initially we can recommend the following:

- Add your additional NICs to the first virtual switch that was created during install and add ports to this switch for your virtual machines, VMotion, NAS storage, and so forth (Figure 3-14).



Add Networking

Ready to Complete

Please verify that all new and modified vSwitches are configured appropriately.

Some random information

Host networking will include the following new and modified vSwitches:

Preview:

VMotion and IP Storage Port
VMotion
192.168.222.1

Service Console Port
Service Console
vswif0 : 9.42.171.66

Virtual Machine Port Group
Dev
0 VMs | VLAN 22

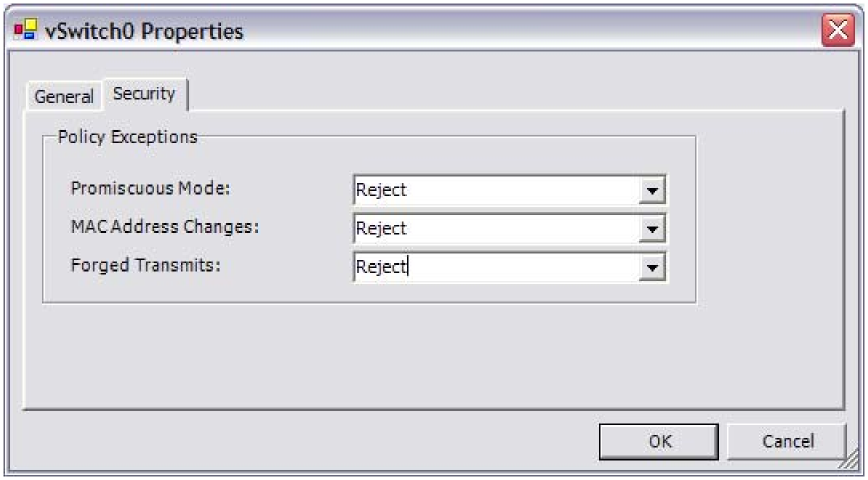
Virtual Machine Port Group
Prod
0 VMs | VLAN 5

Network Adapters
vmnic0

For more information about this wizard, see the online documentation.

Figure 3-14 ESX Server 3 virtual switch port groups

- Change the default security policy on the vSwitch to **Reject** for all three; Promiscuous, MAC Change, and Forged Source (Figure 3-15 on page 110). Unless you have a need for any of these in your organization, it is probably best for security reasons to set to disallow.



Promiscuous Mode: Reject

MAC Address Changes: Reject

Forged Transmits: Reject

Figure 3-15 vSwitch security settings

Storage configuration

One recommendation we can make at this time is that if you plan on using iSCSI storage, you should use the hardware-based initiator with the QLogic QLA4010 HBA and not the software-based initiator built into ESX Server. The iSCSI HBA has a TCP Offload Engine that will be higher performance and much less overhead than using the software-based initiator.

Time synchronization

It is important that ESX Server keep accurate time. To sync ESX Server with a NTP server follow the directions as outlined in VMware KB Answer ID# 1339:

`http://www.vmware.com/support/kb/enduser/std_adp.php?p_faqid=1339`

VMware also recommends that you sync your virtual machine's time with the ESX Server's time. This is done as a function of the VMware Tools that are installed in the virtual machines. For more detailed information about timekeeping see the VMware white paper *Timekeeping in VMware Virtual Machines* available from:

`http://www.vmware.com/pdf/vmware_timekeeping.pdf`

Service console firewall

Another new feature in ESX Server 3 is that the service console firewall is enabled by default. In its default setting, all ports needed for communiction to and from VirtualCenter are open. If you are not going to run any additional applications or agents within the service console, you can leave the firewall as is and get the added security without any configuration. However if you plan to install IBM Director Agent for example, you must reconfigure the firewall. See 5.3.3, “Configuring the ESX Server 3.0 firewall” on page 226 for details.

For initial testing purposes or troubleshooting you might want to disable the firewall temporarily, you can do this by logging onto the service console and entering the following command:

```
esxcfg-firewall -u
```

This will disable the firewall until the next reboot. Figure 3-16 shows the options that can be used with the **esxcfg-firewall** command. This figure displays the output of the **esxcfg-firewall -help command**.

```
esxcfg-firewall <options>
-q|--query                                  Lists current settings.
-q|--query <service>                        Lists setting for the
                                            specified service.
-q|--query incoming|outgoing                Lists setting for non-required
                                            incoming/outgoing ports.
-s|--services                               Lists known services.
-l|--load                                   Loads current settings.
-r|--resetDefaults                          Resets all options to defaults
-e|--enableService <service>                Allows specified service
                                            through the firewall.
-d|--disableService <service>               Blocks specified service
-o|--openPort <port,tcp|udp,in|out,name>    Opens a port.
-c|--closePort <port,tcp|udp,in|out>        Closes a port previously opened
                                            via --openPort.
   --blockIncoming                          Block all non-required incoming
                                            ports  (default value).
   --blockOutgoing                          Block all non-required outgoing
                                            ports (default value).
   --allowIncoming                          Allow all incoming ports.
   --allowOutgoing                          Allow all outgoing ports.
-h|--help                                   Show this message.
```

Figure 3-16 Service Console firewall options

See Table 5-2 on page 227 for a list of what firewall ports you must open for IBM Director.

3.4 Architecture and design

In this section, we focus on the architecture considerations of a scale-up design with the x3950.

Even though there are many new and useful features of ESX Server 3.0, it is likely that many organizations will continue to use and deploy ESX Server 2.5.x for some time. With this in mind, we next discuss architecture and design considerations on both ESX Server 2.5.x and ESX Server 3.0.

3.4.1 ESX Server 2.5.x design

This Redbook is not designed to replace the documentation already available from VMware and other sources. For detailed information about how to install and use ESX Server 2.5.x see the documentation provided by VMware:

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

In discussing architecture and design, we assume that the environment consists of a minimum of two ESX Server systems, shared SAN storage, VirtualCenter and VMotion.

Overview of ESX Server 2.5.x specifications

ESX Server 2.5.x has the following specifications:

- Physical ESX Server:
  - 16 logical processors per system
  - 80 virtual CPUs in all virtual machines per ESX Server system
  - 64 GB of RAM per ESX Server system
  - Up to 8 swap files, with a maximum file size of 64 GB per swap file
  - 64 adapters of all types per system
  - Up to 8 Gigabit Ethernet or 16 10/100 Ethernet ports per system
  - Up to 32 virtual machines per virtual switch
  - 16 host bus adapters per ESX Server system
  - 128 logical unit numbers (LUNs) per storage array
  - 128 LUNs per ESX Server system
- ESX Server 2.5.x virtual machines:
  - Up to two virtual CPUs per virtual machine with the optional vSMP module
  - Up to 3.6 GB of RAM per virtual machine
  - Up to four virtual SCSI adapters and up to 15 SCSI disks
  - Virtual disk sizes up to nine TB
  - Up to four virtual Ethernet network adapters

For the latest list of supported guest operating systems and qualified hardware see the Systems Compatibility Guide:

http://www.vmware.com/vmtn/resources/esx_resources.html

Note: At the time of this writing, there is a known bug with x3950 servers with 64 GB of RAM with ESX 2.5.2. This problem is expected to be fixed in ESX 2.5.3. In the mean time, the work around is to use 60 GB or less RAM in the x3950 with ESX 2.5.2. For details, see RETAIN® tip H185228:

http://www.pc.ibm.com/support?page=MIGR-62310

Virtual Infrastructure with ESX Server 2.5.x

With ESX Server 2.5.x and VirtualCenter 1.3 virtual infrastructure consists of the following components:

- ESX Server 2.5.x
- VirtualCenter 1.3
- vSMP
- VMotion

ESX Server runs on a physical server, while VirtualCenter can either run on a separate physical server or in a virtual machine. One thing to consider if you choose to run VirtualCenter in a VM is that if the parent ESX Server system goes offline, you will not have access to VirtualCenter until the server is back online or you restart the virtual machine on another host. vSMP and VMotion are features already installed and are unlocked with a license key.

VMware offers a Virtual Infrastructure Node (VIN) license that includes the following software licenses:

- ESX Server license
- Virtual SMP license
- VirtualCenter Agent license
- vMotion license

The VIN license offers considerable savings over buying all the individual licenses separately.

Number of servers and server sizing

The number of servers that suit your needs is dependant on a several factors, including:

- Scope of current project
- Future growth estimates
- High availability and disaster recovery plans
- Budgetary constraints

There are a number of different methods to try to calculate the number of ESX Serer systems you will need. Here are two of the more popular methods.

- The easiest rule of thumb is 4-5 virtual CPUs per physical CPU.

  This would result in 16-20 virtual machines per 4 way host, or 32 to 40 per 8 way host, assuming all were one vCPU virtual machines and low to moderate workloads.

  For memory, if you assume 1 GB per virtual machine. That should provide enough memory in most cases for virtual machines, the service console, and virtualization overhead. If you plan on running multiple, memory-intensive workloads, consider increasing this number.

  From these calculations we can arrive at an 8-way (two-node) x3950 with 32 GB of RAM which could support 32 virtual machines, and a 16-way (4-node) x3950 with 64 GB of RAM, which could support 64 virtual machines.

  These calculations assume single-core CPUs. Because a dual-core CPU will not provide 100% the performance of 2 single-core CPUs we recommend you count a dual-core CPU as 1.5 physical CPUs, resulting in 6-7 virtual machines per CPU socket.

- If you are consolidating a number of existing physical servers, then another method that can be employed is to record the average peak CPU utilization of the physical machines and convert this into a total of MHz used.

  For example, if you have a physical server with two 500 MHz CPUs that have an average peak utilization of 50%, then your total would be 500 MHz of CPU for this system.

  To get an average peak utilization, you must record CPU utilization for at least a week during the normal hours when the application is being used. A month is recommended for the most accurate information. If you already use an enterprise monitoring tool such as IBM Tivoli®, HP OpenView, NetIQ, and so on, then you might already have all the data you need.

  The next step is to add up the total CPU clock speeds of your ESX Server system. For example: A two-node 8-way x3950 with 3 GHz CPUs would have a total of 24,000 MHz.

  a. From this total subtract 10% for the console OS, this gives us 21,600 MHz.

  b. Subtract a certain amount for additional peak utilization and overhead; 20% is a safe number to use.

  c. This gives us 17,280 MHz to work with for our virtual machines.

  d. Divide that number against the 500 MHz of average peak utilization we first determined. The yield is about 34 virtual machines (17,280/500=34.5).

  You can do similar calculations to determine how much memory you need, as well. Take the average peak memory utilization of your physical servers, add

54 MB per system for virtualization overhead, add 32 MB for any systems whose average peak is over 512 MB. This is the amount of RAM needed for your VMs. Then add the amount of RAM assigned to the Service Console (512 MB would be an appropriate starting number on an 8-way ESX Server system), add 24 MB for the VMkernel, and this is the total amount of RAM needed.

For example, if you had 10 physical systems to virtualize and each had an average peak memory utilization of 512 MB, then that would equal 5120 MB. Add 54 MB each for virtualization overhead (5120+540=5660 MB). This is the total amount of RAM for the VMs. Add 512 MB for the Service Console (5660+512= 6172 MB) and 24 MB for the VMkernel (6172+24=6196) and this is the total amount of RAM needed to run these 10 VMs: 6 GB of RAM.

Both methods provide very similar results in the number of virtual machines you could support on an 8-way x3950 server. Our experience is that in most organizations, these two methods usually result in a similar number of virtual machines per host. Therefore to save yourself some time, we recommend you use the first method for initial sizing of your ESX servers.

The exact mix of virtual machines and applications running will affect how many virtual machines you can run. Unfortunately there is no one formula that will calculate exactly how many virtual machines you can run. The low end of the recommendations we illustrated here should provide a realistic and conservative target for most organizations. In reality you could end up supporting more or fewer virtual machines.

Future growth is harder to determine. The cycle that happens in many organizations which implement VMware’s Virtual Infrastructure model is:

1. At first, they are resistant to the idea of virtual machines.
2. After they see all the benefits of the virtual infrastructure and that they are not losing performance, the number of requests for new virtual machines can grow rapidly.
3. The result can be an over commitment of processor, memory, and I/O resources and a subsequent loss in overall performance.

To avoid this cycle, one recommendation is that you follow the same purchasing, approval, and change management procedures for virtual machines as you do for physical systems. While the process can usually be streamlined and shorted for virtual machines, having a formal process in place to request virtual machines as well as a way to associate costs to each new virtual machine, you will have much better control over your virtual machine growth and a better idea of future growth.

VMotion considerations

When designing your virtual infrastructure, an important consideration is VMotion. *VMotion* is the feature that allows the migration of a virtual machine from one physical ESX Server system to another while the virtual machine is running. Because VMotion transfers the running architecture state of a virtual machine between two physical hosts, the CPUs of both physical hosts must be able to execute the same instructions. At a bare minimum this means for VMotion to work your servers CPUs must be:

- Same vendor class (Intel or AMD)
- Same processor family (Pentium III, Pentium 4, Opteron, etc)

Sometimes there are significant changes to processors in the same family that have different extended features, such as 64-bit extensions and SSE3. In these cases VMotion might not work, even though the CPUs are in the same processor family. CPU speed and cache level are not an issue, but the extended features will cause problems or VMotion failure if they are different on the target and host servers.

For example, because the x366 and x260 use the same processors as the x3950, these servers would be suitable candidates for joining some x3950s in a VMotion configuration. However other xSeries servers with different processors will not.

VMotion also requires its own dedicated Ethernet controller and network. A Gigabit Ethernet network is listed as required for VMotion by VMware, but it is possible to use VMotion with a 100 Mbps network if was your only option, although migration times will increase significantly.

Another important requirement for VMotion is shared storage. The ESX Server systems that you are going to run VMotion across need to be zoned so that all LUNs are visible to all hosts.

Planning your server farm

With VirtualCenter 1.x, a *farm* is a group of ESX Server systems that can be used to organize your virtual infrastructure. A farm is also a *VMotion boundary*, meaning that all servers in a VMotion configuration must be defined in the one farm. In your planning, you need to think about how many hosts are going to be in each farm. There are a few guidelines provided by VMware to help you decide. VMware recommends:

- No more than 16 ESX Server systems connected to a single VMFS volume
- No more than 32 I/O-intensive virtual machines per LUN, and no more than 100 low-I/O virtual machines per LUN
- No more than 255 files per VMFS volume

- Up to 2 TB limit on storage

Because VMotion requires shared storage, then the upper limit per farm would be 16 ESX Server systems per farm. You might want to create smaller farms for a number of reasons. The lower limit is two servers, assuming you are using VMotion.

Storage sizing

Like server sizing, there is no one universal answer that can be applied to every organization. The previous section lists that there should not be more than 32 I/O-intensive virtual machines per VMFS volume, and staying within this limit should reduce any resource contention or SCSI locking issues.

There are a number of ways to determine the most appropriate size of your VMFS volume. Here is one of the easier ways.

Say you have decided that two 8-way x3950 servers with 32 virtual machines on each server will meet your processing requirements. Using the 32 virtual machines per LUN guideline, this would mean that two LUNs are needed. If you are creating new virtual machines, you can estimate the average size of the virtual disks. If we use 20 GB of disk per VM, this would give us 640 GB per LUN. Consider adding a little additional space for growth, Ten percent is a good rule of thumb, bringing us to 720 GB. If you are planning on using redo logs, you might want to add additional space for that as well.

Planning for networking

There are various options when it comes to designing the networking portion of your server farm. The options chosen are often based on the characteristics of your physical network and networking and security policies of your company. One important factor is if a Gigabit Ethernet network is available. While not absolutely required for ESX Server, a Gigabit network is highly recommended.

In ESX Server 2.5.x there are three basic components you should consider.

- Service console

  It is recommended that the service console have its own dedicated NIC for performance and security reasons. If you have a separate management network in your data center, then this is where you want to locate the service console NIC.

  In a default configuration, a 100 Mbps Ethernet controller is sufficient bandwidth for the service console. If you are planning on also using the service console for backups or other high bandwidth functions, then a Gigabit NIC is recommended.

- Virtual machines

  The virtual machines use a separate network from the service console. Once again a Gigabit network is not required but is highly recommended, because 32 virtual machines will be generating significant network traffic.

  A good rule of thumb is 10-20 virtual machines per Gigabit Ethernet controller. This means we need a minimum of 2 Gigabit Ethernet NICs for an 8-way x3950 running 32 VMs. Remember that this is the minimum recommendation. Adding one or two more should guarantee enough network bandwidth available for all virtual machines.

  Another important consideration is if you have multiple VLANs in your data center that you want to make available to the virtual machines. When using multiple VLANs with ESX Server 2.5.x, you have two options:

  - Install a physically separate NIC for every network you want available. If you only had a few networks you wanted to use, then this would be a viable option. However, if you have 10 different networks, then this is obviously not a practical solution. Remember that ESX Server 2.5.x only supports a maximum of eight gigabit network cards.
  - Use ESX Server's support for VLAN tagging (802.1q). Using this option means you can create a virtual switch with a separate port group for each VLAN you want to use. If your physical switches support this, then this is the recommended option.

  One other consideration is redundancy for your virtual machine's networking. With ESX Server 2.5.x, you can have multiple NICs connected to one virtual switch not only to combine bandwidth, but also to provide redundancy in case of a failure of one of the NICs or a physical cable.

  For higher availability, our recommendation is to have a minimum of two gigabit NICs connected to the virtual switch used by your virtual machines. See 3.3, "Configuration best practices" on page 98 for more information about NIC configuration in the x3950 Server.

- **VMotion**

  VMware lists a separate Gigabit Ethernet network as a requirement for VMotion. It is possible to use VMotion with a 100 Mbps network, but performance might not be acceptable and it is not recommended. You should have a separate physical gigabit NIC for your VMotion network and a separate subnet created for VMotion to use.

  If you only have two systems running ESX Server, then it is possible to use a crossover cable between the two servers for a VMotion network. This is also useful for troubleshooting VMotion problems.

Network load balancing

ESX Server 2.5.x provides two methods for network load balancing for the virtual machines.

- MAC Out is the default method. Using this method requires no additional configuration in ESX Server. Simply connect two physical NICs to the virtual switch. No additional configuration on the physical switches is necessary. The only downside of this method is that is it not very efficient. Often, most virtual machines end up using the same physical NIC and there is no method to select manually what physical NIC each virtual machine uses.
- IP Out is an optional way to configure your networking for better load balancing. The downside to this method is that there is additional configuration steps required in ESX Server as well as your physical switches. You must configure your physical switches for 802.3ad (or EtherChannel in the case of Cisco switches). This is the recommended method for highest performance.

This is a brief overview of networking with ESX Server 2.5.x. Advanced topics such as backup networks, DMZ networks, traffic shaping, and detailed configuration steps are beyond the scope of this Redbook. For in depth information about networking and configuration steps, see the documentation on the VMware Web site:

http://www.vmware.com/support/pubs/esx_pubs.html
http://www.vmware.com/vmtn/resources/esx_resources.html

3.4.2 ESX Server 3 design

Given the early beta release we were working with, it was not possible at the time of this writing to offer concrete best practices for ESX Server 3. There are some changes in ESX Server 3 that will affect how you design your architecture that we highlight here. See 3.1, “What is new with ESX Server and VirtualCenter?” on page 80 for an overview of new features and functionality in ESX Server 3.

Virtual Infrastructure

See 3.1.2, “Virtual Infrastructure Overview” on page 84 to review the basic components of Virtual Infrastructure with ESX Server 3 and VirtualCenter 2.

With the addition of VMware High Availability (HA) in VirtualCenter 2, VMware is now recommending that you run your VirtualCenter 2 server in a virtual machine, because of the higher availability provided by HA. As of this writing, they do not have an official best practice statement on which is better; VirtualCenter on a physical server or in a virtual machine. Unless technical issues are found, we believe that running VirtualCenter within a VM would be the preferred method if you are using HA.

One of the other new components is the licensing server, based on FLEXnet. If you already have a FLEXnet licensing server in your organization, then the VirtualCenter 2 beta documentation indicates you can use that for VirtualCenter as well. This is a low-overhead application and our recommendation would be to install it on the VirtualCenter server if you do not have an existing FLEXnet server.

If you plan to use VMware Consolidated Backup, it will require a physical Windows server™ to act as the proxy backup server. At the time of this writing, we do not have any specific recommendations for the consolidated backup server.

Number of servers and server sizing

ESX Server 3 increases the number of logical CPUs supported to 32 and increase the number of virtual CPUs per physical server supported to 200. Also with 4-way virtual SMP and up to 16 GB of RAM per virtual machine supported this will change the dynamics of how many virtual machines you can run on a physical ESX server.

Not having any definitive answers at this time, the guidelines of 4-5 virtual CPUs on single-core and 6-8 on dual-core CPUs is a good minimum expectation for ESX Server 3.

Planning your server farm and VMotion considerations

In VirtualCenter 2, there is no more farm object. See 3.1.3, “VirtualCenter objects and concepts” on page 85 for an overview of the new objects within VirtualCenter 2. The Datacenter object in VirtualCenter 2 is now the VMotion boundary, so it is similar to the farm in this respect, but it can also contain other objects such as clusters, folders, and resource groups.

You might want to create multiple VMotion boundaries within a datacenter. For example, you could have a datacenter called Boston and in it a folder called Intel and one called AMD because these CPUs would not allow VMotion between them. The same basic rules of VMotion still apply in VirtualCenter 2: the physical servers must have similar CPUs (as we discussed in “VMotion considerations” on page 116) as well as sharing the same storage.

As of this writing VMware did not have a best practices document available, but assured us that there would not be any decreases over what is supported in ESX Server 2.5.x.

There are a few changes in the VMFS 3 file system that are worth noting here:

- VMFS 3 supports 3300 files per VMFS volume and subdirectories.
- VMFS 3 supports 233 files and subdirectories per sub-directory.

- VMFS3 has Improved disk locking to increase scaling for access by a larger number of hosts to shared storage.

Given the increased abilities of ESX Server 3 and the VMFS 3 file system, it is likely the maximum ESX Server hosts per shared LUN and number of virtual machines per LUN will increase, but these number were not yet determined by VMware. We can assume the numbers provided in 3.4.1, “ESX Server 2.5.x design” on page 112 would be the minimum starting point for ESX Server 3.

Storage sizing

As we mentioned in the previous section, with the increased capacities of the VMFS3 file system, we expect increases in the number of ESX Server hosts per LUN and number of virtual machines per VMFS volume.

Because the upper limits had yet to be determined while we were writing, we could not make any additional recommendations over the guidelines for storage sizing in ESX Server 2.5.x in “Storage sizing” on page 117.

There are four important changes to consider in ESX Server 3 that will affect how you size your LUNs and shared storage:

- Swap files

  The virtual machine’s swap files are now stored on the shared storage and not on the local disk. Instead of one giant swap file for all the virtual machines, there is now one swap file per virtual machine equal to the size of RAM allocated to the virtual machine. Each virtual machine has its own folder on the VMFS volume and the swap file is in that folder.

  The result of these changes is that you must increase the size of your VMFS volume to accommodate the additional storage requirements of the swap files. For example if you have 20 virtual machines with an average of 1 GB of RAM allocated to each one, you need 20 GB of additional storage on the VMFS3 volume to accommodate the swap files.

- Configuration files

  The virtual machine’s configuration (.vmx) and NVRAM file are now stored on the shared storage in the same folder as the virtual machine and its swap file, and no longer on the local disks of ESX Server.

  This change was made largely in part to enable the HA feature of VirtualCenter 2,but also to ease the management of the virtual infrastructure. These files do not take much additional space, but you might want to add a small amount of storage per virtual machine.

- iSCSI and NAS support

  iSCSI and NAS are supported with ESX Server 3. This might change your strategy for designing your virtual infrastructure because you might want certain virtual machines on different types of storage. For example you might want to have development virtual machines on lower-cost iSCSI storage. See Table 3-5.

Table 3-5 ESX Server 3 storage comparison

Technology	Protocol	Interface	Performance	Notes
Fibre Channel	FC/SCSI	FC HBA	High	Highest performance due to dedicated network, also highest cost.
iSCSI	IP/SCSI	iSCSI HBA	Medium	Performance dependant on LAN. Low to medium cost.
NAS	IP/NFS	NIC	Medium to low	Performance dependant on LAN. Lowest cost

- Snapshots

  VMware ESX Server 3 supports multiple snapshots similar to the feature in VMware Workstation product. Because these snapshots will be located on the shared storage along with the virtual machine’s disk and configuration files, you must account for additional space if you will be using this feature.

Planning for networking

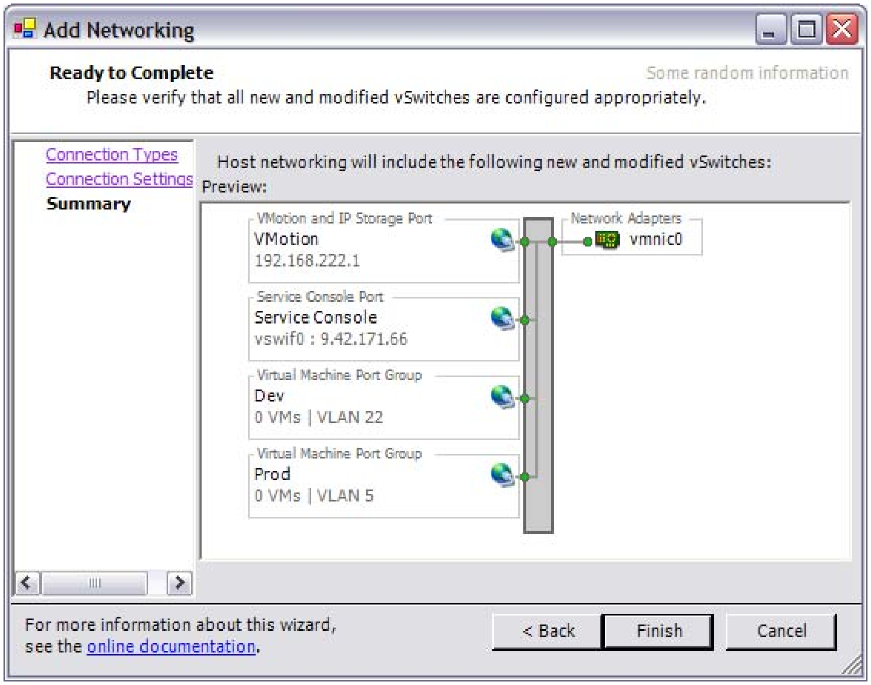
By default, when you install ESX Server 3, there will be one virtual switch created and the service console will be a port group on this virtual switch. You then add additional NICs to this virtual switch and create port groups for VMotion, virtual machines, and iSCSI connections. This is the preferred method according to VMware. The benefit of having this configuration is that you now have network redundancy for the service console, VMotion network, and virtual machines without having to increase the total number of NICs installed in the server.

You still must have one Gigabit NIC for each 10-20 virtual machines, one for VMotion, and one for the service console. The difference now is that they can all be connected to the same virtual switch and provide redundancy for all the port groups from the failure of a single NIC.

There can be reasons, still, to assign a physical NIC to just the service console. For example if your company mandates all management interfaces must be isolated on a management network, you can do this if you need to, it is just no longer a requirement.

To further expand on networking in ESX Server 3, here is an overview of some of the changes.

- All network devices are allocated to the VMkernel (no more vmkpcidivy).
- Service console and virtual machines access network through a virtual switch (vSwitch).
  - No distinction made between bonds, vmnets, and vmnics
  - Up to 32 uplink ports per vSwitch
  - Up to 1024 ports per vSwitch, and up to 1024 vSwitches per ESX server.
  - Up to 20 Gigabit and 26 10/100 NICs supported per ESX server.
- Each virtual switch has three different types of connections available. See Figure 3-17 to see all the different connections on one vSwitch.
  - Service console port allows the service console to access the physical network.
  - VMotion and IP storage port is used by the VMkernel TCP/IP stack to provide functions such as VMotion and access to NAS storage.
  - Virtual machine port group allows virtual machines to connect to the network.



Ready to Complete

Please verify that all new and modified vSwitches are configured appropriately.

Some random information

Host networking will include the following new and modified vSwitches:

Preview:

VMotion and IP Storage Port
VMotion
192.168.222.1

Service Console Port
Service Console
vswif0 : 9.42.171.66

Virtual Machine Port Group
Dev
0 VMs | VLAN 22

Virtual Machine Port Group
Prod
0 VMs | VLAN 5

Network Adapters
vmnic0

For more information about this wizard, see the online documentation.

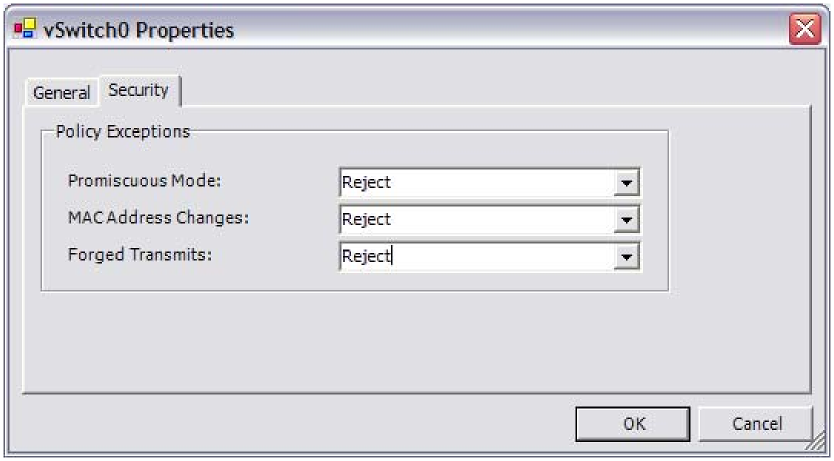
Figure 3-17 virtual switch connections

- Port groups have four uses:
  - VLANs (similar to ESX 2.5.x)
  - NIC teaming policy (now per port instead of per vSwitch)
  - Traffic shaping policy (now per port instead of per vSwitch)
  - Layer 2 security policies
- New NIC Teaming policies:
  - Source vSwitch port based (load balancing based on vSwitch port number)
  - Explicit failover (can have multiple failover links with priorities)
  - Different port groups can have different NIC teaming policies
- New Layer 2 security policy allows for enforcement of security policies at the Ethernet layer.

  There are three different policies as shown in Figure 3-18.

  Note: The default setting for all three is **Allow**.

  - **Promiscuous Mode**: This option prevents virtual machines from using promiscuous mode.
  - **MAC Address Changes**: This option prevents a guest OS from changing its MAC address.
  - **Forged Transmits**: This option prevents virtual machines from sending packets with a different MAC address than that for which they are configured.



Policy Exceptions

Promiscuous Mode: Reject

MAC Address Changes: Reject

Forged Transmits: Reject

Figure 3-18 vSwitch security policy

3.5 Installing ESX Server 3.0 and VirtualCenter 2

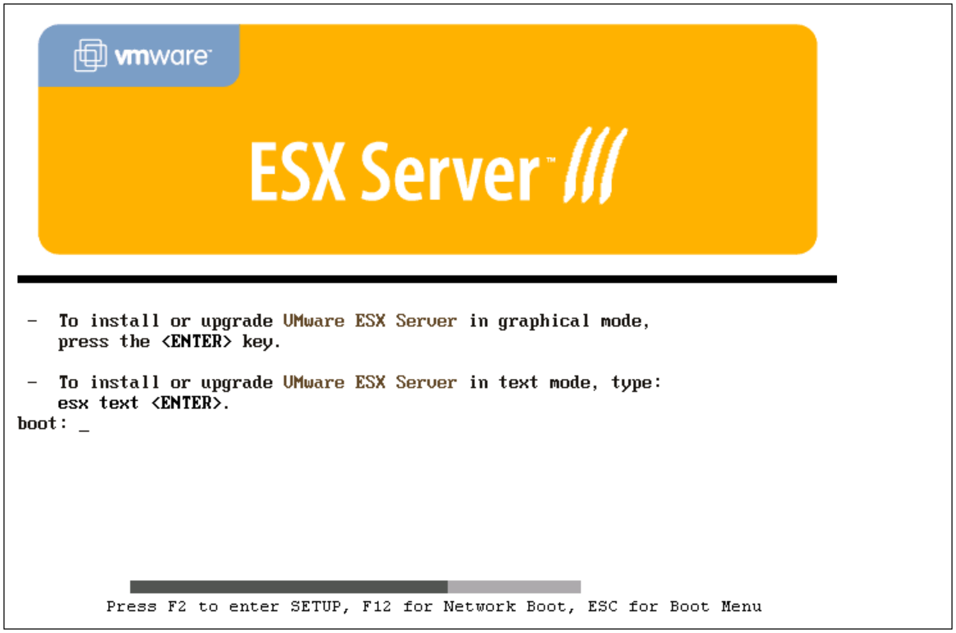
In this section, we describe how to install ESX Server 3.0 and VirtualCenter 2 on an System x3950 configuration.

We have included basic install instructions. Consult VMware's documentation for detailed instructions on installing, configuring, and using ESX Server and VirtualCenter.

3.5.1 ESX Server 3 installation

To perform a basic installation on a local disk, perform the following steps. Consult the *ESX Server 3 Installation Guide* for more complex installation options such as boot from SAN.

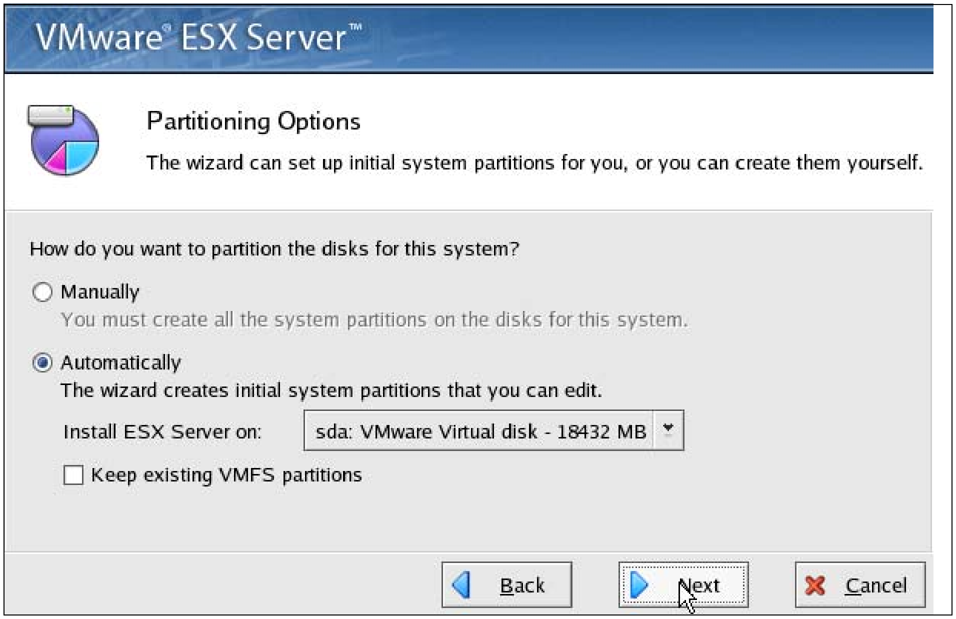
1. Boot from the ESX Server 3 CD-ROM and you can see the initial installation splash screen, as in Figure 3-19. Press Enter to install in graphical mode.



```
 -  To install or upgrade VMware ESX Server in graphical mode,
    press the <ENTER> key.

 -  To install or upgrade VMware ESX Server in text mode, type:
    esx text <ENTER>.
boot: _

        Press F2 to enter SETUP, F12 for Network Boot, ESC for Boot Menu
```

Figure 3-19 ESX Server 3 install splash screen

2. Select **Manually** or **Automatically** (default) on the disk partitioning screen as shown in Figure 3-20 on page 126. Choosing **Automatically** permits you to change partitions. Click **Next** to continue.



VMware® ESX Server™

Partitioning Options

The wizard can set up initial system partitions for you, or you can create them yourself.

How do you want to partition the disks for this system?

○ Manually

You must create all the system partitions on the disks for this system.

● Automatically

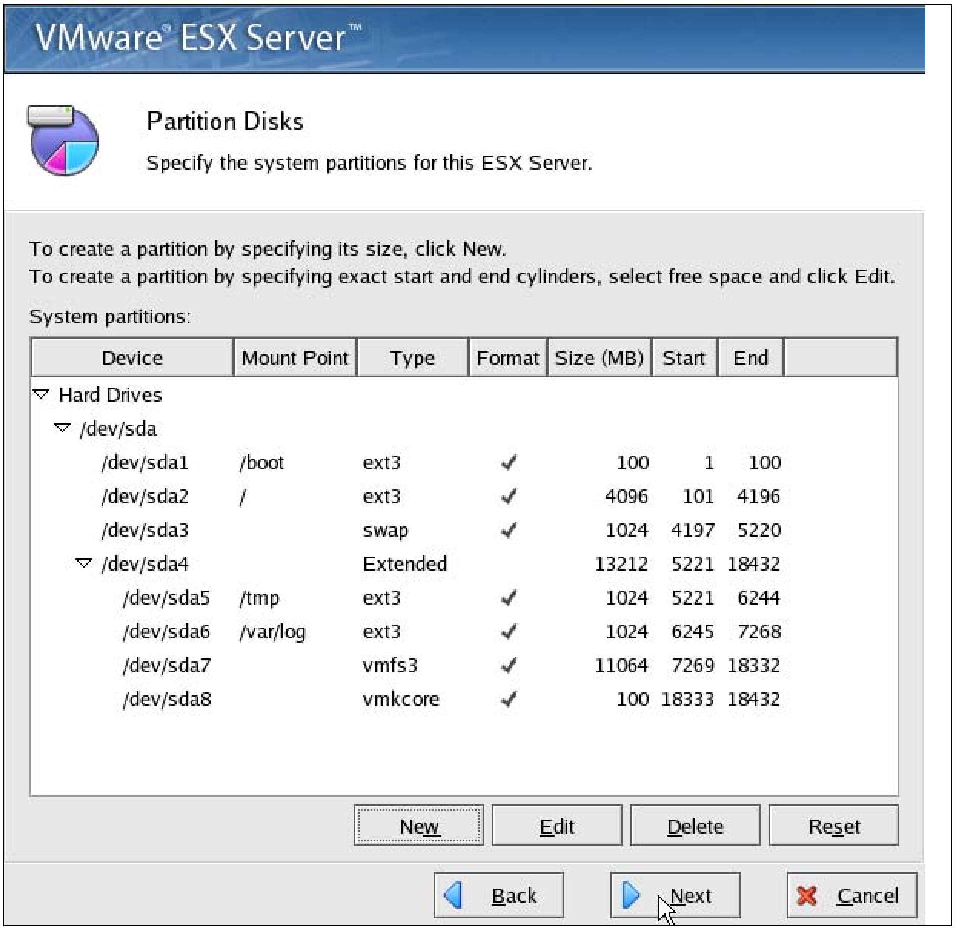
The wizard creates initial system partitions that you can edit.

Install ESX Server on: sda: VMware Virtual disk - 18432 MB

□ Keep existing VMFS partitions

Figure 3-20 Auto or manual disk partitioning

3. Edit your disk partitions to match the recommendations in 3.3.2, “ESX Server 3 configuration best practices” on page 104. Click **Next** when you are finished. See Figure 3-21.



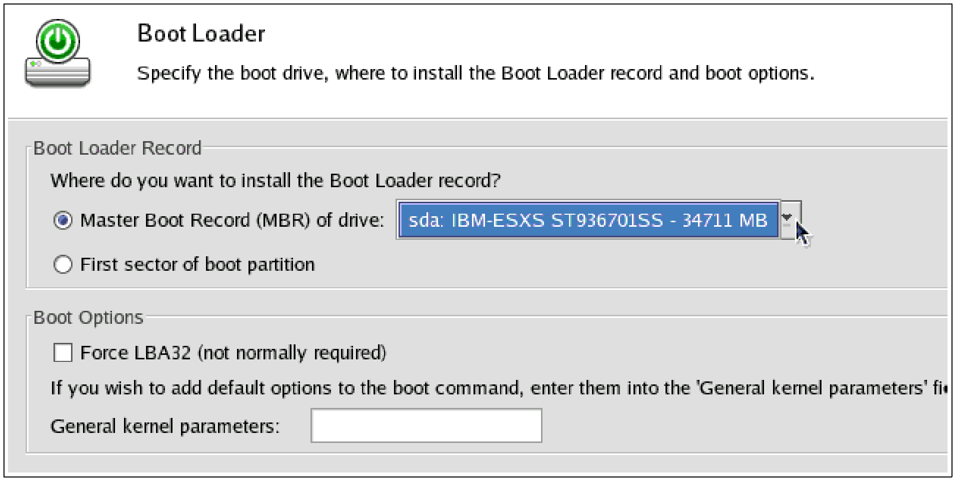
To create a partition by specifying its size, click New.
To create a partition by specifying exact start and end cylinders, select free space and click Edit.

System partitions:

Device	Mount Point	Type	Format	Size (MB)	Start	End
▽ Hard Drives						
▽ /dev/sda						
/dev/sda1	/boot	ext3	✓	100	1	100
/dev/sda2	/	ext3	✓	4096	101	4196
/dev/sda3		swap	✓	1024	4197	5220
▽ /dev/sda4		Extended		13212	5221	18432
/dev/sda5	/tmp	ext3	✓	1024	5221	6244
/dev/sda6	/var/log	ext3	✓	1024	6245	7268
/dev/sda7		vmfs3	✓	11064	7269	18332
/dev/sda8		vmkcore	✓	100	18333	18432

Figure 3-21 Partition disks

4. Select the Boot Loader location. We are using the default of the Master Boot Record (MBR) on our first disk as shown in Figure 3-22. See the *ESX Server 3 Installation Guide* for more information about the other options. Click **Next** to continue.



Boot Loader

Specify the boot drive, where to install the Boot Loader record and boot options.

Boot Loader Record

Where do you want to install the Boot Loader record?

● Master Boot Record (MBR) of drive: sda: IBM-ESXS ST936701SS - 34711 MB

○ First sector of boot partition

Boot Options

□ Force LBA32 (not normally required)

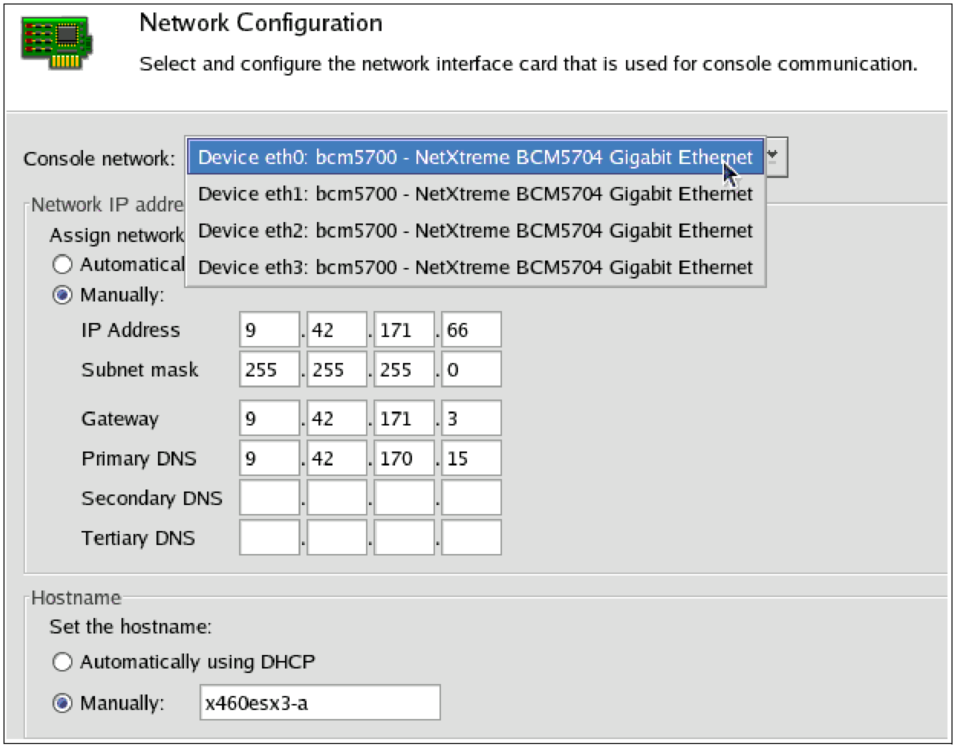
If you wish to add default options to the boot command, enter them into the 'General kernel parameters' fi

General kernel parameters:

Figure 3-22 Select boot loader

5. Select which NIC you want to use for the service console. Enter your IP address information, DNS servers, and host name for your server as shown in Figure 3-23 on page 129. As you can see, you now have the option to use any NIC in the system for the service console during the install. Click **Next** to continue.

   > **Note:** It is recommended always to use a static IP and fully qualified domain name.


Network Configuration

Select and configure the network interface card that is used for console communication.

Console network: Device eth0: bcm5700 - NetXtreme BCM5704 Gigabit Ethernet

Device eth1: bcm5700 - NetXtreme BCM5704 Gigabit Ethernet
Device eth2: bcm5700 - NetXtreme BCM5704 Gigabit Ethernet
Device eth3: bcm5700 - NetXtreme BCM5704 Gigabit Ethernet

Network IP addre

Assign network

○ Automatical

● Manually:

IP Address	9	42	171	66
Subnet mask	255	255	255	0
Gateway	9	42	171	3
Primary DNS	9	42	170	15
Secondary DNS				
Tertiary DNS				

Hostname

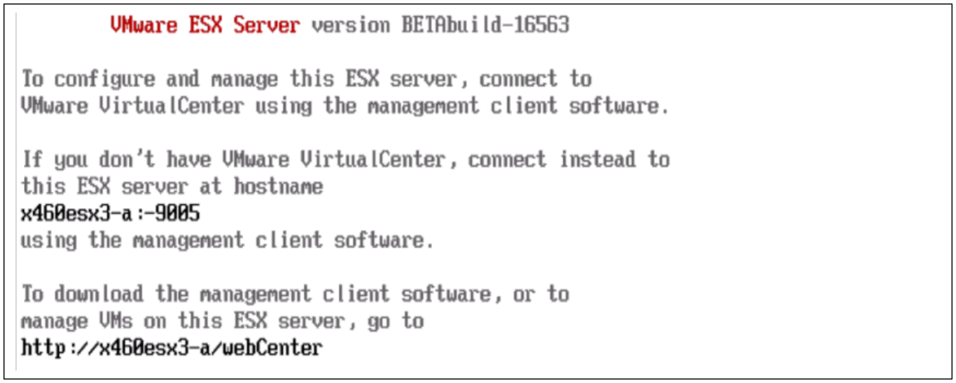
Set the hostname:

○ Automatically using DHCP

● Manually: x460esx3-a

Figure 3-23 Network configuration

6. After the networking portion, the installer reboots and the ESX Server installation is complete.
7. You should see a screen similar to Figure 3-24 on page 130. You can use the VirtualCenter Client to connect directly to ESX server because there is no longer a MUI.



```
        VMware ESX Server version BETAbuild-16563

To configure and manage this ESX server, connect to
VMware VirtualCenter using the management client software.

If you don't have VMware VirtualCenter, connect instead to
this ESX server at hostname
x460esx3-a:-9005
using the management client software.

To download the management client software, or to
manage VMs on this ESX server, go to
http://x460esx3-a/webCenter
```

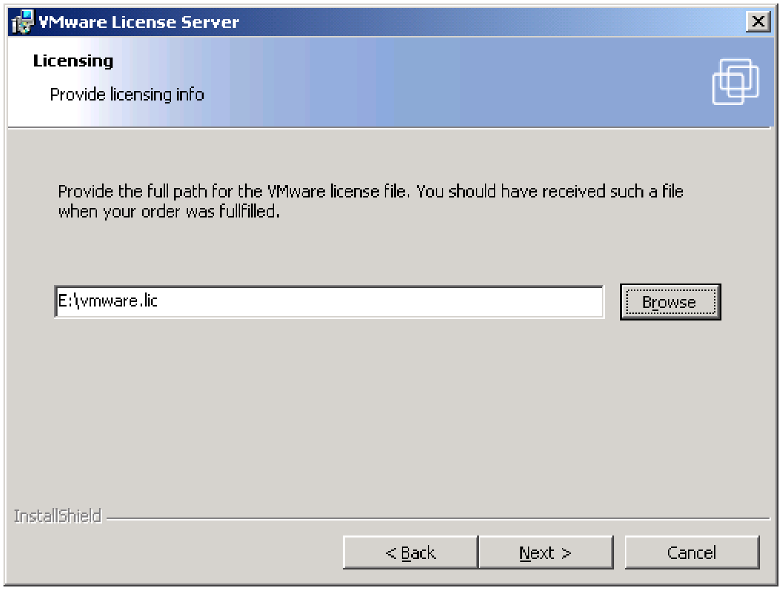
Figure 3-24 ESX Sever 3 install complete

3.5.2 Licensing Server installation

A new component included is the *Licensing Server*. This is the central repository for all your licenses. The license server can be installed on the VirtualCenter server, in a virtual machines, or you can use an existing FLEXnet license server.

> **Note:** It was not clear at the time of this writing if the Licensing Server will remain a separate installer, or be part of the VirtualCenter installation.

1. Run **`vlsetup.exe`** to start the install of the licensing server. You will see the license server install wizard.
2. Accept the license agreement and click **Next** to continue.
3. Enter User Name and Organization in the Customer Information screen and click **Next** to continue.
4. Select the destination folder for installation and click **Next** to continue.
5. Provide a path to your licensing file. Click **Next** to continue. See Figure 3-25 on page 131.



VMware License Server

Licensing

Provide licensing info

Provide the full path for the VMware license file. You should have received such a file when your order was fullfilled.

E:\vmware.lic

Figure 3-25 Path to licensing file

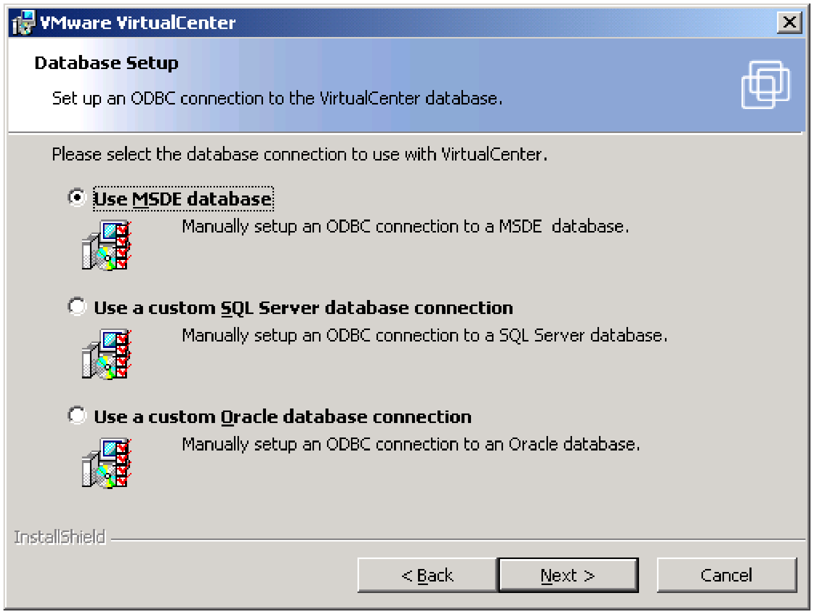
6. Click **Install** to begin the installation.
7. Click **Finish** when the installation is complete.

3.5.3 VirtualCenter 2 installation

In this section, we illustrate a basic installation of VirtualCenter 2. See VMware's documentation for detailed information about installing, configuring, and using VirtualCenter.

1. Launch the VirtualCenter 2 installer and click **Next** to continue on the Welcome screen.
2. Accept the licensing agreement and click **Next** to continue.
3. Enter your User Name and Organization on the Customer Information screen and click **Next** to continue.
4. Select your destination folder for installation and click **Next** to continue.
5. Select Typical or Custom setup. In this example we are doing a **Typical** installation. Click **Next** to continue.
6. Select what database to use. In this example we are using **Microsoft SQL Server 2000 Desktop Engine (MSDE)**. See Figure 3-26 on page 132. Click **Next** to continue.

Note: Microsoft Access is no longer a supported database type, for demo and evaluation purposes you can use MSDE.



Database Setup

Set up an ODBC connection to the VirtualCenter database.

Please select the database connection to use with VirtualCenter.

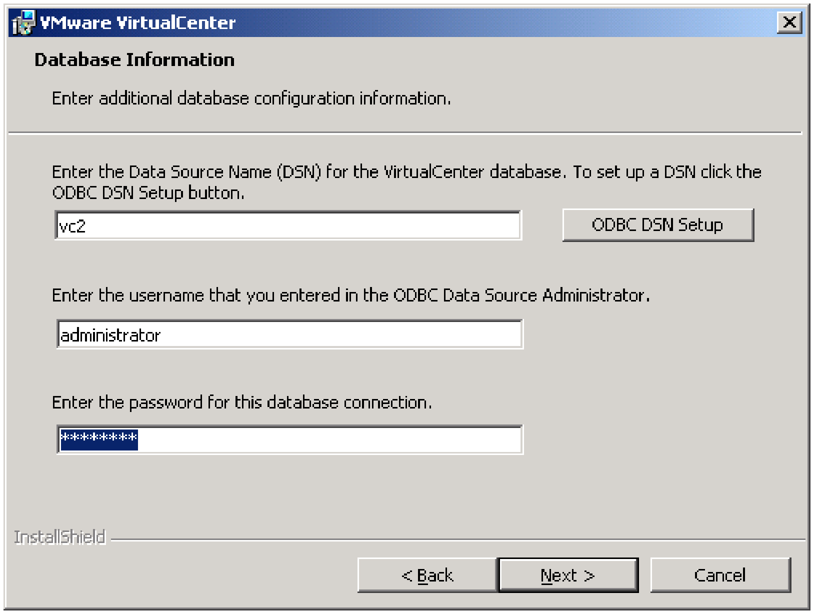
● Use MSDE database
Manually setup an ODBC connection to a MSDE database.

○ Use a custom SQL Server database connection
Manually setup an ODBC connection to a SQL Server database.

○ Use a custom Oracle database connection
Manually setup an ODBC connection to an Oracle database.

Figure 3-26 Select database type

7. Configure your ODBC connection and click **Next** to continue. See Figure 3-27.



VMware VirtualCenter

Database Information

Enter additional database configuration information.

Enter the Data Source Name (DSN) for the VirtualCenter database. To set up a DSN click the ODBC DSN Setup button.

vc2

Enter the username that you entered in the ODBC Data Source Administrator.

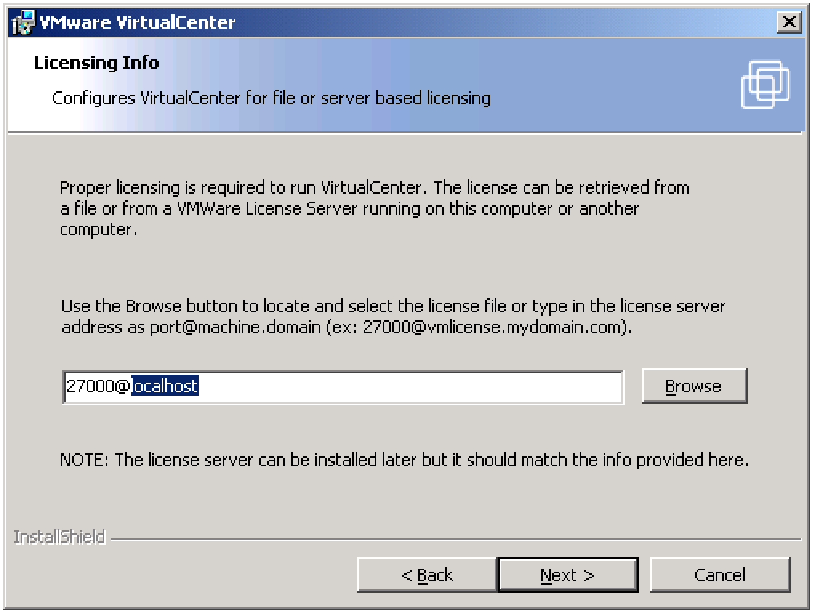
administrator

Enter the password for this database connection.

****

Figure 3-27 Enter database information

8. Enter the name of your Licensing Server as shown in Figure 3-28. You can point to a licensing file, or your licensing server. If the licensing server is on the same system as VirtualCente,r you can enter `27000@localhost` as we have done, or use a fully-qualified domain name.



VMware VirtualCenter

Licensing Info

Configures VirtualCenter for file or server based licensing

Proper licensing is required to run VirtualCenter. The license can be retrieved from a file or from a VMWare License Server running on this computer or another computer.

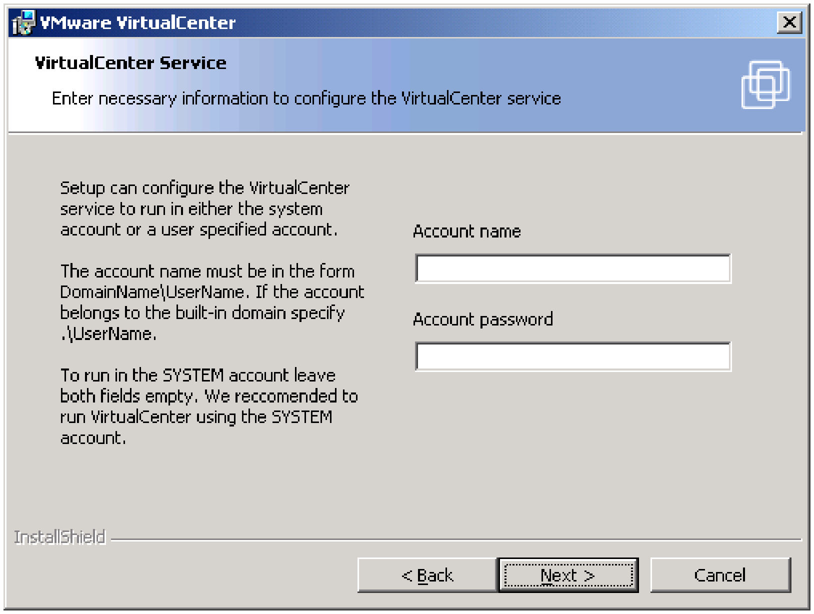
Use the Browse button to locate and select the license file or type in the license server address as port@machine.domain (ex: 27000@vmlicense.mydomain.com).

27000@localhost

NOTE: The license server can be installed later but it should match the info provided here.

Figure 3-28 Enter License Server information

9. Enter VirtualCenter service information. We recommend you run the service as the system account (leave both fields blank) as shown in Figure 3-29.



VMware VirtualCenter

VirtualCenter Service

Enter necessary information to configure the VirtualCenter service

Setup can configure the VirtualCenter service to run in either the system account or a user specified account.

The account name must be in the form DomainName\UserName. If the account belongs to the built-in domain specify .\UserName.

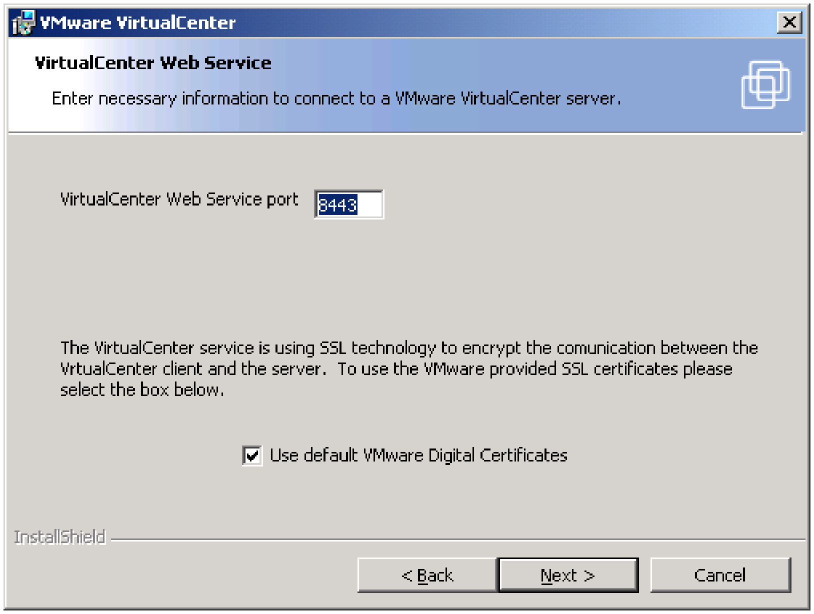
To run in the SYSTEM account leave both fields empty. We reccomended to run VirtualCenter using the SYSTEM account.

Account name:

Account password:

Figure 3-29 VirtualCenter Service account

10. In the VirtualCenter Web Service configuration screen (Figure 3-30), you can change the default port (8443) if you desire. You can also deselect **Use default VMware Digital Certificates** if you want to use your own certificates. See the official VMware documentation for more information. Click **Next** to continue.



VMware VirtualCenter

VirtualCenter Web Service

Enter necessary information to connect to a VMware VirtualCenter server.

VirtualCenter Web Service port 8443

The VirtualCenter service is using SSL technology to encrypt the comunication between the VrtualCenter client and the server. To use the VMware provided SSL certificates please select the box below.

☑ Use default VMware Digital Certificates

Figure 3-30 VirtualCenter Web Service

11. Click **Install** on the ready to install screen. Click **Back** to make any changes.

12. When the installation is complete, click **Finish** to exit.

3.5.4 VirtualCenter client install

You must install the VirtualCenter client not only for connecting to VirtualCenter, but also ESX Server 3 because there is no longer a MUI interface. To install the VirtualCenter client, follow these steps.

1. Launch the installer and click **Next** on the welcome screen to continue.
2. Accept the license agreement and click **Next** to continue.
3. Enter your User Name and Organization in the Customer Information screen; click **Next** to continue.
4. Select the destination path for installation and click **Next** to continue.

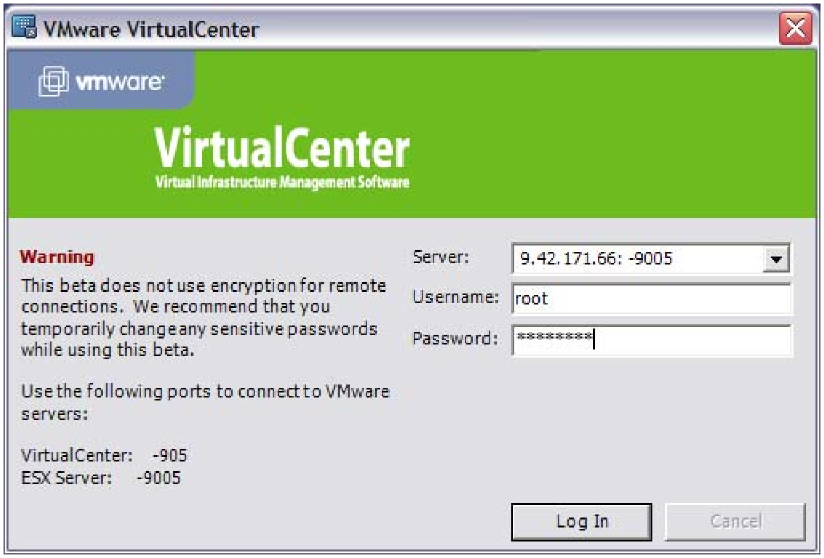
5. Click **Install** on the Ready to Install screen. Click **Back** if you want to make any changes.
6. When the installation is complete, click **Finish** to exit the setup program.

To connect to VirtualCenter or ESX Server, double-click the VirtualCenter Client icon on your desktop and enter one of the following in the Server field of the VirtualCenter window (Figure 3-31):

- To connect to ESX Server enter *servername:* -9005
- To connect to VirtualCenter enter *servername*: -905

Note: You can also use the IP address instead of a server name.

Make sure you enter proper credentials for each one.



Warning

This beta does not use encryption for remote connections. We recommend that you temporarily change any sensitive passwords while using this beta.

Use the following ports to connect to VMware servers:

VirtualCenter: -905
ESX Server: -9005

Server: 9.42.171.66: -9005
Username: root
Password: *********

Figure 3-31 VirtualCenter client

3.6 Load balancing and fault tolerance considerations

The System x3950 is an ideal platform for running ESX Server because of its high availability hardware features as well as its capabilities with very heavy workloads and dynamically adjusting resource assignments across the multi-node configuration.

You can also use the advanced VMware features such as VMware High Availability (formerly Distributed Availability Services) and Distributed Resource

Scheduler (DRS) to enhance the x3950 hardware capabilities to ensure the highest uptime and the best resource balancing on a virtual infrastructure. For more background information regarding HA and DRS, refer to 3.1, “What is new with ESX Server and VirtualCenter?” on page 80.

3.6.1 VMware High Availability

Here we provide you with an overview of the benefits of VMware High Availability (HA) and how they can best fit your x3950 virtual infrastructure implementation.

> **Tip:** VMware High Availability is the new name for Distributed Availability Services.

Figure 3-32 provides an example of a cluster comprised of six System x3950 all running ESX Server 3.0.

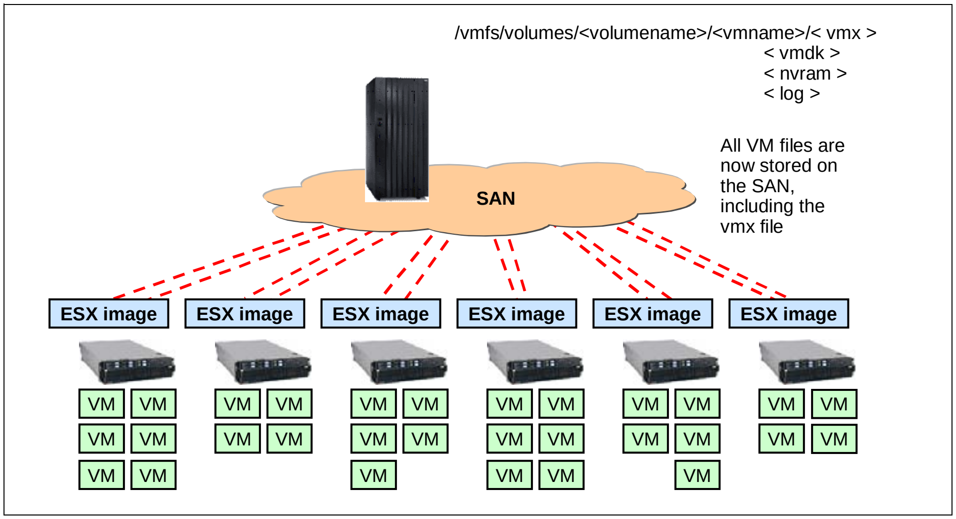

/vmfs/volumes/<volumename>/<vmname>/< vmx >
< vmdk >
< nvram >
< log >
All VM files are now stored on the SAN, including the vmx file
SAN
ESX image
ESX image
ESX image
ESX image
ESX image
ESX image
VM

Figure 3-32 Cluster overview

> **Note:** one of the major difference between ESX Server 2 and ESX Server 3 is that now all the virtual machine files, including the vmx configuration file, are located on the SAN, resulting in even easier management of the infrastructure and easier recovery in the event of a disaster.

Now, consider the situation where one of the six servers in the cluster fails for some reason. Reasons for failure can include:

- Hardware failure of a critical, nonredundant component (loss of power to the one server)
- Kernel panic of the ESX Server image

In this circumstance, the HA service running on the VMware VirtualCenter server that is monitoring the cluster can fail over the virtual machines that were running on the failing system and redistribute them on the surviving hosts. This is shown in Figure 3-33.

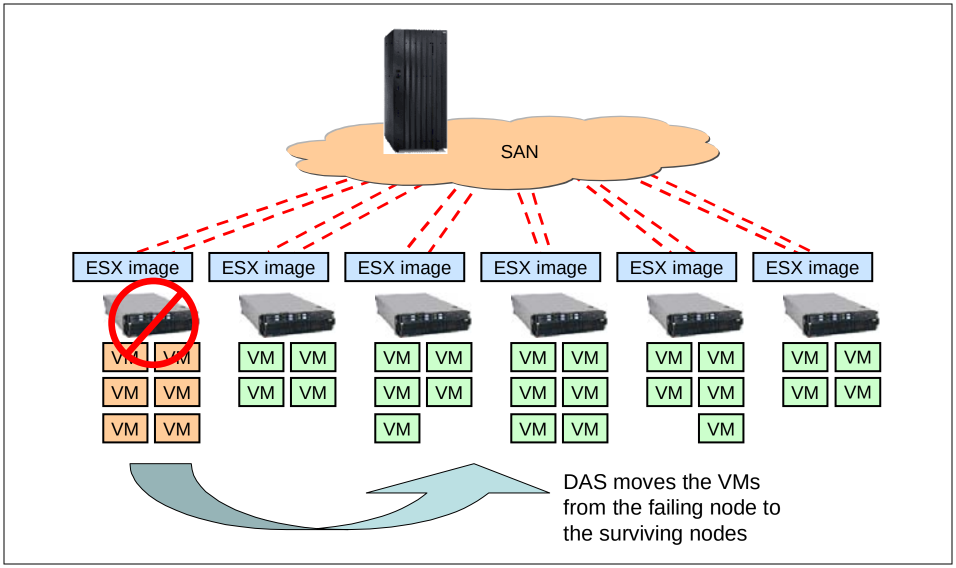

SAN
ESX image
ESX image
ESX image
ESX image
ESX image
ESX image
VM
DAS moves the VMs from the failing node to the surviving nodes

Figure 3-33 HA moves the VMs from the failing node to the surviving nodes

In this case, all virtual machines running on the host that failed in this manner stop as well, but HA will redistribute them on the surviving nodes and restart them as shown in Figure 3-34 on page 140.

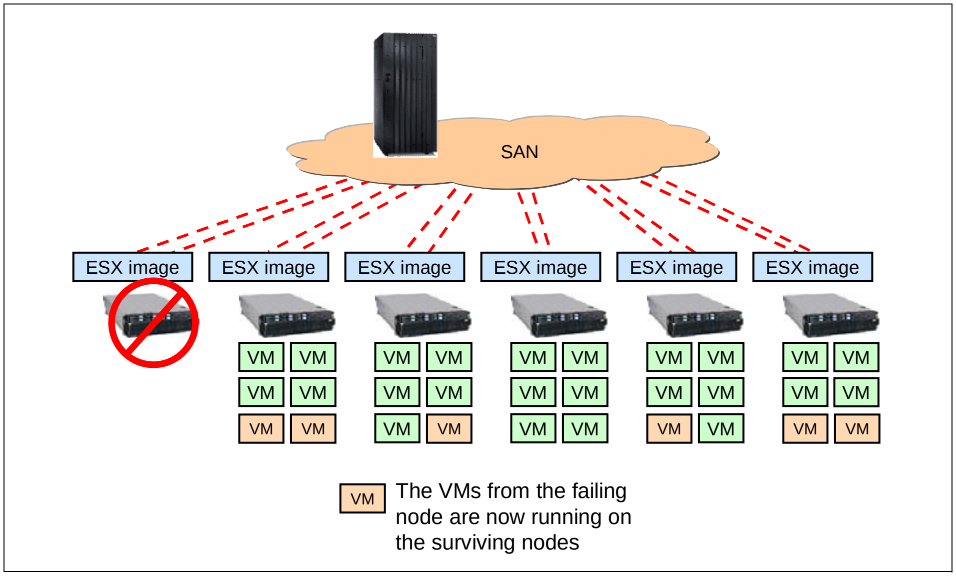

SAN
ESX image
ESX image
ESX image
ESX image
ESX image
ESX image
VM
VM
The VMs from the failing node are now running on the surviving nodes

Figure 3-34 Representation of virtual machine redistribution

> **Note:** When the physical host goes down, the virtual machines on that host go down with it. These VMs will be restarted on other hosts, but the end user will experience some downtime. Depending on the application or the service running in the virtual machines, the downtime might be a few minutes.

Also notice that because all the relevant files of the virtual machines are now on the SAN, the HA service only needs to recatalogue the virtual machines on the proper hosts without a restore of such files. This addresses one of the major concerns of ESX Server 2.x, the architecture of which required the VM configuration files to be owned, typically on local storage, by the ESX Server host that was originally running the virtual machine.

Configuring HA

Perform the following steps to configure VMware HA.

1. Define the cluster boundaries using the VirtualCenter management interface as shown in Figure 3-35 on page 141. As you can see, we have a datacenter that already contains an ESX Server 3.0 system.

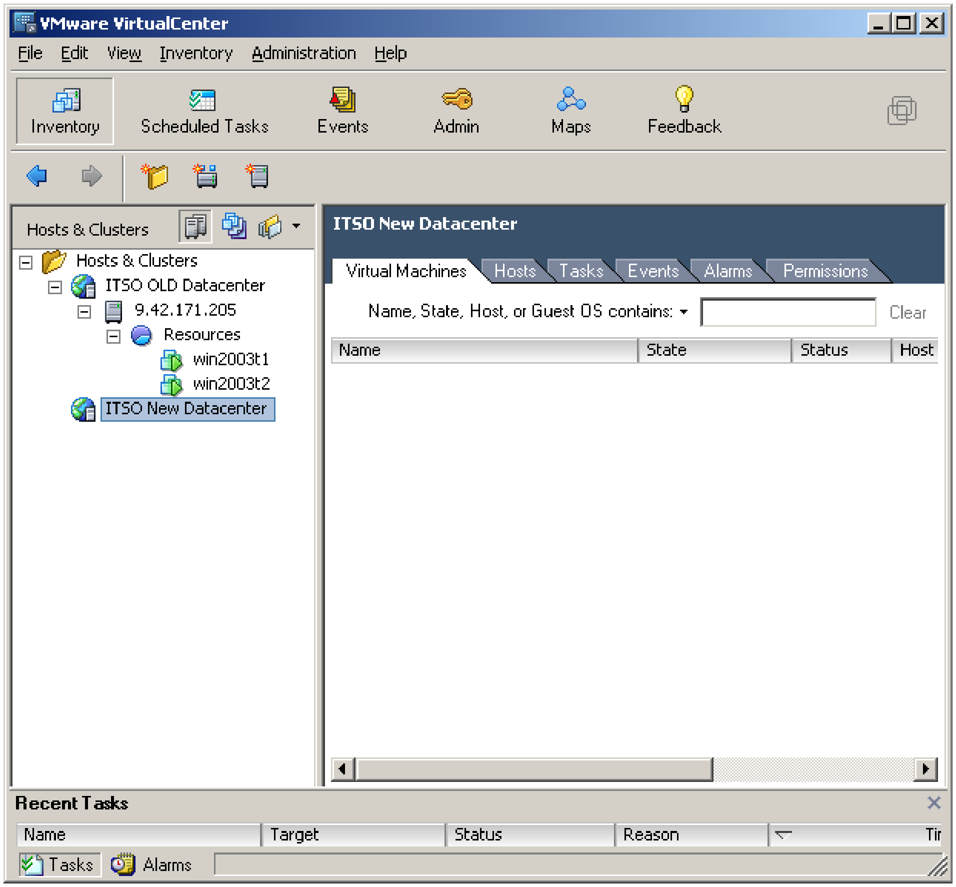
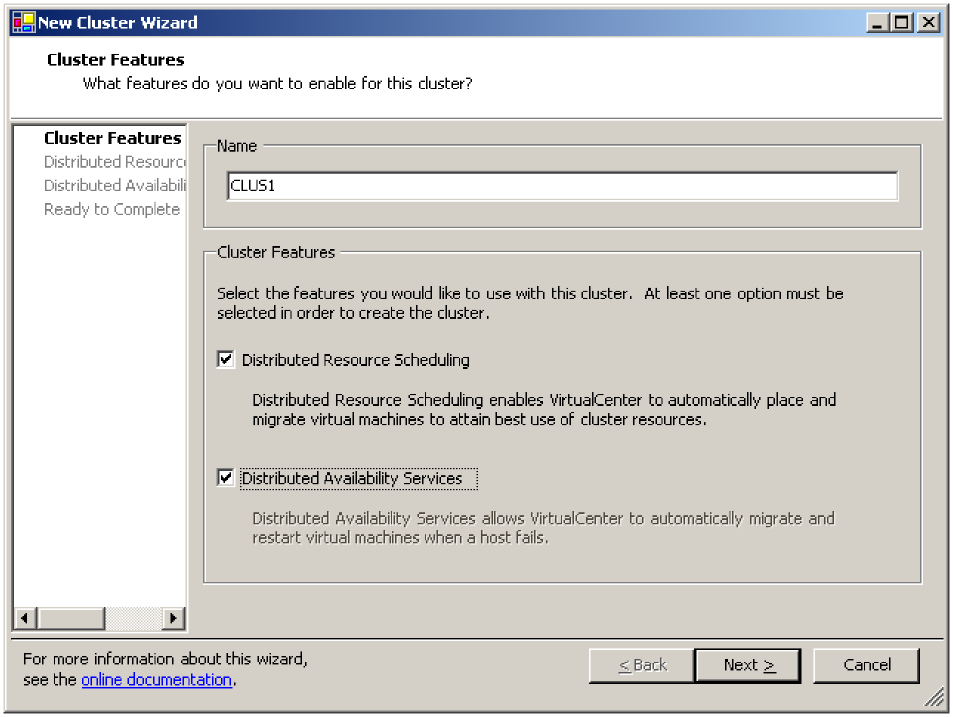


Figure 3-35 The VirtualCenter management interface.

2. Now create a new cluster in the new datacenter that we have defined as shown in Figure 3-35. To create the cluster, right-click the datacenter and click **New Cluster**.



Name: CLUS1

Select the features you would like to use with this cluster. At least one option must be selected in order to create the cluster.

☑ Distributed Resource Scheduling

Distributed Resource Scheduling enables VirtualCenter to automatically place and migrate virtual machines to attain best use of cluster resources.

☑ Distributed Availability Services

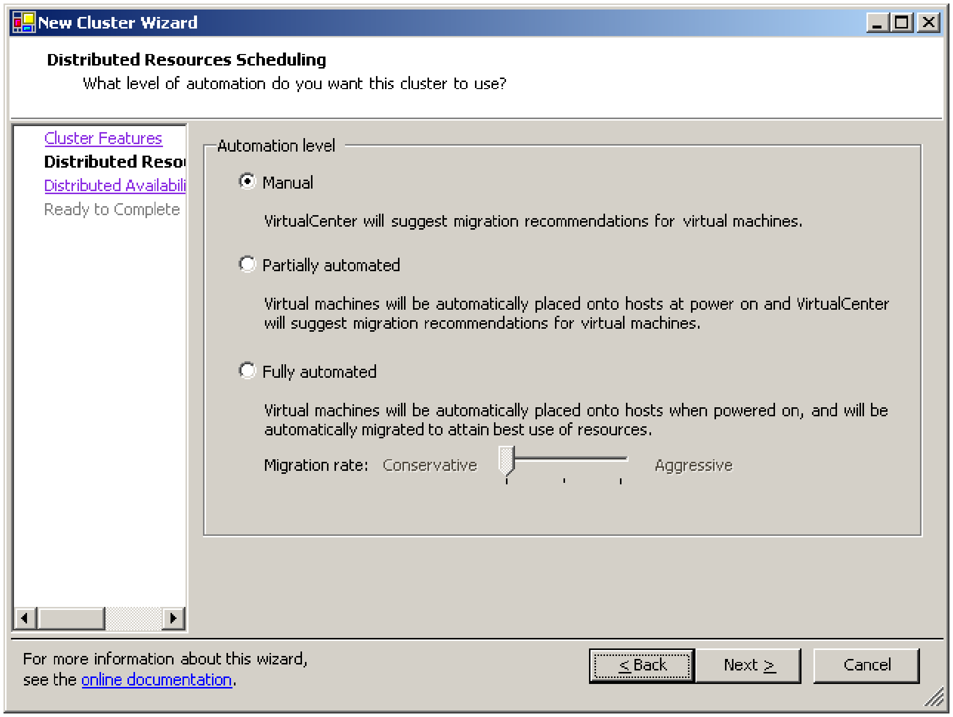
Distributed Availability Services allows VirtualCenter to automatically migrate and restart virtual machines when a host fails.

For more information about this wizard, see the online documentation.

Figure 3-36 New Cluster Wizard - Features

3. Enter the name you want to assign to the cluster as well as whether this cluster should support DRS, HA or both algorithms. We have chosen to support both by adding check marks as shown in Figure 3-36 on page 142. Click **Next**.

> **Note:** The screen captures in this section refer to Distributed Availability Services. This has been renamed to High Availability.



Distributed Resources Scheduling

What level of automation do you want this cluster to use?

Automation level

● Manual

VirtualCenter will suggest migration recommendations for virtual machines.

○ Partially automated

Virtual machines will be automatically placed onto hosts at power on and VirtualCenter will suggest migration recommendations for virtual machines.

○ Fully automated

Virtual machines will be automatically placed onto hosts when powered on, and will be automatically migrated to attain best use of resources.

Migration rate: Conservative — Aggressive

For more information about this wizard, see the online documentation.

Figure 3-37 DRS: Selecting the automation level

4. For the DRS feature, specify the level of automation you want to provide. DRS can be configured in three different modes:

   – **Manual**: VirtualCenter suggests whether to create or move virtual machines, but full control is left to the administrator.

   – **Automated**: VirtualCenter automatically assigns a host, based on workload, to a newly created virtual machine but only suggests to the administrator how to rebalance the workload at run time.

   – **Fully automated**: VirtualCenter decides where to host a newly created virtual machine as well as decides where and when to move virtual machines across the cluster, by means of the VMotion feature, when the workload demands it.

   For the purpose of our test, we set the DRS feature to act in **manual** mode. Click **Next**.

5. The next step is to configure HA but our beta version did not have a configurable HA service. So in our case, the HA window only stated that it will be enabled.

At this point you are presented with some summary screens which bring you to the end of the wizard. You have created your new cluster similar to Figure 3-38.

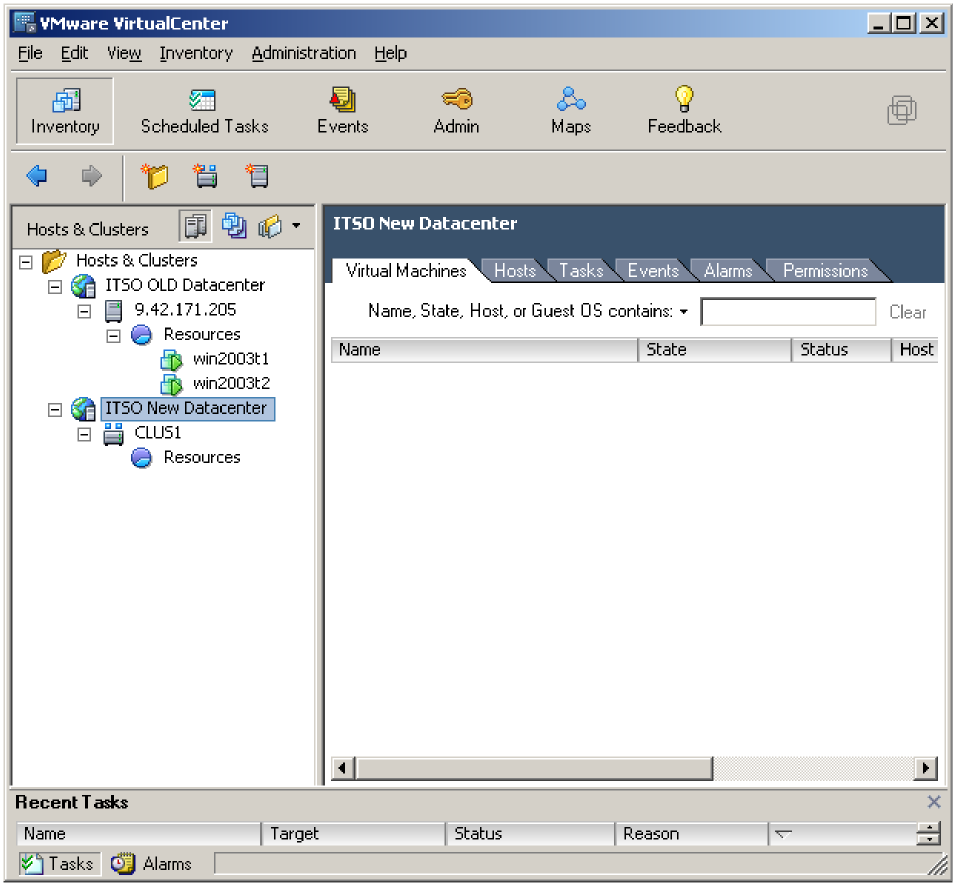
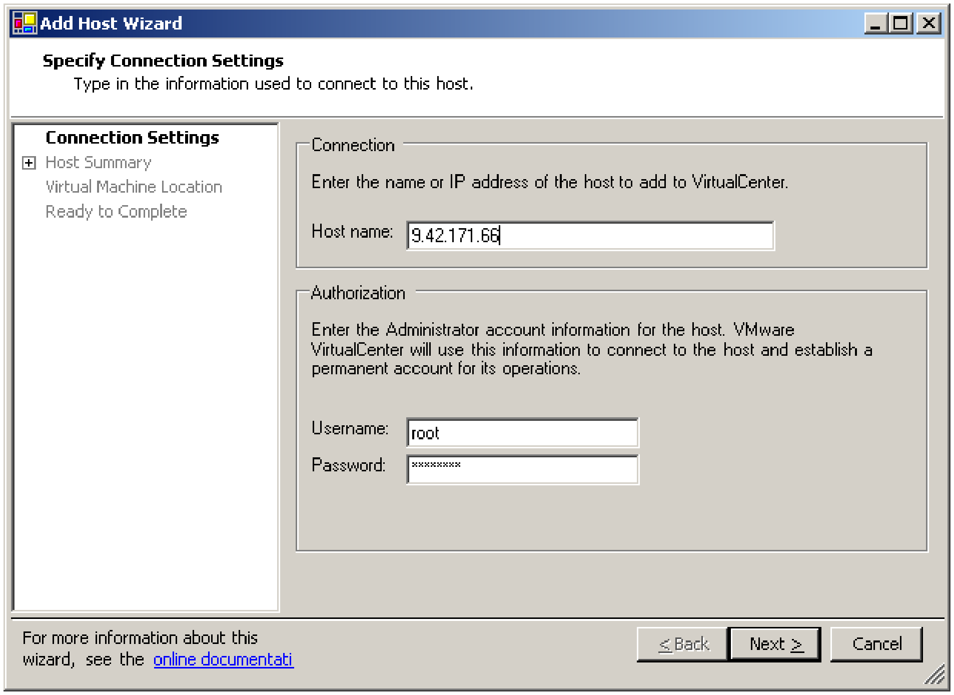


Figure 3-38 The cluster is created

The new cluster is essentially an empty logical entity that is filled with physical servers running ESX Server and that will be subject to the DRS and HA algorithms we have defined in this cluster.

To add ESX Server hosts, follow these steps.

1. Right-click the cluster and click **Add Host**. Figure 3-39 on page 145 opens.



Specify Connection Settings
Type in the information used to connect to this host.

Connection

Enter the name or IP address of the host to add to VirtualCenter.

Host name: 9.42.171.66

Authorization

Enter the Administrator account information for the host. VMware VirtualCenter will use this information to connect to the host and establish a permanent account for its operations.

Username: root

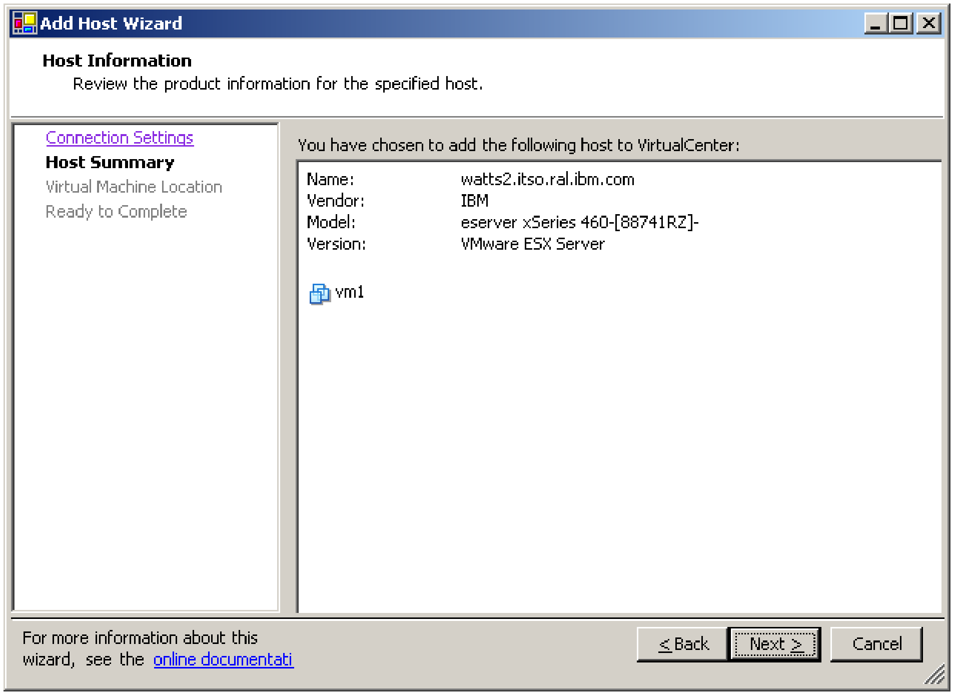
Password: ********

For more information about this wizard, see the online documentati

Figure 3-39 Specify the host name

2. Enter the information for the host you want to add: host name (or IP address) and the password for root. Click **Next**.

   If the information provided is correct, VirtualCenter is granted the ability to connect to the host and a summary page on the host's status is presented, similar to Figure 3-40 on page 146.



Host Information

Review the product information for the specified host.

You have chosen to add the following host to VirtualCenter:

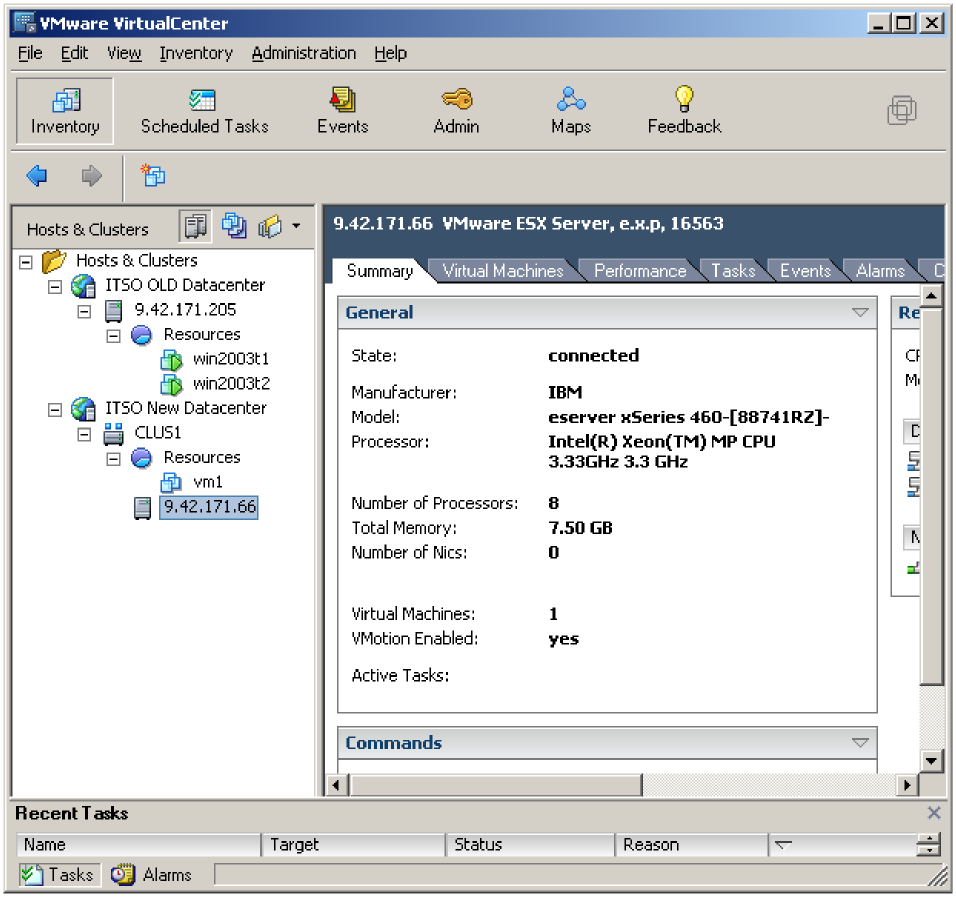
Name:	watts2.itso.ral.ibm.com
Vendor:	IBM
Model:	eserver xSeries 460-[88741RZ]-
Version:	VMware ESX Server

vm1

For more information about this wizard, see the online documentati

Figure 3-40 The host is detected and recognized.

3. Click **Next** to display further summary pages. The host is then added to the cluster. In Figure 3-41 on page 147, we added an 8-way x3950 to cluster CLUS1.



State: connected

Manufacturer: IBM

Model: eserver xSeries 460-[88741RZ]-

Processor: Intel(R) Xeon(TM) MP CPU 3.33GHz 3.3 GHz

Number of Processors: 8

Total Memory: 7.50 GB

Number of Nics: 0

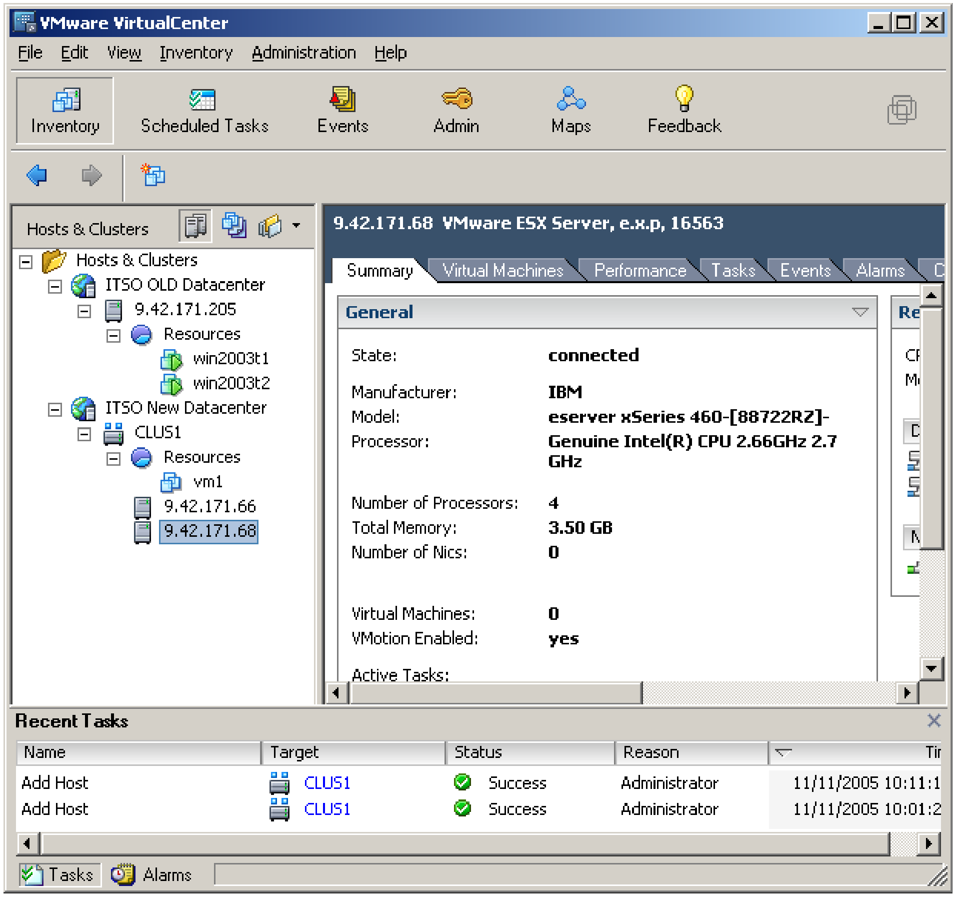
Virtual Machines: 1

VMotion Enabled: yes

Active Tasks:

Figure 3-41 The first node is added.

4. Repeat the same steps to add a second host to the cluster. Figure 3-42 on page 148 shows the result of adding a 4-way x3950 host to CLUS1.



State: connected

Manufacturer: IBM

Model: eserver xSeries 460-[88722RZ]-

Processor: Genuine Intel(R) CPU 2.66GHz 2.7 GHz

Number of Processors: 4

Total Memory: 3.50 GB

Number of Nics: 0

Virtual Machines: 0

VMotion Enabled: yes

Active Tasks:

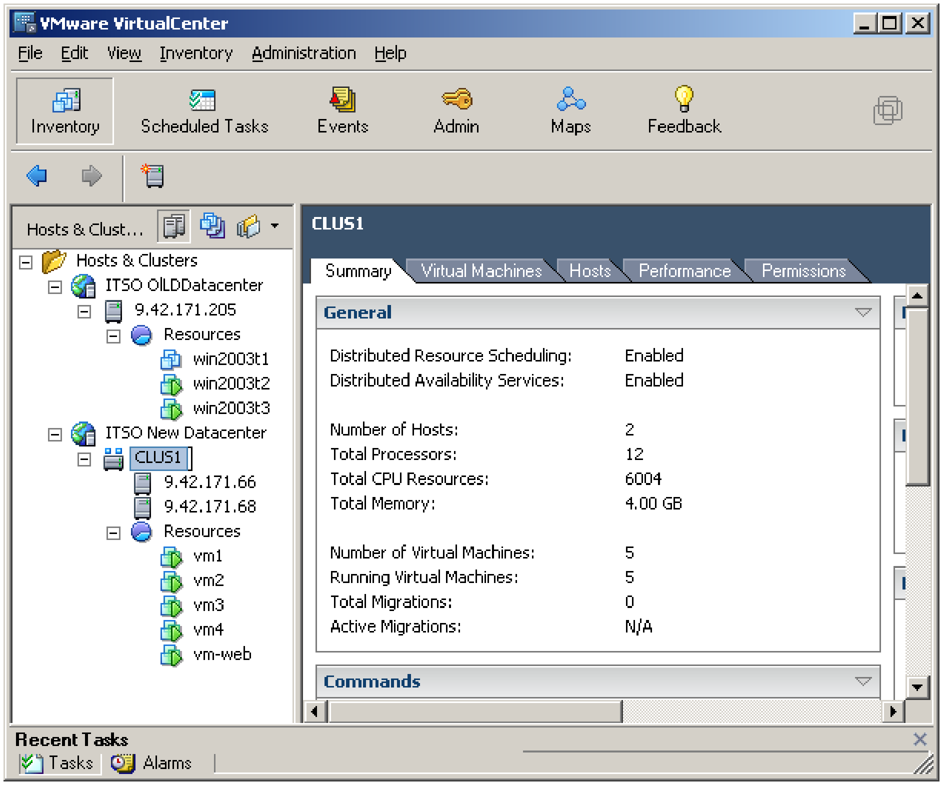
Name	Target	Status	Reason	Tir
Add Host	CLUS1	Success	Administrator	11/11/2005 10:11:1
Add Host	CLUS1	Success	Administrator	11/11/2005 10:01:2

Figure 3-42 The second node is added

Using HA

Now that the cluster has been set up, we can test failover.

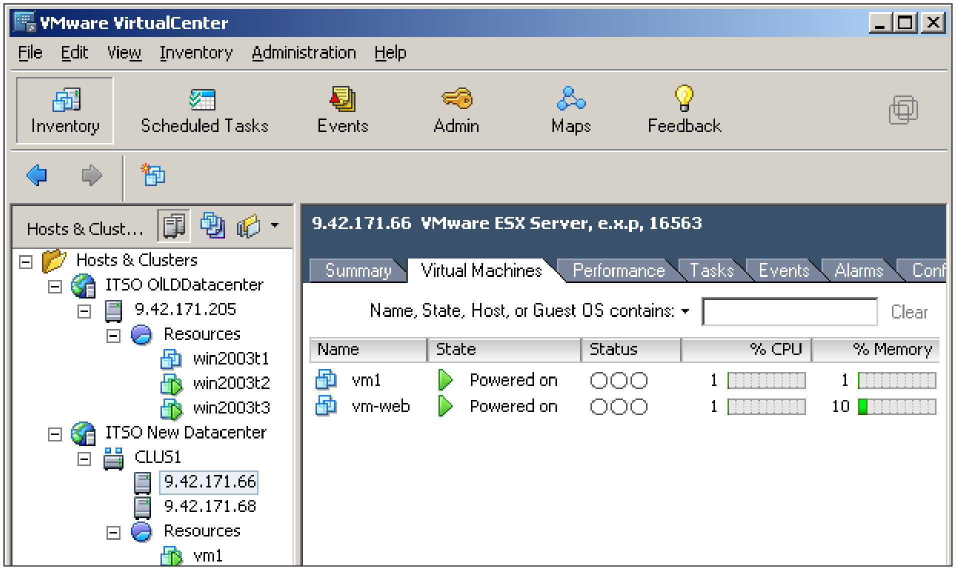
Figure 3-43 on page 149 shows our setup when we tested HA. As you can see, both HA and DRS functions are enabled. There are two hosts that comprise this cluster and a summary of the resources available cluster-wise in terms of CPU and memory.



Distributed Resource Scheduling:	Enabled
Distributed Availability Services:	Enabled
Number of Hosts:	2
Total Processors:	12
Total CPU Resources:	6004
Total Memory:	4.00 GB
Number of Virtual Machines:	5
Running Virtual Machines:	5
Total Migrations:	0
Active Migrations:	N/A

Figure 3-43 Cluster setup

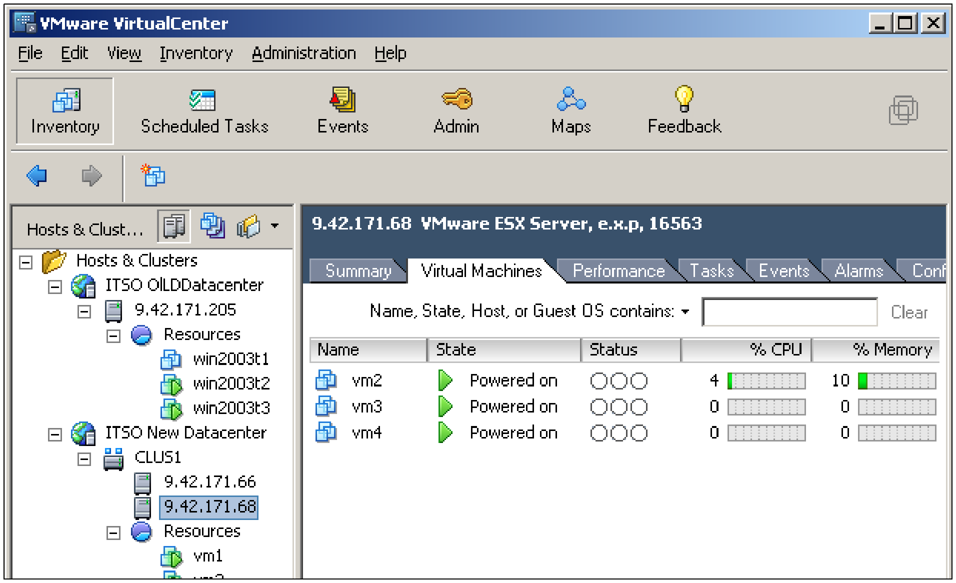
Each of the two hosts support a certain number of running virtual machines. Figure 3-44 on page 150 shows the first host.



Name	State	Status	% CPU	% Memory
vm1	Powered on	○○○	1	1
vm-web	Powered on	○○○	1	10

Figure 3-44 Virtual machines on the first host.

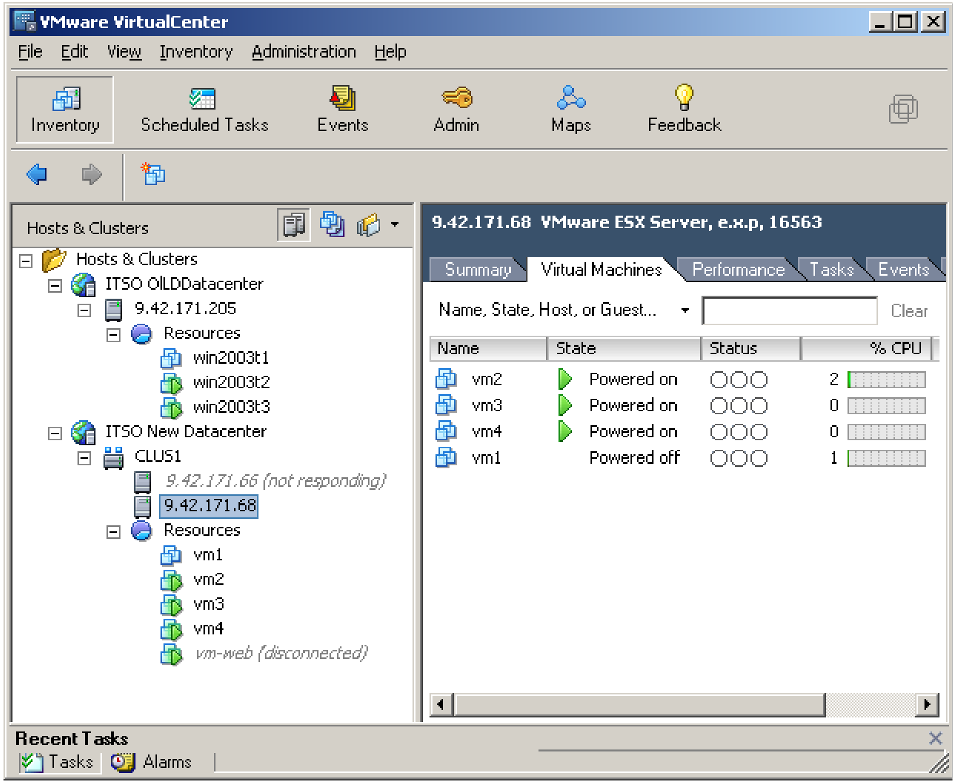
Figure 3-45 shows the second host.



Name	State	Status	% CPU	% Memory
vm2	Powered on	○○○	4	10
vm3	Powered on	○○○	0	0
vm4	Powered on	○○○	0	0

Figure 3-45 Virtual machines on the second host.

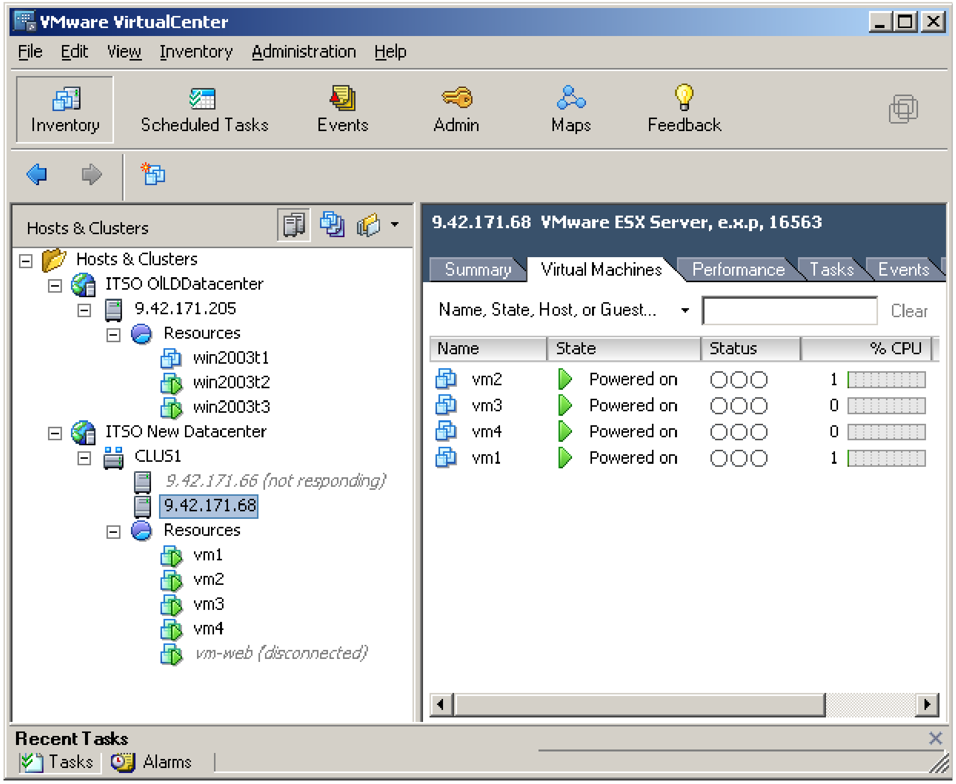
To simulate the crash of a server, we shut-down the first host in the farm. As you can see in Figure 3-46 on page 151, VirtualCenter recognizes that the host is no longer available and the VMs running on that hosts start to be marked as not available. This is the beginning of the failover process.



Name	State	Status	% CPU
vm2	Powered on	○○○	2
vm3	Powered on	○○○	0
vm4	Powered on	○○○	0
vm1	Powered off	○○○	1

Figure 3-46 The first node goes off-line

Figure 3-47 on page 152 shows a point in the middle of the failover process where virtual machine vm1 is restarted on the surviving node.



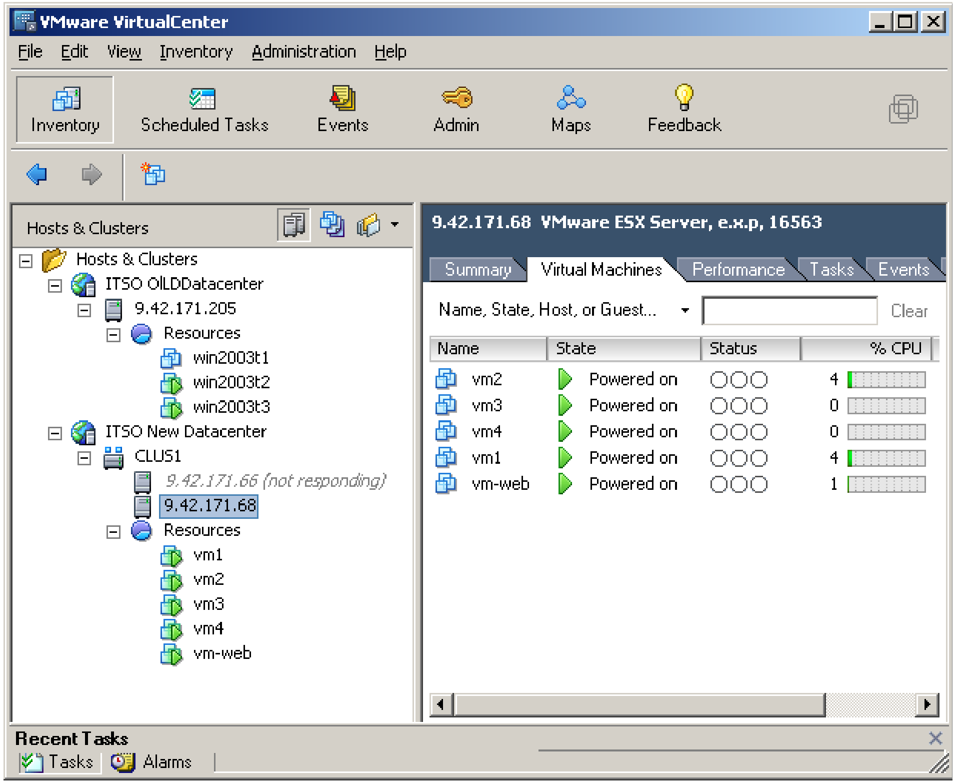
- Hosts & Clusters
  - ITSO OILDDatacenter
    - 9.42.171.205
      - Resources
        - win2003t1
        - win2003t2
        - win2003t3
  - ITSO New Datacenter
    - CLUS1
      - 9.42.171.66 (not responding)
      - 9.42.171.68
      - Resources
        - vm1
        - vm2
        - vm3
        - vm4
        - vm-web (disconnected)

9.42.171.68 VMware ESX Server, e.x.p, 16563

Name	State	Status	% CPU
vm2	Powered on	○○○	1
vm3	Powered on	○○○	0
vm4	Powered on	○○○	0
vm1	Powered on	○○○	1

Figure 3-47 The failover is in progress

After a few seconds, the failover is completed and all virtual machines are running on the remaining node available on the cluster, Figure 3-48 on page 153.

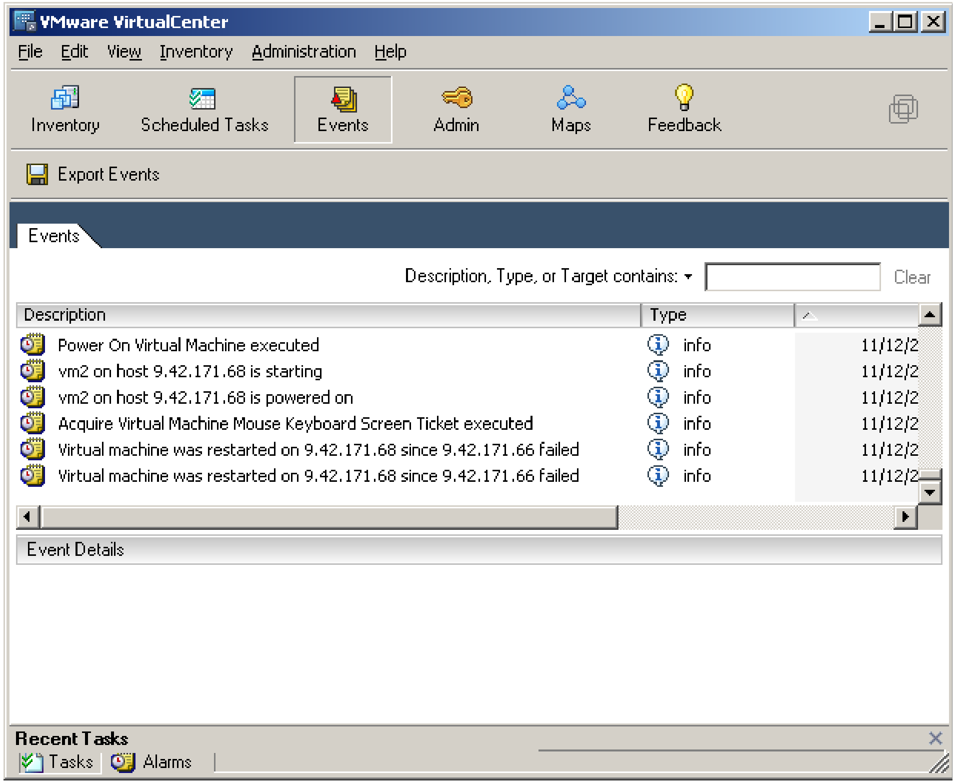


- Hosts & Clusters
  - ITSO OILDDatacenter
    - 9.42.171.205
      - Resources
        - win2003t1
        - win2003t2
        - win2003t3
  - ITSO New Datacenter
    - CLUS1
      - 9.42.171.66 (not responding)
      - 9.42.171.68
      - Resources
        - vm1
        - vm2
        - vm3
        - vm4
        - vm-web

Name	State	Status	% CPU
vm2	Powered on	○○○	4
vm3	Powered on	○○○	0
vm4	Powered on	○○○	0
vm1	Powered on	○○○	4
vm-web	Powered on	○○○	1

Figure 3-48 The HA failover is completed.

The event is also logged in the VirtualCenter logging system, as shown in Figure 3-49 on page 154.



Description	Type	
Power On Virtual Machine executed	info	11/12/2
vm2 on host 9.42.171.68 is starting	info	11/12/2
vm2 on host 9.42.171.68 is powered on	info	11/12/2
Acquire Virtual Machine Mouse Keyboard Screen Ticket executed	info	11/12/2
Virtual machine was restarted on 9.42.171.68 since 9.42.171.66 failed	info	11/12/2
Virtual machine was restarted on 9.42.171.68 since 9.42.171.66 failed	info	11/12/2

Figure 3-49 The failover event is logged.

3.6.2 Distributed Resource Scheduler

In this section we provide information about how Distributed Resource Scheduler (DRS) works as well a real (although simplistic) scenario that we tested in the laboratory.

The initial configuration of the cluster is the same as shown in the Figure 3-32 on page 138.

The idea behind DRS is that if the resources of a host in the cluster become saturated, depending on its configuration DRS will either suggest or initiate a VMotion move onto a host in the cluster that has more resources available. So if one of your hosts is driving resource utilization at 90% for exampl,e and you have other hosts running at a lower utilization rate, DRS will try to rebalance the usage of the resources across all nodes, as you can see in Figure 3-50 on page 155.

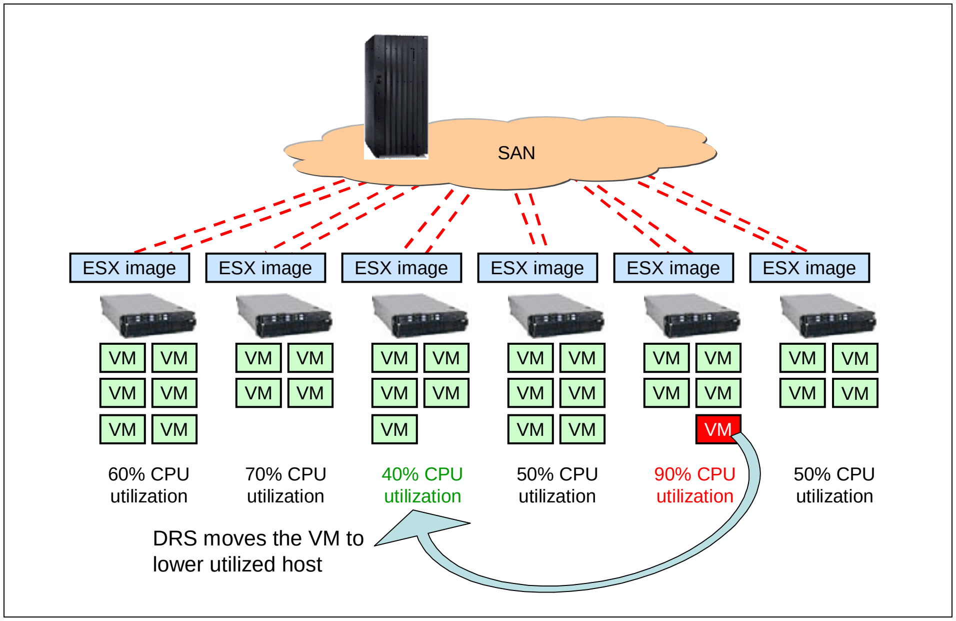


Figure 3-50 DRS initiates or recommends the transfer of the virtual machine.

If the suggestion is accepted by the administrator, or if DRS is set in fully automated mode so that it will perform the transfer automatically, the resource utilization is rebalanced by migrating one or more virtual machines on to other hosts. See Figure 3-51 on page 156.

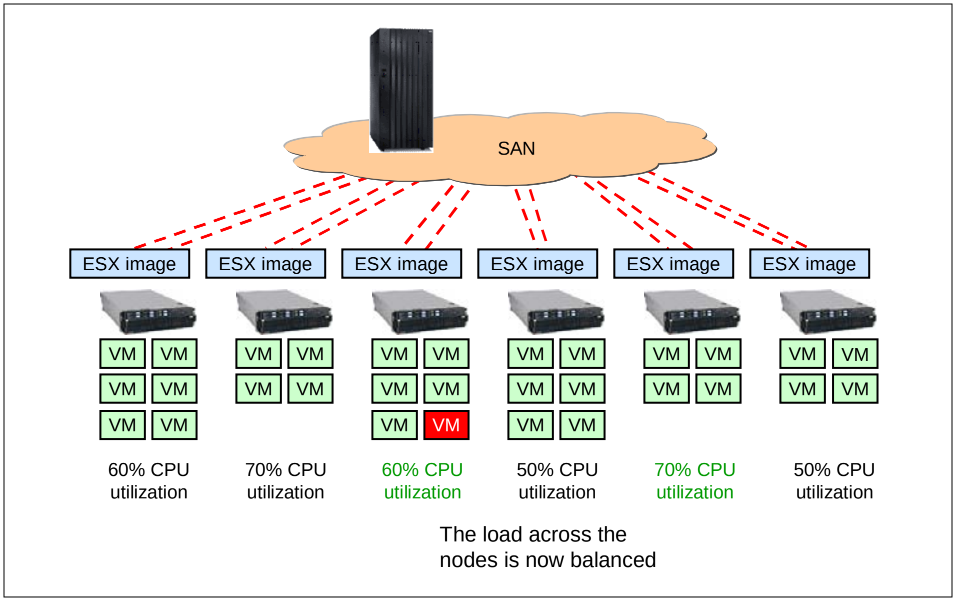
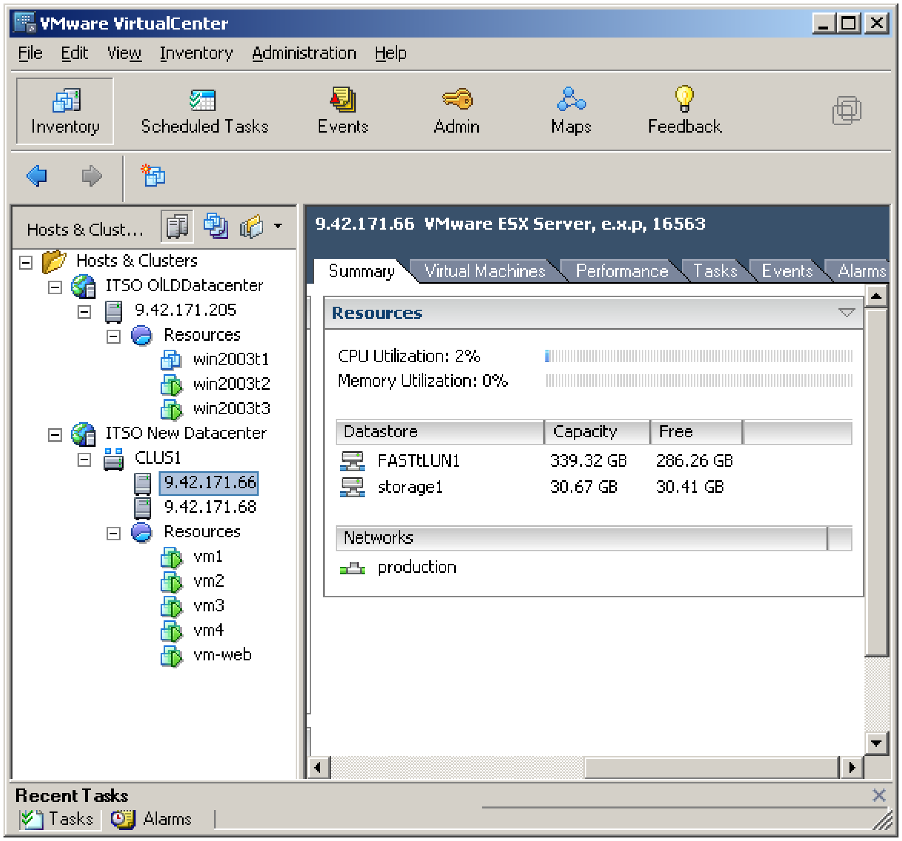


Figure 3-51 The cluster gets rebalanced

Using DRS

Now that we have introduced the concepts of DRS, we use it in this section. We use the same cluster setup as described in 3.6.1, “VMware High Availability” on page 138.

In this case the virtual machines running on the first host, an 8-way x3950, are basically idle and the overall system utilization is very low, as you can see in Figure 3-52 on page 157.



9.42.171.66 VMware ESX Server, e.x.p, 16563

Resources

CPU Utilization: 2%

Memory Utilization: 0%

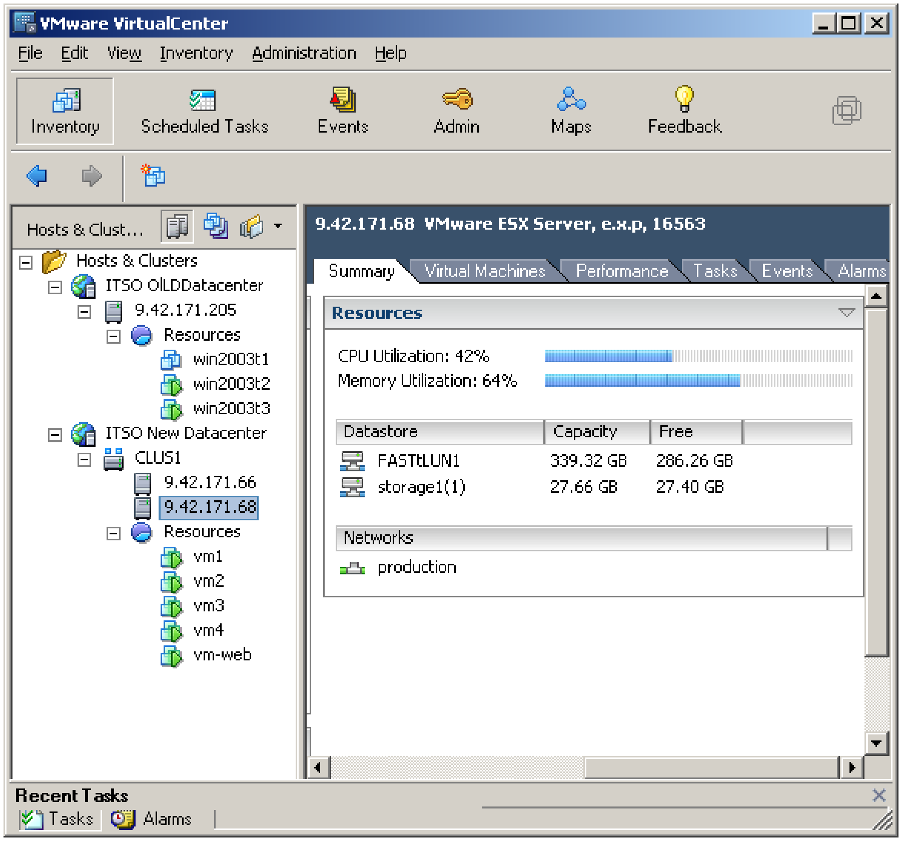
Datastore	Capacity	Free
FASTtLUN1	339.32 GB	286.26 GB
storage1	30.67 GB	30.41 GB

Networks

production

Figure 3-52 CPU utilization of host 1 is low

On the other hand, virtual machine vm2, a 4-way SMP virtual machine running on the second 4-way host in the cluster, has a heavy workload, as shown in Figure 3-53 on page 158.



Hosts & Clusters
- ITSO OlLDDatacenter
  - 9.42.171.205
    - Resources
      - win2003t1
      - win2003t2
      - win2003t3
- ITSO New Datacenter
  - CLUS1
    - 9.42.171.66
    - 9.42.171.68
  - Resources
    - vm1
    - vm2
    - vm3
    - vm4
    - vm-web

9.42.171.68 VMware ESX Server, e.x.p, 16563

Resources

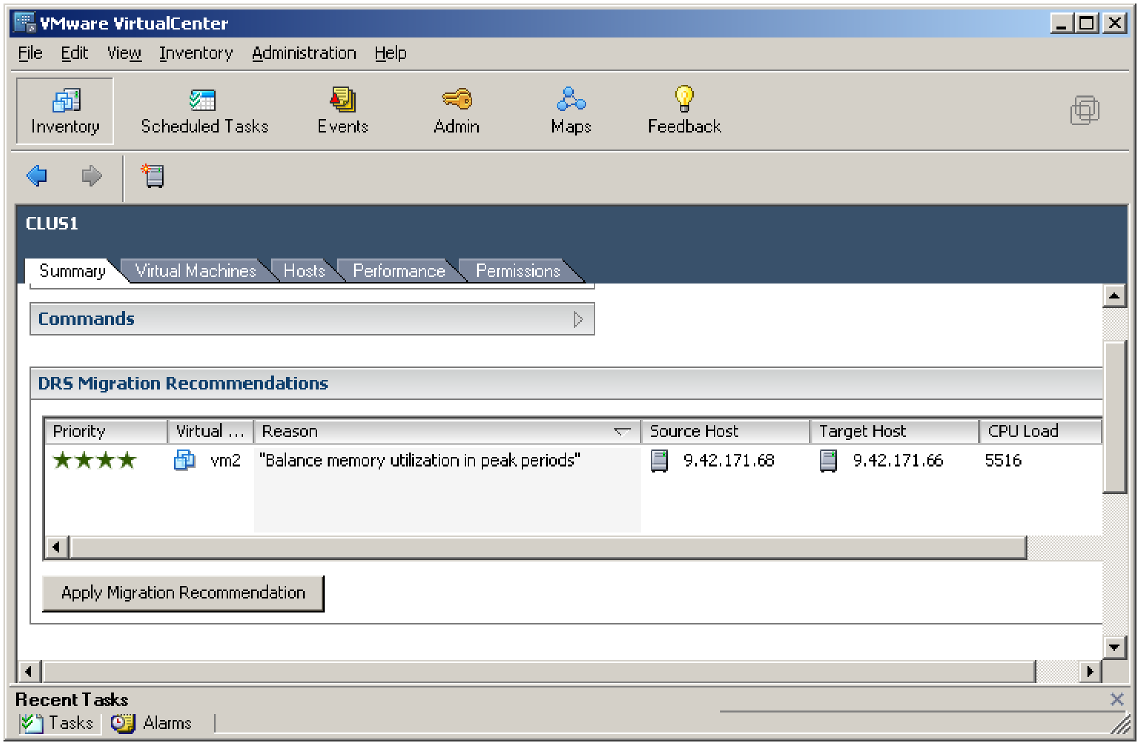
CPU Utilization: 42%
Memory Utilization: 64%

Datastore	Capacity	Free
FASTtLUN1	339.32 GB	286.26 GB
storage1(1)	27.66 GB	27.40 GB

Networks
production

Figure 3-53 CPU and memory utilization of host 2 is high

This system is being pushed hard in terms of CPU and, even more importantly, memory utilization. This is especially true if you consider that the 8-way system in the cluster is idle most of the time. As a result of this, after a few minutes, the DRS algorithms suggest the system administrator move vm2 from the second host to the first one using VMotion. This recommendation is shown in Figure 3-54 on page 159.



Priority	Virtual ...	Reason	Source Host	Target Host	CPU Load
★★★★	vm2	"Balance memory utilization in peak periods"	9.42.171.68	9.42.171.66	5516

Figure 3-54 DRS recommends to move the virtual machine to another host

Clicking **Apply Migration Recommendation** in Figure 3-54 initiates a VMotion task to move vm2 from the 4-way system to the 8-way idle system.

Note: if we set DRS to **fully automated** mode, VMotion would move the virtual machine without any sort of confirmation from the administrator and the action would simply be recorded in the VirtualCenter event log.

3.7 Four-way virtual machines

One of the innovative features included with ESX Server 3.0 is the support for four-way (4-way) virtual machines. With this new feature, you can create virtual machines that are capable of running enterprise applications such as databases, ERP or CRM services that usually require more than one and often more than two CPUs.

Figure 3-55 on page 160 shows a 4-way SMP virtual machine.

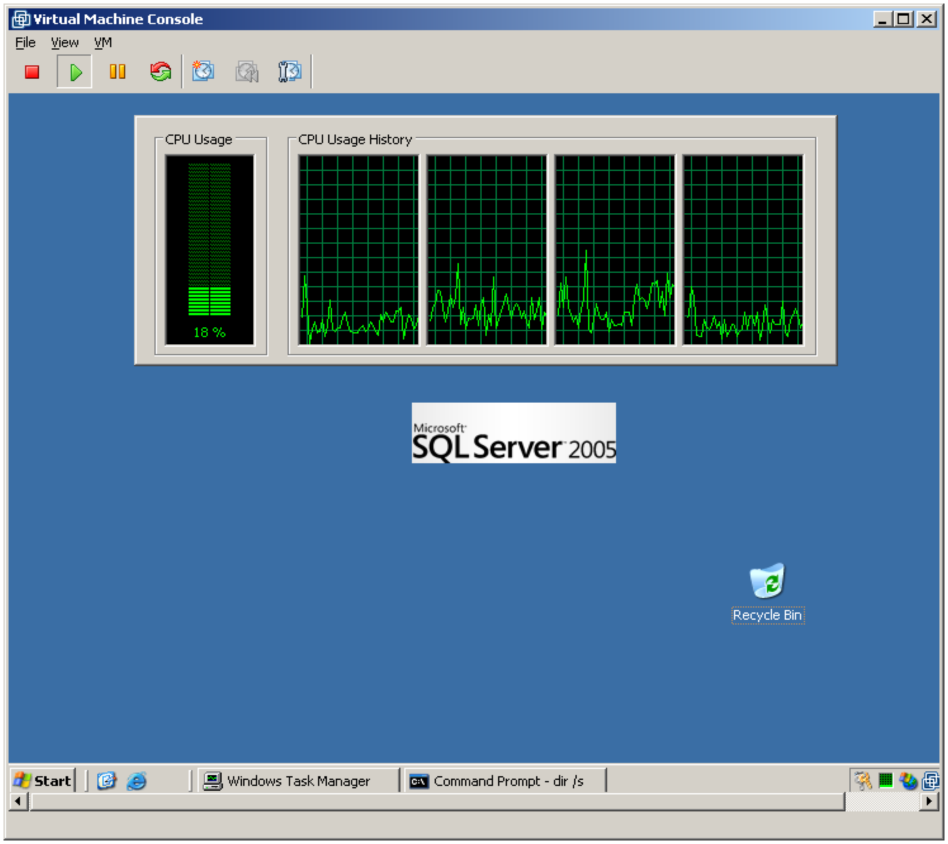


Figure 3-55 4-way virtual SMP virtual machine

The System x3950 is not only a superb platform to run standard ESX Server workloads, the x3950, especially in multi-node configuration, is also the reference platform when it comes to run scalable and enterprise workloads inside virtual machines. There are at least two reasons for this, discussed in the following sections:

- 3.7.1, “Big virtual machines require big hosts” on page 160
- 3.7.2, “NUMA-enabled 4-way virtual machines” on page 163

3.7.1 Big virtual machines require big hosts

The way that ESX Server runs multiple virtual machines on the one server is good for a number of reasons, including higher resource utilization, server consolidation and so forth. One restriction is that the virtual machines cannot be assigned more processor power (virtual processors) than there is installed in the server (logical processors). This means that if you want to create four-way virtual machine (each with four virtual processors), you must have a server with four logical processors (for example, at least two cores and Hyper-Threading enabled). To take advantage of the new ability of creating four-way VMs, you need a server with many processors. The x3950 is ideal in this regard.

Dual-core processors do help in this trend but they do not solve all the problems associated with running multiple enterprise workloads on the same physical server, especially when there is an overlapping of utilization peaks of the virtual machines.

One option would be to create a *high priority* resource pool on the server where all the enterprise virtual machines can be placed. Then they can benefit from all the resources of the server whenever needed. You can also create a *low priority* resource pool on the same x3950 where all the other virtual machines get their resources from so that when the high priority virtual machines are not fighting for resources, the other low priority virtual machines can get them if needed.

Using the x3950 in multi-node configurations as a foundation for this scenario has a number of benefits:

- You ensure that enterprise virtual machines always get the highest priority by means of the different resource pools.
- You ensure adherence to the service level agreements of those enterprise virtual machines by means of the scalable multi-node configuration, even though they might peak at the same time.
- You ensure that when the enterprise virtual machines are not consuming all available resources in the multi-node configuration, the lower-priority virtual machines use any unused cycles and memory.

This scenario does not require any manual user intervention and it is provided by the basic VMkernel scheduling algorithms.

Note: Although it is possible to implement such a scenario using DRS to balance virtual machines across multiple physical hosts, it is suggested to keep the administrators in the loop for applying the DRS algorithms to critical virtual machines.

Of course the number of systems, their configuration and their workload is provided as an example for the purpose of the discussion. You certainly do not want to drive them close to 100% utilization, for example.

The message is that these setups could give you the confidence for running multiple 4-way virtual machines peaking at the same time. On a 2-node x3950 single-core configuration, you could have, for example, five or ten 4-way virtual machines running and provide them full performance when two of them peak at the very same time.

Table 3-6 is a summary of our recommendations, showing the maximum number of 4-way enterprise virtual machines that should be allowed to peak at the same time, depending on the x3950 configuration:

Table 3-6 4-way virtual machine flexibility in dealing with peaks

x3950 configuration	4-way VMs allowed to peak concurrently
4-way single core	1
4-way dual core	2
8-way single core	2
8-way dual core	4
16-way single core	4
16-way dual core	8

For example, if you have a 4-way single core x3950 (the first row in the table), you can only allow a single 4-way vm to peak at any given point in time. At that point, the other virtual machines are basically not given any resources. With more cores (as with a multi-node x3950 configuration), you will have more flexibility because if you have four 4-way virtual machines on a 8-way dual core x3950 complex, it is likely the virtual machines will not peak at the same time and would therefore allow the other VMs to continue to be scheduled.

Note: For the sake of streamlining the discussion, we have not considered enabling Hyper-Threading on this configuration and counting the HT logical processors for this exercise.

Different applications have different workloads throughout the day and, as a result, it is possible to consolidate them as described here. Consider different types of database purposes such as an SQL Server database extensively used for OLTP support during the day and an Oracle database for data warehousing used only during the night to generate reports. Consider different types of services such as SAP or Siebel application servers stressed during the day as well as heavy batch programs that runs early in the morning.

Whenever you can find enterprise workloads that have different load patterns and therefore are not hostile to each other in terms resource utilization for longer periods of time, those workloads are good candidates for consolidation on a multi-node x3950 configuration.

It is also important to note that the VMware Virtual Infrastructure might not always be the best fit to host all 4-way enterprise workloads. Specifically if you have an enterprise workload that requires 90% of CPU utilization 24 hours a day, you might want to consider moving it to a separate physical system.

3.7.2 NUMA-enabled 4-way virtual machines

We discussed NUMA technology and how ESX Server takes advantage of it in 3.2.1, “ESX Server and NUMA” on page 89.

The NUMA algorithms of the VMkernel have the goal of containing a virtual machine within a single NUMA node. Should the VMkernel find that the number of virtual CPUs assigned to a virtual machine exceeds the capabilities of the NUMA node in the system, the VMkernel will *disable* the NUMA algorithms for that specific virtual machine. This means that this virtual machine will be assigned memory as though it were a flat SMP system. This can affect negatively the performance of the virtual machine. Fortunately this does not occur with the x3950 as shown in Table 3-7.

This is particularly important when it comes to 4-way vSMP virtual machines because the whole idea behind creating a 4-way virtual machine is because you are running an enterprise service, such as a backend SQL Server database for example, that requires nothing but the highest performance possible.

The Intel Xeon architecture is the only x86 architecture currently available to satisfy the requirement of having the size of the NUMA node that is larger than a 4-way virtual machine. This means that on the x3950, the VMkernel will use the NUMA algorithms to optimize CPU and memory locality for all 4-way virtual machines that get created on the system.

Table 3-7 summarize the size of the NUMA node in terms of logical processors available depending on the configuration:

Table 3-7 NUMA node characteristics

x3950 configuration	Hyper-Threading	Sockets	Cores	=> Logical CPUs per NUMA node
1 node (one x3950)	Disabled	4	4	4
1 node (one x3950)	Enabled	4	4	8
1 node (one x3950)	Disabled	4	8	8
1 node (one x3950)	Enabled	4	8	16

3.8 How to scale-up an x3950 ESX Server farm

As discussed in 1.5, "Server consolidation" on page 20, one of the advantages of implementing ESX Server on x3950 configurations is lowering the number of physical servers to administer. In this section, we discuss those scenarios for which it is possible to maintain a stable IT infrastructure while increasing the *computational power* of the Virtual Infrastructure.

Our example in Figure 3-56 is an existing configuration of 100 virtual machines on six ESX Server systems running on scalable x3950 platforms.

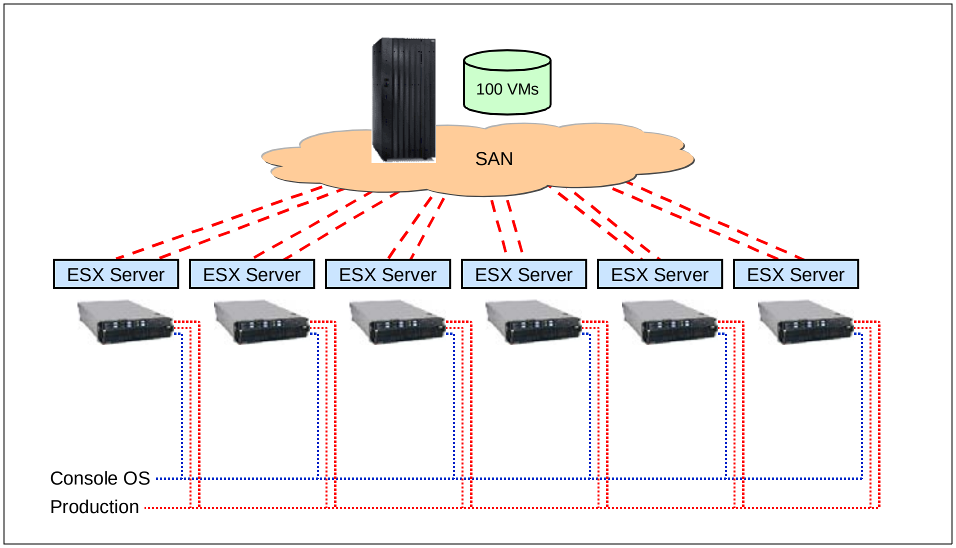


Figure 3-56 Initial deployment

As business objectives require new systems, virtual machines are created to support the new services. Because the current physical servers are already being used at 70-80% of their capabilities, there are two options:

- Buy new servers, install ESX Server on them, and start deploying new virtual machines on those new servers. This is usually referred to as a *scale-out* approach.
- With the x3950 though, we can approach this requirement in a different manner. Instead of adding new servers to the current infrastructure, you could upgrade the existing x3950 to support more virtual machines, a *scale-up* approach. This method is how we will proceed.

By adding x3950 (or x3950 E) nodes to the existing x3950 servers, as shown in Figure 3-57 on page 165, we are able to support twice the number of virtual machines without changing the architecture of their farms and or clusters.

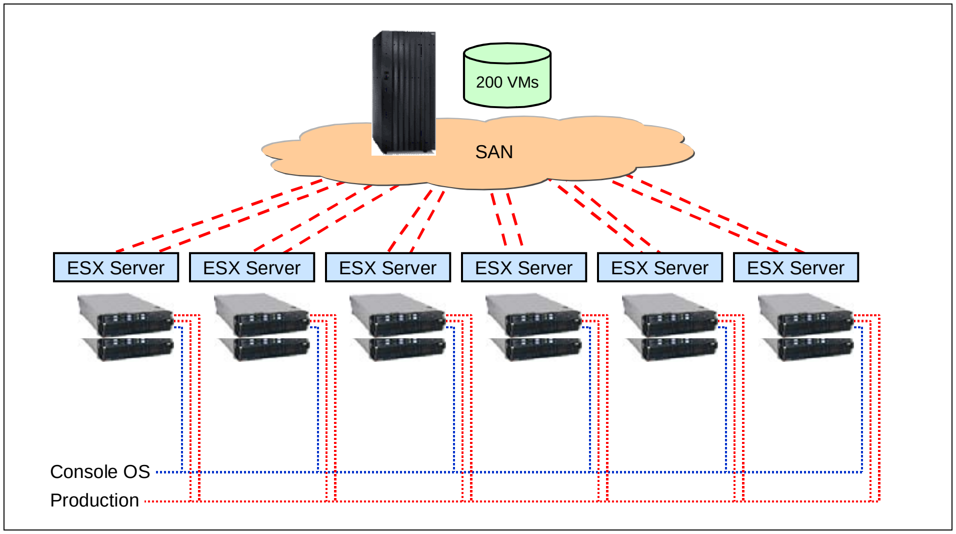


Figure 3-57 Adding nodes to the existing x3950 servers doubles the supported VMs

The configuration can be extended further by simply adding more nodes to the existing servers, upgrading some ESX Servers systems from 8-way (two nodes) to 16-way (four nodes), as shown in Figure 3-58.

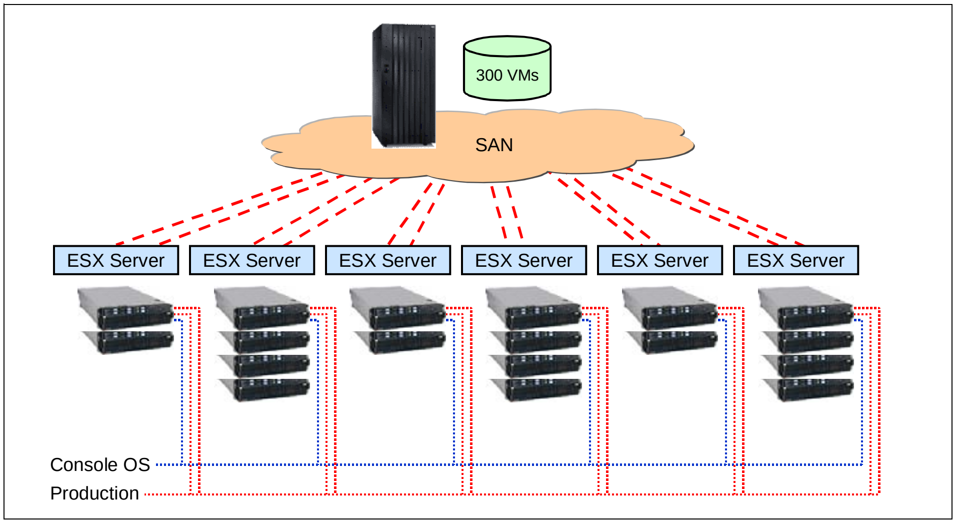


Figure 3-58 Upgrading some ESX Server systems to 16-way (four nodes)

Note: Upgrading a 4-way x3950 to an 8-way configuration is usually not much more expensive than adding another standalone 4-way system, both from a licensing and hardware perspective.

You might have noticed in this example that we have not considered upgrading the Ethernet and Fibre Channel connections. Depending on the pattern of the workloads being deployed in the virtual machines, you should consider adding physical Ethernet connections to the virtual switches as well as Fibre Channel adapters to increase the I/O throughput.

The following examples in Table 3-8 are of situations with which many systems administrators might find themselves involved. Here, we assume that the current configuration is based on single-node 4-way x3950 servers.

Table 3-8 Common scale-up situations and solutions for x3950

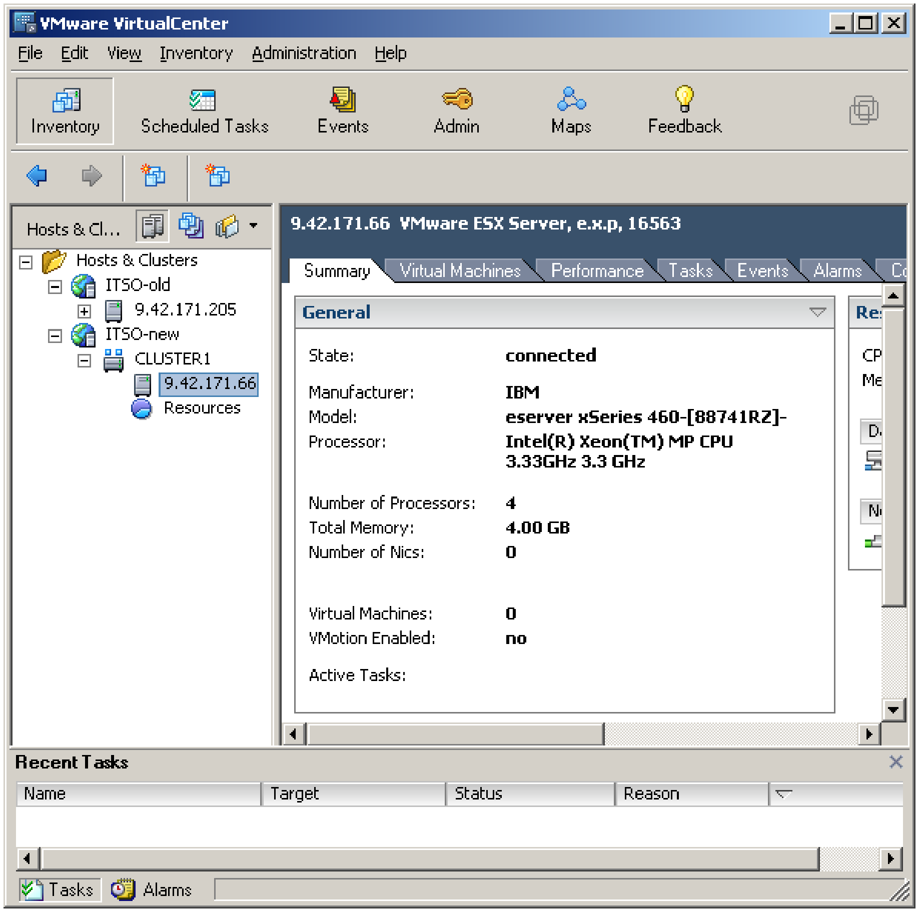
Situation	Suggested Solution
I have 300 virtual machines running on six 4-way x3950 systems. I need to grow my infrastructure adding new virtual machines but I am short on CPU capacity and RAM across my farms and clusters. Network and storage throughput are not a problem at the moment. What do I do?	Update your six servers by adding an extra x3950 (or x3950 E) node to each. This will provide you the ability to host more virtual machines without changing or adjusting anything in the way you operate today your infrastructure.
I have 300 virtual machines running on six 4-way x3950 systems. I need to grow my infrastructure adding new virtual machines but I am short on CPU capacity, RAM and I/O throughput across my farms and clusters. What do I do?	Update your six servers by adding an extra x3950 (or x3950 E) node to each. In each new node connect the onboard Gigabit Ethernet controllers and install and connect a Fibre Channel adapter. This will provide you the ability to host more virtual machines.
I have 300 virtual machines running on six 4-way x3950 systems. I need to grow my infrastructure adding new virtual machines but I am short on I/O throughput across my farms and clusters. CPU and RAM are not a problem at the moment. What do I do?	Add new I/O cards to your existing 4-way systems.

3.8.1 Scale-up step-by-step

The previous section describes the concepts of scale-up with ESX Server on the x3950. In this section, we describe how such upgrades are implemented in ESX

Server. We use as our example, an upgrade of a 4-way (single node) x3950 running ESX Server 3.0 to an 8-way (two node) x3950.

Figure 3-59 on page 167 shows the CPU and memory of the existing 4-way configuration.



9.42.171.66 VMware ESX Server, e.x.p, 16563

General

State: connected

Manufacturer: IBM
Model: eserver xSeries 460-[88741RZ]-
Processor: Intel(R) Xeon(TM) MP CPU 3.33GHz 3.3 GHz

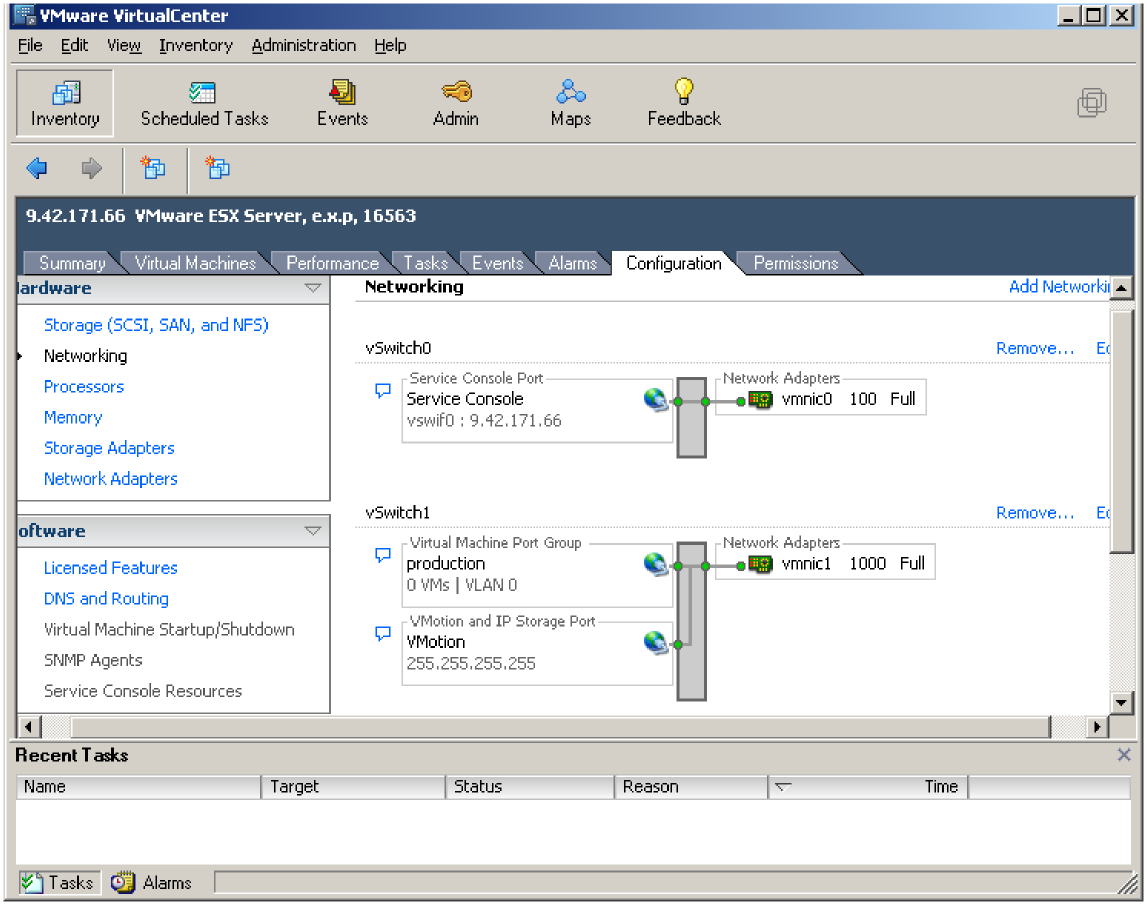
Number of Processors: 4
Total Memory: 4.00 GB
Number of Nics: 0

Virtual Machines: 0
VMotion Enabled: no

Active Tasks:

Figure 3-59 x3950 4-way CPU and memory

Figure 3-60 on page 168 shows the current networking configuration:



9.42.171.66 VMware ESX Server, e.x.p, 16563

Networking

vSwitch0

Service Console Port
Service Console
vswif0 : 9.42.171.66

Network Adapters
vmnic0 100 Full

vSwitch1

Virtual Machine Port Group
production
0 VMs | VLAN 0

VMotion and IP Storage Port
VMotion
255.255.255.255

Network Adapters
vmnic1 1000 Full

Figure 3-60 x3950 4-way current networking configuration

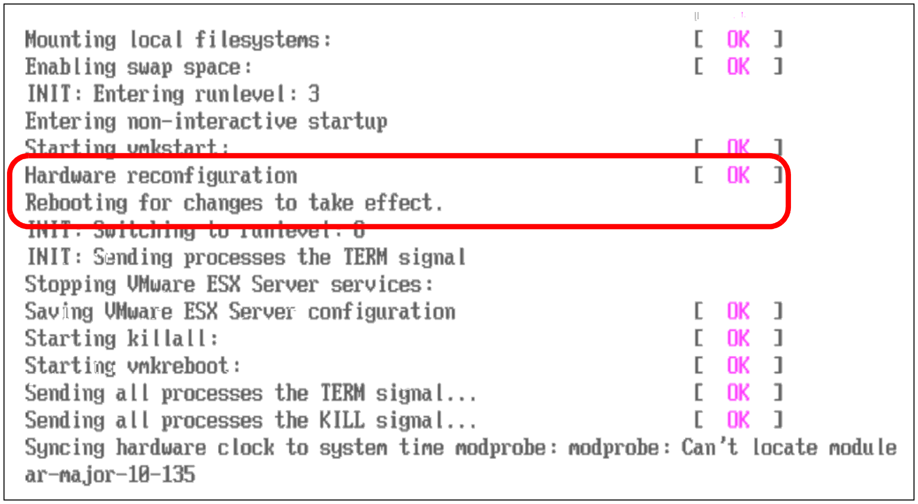
In our example, we are using the two integrated Gigabit Ethernet controllers. As you can see from Figure 3-60, the first connection is used to support the Console OS, while the second connection is shared between the VMotion port group and the production port group. Note that this is not an optimal networking configuration, simply a test configuration.

The procedure to upgrade our configuration to 8-way is:

1. Shutdown ESX Server and the x3950. You must power off the x3950 before connecting additional nodes.
2. Connect the new x3950 or x3950 E to the existing x3950 using the scalability cables and ensure all BIOS and firmware levels are matching and up-to-date. See 2.5, “Multi-node configurations” on page 46 for details.

3. Configure the two-node partition using the Web interface of the service processor in the first x3950 as described in 2.6, “Scalable system setup” on page 54.
4. Power on the first x3950 node, which should automatically power on the second node as well. The system should boot to ESX Server from the local drives in the first node, just as it did as a single-node configuration.

   During boot, ESX Server detects the new resources: new CPUs, memory, PCI-X slots, and I/O devices such as the Ethernet controllers in the second node. ESX Server requires a reboot before the changes in configuration can take effect, as circled in Figure 3-61.

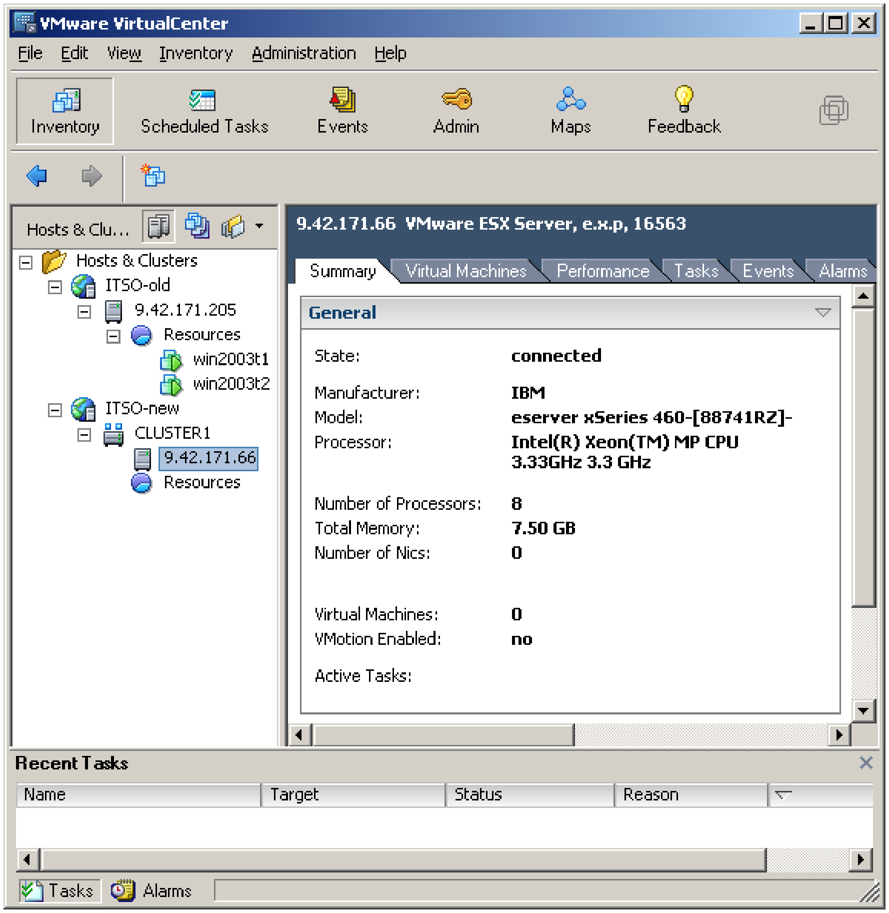


```
Mounting local filesystems:                                  [  OK  ]
Enabling swap space:                                         [  OK  ]
INIT: Entering runlevel: 3
Entering non-interactive startup
Starting vmkstart:                                           [  OK  ]
Hardware reconfiguration                                     [  OK  ]
Rebooting for changes to take effect.
INIT: Switching to runlevel: 0
INIT: Sending processes the TERM signal
Stopping VMware ESX Server services:
Saving VMware ESX Server configuration                       [  OK  ]
Starting killall:                                            [  OK  ]
Starting vmkreboot:                                          [  OK  ]
Sending all processes the TERM signal...                     [  OK  ]
Sending all processes the KILL signal...                     [  OK  ]
Syncing hardware clock to system time modprobe: modprobe: Can't locate module
ar-major-10-135
```

Figure 3-61 ESX Server reboots so the hardware reconfiguration can take effect

5. After the two-node system has rebooted, the memory and CPU configuration is correctly updated, as shown in Figure 3-62 on page 170.

> **Note:** You might need to add additional licenses to your server to accommodate the larger number of processors.



9.42.171.66 VMware ESX Server, e.x.p, 16563

State: connected

Manufacturer: IBM

Model: eserver xSeries 460-[88741RZ]-

Processor: Intel(R) Xeon(TM) MP CPU 3.33GHz 3.3 GHz

Number of Processors: 8

Total Memory: 7.50 GB

Number of Nics: 0

Virtual Machines: 0

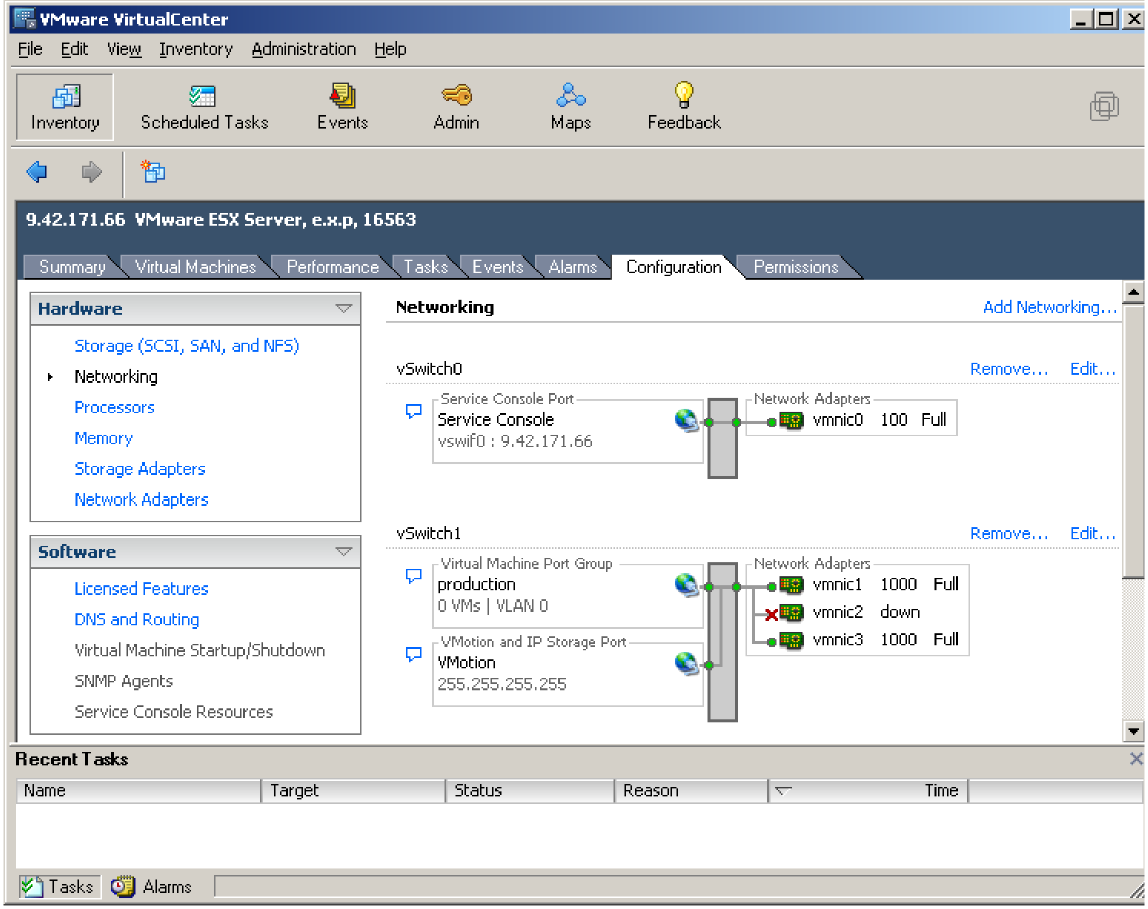
VMotion Enabled: no

Active Tasks:

Figure 3-62 x3950 8-way new CPU and memory count

6. The networking reconfiguration is similar. The two new integrated Gigabit Ethernet controllers in node 2 were detected. For our configuration, we have decided to assign these two new controllers to the switch where the VMotion and production port groups were defined as shown in Figure 3-63 on page 171.

   The virtual switch vSwitch1 can now balance across two network controllers, one in the primary node and the other in the secondary. In our configuration, the first NIC in the secondary node, vmnic2, is not connected to Ethernet network.



9.42.171.66 VMware ESX Server, e.x.p, 16563

Hardware

- Storage (SCSI, SAN, and NFS)
- Networking
- Processors
- Memory
- Storage Adapters
- Network Adapters

Software

- Licensed Features
- DNS and Routing
- Virtual Machine Startup/Shutdown
- SNMP Agents
- Service Console Resources

Networking

vSwitch0

Service Console Port
Service Console
vswif0 : 9.42.171.66

Network Adapters
vmnic0 100 Full

vSwitch1

Virtual Machine Port Group
production
0 VMs | VLAN 0

VMotion and IP Storage Port
VMotion
255.255.255.255

Network Adapters
vmnic1 1000 Full
vmnic2 down
vmnic3 1000 Full

Name	Target	Status	Reason	Time

Figure 3-63 x3950 48way new networking configuration

4

Microsoft Virtual Server

This chapter describes how to configure Microsoft Virtual Server to take advantage of the processing power and scalability features of the x3950.

The topics covered in this chapter are as follows:

4.1 Overview

Microsoft Virtual Server 2005 is hosted on the Microsoft Windows Server 2003 or Windows XP (for nonproduction use only). It is also capable of running on a 32-bit or 64-bit platform. Virtual Server runs better in a fully 64-bit platform beca4se it offers better scaling from larger kernel address space and a 64-bit system can typically have more RAM.

Virtual Server offers cost saving tasks through virtual machines with advanced levels of scalability, manageability and reliability. It is designed to deliver cost saving tasks in software test and development, legacy rehosting and server consolidation.

The concept of implementing a Microsoft Virtual Server solution on the x3950 platform is shown in Figure 4-1.

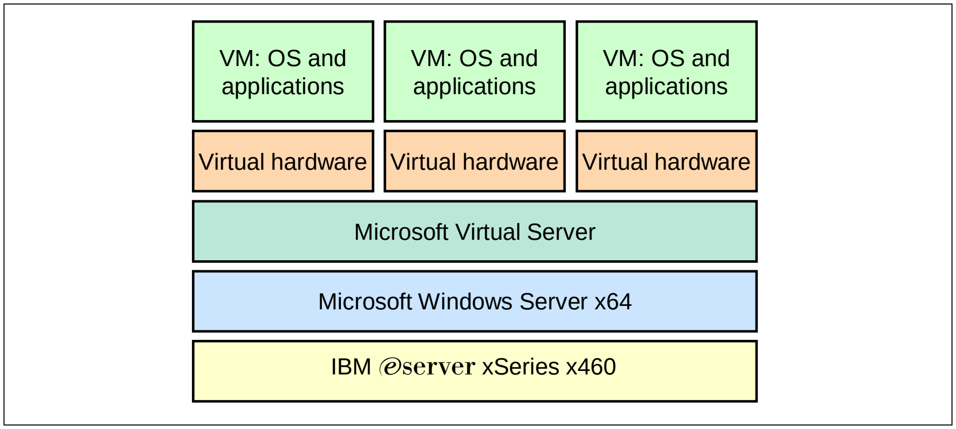


Figure 4-1 Microsoft Virtual Server concept

Microsoft Virtual Server is a multi-threaded application and runs as a Windows system service. Each VM is running its own thread of execution. Virtual Server derives two core functions from the host operating system:

- The underlying host operating system kernel schedules CPU resources.
- The device drivers of the host operating system provide access to system devices.

The Virtual Machine Manager (VMM) provides the software infrastructure to create and emulate virtual machines, manage instances, and interact with guest operating systems.

See the *Microsoft Virtual Server 2005 Technical Overview* for a complete list of what hardware that is emulated in the VM, available from:

http://www.microsoft.com/windowsserversystem/virtualserver/overview/vs2005tech.mspx

The following operating systems are supported as the host for Microsoft Virtual Server:

- 32-bit operating systems
  - Windows Server 2003 Standard Edition or later
  - Windows Server 2003 Enterprise Edition or later
  - Windows Server 2003 Datacenter Edition or later
  - Windows Small Business Server 2003 Standard Edition or later
  - Windows Small Business Server 2003 Premium Edition or later
  - Windows XP Professional (for non-production use only)
- 64-bit operating systems (64-bit support only for Virtual Server 2005 R2)
  - Windows Server 2003 x64 Standard Edition or later
  - Windows Server 2003 x64 Enterprise Edition or later
  - Windows Server 2003 x64 Datacenter Edition or later
  - Windows XP Professional x64 (for non-production use only)

The guest operating systems running on the virtual machines can only be 32-bit (64-bit virtual machines currently not supported) and the following Operating Systems are supported;

- Windows Server 2003, Standard, Enterprise, and Web Editions.

  Note: Windows Server 2003 SP1 will run only as a guest in Virtual Server 2005 R2 only.
- Windows Small Business Server 2003, Standard & Premium Editions
- Windows 2000 Server and Advanced Server
- Windows NT Server 4.0, Service Pack 6a
- Windows XP SP2 (only on Virtual Server 2005 R2)
- Linux

  A number of Red Hat and SUSE Linux distributions are supported as guests for Virtual Server 2005 R2. The complete list is available from:

  http://www.microsoft.com/windowsserversystem/virtualserver/evaluation/linuxguestsupport

The Microsoft Virtual Server Migration Toolkit (VSMT) is a free toolkit for Virtual Server that allows businesses to migrate physical platforms to a virtual server environment, without manually reinstalling the software, as shown in Figure 4-2 on page 176.

The Virtual Server Migration Toolkit is available from:

http://www.microsoft.com/windowsserversystem/virtualserver/downloads

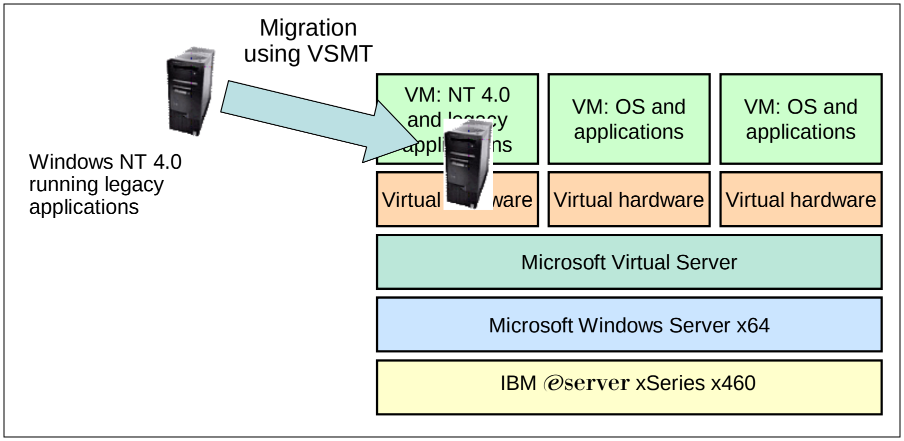


Figure 4-2 Migration using the Virtual Server Migration Toolkit

High availability is offered through clustering of virtual machines across Virtual Server hosts, as shown in the Figure 4-3.

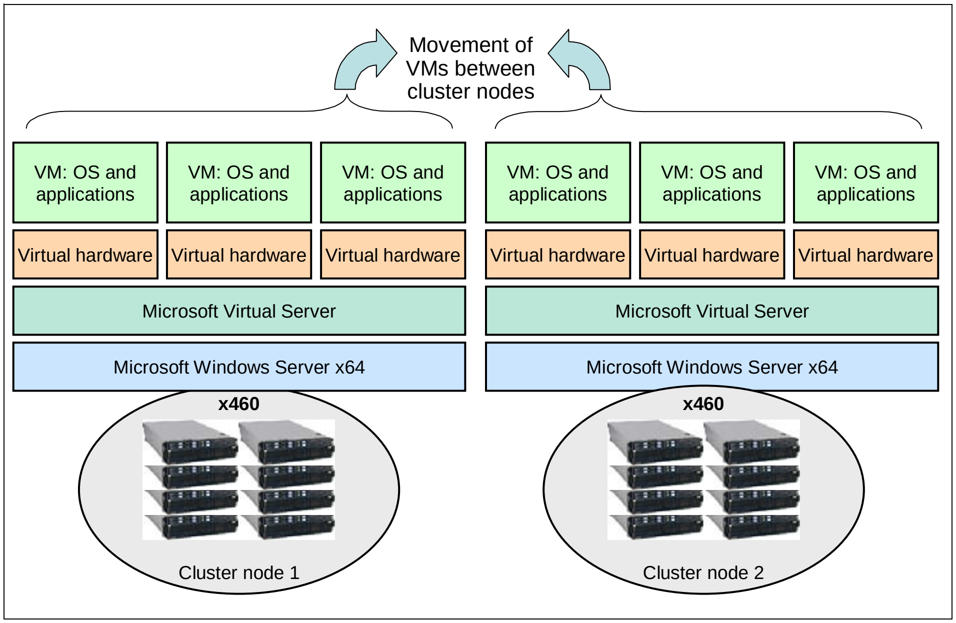


Figure 4-3 Clustering of Virtual Server systems

Virtual Server host clustering uses Windows Server 2003 Enterprise Edition or Datacenter Edition clustering and about 250 lines of Visual Basic® scripts. It supports up to eight cluster nodes and can utilize SAN, iSCSI, or direct attached storage.

With this configuration you can do the following:

- Service the host hardware or patching the host operating system by migrating all VMs to the other cluster node.
- Migrade virtual machines from one cluster node to another with minimal downtime. It takes less than 10 seconds to move a 128 MB virtual machine on Gigabit Ethernet iSCSI.
- Failover VMs to other cluster nodes due to hardware failure. The VM is suspended during failover. After the LUNs have failed over, the VM resumes on the second node. This is accomplished through scripting and failover takes around 10-15 seconds.

The Virtual Server host clustering code is based from the clustering service that is part of Windows Enterprise or DataCenter edition. See section 4.7.2, “Clustered installation” on page 184 for more information.

4.2 What is new for Microsoft Virtual Server 2005 R2?

There are two different versions of Microsoft Virtual Server:

- Microsoft Virtual Server 2005
- Microsoft Virtual Server 2005 R2

Here is the main features that are offered with R2:

- Greater scalability with 64-bit
  - Support for x64 hosts running 32-bit guest operating systems
    - Windows Server 2003 Standard x64 Edition
    - Windows Server 2003 Enterprise x64 Edition
    - Windows XP Professional x64 Edition
  - The VMM service is 64-bit if using 64-bit version on 64-bit OS
    - Better scaling from larger kernel address space
    - x64 systems typically can have more RAM
- Increased performance
  - Improved shadow page table management
  - Improved performance of guest process switching and memory intensive applications
  - Increase of 65% in internal TPC-C in memory tests
  - Early customer saw a 50% drop in CPU utilization
- Higher Availability
  - Support for iSCSI allows clustering virtual machines across hosts
  - Host cluster support
    - The ability to cluster Virtual Server (VS) hosts
    - VS host clustering uses Windows Server 2003 EE/DC clustering
    - VS host clustering supports SAN, iSCSI or direct attached storage
  - Planned downtime:
    - Servicing the host hardware or patching the host operating system
    - Virtual machine migration using clustering
    - Less than 10 seconds to move 128 MB VM via Gigabit Ethernet iSCSI
    - Faster on SAN
  - Unplanned downtime:
    - Failover to another cluster node due to hardware failure
- PXE booting
  - Emulated Ethernet card now supports PXE booting

- Integrates virtual machines into deployment infrastructure

- Better interoperability
  - Support for Linux guest operating systems
  - Additional Windows guests supported
    - Windows Server 2003 Standard Edition Service Pack 1
    - Windows XP Professional Service Pack 2
- Other improvements
  - Improved security features
  - Improved Hyper-Threading support
  - SCSI disk driver support via F6 at boot
  - Virtual disk precompactor

    This utility is designed to zero out any available blank space on a virtual hard disk. This will compact your dynamically expanding virtual hard disks (VHDs) and reduces the amount of disk space that they use.
  - Opens necessary ports at install time
  - Reserve space for saved states

4.3 NUMA and Microsoft Virtual Server

Microsoft Virtual Server is able to utilize the Non-Uniform Memory Access (NUMA) architecture on the x3950. Virtual Server uses the Static Resource Affinity Table (SRAT) to determine the affinity of the CPU and memory resources.

The use of the SRAT table ensures that whereever possible, processes are run in the same server as where the memory is allocated. The SRAT describes both installed and hot-pluggable memory, that is, memory that can be added or removed while the system is running, without requiring a reboot.

The NUMA architecture is a way of building very large multi-processor systems without jeopardizing hardware scalability. The name NUMA is not completely correct because not only memory can be accessed in a nonuniform manner but also I/O resources. NUMA effectively means that every processor or every group of processors has a certain amount of memory local to it.

Multiple processors or multiple groups of processors are then connected together using special bus systems (for example the scalability ports of a x3950) to provide processor data coherency. The essence of the NUMA architecture is the existence of multiple memory subsystems, as opposed to a single one on a SMP system.

The so called local or near memory has the very same characteristics as the memory subsystem in a SMP system. But by limiting the number of processors that directly access that memory, performance is improved because of the much shorter queue of requests. Because each group of processors has its local memory, memory on another group of processors would be considered to be remote to the local processor. This remote memory can be accessed, but at a longer latency than local memory. All requests between local and remote memory flow over the inter-processor connection (scalability ports). See 2.4, "NUMA Architecture" on page 45 for more information about how NUMA works on the x3950 platform.

If memory is not local at any time during normal operation, the Microsoft Virtual Server logs an error in the event log indicating that it is not using local memory.

Although there is no processor affinity within Microsoft Virtual Server, the maximum that you can allocate to any given virtual machine is 100% of one processor (SMP in VMs is not supported). If you want to allocate the equivalent of one entire processor to a virtual machine, specify 100% as its reserved capacity. The VMs are populated through the processors on node 0 first. When node 0 has been fully populated, it will move on to next node to populate the additional VMs. This is handled through the Windows scheduler.

4.4 Why x3950 for Microsoft Virtual Server?

The x3950 is capable of scaling up to eight nodes with four processors in each node, making a 32-way with single or dual core processors. This means that a x3950 is capable of providing 64 logical processors across 32 processor sockets (or 128 with Hyper-Threading enabled).

The x3950 and Microsoft Virtual Server together is capable of adding nodes and processors as you grow. For example, a two-node, 8-way solution could grow to a 16-way configuration by adding two nodes each with four processors. Currently, Microsoft Virtual Server Standard Edition supports up to four processors and Enterprise Edition can support up to 32 processors. Both editions support up to 64 GB of RAM.

Important: Virtual Server provides its own resource manager for allocating system resources to virtual machines. You should not use any other resource manager, such as Windows System Resource Manager, with Microsoft Virtual Server.

Each VM can have up to 3712 MB of RAM. It is highly recommended that you ensure that there is enough RAM in the local node for this VM to ensure that memory will always be local within the node. This will increase the overall performance of the x3950.

The x3950 offers up to 64 GB SDRAM per chassis. Memory can be added while the server is up and running with certain limitations. Hot-add memory is supported with Microsoft Windows Server 2003 and it powers the dynamic addition of main memory to increase performance. Both Microsoft Virtual Server and the virtual machines running under Microsoft Virtual Server do not support hot-add memory. However, by restarting the Virtual Server service, it should be able to access the added memory.

Consult the Microsoft Article 903748 for Microsoft Virtual Server performance tips, available at:

http://support.microsoft.com/default.aspx?scid=kb;en-us;903748

4.5 Dual-core processors and Hyper-Threading

The x3950 offers both single and dual-core processors and the dual-core processor improve performance by up to 50%.

Microsoft Virtual Server is licensed by the socket, so Microsoft Virtual Server 2005 R2 Enterprise Edition will support up to 32 sockets, regardless of the number of cores or whether Hyper-Threading is enabled. This means that Enterprise Edition 2005 R2 can handle up to 128 logical processor (32 dual core processors with Hyper-Threading enabled).

Consult the Virtual Server 2005 R2 pricing and licensing information from Microsoft for additional information, available from:

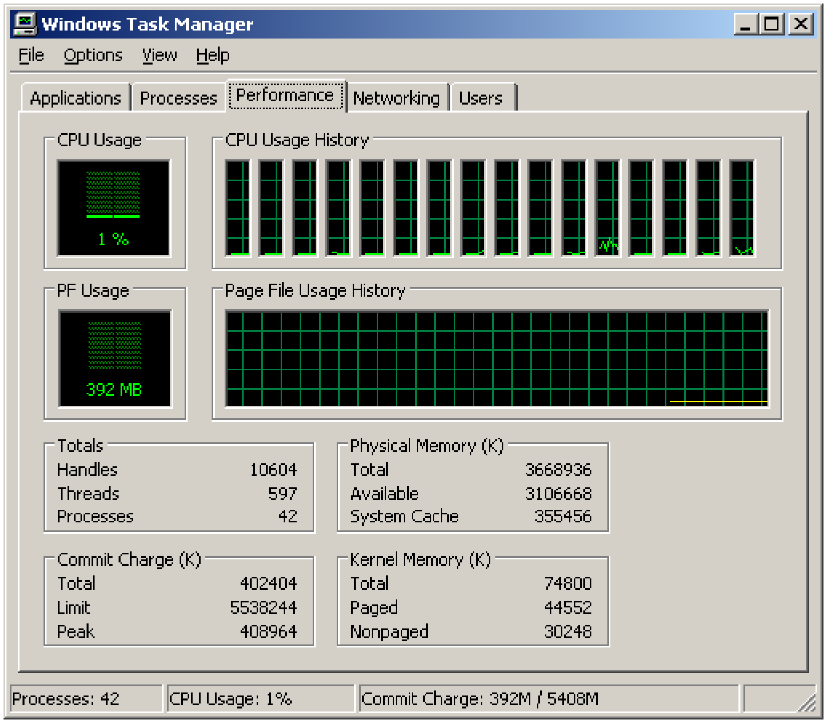
http://www.microsoft.com/windowsserversystem/virtualserver/howtobuy

For example, consider an 8-node x3950 configuration with its 32 processor sockets. Each socket has a dual-core processor installed and Hyper-Threading is enabled. This results in the operating system seeing 128 logical processors. because Microsoft Virtual Server 2005 R2 is licensed by the socket, all 128 logical processors are available for use. Virtual Server Standard Edition is licensed for four processor sockets and Enterprise Edition is licensed for 32 processor sockets.

With Microsoft Virtual Server 2005 (R1), Microsoft recommends that Hyper-Threading be disabled. However, that no longer applies to Virtual Server 2005 R2. Microsoft has changed the way that threads are scheduled, and no longer schedules the main virtual machine thread on a logical processor.

Each VM has is its own threads of execution and the task scheduler in Virtual Server 2005 R2 always attempts to schedule threads to different processor cores, thereby avoiding scheduling threads to the logical processors on the same core. This means that on an x3950 system with four dual-core processors and eight virtual machines running, the VMs will be spread across all eight processor cores, even though Hyper-Threading might be enabled and the operating system recognizing 16 logical processors.

As shown in Figure 4-4, you can see that a four-way dual-core x3950 system will present with 16 processors to the operating system.



Totals	
Handles	10604
Threads	597
Processes	42

Physical Memory (K)	
Total	3668936
Available	3106668
System Cache	355456

Commit Charge (K)	
Total	402404
Limit	5538244
Peak	408964

Kernel Memory (K)	
Total	74800
Paged	44552
Nonpaged	30248

Processes: 42 CPU Usage: 1% Commit Charge: 392M / 5408M

Figure 4-4 x3950 with 4 dual core processor and Hyper-Threading enabled

If you look at the Device Manager, you will see that the operating system recognizes 16 processors. If you disable Hyper-Threading, you will see eight processors instead of 16.

4.6 Sizing

The rule of thumb is that each VM requires the same amount of hardware as compared to a physical machine. Microsoft Virtual Server allows up to 64 VMs to be created, or as many as the hardware allows, depending on the amount of available RAM, disk, and system resources, or up to 64 virtual machines. The maximum amount of RAM that Virtual Server can use is 64 GB. Each VM can have up to 3712 MB of RAM and a maximum of one processor core.

Some general rules of thumb for Virtual Server configurations are:

- Memory
  - A virtual machine needs as much memory as a physical machine, plus an overhead (about 25 MB per virtual machine)
  - Overhead for MMU virtualization, video emulation, and so on
- CPU
  - Virtual Server provides CPU allocation controls
    - Maximum (not to exceed)
    - Reserve (will always have enough)
    - Weights (weighted average)
  - One virtual processor per virtual machine
- I/O
  - Virtual machines will share disk and networking paths
  - A virtual machine does not do less I/O than a physical machine
  - Configure systems with multiple I/O paths
    - Multiple spindles for disks
    - Controllers can become the bottleneck
    - Multiple NICs for performance, isolation and resilience

4.7 Virtual Server installation

This section describes how to install Virtual Server, both in a nonclustered and a clustered configuration.

4.7.1 Nonclustered installation

As we mention in the overview, we highly recommend that you install Virtual Server on a 64-bit version of Windows because it offers better scaling from larger kernel address space.

To install the host operating system, follow one of the following sets of instructions:

- Installing Windows Server 2003 x64 Edition on the x3950

  http://www.pc.ibm.com/support?page=MIGR-60676

- Installing Windows Server 2003 (32-bit) on the x3950

  http://www.pc.ibm.com/support?page=MIGR-61178

You also need to install the Internet Information Services (IIS) component of the operating system:

1. Open the **Control Panel**.
2. Open **Add or Remove Programs**.
3. Select **Add/Remove Windows Components**.
4. Select **Application Server** and select **Internet Information Services**.

To install Microsoft Virtual Server, follow these instructions:

http://www.pc.ibm.com/support?page=MIGR-61438

The Microsoft Virtual Server Administrator's Guide is useful when configuring and managing the Microsoft Virtual Server. It is available at:

http://www.microsoft.com/technet/prodtechnol/virtualserver/2005/proddocs

4.7.2 Clustered installation

This section provides an overview to Virtual Server host clustering. With Virtual Server host clustering, you can provide computing capacity across a group of physical servers and, at the same time, maintain availability of the virtual machines. If one server requires scheduled or unscheduled downtime, another server is ready to begin supporting the virtual machines quickly. Users experience minimal disruptions in service.

Virtual Server host clustering is a way of combining Microsoft Virtual Server 2005 R2 with the server cluster feature in Microsoft Windows Server 2003. The setup in this example is shown in the Figure 4-5.

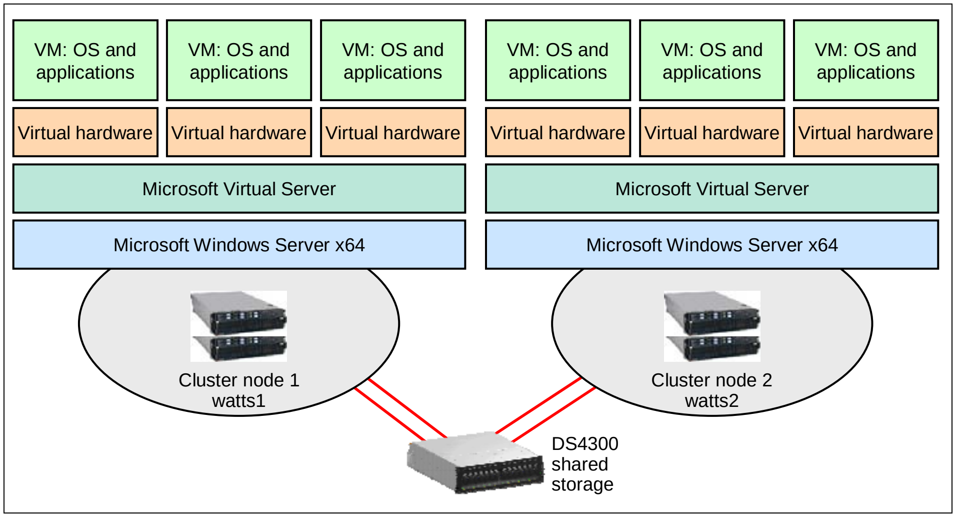


Figure 4-5 Cluster setup

As you can see in Figure 4-5, each cluster node consist of two x3950 cabled together to form a single 8-way server. Both of the clustering nodes share an IBM TotalStorage® DS4300 for storage. In this example we are using the clustering feature in Microsoft Windows Server 2003 x64 Enterprise Edition and the Microsoft Virtual Server 2005R2 Enterprise edition.

Follow these instructions to install to Virtual Server host clustering:

> **Note:** These instructions are based from the *Virtual Server Host Clustering Step-by-Step Guide for Virtual Server 2005 R2*, available from:
>
> `http://www.microsoft.com/downloads/details.aspx?FamilyID=09cc042b-154f-4eba-a548-89282d6eb1b3&displaylang=en`

1. Configure the x3950 systems' two 8-way complexes. See 2.5, "Multi-node configurations" on page 46 for instructions
2. Install Windows Server 2003 x64 Enterprise Edition using these instructions:

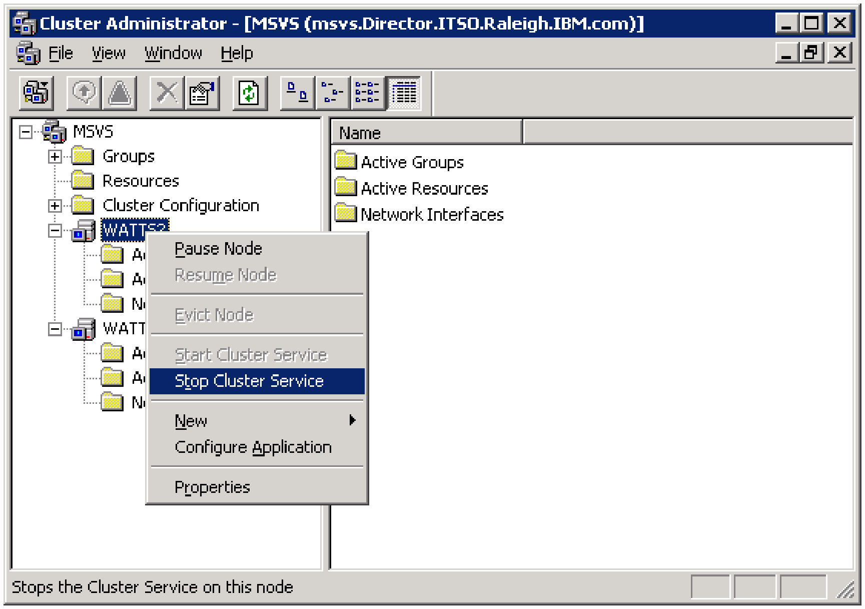
   `http://www.pc.ibm.com/support?page=MIGR-60676`
3. On each cluster node, install Internet Information Services (IIS). To install IIS, open the **Control Panel**. Select **Add or Remove Programs** and then select **Add/Remove Windows Components**. Select the check box for **IIS** under Application Server.

4. Install Microsoft cluster and form a cluster between the two x3950. Instructions are available from:

   http://go.microsoft.com/fwlink/?LinkId=55162

5. Install Microsoft Virtual Server on the first node:

   a. Stop the cluster service on the first node. From within Cluster Administrator, right-click the node and select **Stop Cluster Service**, as shown in Figure 4-6. Leave the cluster service on the second node running.



Cluster Administrator - [MSVS (msvs.Director.ITSO.Raleigh.IBM.com)]

Stops the Cluster Service on this node

Figure 4-6 Stop cluster service on node 1

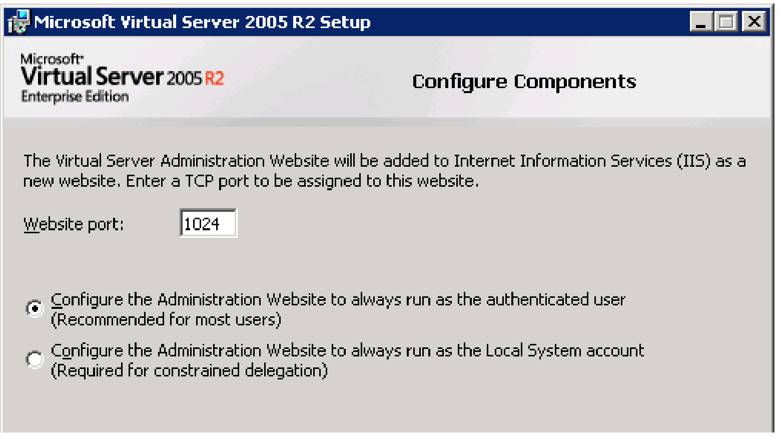
   b. Start the installation of Microsoft Virtual Server by inserting the CD-ROM.

   c. Accept the license agreement and select **Next** to continue.

   d. Enter your license information and select **Next** to continue.

   e. Select to do a **Complete** installation and select **Next** to continue.

   f. As shown in Figure 4-7 on page 187, type in the TCP port that will be used for remote administration. Ensure that this port is enabled in your firewall. In this example, we are using default values. Click **Next** to continue.


Microsoft Virtual Server 2005 R2 Setup
Microsoft Virtual Server 2005 R2 Enterprise Edition

Configure Components

The Virtual Server Administration Website will be added to Internet Information Services (IIS) as a new website. Enter a TCP port to be assigned to this website.

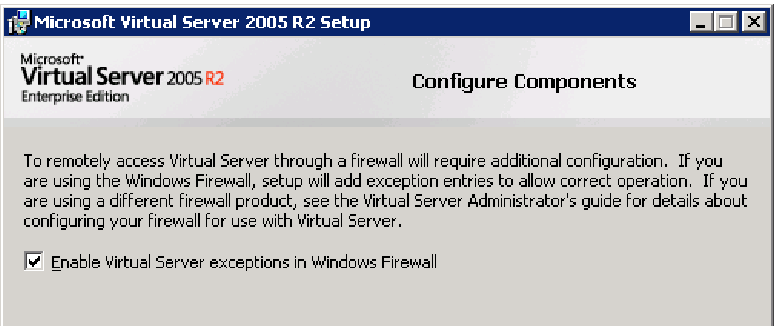
Website port: 1024

● Configure the Administration Website to always run as the authenticated user (Recommended for most users)

○ Configure the Administration Website to always run as the Local System account (Required for constrained delegation)

Figure 4-7 MSVS Administration Web site port

g. Select whether you want to allow Virtual Server to enable the firewall locally on the server for remote administration, as shown in Figure 4-8. In this example we are using defaults. select **Next**.


Microsoft Virtual Server 2005 R2 Setup
Microsoft Virtual Server 2005 R2 Enterprise Edition

Configure Components

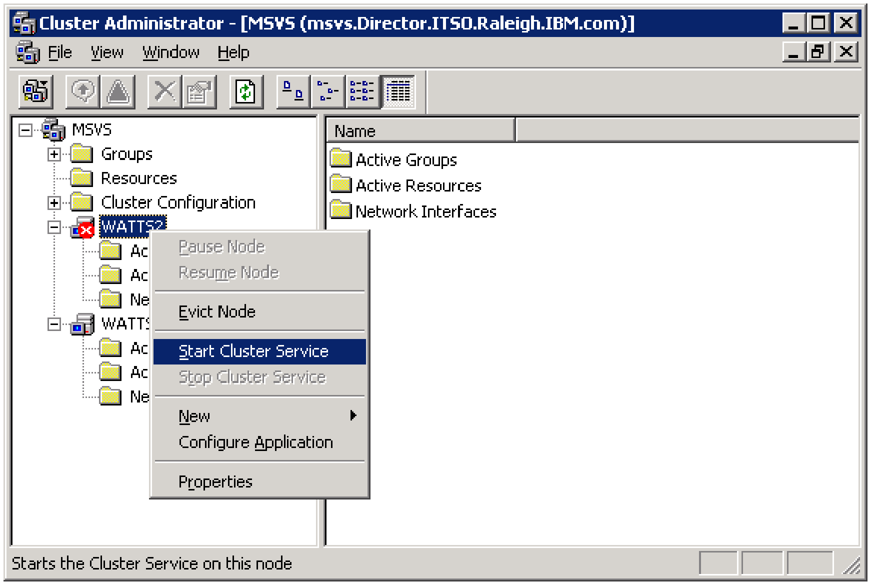
To remotely access Virtual Server through a firewall will require additional configuration. If you are using the Windows Firewall, setup will add exception entries to allow correct operation. If you are using a different firewall product, see the Virtual Server Administrator's guide for details about configuring your firewall for use with Virtual Server.

☑ Enable Virtual Server exceptions in Windows Firewall

Figure 4-8 MSVS Enable firewall

h. Click **Install** to begin the installation. Click **Finish** when prompted that the installation has complete.

i. Using the Cluster administrator, start the Cluster service on the first node where you just installed Microsoft Virtual Server, as shown in Figure 4-9 on page 188.

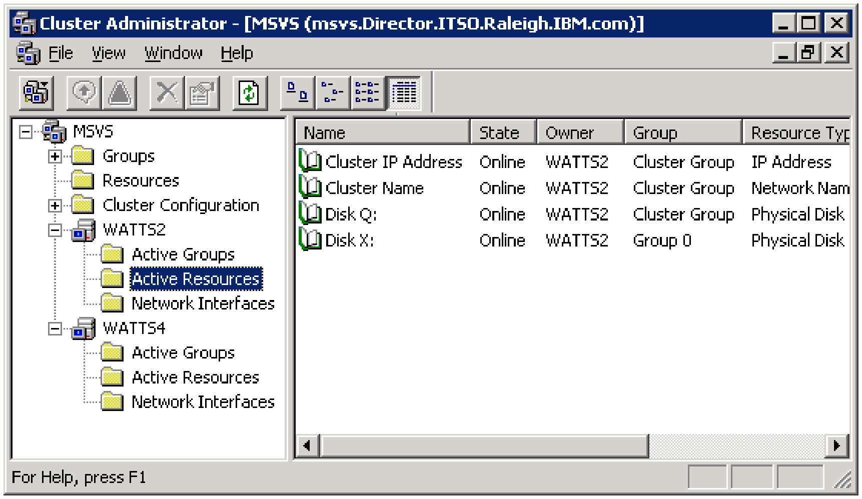


Starts the Cluster Service on this node

Figure 4-9 Start Cluster service on first node

6. Repeat all of step 5 on page 186 to install Microsoft Virtual Server on the second node in the cluster and then restart the cluster service on that node.

7. Verify that both of the nodes are online and that the cluster resources are online, as shown in the Figure 4-10.



Name	State	Owner	Group	Resource Typ
Cluster IP Address	Online	WATTS2	Cluster Group	IP Address
Cluster Name	Online	WATTS2	Cluster Group	Network Nam
Disk Q:	Online	WATTS2	Cluster Group	Physical Disk
Disk X:	Online	WATTS2	Group 0	Physical Disk

For Help, press F1

Figure 4-10 Verify cluster resources

8. Configure a shutdown script on both of the nodes to stop the cluster service on shutdown. Do the following on both nodes to accomplish this.

   a. In the root directory of the local hard disk on each node, create a batch file and name it `stop_clustersvc.cmd`.

   b. In the batch file, enter the following:

   ```
   net stop clussvc
   ```

   c. Save the batch file changes.

   d. Select **Start** → **Run**, and enter `gpedit.msc`.

   e. In the left panel, select **Local Computer Policy** → **Computer Configuration** → **Windows Settings** → **Scripts (Startup/Shutdown)**.

   f. In the right panel, double-click **Shutdown**.

   g. In the Shutdown Properties dialog box, select **Add**. For the Script Name, enter:

   ```
   c:\stop_clustersvc.cmd
   ```

   h. Close the Policy window.

9. Now configure the disk resource, resource group, and guest control script for the first virtual machine. First you need to decide on what node you want the virtual machine to be hosted. In this example we use node 2 as primary owner of this virtual machine.

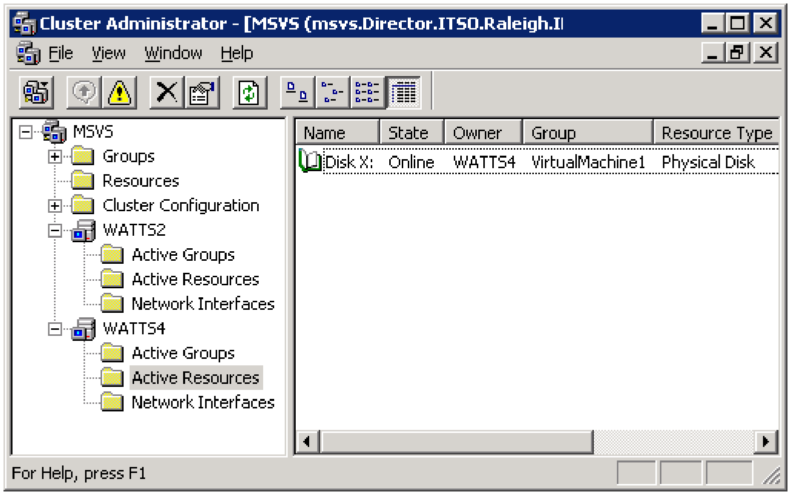
   Do the following to configure resources for the first VM:

   a. In Cluster Administrator, create a new resource group and name it `VirtualMachine1`. Specify node2 as the preferred owner. The name that you give the group is for administrative purposes only. It is not the same name that clients will use to connect to the group.

   b. Create a physical disk resource. For Possible Owners, make sure both cluster nodes are listed. Do not specify any dependencies.

   The properties of the physical disk in this example are shown in Figure 4-11 on page 190. As you can see, the disk has drive letter X: and associated with VirtualMachine1 group. The disk resource is online on the second node, WATTS4 in this example.

   > **Tip:** You may specify additional physical disk resources for each virtual machine. If you do specify additional disks, you need to ensure that the physical disk for data to be dependent for the physical disk resource with the guest operating system. This will ensure that all of the resources that are associated with the data disk are online before the guest operating system attempt to access the data on them.



Name	State	Owner	Group	Resource Type
Disk X:	Online	WATTS4	VirtualMachine1	Physical Disk

Figure 4-11 Physical disk

c. From the second node, create a folder on disk X: called Guest1, as shown in the Example 4-1. This folder will be used in a future step when installing and configuring the guest VM.

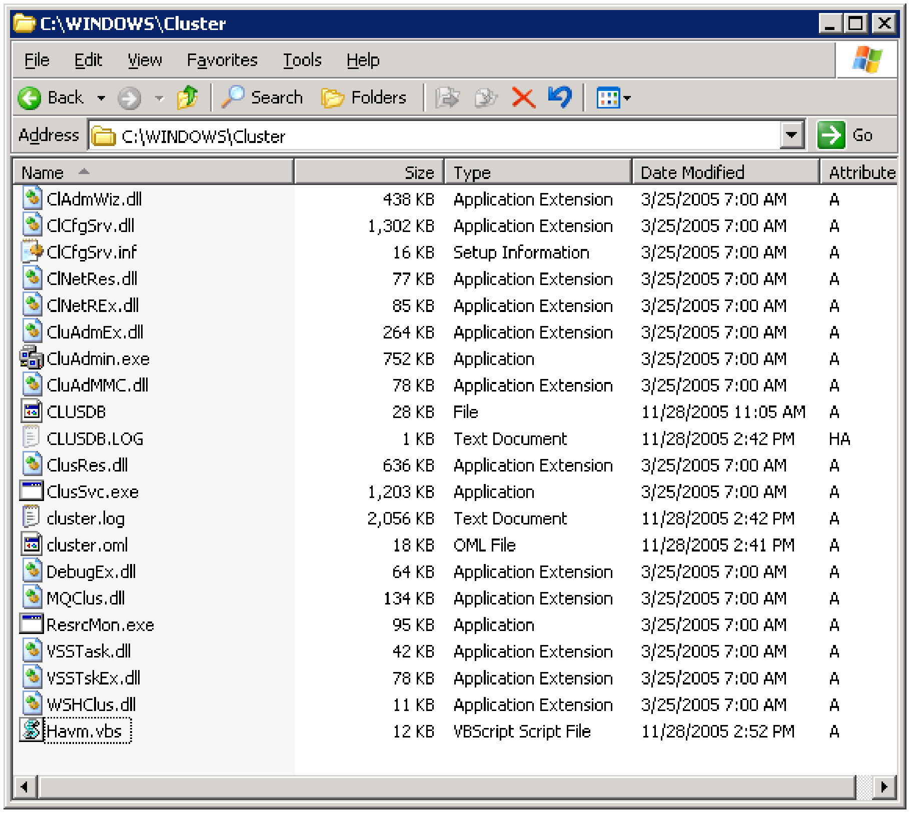
Example 4-1 Create folder called Guest1

```
X:\>mkdir Guest1
X:\>dir
 Volume in drive X is MSVS
 Volume Serial Number is 188D-72DE
 Directory of X:\
11/28/2005  02:48 PM    <DIR> Guest1
               0 File(s)              0 bytes
               1 Dir(s)   4,269,367,296 bytes free
X:\>
```

10. On each node's local disk, in the systemroot\Cluster folder, copy the script Havm.vbs, as shown in the Figure 4-12 on page 191. The script is available in Appendix B of *Virtual Server Host Clustering Step-by-Step Guide for Virtual Server 2005 R2*, available from:

    http://www.microsoft.com/downloads/details.aspx?FamilyID=09cc042b-154f-4eba-a548-89282d6eb1b3&displaylang=en

> **Note:** The script must be copied to the correct folder on each node's local hard disk, not to a disk in the cluster storage.



Name	Size	Type	Date Modified	Attribute
ClAdmWiz.dll	438 KB	Application Extension	3/25/2005 7:00 AM	A
ClCfgSrv.dll	1,302 KB	Application Extension	3/25/2005 7:00 AM	A
ClCfgSrv.inf	16 KB	Setup Information	3/25/2005 7:00 AM	A
ClNetRes.dll	77 KB	Application Extension	3/25/2005 7:00 AM	A
ClNetREx.dll	85 KB	Application Extension	3/25/2005 7:00 AM	A
CluAdmEx.dll	264 KB	Application Extension	3/25/2005 7:00 AM	A
CluAdmin.exe	752 KB	Application	3/25/2005 7:00 AM	A
CluAdMMC.dll	78 KB	Application Extension	3/25/2005 7:00 AM	A
CLUSDB	28 KB	File	11/28/2005 11:05 AM	A
CLUSDB.LOG	1 KB	Text Document	11/28/2005 2:42 PM	HA
ClusRes.dll	636 KB	Application Extension	3/25/2005 7:00 AM	A
ClusSvc.exe	1,203 KB	Application	3/25/2005 7:00 AM	A
cluster.log	2,056 KB	Text Document	11/28/2005 2:42 PM	A
cluster.oml	18 KB	OML File	11/28/2005 2:41 PM	A
DebugEx.dll	64 KB	Application Extension	3/25/2005 7:00 AM	A
MQClus.dll	134 KB	Application Extension	3/25/2005 7:00 AM	A
ResrcMon.exe	95 KB	Application	3/25/2005 7:00 AM	A
VSSTask.dll	42 KB	Application Extension	3/25/2005 7:00 AM	A
VSSTskEx.dll	78 KB	Application Extension	3/25/2005 7:00 AM	A
WSHClus.dll	11 KB	Application Extension	3/25/2005 7:00 AM	A
Havm.vbs	12 KB	VBScript Script File	11/28/2005 2:52 PM	A

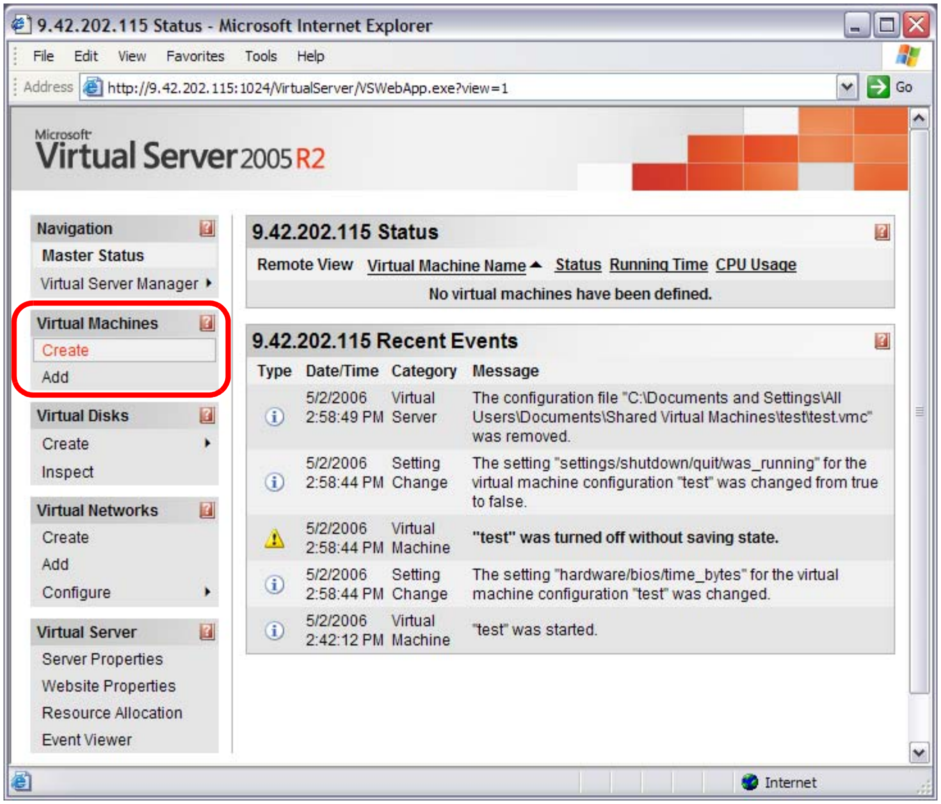
Figure 4-12 Copy the HAVM.VBS file to the C:\Windows\Cluster directory

The HAVM.VBS is a Visual Basic script that ensures that the guest VM functions correctly when a failover or other cluster-related processes occurs. The script also triggers restart of the guest VM if the guest VM stops running. The script is configured as a Generic Script resource in the cluster.

11. Now you need to install and configure the guest VM. All the configuration files of the guest VM must reside on the shared cluster disk resource, in this case drive X.

    a. On the computer that contains the management tool for Virtual Server 2005 R2, click **Start** → **Programs** → **Microsoft Virtual Server** → **Virtual Server Administration Web site**. Highlight the cluster node that currently owns the shared cluster disk resource X: (in Guest1Group).

    b. In the navigation pane, under Virtual Networks, click **Create**, as shown in Figure 4-13 on page 192.



9.42.202.115 Status

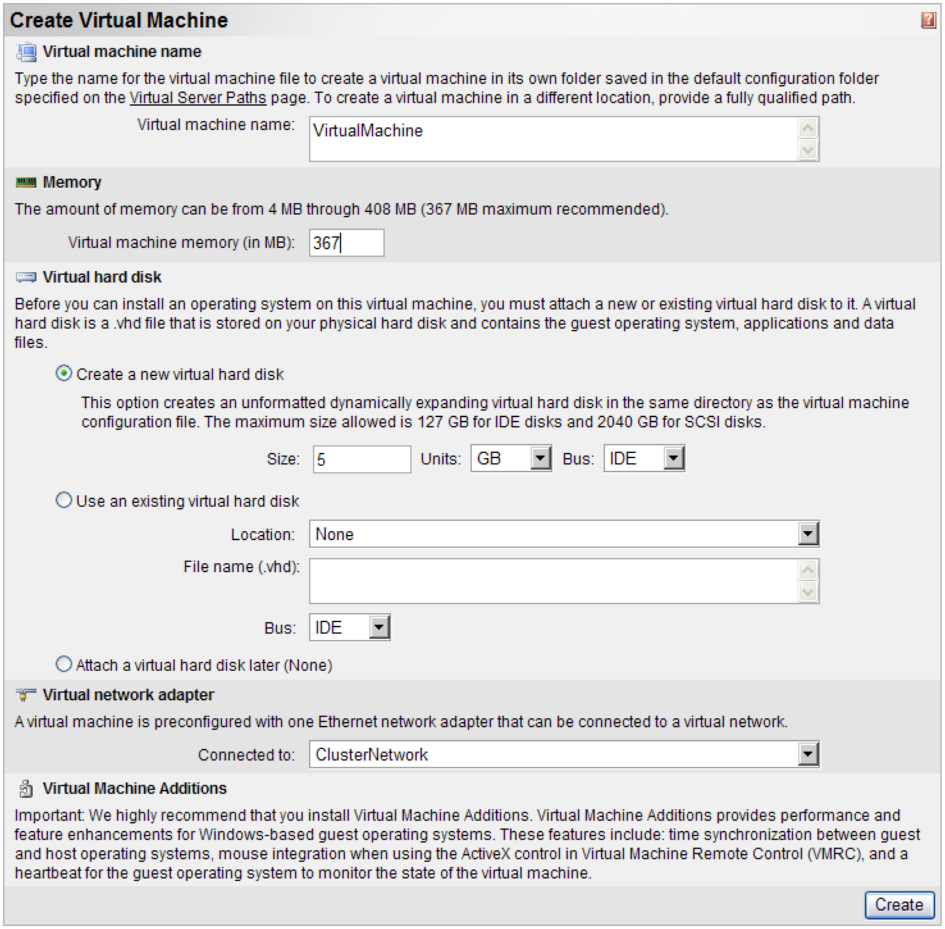
Remote View	Virtual Machine Name	Status	Running Time	CPU Usage
No virtual machines have been defined.				

9.42.202.115 Recent Events

Type	Date/Time	Category	Message
ⓘ	5/2/2006 2:58:49 PM	Virtual Server	The configuration file "C:\Documents and Settings\All Users\Documents\Shared Virtual Machines\test\test.vmc" was removed.
ⓘ	5/2/2006 2:58:44 PM	Setting Change	The setting "settings/shutdown/quit/was_running" for the virtual machine configuration "test" was changed from true to false.
⚠	5/2/2006 2:58:44 PM	Virtual Machine	**"test" was turned off without saving state.**
ⓘ	5/2/2006 2:58:44 PM	Setting Change	The setting "hardware/bios/time_bytes" for the virtual machine configuration "test" was changed.
ⓘ	5/2/2006 2:42:12 PM	Virtual Machine	"test" was started.

Figure 4-13 Create Virtual Machine - Step 1

c. Create the VMM, as shown in Figure 4-14 on page 193. Ensure that in the Network adapter on the physical computer, you select the network adapter associated with the public network (not the private network) and then click **Create**.



Create Virtual Machine

Virtual machine name

Type the name for the virtual machine file to create a virtual machine in its own folder saved in the default configuration folder specified on the Virtual Server Paths page. To create a virtual machine in a different location, provide a fully qualified path.

Virtual machine name: VirtualMachine

Memory

The amount of memory can be from 4 MB through 408 MB (367 MB maximum recommended).

Virtual machine memory (in MB): 367

Virtual hard disk

Before you can install an operating system on this virtual machine, you must attach a new or existing virtual hard disk to it. A virtual hard disk is a .vhd file that is stored on your physical hard disk and contains the guest operating system, applications and data files.

● Create a new virtual hard disk

This option creates an unformatted dynamically expanding virtual hard disk in the same directory as the virtual machine configuration file. The maximum size allowed is 127 GB for IDE disks and 2040 GB for SCSI disks.

Size: 5 Units: GB Bus: IDE

○ Use an existing virtual hard disk

Location: None

File name (.vhd):

Bus: IDE

○ Attach a virtual hard disk later (None)

Virtual network adapter

A virtual machine is preconfigured with one Ethernet network adapter that can be connected to a virtual network.

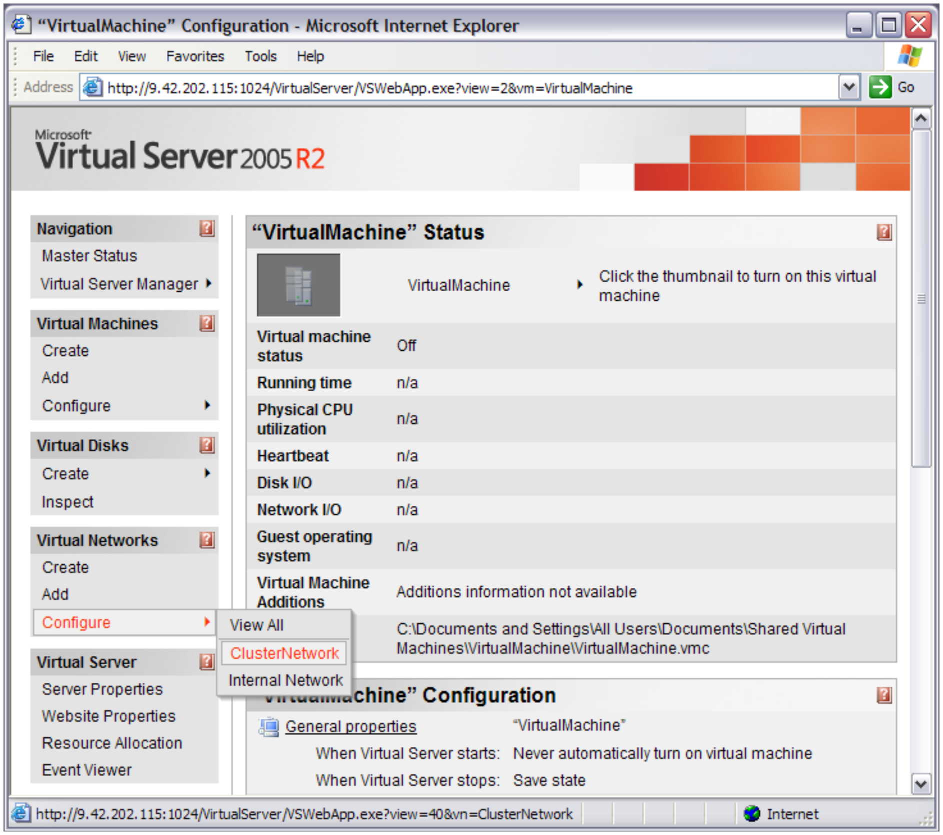
Connected to: ClusterNetwork

Virtual Machine Additions

Important: We highly recommend that you install Virtual Machine Additions. Virtual Machine Additions provides performance and feature enhancements for Windows-based guest operating systems. These features include: time synchronization between guest and host operating systems, mouse integration when using the ActiveX control in Virtual Machine Remote Control (VMRC), and a heartbeat for the guest operating system to monitor the state of the virtual machine.

Figure 4-14 Create Virtual Machine - step 2

d. Select **Virtual Networks** → **Configure** → **ClusterNetwork**, as shown in the Figure 4-15 on page 194.



Address: http://9.42.202.115:1024/VirtualServer/VSWebApp.exe?view=2&vm=VirtualMachine

"VirtualMachine" Status

VirtualMachine — Click the thumbnail to turn on this virtual machine

Virtual machine status	Off
Running time	n/a
Physical CPU utilization	n/a
Heartbeat	n/a
Disk I/O	n/a
Network I/O	n/a
Guest operating system	n/a
Virtual Machine Additions	Additions information not available
	C:\Documents and Settings\All Users\Documents\Shared Virtual Machines\VirtualMachine\VirtualMachine.vmc

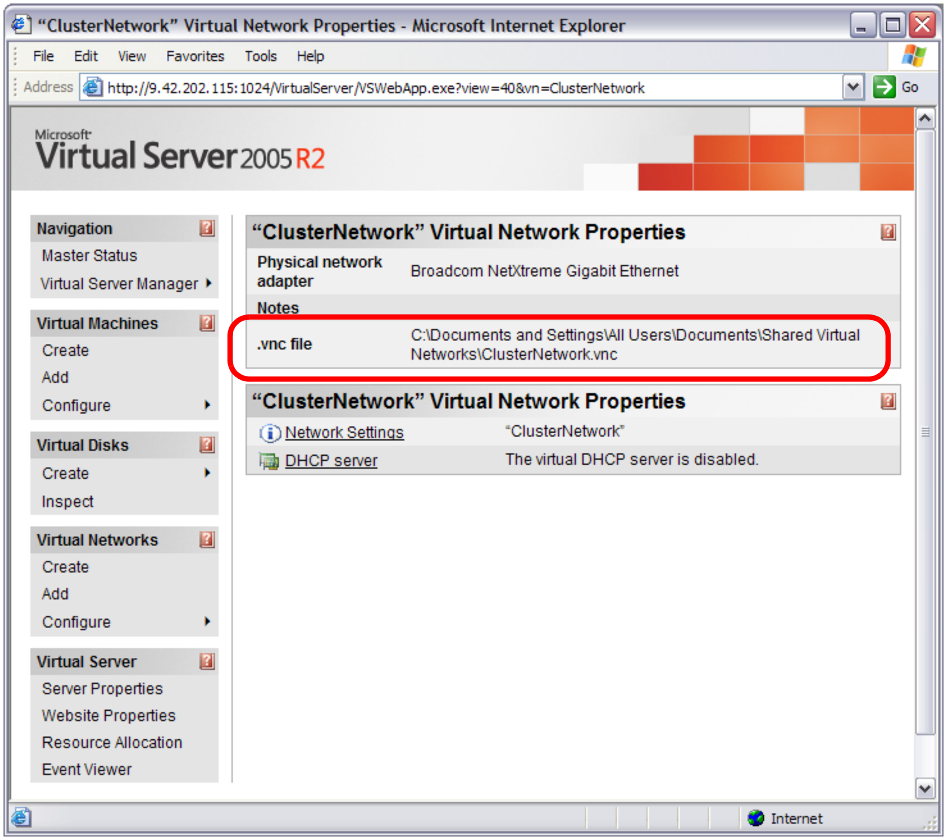
"VirtualMachine" Configuration

General properties: "VirtualMachine"
When Virtual Server starts: Never automatically turn on virtual machine
When Virtual Server stops: Save state

http://9.42.202.115:1024/VirtualServer/VSWebApp.exe?view=40&vn=ClusterNetwork

Figure 4-15 Configure External Network

e. In the line labeled **.vnc file** in Figure 4-16 on page 195, select the path, then copy and paste it into a text editor such as Notepad for later use.



"ClusterNetwork" Virtual Network Properties

Physical network adapter	Broadcom NetXtreme Gigabit Ethernet
Notes	
.vnc file	C:\Documents and Settings\All Users\Documents\Shared Virtual Networks\ClusterNetwork.vnc

"ClusterNetwork" Virtual Network Properties

Network Settings	"ClusterNetwork"
DHCP server	The virtual DHCP server is disabled.

Figure 4-16 Note the location of the .vnc file

f. In the Virtual Server Administration Web site, select **Virtual Networks** → **View All**.

g. Select the external virtual network you just created, and then click **Remove**, as shown in Figure 4-17 on page 196.

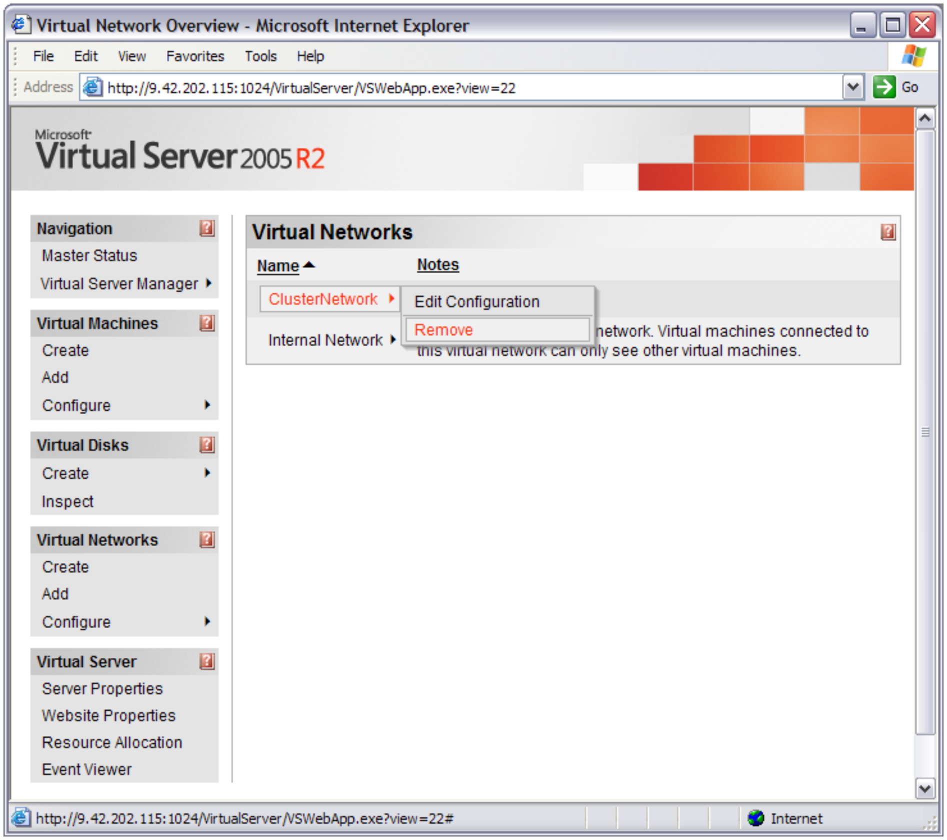

Virtual Network Overview - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://9.42.202.115:1024/VirtualServer/VSWebApp.exe?view=22
Go
Microsoft Virtual Server 2005 R2
Navigation
Master Status
Virtual Server Manager
Virtual Machines
Create
Add
Configure
Virtual Disks
Create
Inspect
Virtual Networks
Create
Add
Configure
Virtual Server
Server Properties
Website Properties
Resource Allocation
Event Viewer
Virtual Networks
Name
Notes
ClusterNetwork
Edit Configuration
Remove
Internal Network
network. Virtual machines connected to this virtual network can only see other virtual machines.
http://9.42.202.115:1024/VirtualServer/VSWebApp.exe?view=22#
Internet

Figure 4-17 Remove network connection

> **Tip:** The purpose of this step is not to undo the creation of the virtual network, but to clear Virtual Server of information that will prevent you from moving the configuration file for the virtual network (the .vnc file) to the cluster storage.

h. On the cluster node on which you created the .vnc file, open the command prompt, and then navigate to the path that you copied into a text file in the previous step. Move the *.vnc file to x:, as shown in Example 4-2.

Example 4-2 Move the .vnc file to x:

```
X:\>mkdir Guest1

x:\cd Documents and Settings\All Users\Documents\Shared Virtual
Networks
```

```
C:\Documents and Settings\All Users\Documents\Shared Virtual
Networks>dir
 Volume in drive C has no label.
 Volume Serial Number is AC79-5CA1
Directory of C:\Documents and Settings\All Users\Documents\Shared
Virtual Networks
03/20/2006  04:09 PM    <DIR>          .
03/20/2006  04:09 PM    <DIR>          ..
03/20/2006  04:09 PM             3,242 ClusterNetwork.vnc.vnc
03/20/2006  04:09 PM             2,938 Internal Network.vnc
2 File(s)          6,180 bytes
2 Dir(s)  76,374,700,032 bytes free

C:\Documents and Settings\All Users\Documents\Shared Virtual
Networks>move ClusterNetwork.vnc.vnc x:\Guest1\ClusterNetwork.vnc
C:\Documents and Settings\All Users\Documents\Shared Virtual
Networks\ClusterNetwork.vnc
1 file(s) moved.
C:\Documents and Settings\All Users\Documents\Shared Virtual
Networks>x:

X:\>cd Guest1

X:\Guest1>dir
 Volume in drive X is DATA1
 Volume Serial Number is B853-D7BC
Directory of X:\Guest1
05/01/2006  01:13 PM    <DIR>          .
05/01/2006  01:13 PM    <DIR>          ..
03/20/2006  04:09 PM 3,242 ClusterNetwork.vnc
1 File(s)          3,242 bytes
2 Dir(s)  697,157,910,528 bytes free
```

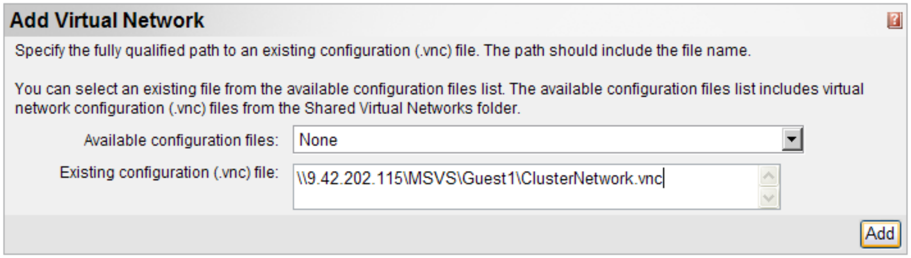
> **Note:** You must move the file using **Move**, not copy using **Copy**.

i. In the Virtual Server Administration Website, under Virtual Networks, click **Add**.

j. In the box next to Existing configuration (.vnc) file, type: `\\`*IPofMSVS*`\`*SharedFolderName*`\Guest1\ClusterNetwork.vnc`, as shown in Figure 4-18 on page 198. When complete, select **Add**.

Note: When typing paths to folders or files, note that local file paths (for example, C:\my disk files\my disk.vhd) reference the computer running the Virtual Server service. If the file or folder that you want to reference is on a computer other than the one running the Virtual Server service, you must use a Universal Naming Convention (UNC) path (for example, \\computer name\share\folder\file name).In this example we are showing how to use the UNC path.

Also, it has to point to the clustered disk volume, in this case disk X:.



Add Virtual Network

Specify the fully qualified path to an existing configuration (.vnc) file. The path should include the file name.

You can select an existing file from the available configuration files list. The available configuration files list includes virtual network configuration (.vnc) files from the Shared Virtual Networks folder.

Available configuration files: None

Existing configuration (.vnc) file: \\9.42.202.115\MSVS\Guest1\ClusterNetwork.vnc

Figure 4-18 Add network connection

k. In the navigation pane, under Virtual Machines, select **Create**.

l. In Virtual machine name, instead of simply typing the name, type the following path, which not only names the virtual machine Guest1, but places the virtual machine's configuration file on the cluster storage:

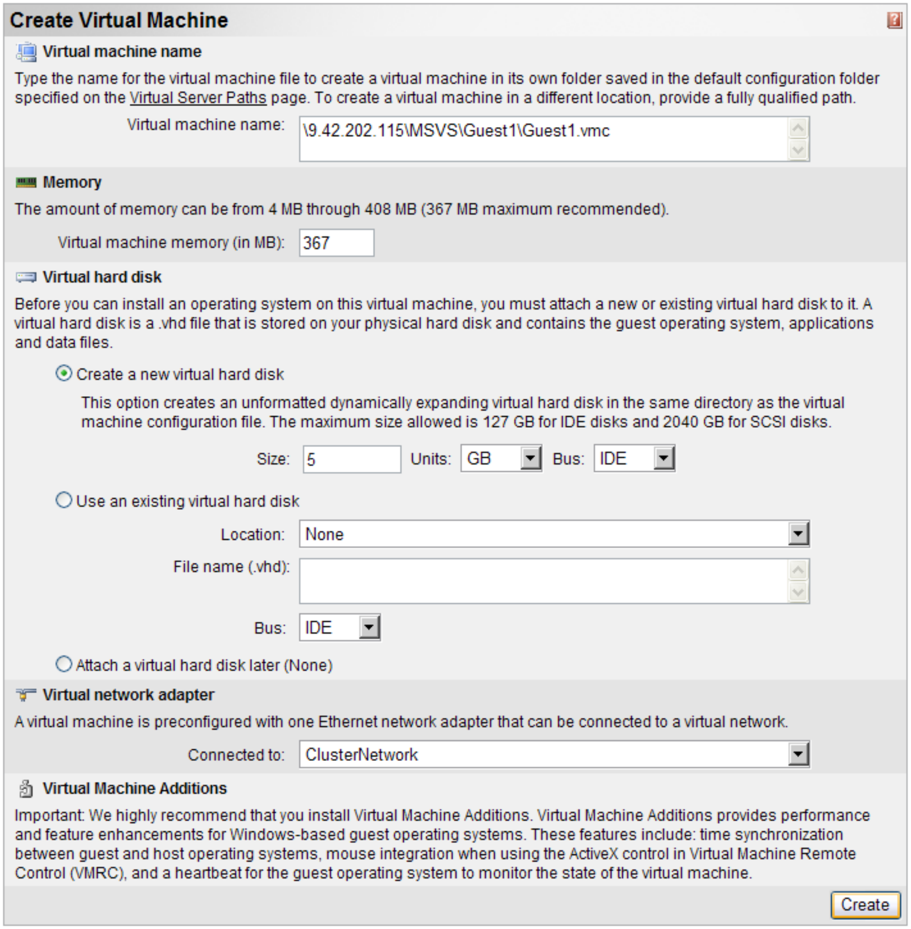
```
X:\Guest1\Guest1.vmc
```

m. In Memory, type a value in megabytes for the amount of RAM used by the virtual machine.

n. If you plan to create other virtual machines on this physical host, be sure to use only part of the physical RAM for Guest1 VM.

o. In Virtual hard disk, select **Create a new virtual hard disk**. To set the size of the virtual hard disk, specify a value in Size, and then select either MB for megabytes or GB for gigabytes. This size must be smaller than or equal to the size of disk X.

p. In Virtual network adapter, select **ClusterNetwork** and then click **Create**, as shown in the Figure 4-19 on page 199.



Create Virtual Machine

Virtual machine name

Type the name for the virtual machine file to create a virtual machine in its own folder saved in the default configuration folder specified on the Virtual Server Paths page. To create a virtual machine in a different location, provide a fully qualified path.

Virtual machine name: \9.42.202.115\MSVS\Guest1\Guest1.vmc

Memory

The amount of memory can be from 4 MB through 408 MB (367 MB maximum recommended).

Virtual machine memory (in MB): 367

Virtual hard disk

Before you can install an operating system on this virtual machine, you must attach a new or existing virtual hard disk to it. A virtual hard disk is a .vhd file that is stored on your physical hard disk and contains the guest operating system, applications and data files.

● Create a new virtual hard disk

This option creates an unformatted dynamically expanding virtual hard disk in the same directory as the virtual machine configuration file. The maximum size allowed is 127 GB for IDE disks and 2040 GB for SCSI disks.

Size: 5 Units: GB Bus: IDE

○ Use an existing virtual hard disk

Location: None

File name (.vhd):

Bus: IDE

○ Attach a virtual hard disk later (None)

Virtual network adapter

A virtual machine is preconfigured with one Ethernet network adapter that can be connected to a virtual network.

Connected to: ClusterNetwork

Virtual Machine Additions

Important: We highly recommend that you install Virtual Machine Additions. Virtual Machine Additions provides performance and feature enhancements for Windows-based guest operating systems. These features include: time synchronization between guest and host operating systems, mouse integration when using the ActiveX control in Virtual Machine Remote Control (VMRC), and a heartbeat for the guest operating system to monitor the state of the virtual machine.

Figure 4-19 Create Virtual Machine on Shared Storage

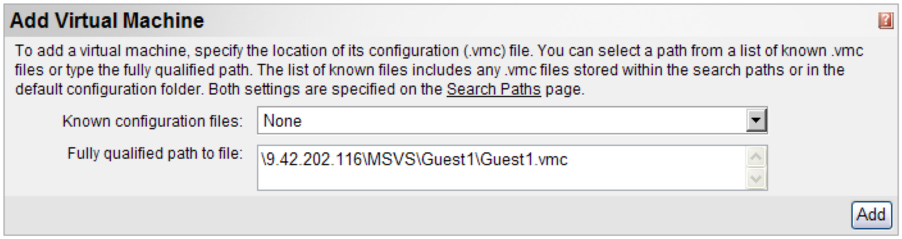
12. The last step is to configure the Guest1 VM for failover. Perform the following steps.

    a. In Cluster Administrator, move Guest1Group to the other node (not the node on which you were working in the previous procedure).

    b. For the cluster node on which Guest1Group is currently located, open the Virtual Server Administration Web site.

    c. In the navigation pane, under Virtual Machines, click **Add**.

    d. In Fully qualified path to file, type: `\\`*`IPofMSVS`*`\`*`SharedFolderName`*`\Guest1\ClusterNetwork.vnc`, as shown in Figure 4-20 on page 200.



Add Virtual Machine

To add a virtual machine, specify the location of its configuration (.vmc) file. You can select a path from a list of known .vmc files or type the fully qualified path. The list of known files includes any .vmc files stored within the search paths or in the default configuration folder. Both settings are specified on the Search Paths page.

Known configuration files: None

Fully qualified path to file: \9.42.202.116\MSVS\Guest1\Guest1.vmc

Figure 4-20 Add Virtual Machine on Second Cluster Node

e. Click **Add**.

f. On either cluster node, in Cluster Administrator, create a new script resource with the properties in the following list.

> **Note:** Do not bring this new resource online until you have completed step h.

g. Apply the following for the script, as shown in the Figure 4-21 on page 201:

i. Call it Guest1Script.

ii. Make it a Generic Script resource.

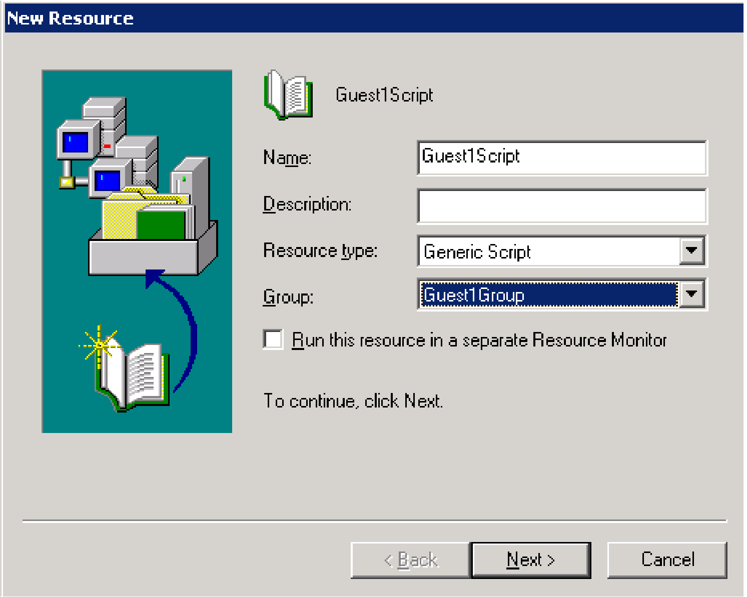
iii. Assign the resource to Guest1Group.

iv. For Possible Owners, make sure both cluster nodes are listed.

v. Add DiskResourceX to the list of resource dependencies.

vi. For the Script filepath, specify the following:

```
%windir%\Cluster\Havm.vbs
```


New Resource

Guest1Script

Name: Guest1Script

Description:

Resource type: Generic Script

Group: Guest1Group

□ Run this resource in a separate Resource Monitor

To continue, click Next.

Figure 4-21 Generic Script Resource

h. With Guest1Script in the Offline state, on the same node as in the previous step, click **Start** → **Run** and enter the following command:

```
cluster res "Guest1Script" /priv VirtualMachineName=Guest1
```

This command associates the Guest1Script resource with the guest named Guest1.

i. In Cluster Administrator, bring Guest1Group online. If you use the Virtual Server Administration Website to view the node that is the owner of Guest1Group, in Master Status, Guest1 will now have a status of Running.

13. Now it is time to install an Operating System on the Guest1 VM.

14. After the OS is installed on the Guest1 VM, you must install Virtual Machine Additions on the guest. Virtual Machine Additions is included in Virtual Server 2005 R2 and improves the integration and performance of a virtual machine running certain Windows operating systems.

    Virtual Machine Additions will also improve many aspects of your experience when using Virtual Server. For example, if Virtual Machine Additions is installed on the virtual machine, you can move the pointer freely between the virtual machine window and the host operating system when using the Virtual Machine Remote Control (VMRC) client.

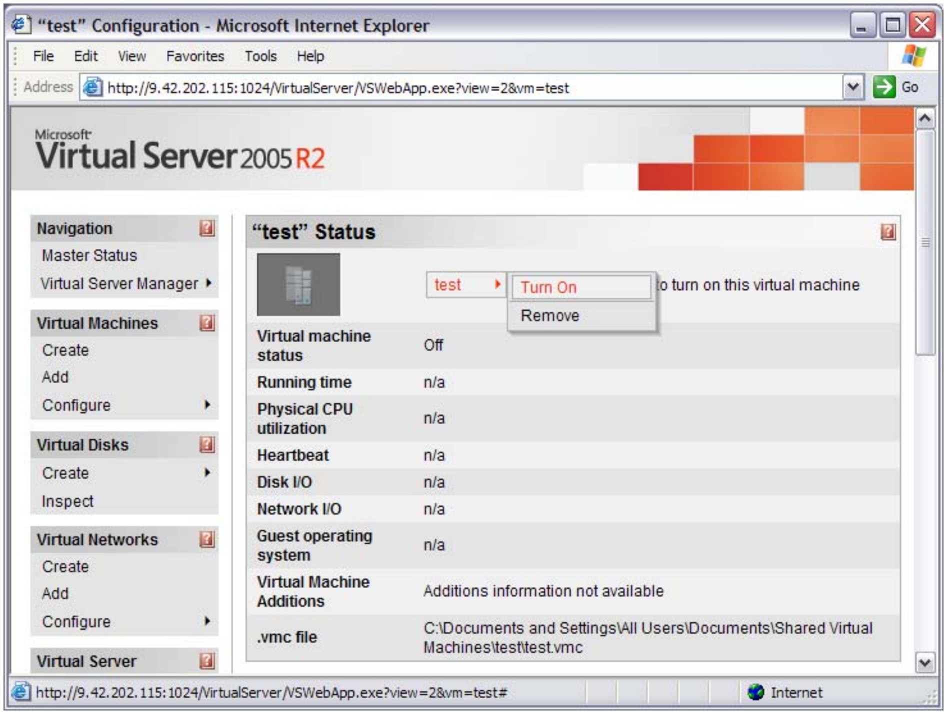
Virtual Machine Additions is installed from the guest operating system when it is running a supported operating system. It is not installed on the host operating system.

To install Virtual Machine Additions, perform the following steps:

a. Open the Administration Web site.

b. In the navigation pane, under Virtual Machines, point to Configure and then click the appropriate virtual machine.

c. In Status, point to the virtual machine name, and then click **Turn On**, as shown in Figure 4-22.



http://9.42.202.115:1024/VirtualServer/VSWebApp.exe?view=2&vm=test

"test" Status

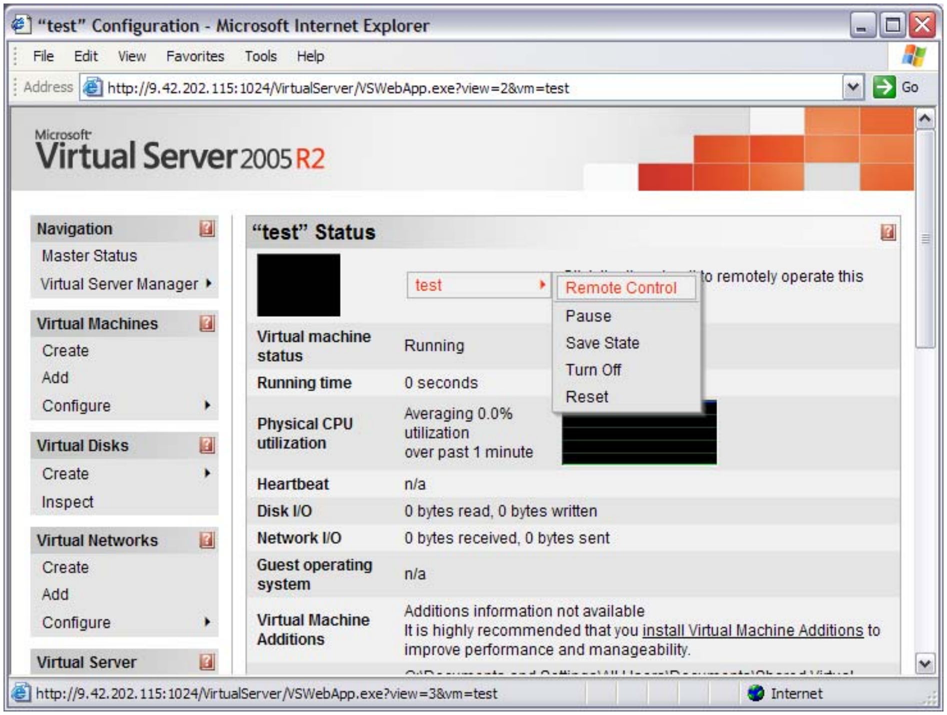
o turn on this virtual machine

Virtual machine status	Off
Running time	n/a
Physical CPU utilization	n/a
Heartbeat	n/a
Disk I/O	n/a
Network I/O	n/a
Guest operating system	n/a
Virtual Machine Additions	Additions information not available
.vmc file	C:\Documents and Settings\All Users\Documents\Shared Virtual Machines\test\test.vmc

http://9.42.202.115:1024/VirtualServer/VSWebApp.exe?view=2&vm=test#

Figure 4-22 Turn on Virtual Machine

d. Once the virtual machine has started, point to the virtual machine name, and then click Remote Control, as shown in the Figure 4-23 on page 203.


"test" Configuration - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://9.42.202.115:1024/VirtualServer/VSWebApp.exe?view=2&vm=test
Go
Microsoft Virtual Server 2005 R2
Navigation
Master Status
Virtual Server Manager
Virtual Machines
Create
Add
Configure
Virtual Disks
Create
Inspect
Virtual Networks
Create
Add
Configure
Virtual Server
test
Remote Control
Pause
Save State
Turn Off
Reset
http://9.42.202.115:1024/VirtualServer/VSWebApp.exe?view=3&vm=test
Internet

"test" Status

to remotely operate this

Virtual machine status	Running
Running time	0 seconds
Physical CPU utilization	Averaging 0.0% utilization over past 1 minute
Heartbeat	n/a
Disk I/O	0 bytes read, 0 bytes written
Network I/O	0 bytes received, 0 bytes sent
Guest operating system	n/a
Virtual Machine Additions	Additions information not available It is highly recommended that you install Virtual Machine Additions to improve performance and manageability.

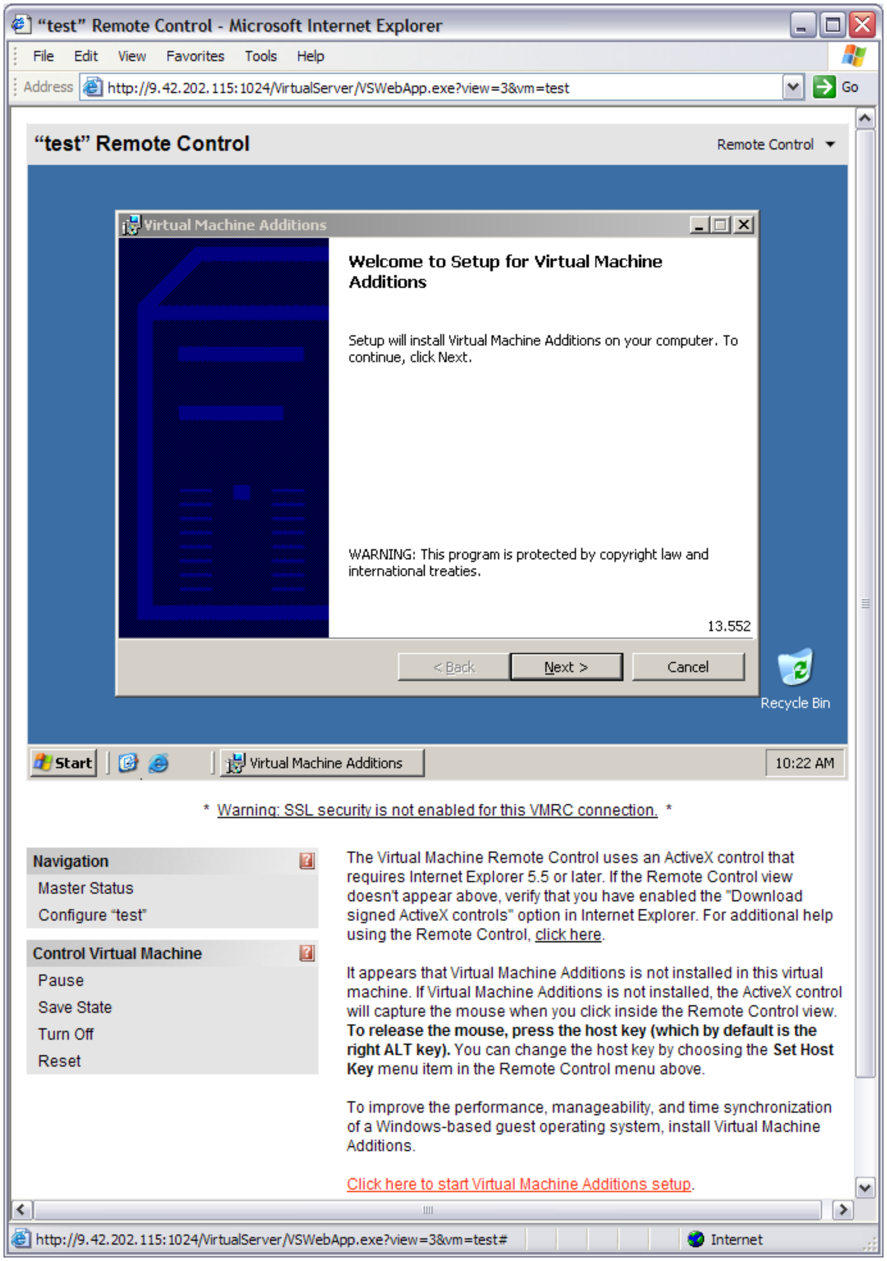
Figure 4-23 Virtual Machine Remote Control

e. Log on to the virtual machine as an administrator or member of the Administrators group.

f. When the guest operating system is loaded, press the HOST KEY to release the mouse pointer, and then in the lower-left corner under Navigation, click Configure virtual_machine_name.

g. In Configuration, click **Virtual Machine Additions**, click **Install Virtual Machine Additions**, and then click **OK**, as shown in Figure 4-24 on page 204.

h. Click in the Remote Control window to return to the guest operating system. Select **Click here to start Virtual Machine Additions Setup**, as shown in the Figure 4-24 on page 204. The Virtual Machine Additions installation wizard will start, as shown in the Figure 4-24 on page 204.


"test" Remote Control - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Address
http://9.42.202.115:1024/VirtualServer/VSWebApp.exe?view=3&vm=test
Go
"test" Remote Control
Remote Control
Virtual Machine Additions
< Back
Next >
Cancel
Recycle Bin
Start
Virtual Machine Additions
10:22 AM
Navigation
Master Status
Configure "test"
Control Virtual Machine
Pause
Save State
Turn Off
Reset
http://9.42.202.115:1024/VirtualServer/VSWebApp.exe?view=3&vm=test#
Internet

Welcome to Setup for Virtual Machine Additions

Setup will install Virtual Machine Additions on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.

13.552

* Warning: SSL security is not enabled for this VMRC connection. *

The Virtual Machine Remote Control uses an ActiveX control that requires Internet Explorer 5.5 or later. If the Remote Control view doesn't appear above, verify that you have enabled the "Download signed ActiveX controls" option in Internet Explorer. For additional help using the Remote Control, click here.

It appears that Virtual Machine Additions is not installed in this virtual machine. If Virtual Machine Additions is not installed, the ActiveX control will capture the mouse when you click inside the Remote Control view. **To release the mouse, press the host key (which by default is the right ALT key).** You can change the host key by choosing the **Set Host Key** menu item in the Remote Control menu above.

To improve the performance, manageability, and time synchronization of a Windows-based guest operating system, install Virtual Machine Additions.

Click here to start Virtual Machine Additions setup.

Figure 4-24 Virtual Machine Additions setup

i. Proceed through the wizard. When the wizard is complete, you are prompted to restart the virtual machine to complete the installation.

4.8 Adding resources to Virtual Server

As discussed in 2.5, “Multi-node configurations” on page 46, the number of processors, RAM and PCI-X slots in the x3950 can be expanded by adding additional nodes, up to a total of eight x3950 node.

When Virtual Server is restarted, it detects any new hardware and, without any further configuration, is able to use the new resources. You can then add and configure extra VMs as needed. You also might want to configure the new onboard Gigabith Ethernet adapters and any extra PCI adapters that you wish to add to virtual machines.

One of the advantages of implementing Microsoft Virtual Server on x3950 configurations is lowering the number of physical servers to administer. In this section we discuss those scenarios for which it is possible to maintain a stable IT infrastructure while increasing the *computational power* of the overall installation.

Our example is an existing configuration of 150 virtual machines on six installations of Microsoft Virtual Server running on x3950 servers. Figure 4-25 illustrates this.

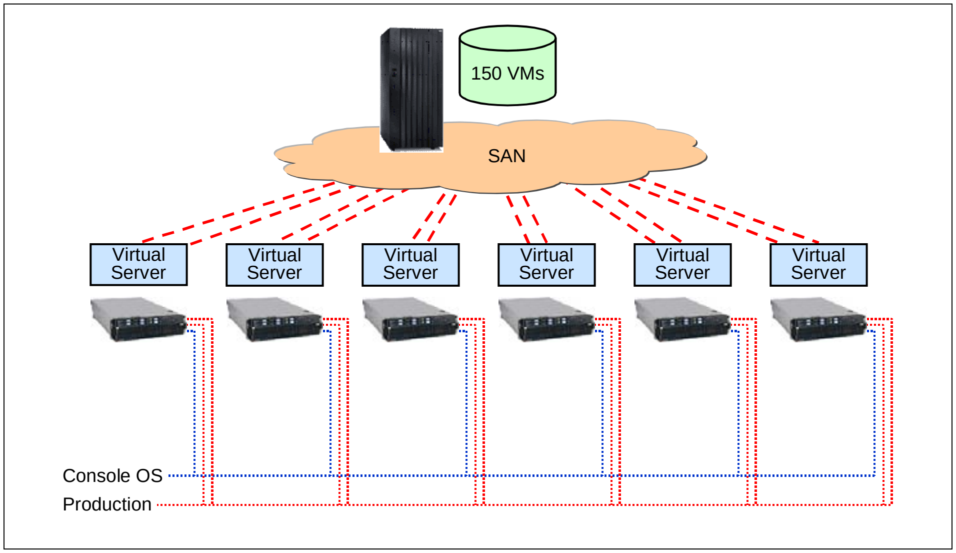


Figure 4-25 Initial deployment

As business objectives require new systems, virtual machines are created to support the new services. because the current physical servers are already being used at 75-85% of their capabilities, there are two options:

- Buy new servers, install Virtual Server on them, and start deploying new virtual machines on those new servers. This is what is usually referred to as a *scale-out* approach.
- With the x3950 though, we can approach this requirement in a different manner. Instead of adding new servers to the current infrastructure, you could upgrade the existing x3950 to support more virtual machines, a *scale-up* approach. This is how we will proceed.

By adding x3950 (or x3950 E) nodes to the existing x3950 servers, as shown in Figure 4-26, we are able to support twice the number of virtual machines without changing the architecture of their farms and or clusters.

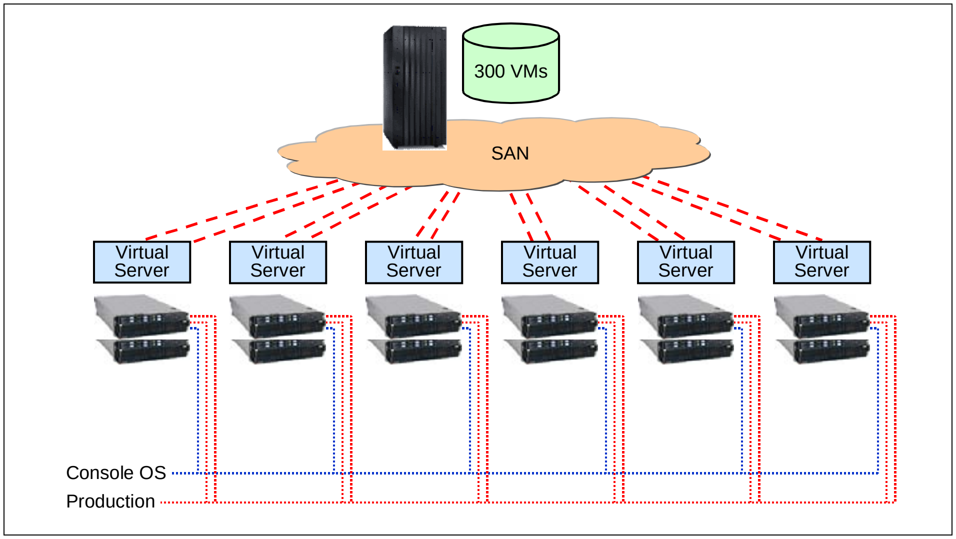


Figure 4-26 Adding nodes to the existing x3950 servers doubles the supported VMs

The configuration can be extended further by simply adding more nodes to the existing servers, upgrading some systems from 8-way (two nodes) to 16-way (four nodes), as shown in Figure 4-27 on page 207.

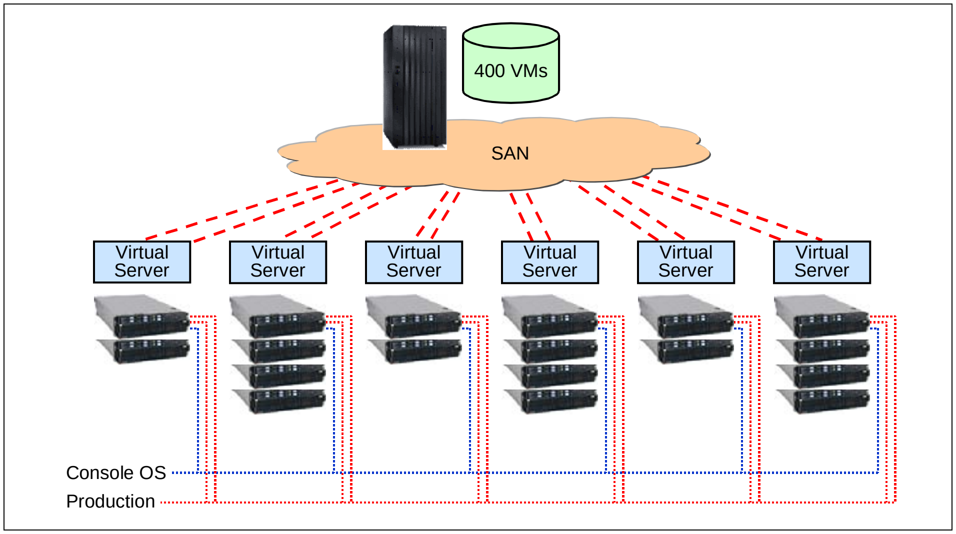


Figure 4-27 Upgrading some Virtual Server systems to 16-way (four nodes)

You might have noticed in the above example, we have not considered upgrading the Ethernet and Fibre Channel connections. Depending on the pattern of the workloads being deployed in the virtual machines, you should consider adding physical Ethernet connections to the virtual switches as well as Fibre Channel adapters to increase the I/O throughput.

Table 4-1 shows examples of situations with which many systems administrators might find themselves involved. Here, we assume that the current configuration is based on single-node, 4-way x3950 servers.

Table 4-1 Common situations and suggested solutions for virtual servers

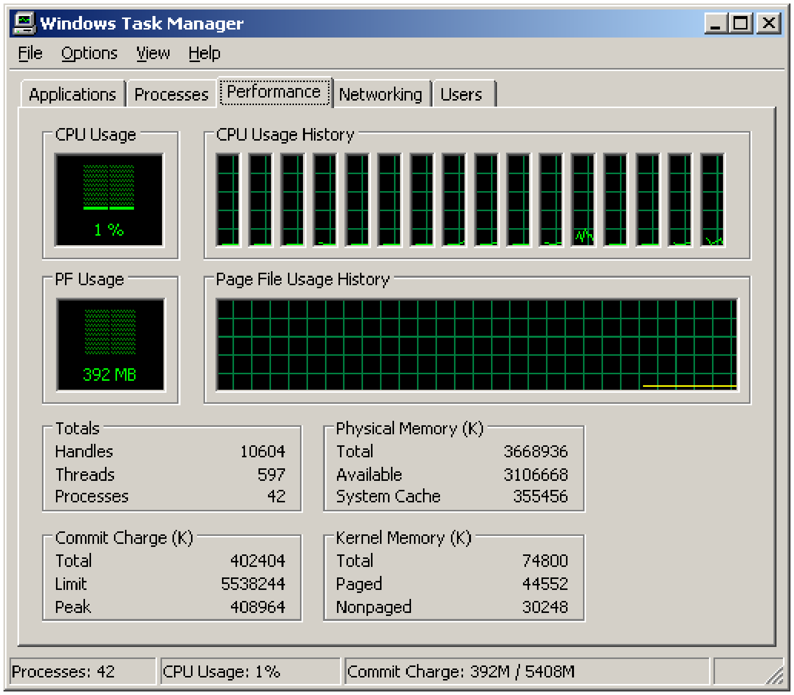
Situation	Suggested Solution
I have 24 virtual machines running on six 4-way x3950 systems. I need to grow my infrastructure adding new virtual machines but I am short on CPU capacity and RAM across my farms and clusters. Network and storage throughput are not a problem at the moment. What do I do?	Update your six servers by adding an extra x3950 (or x3950 E) node to each. This will provide you the ability to host more virtual machines without changing or adjusting anything in the way you operate today your infrastructure.

Situation	Suggested Solution
I have 24 virtual machines running on six 4-way x3950 systems. I need to grow my infrastructure adding new virtual machines but I am short on CPU capacity, RAM and I/O throughput across my farms and clusters. What do I do?	Update your six servers by adding an extra x3950 (or x3950 E) node to each. In each new node connect the onboard Gigabit Ethernet controllers and install and connect a Fibre Channel adapter. This will provide you the ability to host more virtual machines
I have 24 virtual machines running on six 4-way x3950 systems. I need to grow my infrastructure adding new virtual machines but I am short on I/O throughput across my farms and clusters. CPU and RAM are not a problem at the moment. What do I do?	Add new I/O cards to your existing 4-way systems.
I have 24 virtual machines running on six 4-way x3950 systems. I need to increase the high-availability and prevent from systems failing.	Adding a number of 4-way x3950 systems and cluster all of the Virtual Servers installations using Windows Server clustering.

4.8.1 Scale-up step-by-step

The previous section describes the concepts of scale-up with Virtual Server on the x3950. In this section, we describe how such upgrades are implemented. We use as our example, an upgrade of a 4-way (single node) x3950 running Microsoft Virtual Server to an 8-way (two node) x3950.

Figure 4-28 on page 209 shows the CPU and memory of the existing 4-way configuration.



Totals	
Handles	10604
Threads	597
Processes	42

Physical Memory (K)	
Total	3668936
Available	3106668
System Cache	355456

Commit Charge (K)	
Total	402404
Limit	5538244
Peak	408964

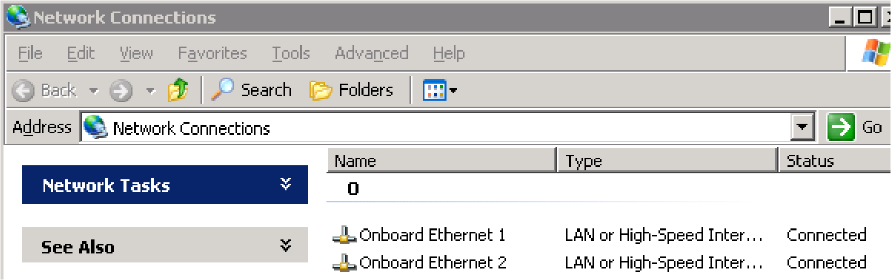
Kernel Memory (K)	
Total	74800
Paged	44552
Nonpaged	30248

Processes: 42 | CPU Usage: 1% | Commit Charge: 392M / 5408M

16 CPUs

Figure 4-28 x3950 4-way current dual-core CPU and memory count and Hyper-Threading enabled

Figure 4-29 shows the current networking configuration.


Network Connections
File Edit View Favorites Tools Advanced Help
Back Search Folders
Address Network Connections Go
Network Tasks
See Also

Name	Type	Status
0		
Onboard Ethernet 1	LAN or High-Speed Inter...	Connected
Onboard Ethernet 2	LAN or High-Speed Inter...	Connected

Figure 4-29 x3950 4-way current networking configuration

Note that for the purpose of the test we are only using the two integrated Broadcom network adapters.

The procedure to upgrade our configuration to 8-way is:

1. Shutdown Virtual Server, Windows, and the x3950. You must power off the x3950 before connecting additional nodes.

2. Connect the new x3950 or x3950 E to the existing x3950 using the scalability cables and ensure all BIOS and firmware levels are matching and up-to-date. See 2.5, “Multi-node configurations” on page 46 for details.
3. Configure the two-node partition using the Web interface of the service processor in the first x3950 as described in 2.6, “Scalable system setup” on page 54.
4. Power on the first x3950 node, which should power on automatically the second node as well. The system should boot to ESX Server from the local drives in the first node, just as it did as a single-node configuration.

   During boot, Windows Server 2003 detects the new resources: new CPUs, memory, PCI-X slots, and I/O devices such as the Ethernet controllers in the second node.

   After the system has rebooted and the new devices has been recognized and installed, the configuration is shown in the Figure 4-30.

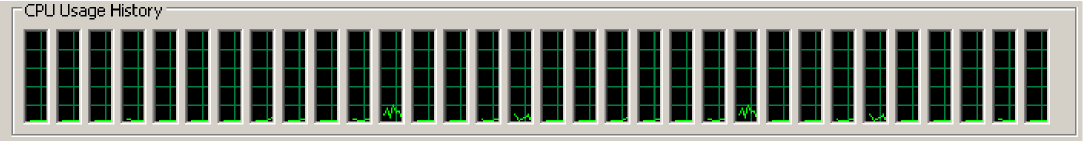
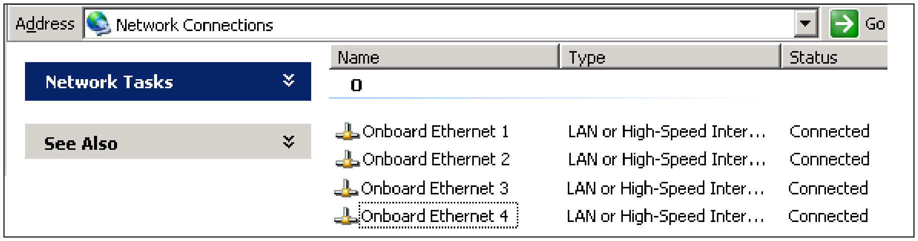


Figure 4-30 x3950 8-way new CPU count (32 processors)

The networking reconfiguration is similar. The two new integrated Gigabit Ethernet controllers in node 2 were detected (Figure 4-31).



Name	Type	Status
0		
Onboard Ethernet 1	LAN or High-Speed Inter...	Connected
Onboard Ethernet 2	LAN or High-Speed Inter...	Connected
Onboard Ethernet 3	LAN or High-Speed Inter...	Connected
Onboard Ethernet 4	LAN or High-Speed Inter...	Connected

Figure 4-31 x3950 8-way new networking configuration.

There is nothing that needs to be configured within MSVS to recognize the new hardware. All of the hardware changes are managed by the host Operating System.

5

Management with IBM Director

This chapter describes how to install the RSA II driver, IBM Director Agent and the VMM software, and how to use VMM to perform management tasks on ESX Server systems and Microsoft Virtual Server systems.

Topics in this chapter are as follows:

5.1 Introduction to IBM Director

IBM Director is an industry-leading hardware management solution that provides an integrated suite of software tools for a consistent, single point of management and automation.

IBM Director supports industry standards which enable heterogeneous hardware management with broad platform and operating system support, including Intel and POWER systems that support Windows, Linux, NetWare, VMware ESX Server, AIX 5L™ and i5/OS® all from a single, Java-based user interface. IBM Director also provides management of both VMware ESX Server and Microsoft Virtual Server using the Virtual Machine Manager extension.

IBM Director is designed to manage a complex environment that contains numerous servers, desktop computers, workstations, notebook computers, storage subsystems and various types of SNMP-based devices. Figure 5-1 on page 213 shows a simple diagram of the major components you might find in an IBM Director managed environment, as well as the IBM Director software components that would be installed (if any) on each type of hardware.

The hardware in an IBM Director environment can be divided into the following groups:

- Management servers: one or more servers on which IBM Director Server is installed
- Managed systems: servers (physical systems and virtual machines), workstations, desktop computers, and notebook computers that are managed by IBM Director
- Management consoles: servers, workstations, desktop computers, and notebook computers from which you communicate with one or more IBM Director Servers.
- SNMP devices: network devices, printers, or computers that have SNMP agents installed or embedded

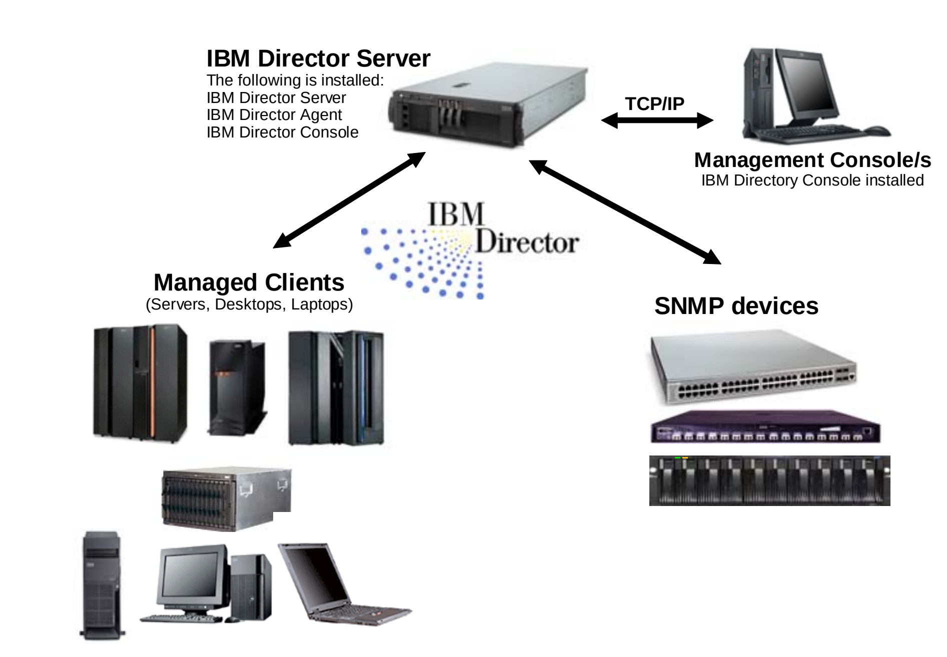


Figure 5-1 Typical IBM Director management environment

Each system in your IBM Director environment will have one or more of these components installed:

- IBM Director Server is installed on the system that is to become the management server. Ideally, this is a single system in the environment, but this is not always possible.
- IBM Director Agent is installed on each managed system or virtual machine (including the Console OS
- IBM Director Console is installed on any system from which a system administrator will remotely access the management server (called a *management console*).

In the following sections, we provide a brief description of each of these components.

5.1.1 IBM Director Server

IBM Director Server is the main component of IBM Director and contains the management data, the server engine, and the application logic. IBM Director Server provides basic functions such as discovery of the managed systems, persistent storage of inventory data, SQL database support, presence checking, security and authentication, management console support, and administrative tasks.

In the default installation under Windows, Linux, and AIX, IBM Director Server stores management information in an embedded Apache Derby database. You can access information that is stored in this integrated, centralized, relational database even when the managed systems are not available. For large-scale IBM Director solutions, you can use a stand-alone database application, such as IBM DB2® Universal Database™, Oracle, or Microsoft SQL Server.

IBM Director Server can be installed on the following operating systems:

- Microsoft Windows 2000 Server and Advanced Server with Service Pack 3 or later
- Microsoft Windows Server 2003 Standard, Enterprise and Web Editions, 32-bit and 64-bit
- Red Hat Enterprise Linux 3.0, AS and ES, 32-bit and 64-bit
- Red Hat Enterprise Linux 4.0, AS and ES, 32-bit and 64-bit
- SUSE Linux Enterprise Server 8.0, 32-bit only
- SUSE Linux Enterprise Server 9.0, 32-bit and 64-bit
- AIX v5.2
- AIX v5.3
- i5/OS V5R3

Note: IBM Director Server is not supported running in a virtual machine (VMware products or Microsoft Virtual Server), nor in the ESX Server console.

See the *IBM Director 5.10 Hardware and Software Compatibility Guide* for a complete list of supported operating systems by platform, available from:

http://www.pc.ibm.com/support?page=MIGR-61788

5.1.2 IBM Director Agent

IBM Director Agent provides management data to the management server through various network protocols. When installed on systems in your network, IBM Director Agent permits the management server to communicate with these managed systems in your network.

IBM Director Agent is supported in the following virtualized environments:

- In the Console OS or a supported guest operating system in:
  - VMware ESX Server 2.1
  - VMware ESX Server 2.5
  - VMware ESX Server 2.51
  - VMware ESX Server 2.52 (IBM Director 5.10.2)
  - VMware ESX Server 3.0 (planned for support with IBM Director 5.10.3)

  Supported guest operating systems are those that are supported by both IBM Director 5.10.1 as listed in the *IBM Director 5.10 Hardware and Software Compatibility Guide* and by the specific ESX Server version.
- Microsoft Virtual Server 2005, with the following guest operating systems:
  - Windows 2000 Server and Advanced Server (SP3 or SP4)
  - Windows Server 2003, Enterprise, Standard, and Web Editions
- Microsoft Virtual Server 2005 (Service Pack 1), with the following guest operating systems:
  - Windows 2000 Server and Advanced Server (SP3 or SP4)
  - Windows Server 2003, Enterprise, Standard, and Web Editions
  - Windows Server 2003, Enterprise, Standard, and Web Editions, x64
  - Windows XP Professional Edition (SP2 required)
  - Windows XP Professional x64 Edition
- Microsoft Virtual Server 2005 R2 is also planned to be supported with guest operating systems.

A full list of the supported operating systems and virtualization products can be found in the *IBM Director 5.10 Hardware and Software Compatibility Guide*:

http://www.pc.ibm.com/support?page=SERV-DIRECT

5.1.3 IBM Director Console

IBM Director Console is the graphical user interface (GUI) for IBM Director Server. Using IBM Director Console, system administrators can conduct comprehensive hardware management using either a drag-and-drop action or a single click. The Console has undergone significant improvements in the latest release of IBM Director.

When you install IBM Director Console on a system, IBM Director Agent is not installed automatically. If you want to manage the system on which you have installed IBM Director Console, you must also install IBM Director Agent on that system.

IBM Director Console can be installed on the following operating systems:

- Windows XP Professional, 32-bit only
- Windows 2000 Professional, Server, Advanced Server, Datacenter Server with Service Pack 3 or later
- Windows Server 2003 Standard, Enterprise and Web Editions, 32-bit only
- Red Hat Enterprise Linux 3.0 and 4.0, AS and ES, 32-bit and 64-bit
- SUSE LINUX Enterprise Server 8.0, 32-bit only
- SUSE LINUX Enterprise Server 9.0, 32-bit and 64-bit
- AIX v5.2 and v5.3

You can install IBM Director Console on as many systems as needed. The license is available at no charge.

5.1.4 IBM Director Extensions

Several plug-in modules to IBM Director, which are collectively called IBM Director Extensions, are available from IBM and third parties. Some of these tools are fee-based and require a license, such as Capacity Manager and Application Workload Manager. However, many IBM Director Extensions can be obtained by downloading them from the Web without charge.

IBM Director extensions includes Virtual Machine Manager, which we describe in 5.2, "Virtual Machine Manager" on page 216.

For a complete discussion of other plug-ins, see the redbook *IBM Director 5.10*, SG24-6188.

5.2 Virtual Machine Manager

IBM Virtual Machine Manager (VMM) is an extension to IBM Director that allows you to manage both physical and virtual machines from a single console. With VMM, you can manage both ESX Server and Virtual Server environments using IBM Director. VMM also integrates VMware VirtualCenter and IBM Director for advanced virtual machine management.

The VMM agent uses the standard APIs provided by the VMware and Microsoft products. With VMM installed, you can perform the following tasks from IBM Director Console:

- Correlate relationships between physical platforms and virtual components.

- Report status of physical platforms and their corresponding virtual components.
- Log in to the management interface of the virtualization application.
- Discover virtual components.
- Perform power operations on virtual machines.
- Create event action plans that involve virtual components.

With VMM, IBM Director can recognize systems that contain virtual components and create the following managed objects:

- Coordinators (VirtualCenter)
- Farm (VirtualCenter farms)
- Hosts (ESX Server, GSX Server, Virtual Server)
- Guest operating systems

In addition, VMM can be used for the creation of an event action plan to initiate the transfer of virtual machines. More information about VMM can be found at:

http://www.ibm.com/servers/eserver/xseries/systems_management/ibm_director/extensions/vmm.html

5.2.1 What is new in IBM Virtual Machine Manager 2.1?

IBM Virtual Machine Manager 2.1 now supports IBM Director 5.10 and offers the following new features for VMware ESX Server users:

- Planned support both VMware ESX Server 3.0 and VirtualCenter 2.0

> **Note:** We were successfully able to use VMM 2.1 with ESX Server 3, however at the time of writing, the level of support for ESX Server 3 was still being determined (since ESX Server 3 had not been released).

- Support modifying the Migration Enabled attribute in VirtualCenter for ESX Server hosts
- Support modifying undo disk action while VM is powered ON

For Microsoft Virtual Server users, VMM 2.1 has these new features:

- Support Microsoft Virtual Server 2005 R2
- Allow virtual machines to be created using 3.6 GB of RAM and 12 GB of disk

5.2.2 Where VMM is installed

Like IBM Director, VMM has three components that are installed on systems running:

- The VMM server extension installs on the system running IBM Director Server. Only Windows operating systems are supported. The VMM console is also installed automatically.
- The VMM console extension is installed on all systems running IBM Director Console. Only Windows operating systems are supported.
- The VMM agent extension is installed on systems running IBM Director Agent and the following virtualization product:
  - VMware VirtualCenter (Windows systems only)
  - VMware ESX Server console OS
  - VMware GSX Server
  - Microsoft Virtual Server

  If you are managing an ESX Server system using VirtualCenter, then you do not install the VMM agent on that system. Instead, VirtualCenter will perform all management tasks by communicating directly with the ESX Server console OS.

> **Note:** VMM only runs on Windows systems, or the ESX Server Console OS. Linux systems are not supported.

VMM 2.1 can be downloaded from:

`http://www.pc.ibm.com/support?page=MIGR-56914`

There are two components available for download:

- Windows software for the server and console extensions, as well as the agent extension for VirtualCenter, GSX Server and Microsoft Virtual Server
- RPM package for installing the agent extension in the console OS of ESX Server

Installation instructions can be found in the *IBM Virtual Machine Manager Installation and User's Guide*, available from:

`http://www.pc.ibm.com/support?page=MIGR-56955`

As shown in the Figure 5-2 on page 219, VMM can manage ESX Server systems directly or through VirtualCenter. It also directly manage systems running GSX Server and Microsoft Virtual Servers.

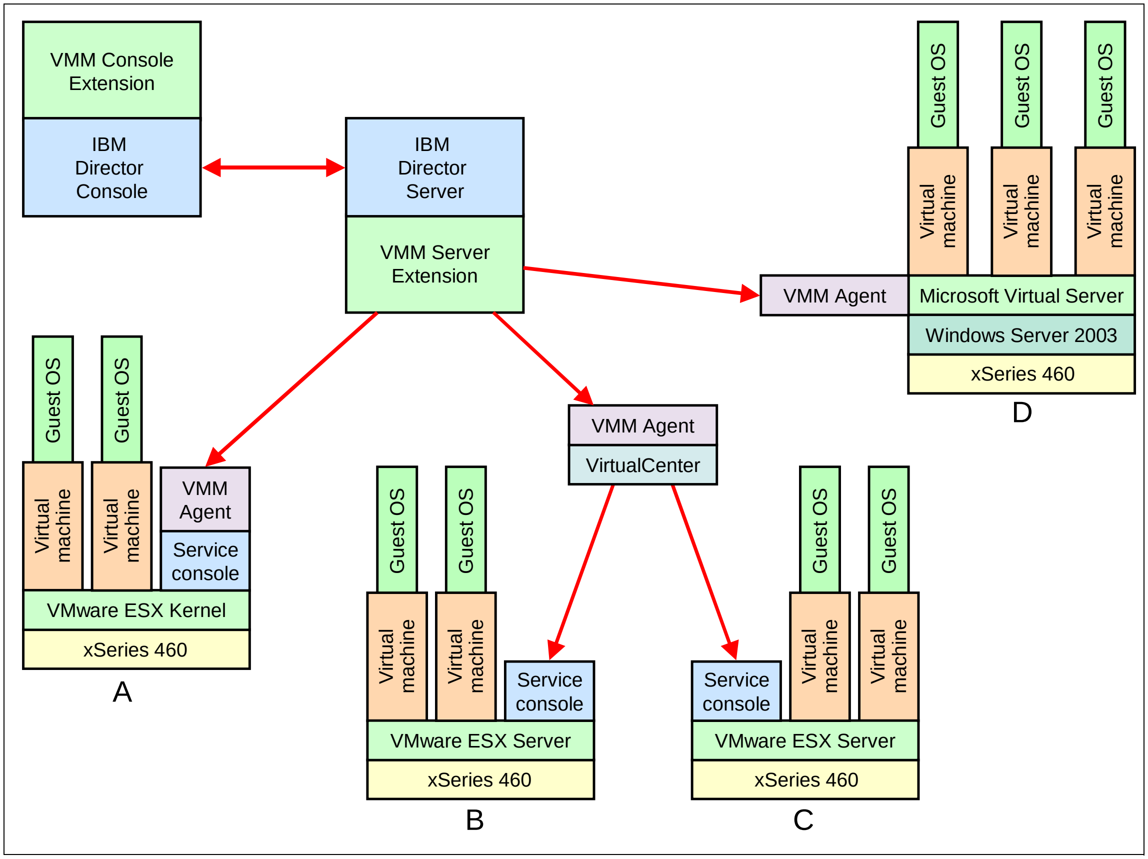


Figure 5-2 IBM Virtual Machine Manager architecture

In Figure 5-2 we have four x3950 complexes running virtualization software:

- System A is running ESX Server but it is not managed by VirtualCenter. This system has the VMM agent extension installed in the service console.
- Systems B and C are running ESX Server and are managed by another Windows system running VirtualCenter. The VirtualCenter system has the VMM agent installed, but the ESX Server systems do not have the VMM agent installed.
- System D has Microsoft Virtual Server and the VMM agent installed.

Important: If you are using VirtualCenter to manage your ESX Server installations, you install the VMM agent only on the VirtualCenter system. In this configuration, you do not install the VMM agent on the ESX Server systems.

5.3 Installation steps for ESX Server

Follow these steps to install IBM Director and VMM on your ESX Server systems:

1. Install the RSA II daemon in the Console OS.
2. Install IBM Director Agent in the Console OS.
3. Configure the ESX Server 3.0 firewall.

With these components installed, you will be able to receive hardware alerts from your x3950 and have those sent to the IBM Director management server for processing.

5.3.1 Installing the RSA II daemon

Because the x3950 has an RSA II service processor installed, you should install the RSA II driver for it.

ESX Server 3.0 now supports the use of the RSA II daemon for in-band alerting for hardware events to IBM Director Agent.

At the time of writing, however, there was no compiled RSA II daemon for use in ESX Server 3.0. This section describes how to compile the daemon and install it. You must install the daemon before installing IBM Director Agent and the VMM agent on the ESX Server console OS.

Note: Compiling the daemon requires files from a Red Hat distribution.

To install the RSAII daemon, follow these steps:

1. Log into the ESX Server console OS as root.
2. Copy the rpm files required to compile the driver from the Red Hat Enterprise 3.0 CDs to a temporary directory. Table 5-1 lists the files.

Table 5-1 RPMs required from the Red Hat Enterprise CDs

RHEL 3 CD #	Directory on CD	Filename
RHEL 3 CD #2	\RedHat\RPM	libusb-0.1.6-3.i386.rpm
RHEL 3 CD #2	\RedHat\RPM	rpm-4.2.3-10.i386.rpm
RHEL 3 CD #2	\RedHat\RPM	rpm-libs-4.2.3-10.i386.rpm
RHEL 3 CD #2	\RedHat\RPM	rpm-python-4.2.3-10.i386.rpm
RHEL 3 CD #2	\RedHat\RPM	rpmdb-redhat-3-0.20040902.i386.rpm

RHEL 3 CD #	Directory on CD	Filename
RHEL3 CD #3	\RedHat\RPM	libusb-devel-0.1.6-3.i386.rpm
RHEL 3 CD #3	\RedHat\RPM	rpm-build-4.2.3-10.i386.rpm
RHEL 3 CD #3	\RedHat\RPM	rpm-devel-4.2.3-10.i386.rpm

3. Login to the ESX Server service console and change to the temp directory where all the RPMs have been copied to and install the RPMs using the following command:

   ```
   rpm -ivh *.rpm
   ```

   The output is shown in Example 5-1.

   Example 5-1 Install RPMs on ESX console

   ```
   [root@watts2 ibm]# ls
   libusb-0.1.6-3.i386.rpm        rpm-devel-4.2.3-10.i386.rpm
   libusb-devel-0.1.6-3.i386.rpm  rpm-libs-4.2.3-10.i386.rpm
   rpm-4.2.3-10.i386.rpm          rpm-python-4.2.3-10.i386.rpm
   rpm-build-4.2.3-10.i386.rpm    rpmdb-redhat-3-0.20040902.i386.rpm
   [root@watts2 ibm]#
   [root@watts2 ibm]# rpm -ivh *.rpm
   warning: libusb-0.1.6-3.i386.rpm: V3 DSA signature: NOKEY, key ID
   db42a60e
   Preparing...                ################################### [100%]
      1:libusb                 #################################### [ 13%]
      2:libusb-devel           #################################### [ 25%]
      3:rpmdb-redhat           #################################### [ 38%]
      4:rpm-libs               #################################### [ 50%]
      5:rpm                    #################################### [ 63%]
      6:rpm-build              #################################### [ 75%]
      7:rpm-devel              #################################### [ 88%]
      8:rpm-python             #################################### [100%]
   [root@watts2 ibm]#
   ```

4. Download and copy the source RPM for the latest USB daemon for Linux to the ESX console, the latest code is available from:

   http://www.pc.ibm.com/support?page=MIGR-59454

   At the time of writing, the downloaded file is named `ibmusbasm-1.18-2.src.rpm`.

5. Login to the ESX Server service console and enter the directory onto which re you copied the drivers. Use the following command to compile and build the driver:

   ```
   rpmbuild --rebuild ibmusbasm-1.18-2.src.rpm
   ```

   The output is shown in Example 5-2.

Example 5-2 Build RSAII driver using rpmbuild

```
[root@watts2 tmp]# rpmbuild --rebuild ibmusbasm-1.18-2.src.rpm
Installing ibmusbasm-1.18-2.src.rpm
Executing(%prep): /bin/sh -e /var/tmp/rpm-tmp.13782
+ umask 022
+ cd /usr/src/redhat/BUILD
+ cd /usr/src/redhat/BUILD
+ rm -rf ibmusbasm-src
+ /usr/bin/gzip -dc /usr/src/redhat/SOURCES/ibmusbasm-src.tgz
+ tar -xvvf -
drwxr-xr-x root/root         0 2005-01-14 06:20:47 ibmusbasm-src/
-r-xr-xr-x root/root       839 2005-01-14 06:20:47 ibmusbasm-src/ibmasm.initscript
-r-xr-xr-x root/root       219 2005-01-14 06:20:47 ibmusbasm-src/ibmspdown
-r-xr-xr-x root/root       283 2005-01-14 06:20:47 ibmusbasm-src/ibmspup
-r-xr-xr-x root/root      3247 2005-01-14 06:20:47 ibmusbasm-src/ibmusbasm.spec
-r-xr-xr-x root/root      5303 2005-01-14 06:20:47 ibmusbasm-src/setup
drw-r--r-- root/root         0 2005-01-14 06:20:47 ibmusbasm-src/shlib/
-rw-r--r-- root/root     60682 2005-01-14 06:20:47 ibmusbasm-src/shlib/uwiapi.c
drw-r--r-- root/root         0 2005-01-14 06:20:47 ibmusbasm-src/src/
-rw-r--r-- root/root    106176 2005-01-14 06:20:47 ibmusbasm-src/src/ibmasm.c
-rw-r--r-- root/root     14369 2005-01-14 06:20:47 ibmusbasm-src/src/ibmasm.h
-rw-r--r-- root/root       312 2005-01-14 06:20:47 ibmusbasm-src/src/Makefile
drw-r--r-- root/root         0 2005-01-14 06:20:47 ibmusbasm-src/exe/
-rw-r--r-- root/root      8974 2005-01-14 06:20:47 ibmusbasm-src/README.TXT
+ STATUS=0
+ '[' 0 -ne 0 ']'
+ cd ibmusbasm-src
++ /usr/bin/id -u
+ '[' 0 = 0 ']'
+ /bin/chown -Rhf root .
++ /usr/bin/id -u
+ '[' 0 = 0 ']'
+ /bin/chgrp -Rhf root .
+ /bin/chmod -Rf a+rX,g-w,o-w .
+ cp /usr/src/redhat/SOURCES/README.TXT .
+ exit 0
Executing(%build): /bin/sh -e /var/tmp/rpm-tmp.13782
+ umask 022
```

```
+ cd /usr/src/redhat/BUILD
+ cd ibmusbasm-src
+ cd shlib
+ gcc -D__IBMLINUX__ -fPIC -shared -I ../src -o libsysSp.so.1 uwiapi.c
+ cd ../exe
+ gcc -I ../src -o ibmasm ../src/ibmasm.c -ldl
+ exit 0
Executing(%install): /bin/sh -e /var/tmp/rpm-tmp.49295
+ umask 022
+ cd /usr/src/redhat/BUILD
+ cd ibmusbasm-src
+ install -m 755 shlib/libsysSp.so.1 /lib
+ install -m 700 exe/ibmasm /sbin
+ install -m 700 ibmspup ibmspdown /sbin
+ install -m 700 /usr/src/redhat/SOURCES/ibmasm.initscript /etc/init.d/ibmasm
+ /usr/lib/rpm/brp-compress
+ /usr/lib/rpm/brp-strip
+ /usr/lib/rpm/brp-strip-static-archive
+ /usr/lib/rpm/brp-strip-comment-note
Processing files: ibmusbasm-1.18-2
Executing(%doc): /bin/sh -e /var/tmp/rpm-tmp.49295
+ umask 022
+ cd /usr/src/redhat/BUILD
+ cd ibmusbasm-src
+ DOCDIR=/usr/share/doc/ibmusbasm-1.18
+ export DOCDIR
+ rm -rf /usr/share/doc/ibmusbasm-1.18
+ /bin/mkdir -p /usr/share/doc/ibmusbasm-1.18
+ cp -pr README.TXT /usr/share/doc/ibmusbasm-1.18
+ exit 0
Provides: config(ibmusbasm) = 1.18-2 libsysSp.so.1
Requires(interp): /bin/sh /bin/sh /bin/sh
Requires(rpmlib): rpmlib(CompressedFileNames) <= 3.0.4-1 rpmlib(PayloadFilesHave
Prefix) <= 4.0-1
Requires(post): /bin/sh
Requires(preun): /bin/sh
Requires(postun): /bin/sh
Requires: /bin/sh config(ibmusbasm) = 1.18-2 libc.so.6 libc.so.6(GLIBC_2.0) libc
.so.6(GLIBC_2.1) libc.so.6(GLIBC_2.1.3) libc.so.6(GLIBC_2.2) libdl.so.2 libdl.so
.2(GLIBC_2.0) libdl.so.2(GLIBC_2.1) libusb
Conflicts: ibmasm ibmasr ibmasm-src-redhat ibmasm-src-suse
Wrote: /usr/src/redhat/RPMS/i386/ibmusbasm-1.18-2.i386.rpm
Executing(--clean): /bin/sh -e /var/tmp/rpm-tmp.49295
+ umask 022
+ cd /usr/src/redhat/BUILD
+ rm -rf ibmusbasm-src
+ exit 0
[root@watts2 tmp]#
```

6. Install the RSAII daemon using the rpm command from the /usr/src/redhat/RPMS/i386 directory, as shown in the Example 5-3.

   `rpm -ivh ibmusbasm-1.18-2.i386.rpm`

Example 5-3 Install the RSAII daemon using rpm

```
[root@watts2 i386]# pwd
/usr/src/redhat/RPMS/i386
[root@watts2 i386]# rpm -ivh ibmusbasm-1.18-2.i386.rpm
Preparing...                ########################################### [100%]
   1:ibmusbasm              ########################################### [100%]
Found IBM Remote Supervisor Adaptor II.
Installing the USB Service Processor driver.
```

5.3.2 Installing IBM Director Agent on ESX Server

IBM Director Agent can be installed in the console OS of ESX Server. Install IBM Director Agent after you have installed the RSA II driver/daemon as described in 5.3.1, “Installing the RSA II daemon” on page 220.

Tip: If you also want to manage the operating system and applications installed in a virtual machines, you will also need to install IBM Director Agent there as well. See the *IBM Director Installation and Configuration Guide* for installation procedures.

IBM Director Agent can be downloaded from:

`http://www.pc.ibm.com/support?page=SERV-DIRECT`

To install IBM Director Agent in the Console OS of ESX Server, follow these steps:

1. Download and the Linux version of IBM Director Agent for Linux to the ESX console, available from:

   `http://www.pc.ibm.com/support?page=SERV-DIRECT`

2. Unpack the tar file and install the agent using the **`dir5.10_agent_linux.sh`** script, as shown in Example 5-4.

Example 5-4 Installing IBM Director Agent on ESX console

```
[root@watts2 usb]# mkdir /agent
```

```
[root@watts2 usb]# cp dir5.10_agent_linux.tar /agent
[root@watts2 usb]# cd /agent/
[root@watts2 agent]# ls
dir5.10_agent_linux.tar
[root@watts2 agent]#
[root@watts2 agent]# tar -xf dir5.10_agent_linux.tar
[root@watts2 agent]# ls
FILES  META-INF  dir5.10_agent_linux.tar
[root@watts2 agent]# cd FILES
[root@watts2 FILES]# ls
dir5.10_agent_linux.sh  diragent.rsp
[root@watts2 agent]#
[root@watts2 FILES]# ./dir5.10_agent_linux.sh

./dir5.10_agent_linux.sh self-extracting installation program... Please
wait..

Attempting to install ITDAgent-5.10-1.i386.rpm
Preparing...
##################################################
ITDAgent
##################################################
Interprocess communications data are encrypted by default. Run
  /opt/ibm/director/bin/cfgsecurity to remove this setting.
To start the IBM Director Agent manually, run
/opt/ibm/director/bin/twgstart
Attempting to install DirectorCimCore-5.10-1_RHEL3.i386.rpm
Preparing...
##################################################
DirectorCimCore
##################################################
Creating SSL certificate and private key
Compiling MOF files..
xSeriesCoreServices-level1
##################################################
Compiling MOF files...
Finished compiling MOF files.
Starting Pegasus CIMOM
Setting default events...
Finished setting default events.
Installation of selected components is successful.
To start IBM Director Agent manually, run
/opt/ibm/director/bin/twgstart
```

3. Encryption between the IBM Director agent and server is enabled by default. If you want to disable encryption, use the following commands:

   ```
   /opt/ibm/director/bin/cfgsecurity
   ```

   Output from this command is shown in Example 5-5.

Example 5-5 Disable encryption

```
[root@watts2 FILES]# /opt/ibm/director/bin/cfgsecurity
Do you want IBM Director to encrypt data communications on this system?
     0 - No
     1 - Yes
Enter number corresponding to desired encryption setting (1 is
default):
0
Encryption set or unset successfully.
Do you wish your managed system to be secured?  (Director UNIX systems
are "secured" by default.)
     0 - No
     1 - Yes
Enter number corresponding to desired security setting (1 is default):
0
Setting secured or unsecured system successful.
```

4. Start the Director agent using the following command:

   ```
   /opt/ibm/director/bin/twgstart
   ```

Tip: You can confirm the status of IBM Director Agent by issuing the following command:

```
/opt/IBM/director/bin/twgstat
```

The document *Managing VMware ESX Server using IBM Director*, can provide additional information, although it is based on IBM Director 4.21 and is available from:

http://www.pc.ibm.com/support?page=MIGR-59701

5.3.3 Configuring the ESX Server 3.0 firewall

ESX Server 3.0 includes a firewall that by default does not allow IBM Director traffic to pass. However, the firewall can be easily enabled to support IBM Director using the command:

```
esxcfg-firewall ---enableservice ibmdirector
```

If you prefer to have greater control over port access, you can review the ports used by IBM Director as shown in Table 5-2.

Table 5-2 IBM Director network ports

Connection	IP Port	IPX™ Port
IBM Director		
IBM Director Server to BladeCenter switch module	80 TCP	
IBM Director Server to BladeCenter chassis	427 UDP and TCP	
IBM Director Server to IBM Director Agent	14247 UDP and TCP, 14248 UDP (LINUX only)	4490 (hex) read, 4491 (hex) write
IBM Director Server to IBM Director Agent	Random TCP (see note 1)	
IBM Director Agent to IBM Director Server	14247 UDP and TCP	4490 (hex) read, 4491 (hex) write
IBM Director Server to IBM Director Console	Random (see note 2)	
IBM Director Console to IBM Director Server	2033 TCP (see note 2)	
SNMP access	161 UDP	
SNMP traps	162 UDP	
Remote session on SNMP devices	23	
Web-based access to IBM Director Agent		
IBM Director Web server (configured during installation of IBM Director Agent)	411 HTTP, 423 (HTTPS), 8009 (internal use)	
Service processors		
Telnet to service processors	23 TCP	
IBM Director to service processor	6090 TCP	

Connection	IP Port	IPX™ Port
Web-based access	80	
SNMP agent	161 UDP	
SNMP traps	162 UDP	
LAN alerts	13991 UDP	

NOTES:

1. A random TCP session is used for some tasks (such as file transfer) between the IBM Director Agent and IBM Director Server. The operating system returns a port in the range 1024-65535.
2. IBM Director Console opens a port in the 1024-65535 range then connects through TCP to IBM Director Server using port 2033. When IBM Director Server responds to IBM Director Console, it communicates to the random port in the 1024-65535 range that IBM Director Console opened.

The following command can be used to open ports:

```
esxcfg-firewall --openports
```

To list known services, use:

```
esxcfg-firewall -s
```

Example 5-6 Open network ports on ESX Server

```
[root@watts2 root]# esxcfg-firewall -s
Known services: nfsClient ftpServer ntpClient dellom nisClient
vncServer tmpLicenseClient swISCSIClient CIMHttpsServer sshClient snmpd
tmpAAMClient vpxHeartbeats smbClient hpim tmpHostdVmdbServer
tmpHostdSOAPServer ftpClient sshServer ibmdirector CIMHttpServer
telnetClient
[root@watts2 root]#
[root@watts2 root]# esxcfg-firewall --enableservice ibmdirector
```

5.3.4 Installing Virtual Machine Manager

Section 5.2.2, “Where VMM is installed” on page 218 describes where the server, console and agent components of VMM are to be installed. A key factor is whether or not you are using VMware VirtualCenter:

- If your ESX Server systems are managed by VirtualCenter, install the VMM agent on the VirtualCenter system only (not the ESX Server systems)

- If you do not use VirtualCenter, install the VMM agent on each ESX Server system.

In addition, you must install VMM on both the IBM Director management server and any systems with IBM Director Console installed.

The VMM software can be downloaded by following the links in:

`http://www.ibm.com/servers/eserver/xseries/systems_management/ibm_director/extensions/vmm.html`

Alternatively, you can download the code directly from:

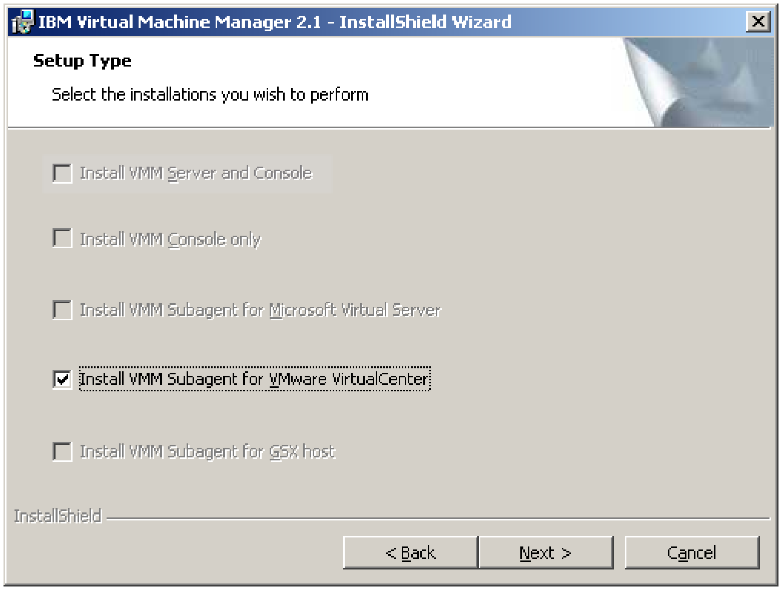
`http://www.pc.ibm.com/support?page=MIGR-56914`

Installing VMM on the VirtualCenter server

Note: Install IBM Director Agent on the system before installing VMM.

To install VMM on your server running VirtualCenter, download the exe from the above URL and run it.

During the installation, accept the defaults. VMM will detect that VirtualCenter is installed, as shown in Figure 5-3.


IBM Virtual Machine Manager 2.1 - InstallShield Wizard

Setup Type

Select the installations you wish to perform

□ Install VMM Server and Console

□ Install VMM Console only

□ Install VMM Subagent for Microsoft Virtual Server

☑ Install VMM Subagent for VMware VirtualCenter

□ Install VMM Subagent for GSX host

Figure 5-3 Installing the VMM agent on your VirtualCenter system

As previously stated, if you are running VirtualCenter, the you do not install VMM on any of the ESX Server systems.

Installing VMM on the ESX Server Console OS

> **Note:** Install IBM Director Agent on the Console OS before installing VMM. See 5.3.2, “Installing IBM Director Agent on ESX Server” on page 224.

If you are not running VirtualCenter, then install the VMM agent on the ESX Server Console OS.

The steps to install VMM to the Console OS are:

1. Download the VMM agent RPM file from the above URL to the Console OS. At the time of writing, the file is 40k1448.rpm.
2. Log onto the ESX Server Console OS as root.
3. Stop IBM Director Agent using the command:

   ```
   /opt/ibm/director/bin/twgstop
   ```

4. Install the package using the following command:

   ```
   rpm -iv 40k1448.rpm
   ```

5. Restart IBM Director Agent using the command:

   ```
   /opt/ibm/director/bin/twgstart
   ```

Installing VMM on the IBM Director management server

You will also need to install the server and console components:

- Systems running IBM Director Server

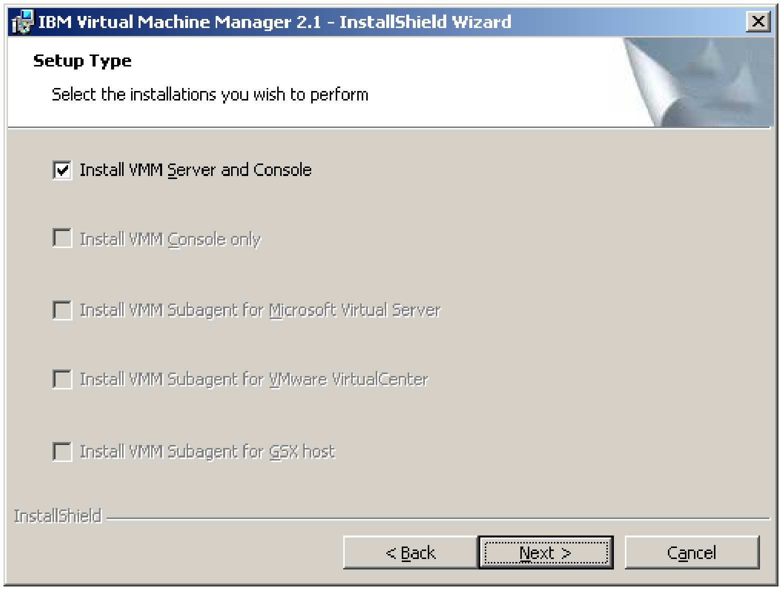
  Install the VMM server component (The VMM console component will be installed automatically at the same time).

- Systems with IBM Director Console installed

  Install the VMM console component.

You will need to have IBM Director installed before installing VMM.

Download and run the VMM installer from the link on the previous page. During the installation, the installer detects that either the IBM Director Server and Console are installed (Figure 5-4 on page 231) or just the Console is installed.



Setup Type

Select the installations you wish to perform

☑ Install VMM Server and Console

□ Install VMM Console only

□ Install VMM Subagent for Microsoft Virtual Server

□ Install VMM Subagent for VMware VirtualCenter

□ Install VMM Subagent for GSX host

Figure 5-4 Install VMM Server and Console on IBM Director server

5.4 Installation steps for Microsoft Virtual Server

You must perform the following steps to install IBM Director and VMM on your Microsoft Virtual Server systems:

1. Install the RSA II driver on the MSVS system.
2. Install IBM Director Agent on the MSVS system.
3. Install the VMM agent.

5.4.1 Installing the RSA II driver in Windows

Before installing IBM Director Agent on your server to run Microsoft Virtual Server, you should install the RSA II driver. The driver can be downloaded from:

`http://www.pc.ibm.com/support?page=MIGR-62147`

Simply download the installer and follow the instructions. More correctly, in Windows, the RSA II is in fact a service. When it is installed, you will see it listed in Services in the Control Panel.

5.4.2 Installing IBM Director Agent on Windows

IBM Director Agent can be installed on the system running Microsoft Virtual Server. Install IBM Director Agent after you have installed the RSA II driver as described in 5.4.1, "Installing the RSA II driver in Windows" on page 231.

IBM Director Agent can be downloaded from:

http://www.pc.ibm.com/support?page=SERV-DIRECT

To install IBM Director Agent, download and run the installer from this URL and following the onscreen prompts. Details of the installation procedure and the dialog boxes you are presented with are fully documented in the *IBM Director Installation and Configuration Guide*, available in PDF format or online in the IBM InfoCenter:

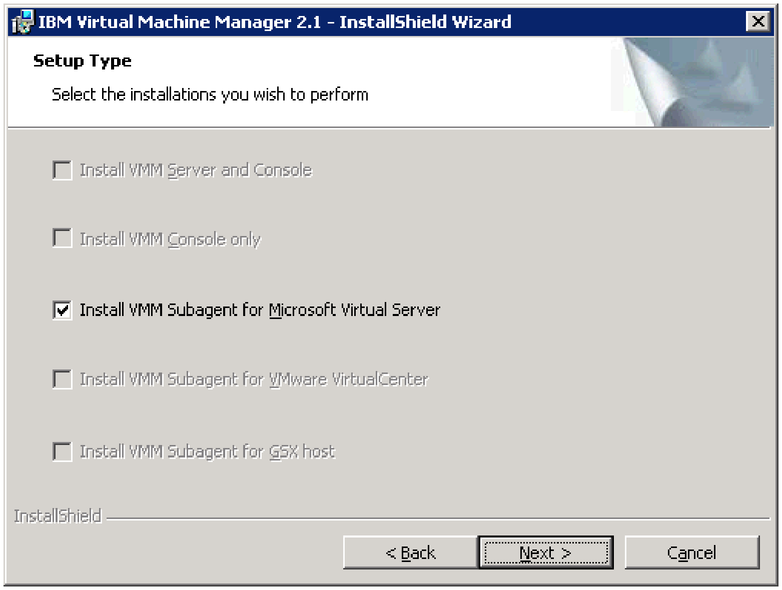
http://www.ibm.com/servers/eserver/xseries/systems_management/ibm_director/resources

5.4.3 Installing VMM

Note: Install IBM Director Agent on the system before installing VMM.

To install VMM on your server, download the executable file from the URL on 229 and run it.

During the installation, accept the defaults. VMM detect that Microsoft Virtual Server is installed, as shown in Figure 5-5 on page 233.


IBM Virtual Machine Manager 2.1 - InstallShield Wizard

Setup Type

Select the installations you wish to perform

□ Install VMM Server and Console

□ Install VMM Console only

☑ Install VMM Subagent for Microsoft Virtual Server

□ Install VMM Subagent for VMware VirtualCenter

□ Install VMM Subagent for GSX host

Figure 5-5   Installing the VMM agent on the Microsoft Virtual Server system

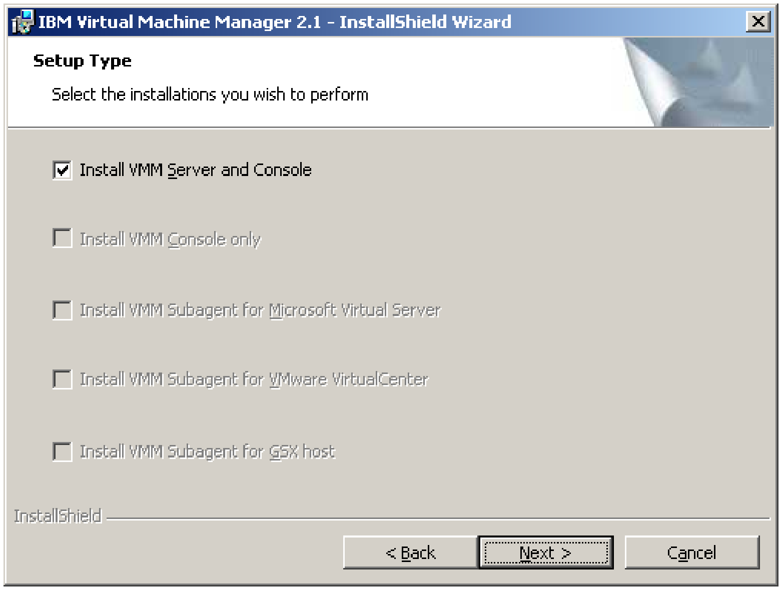
Installing VMM on the IBM Director management server

You will also need to install the server and console components:

- Systems running IBM Director Server: Install the VMM server component. The VMM console component will be installed automatically at the same time.
- Systems with IBM Director Console installed: Install the VMM console component.

You will need to have IBM Director installed before installing VMM.

Download and run the VMM installer from the link on page 229. During the installation, the installer detects that either the IBM Director Server and Console are installed (Figure 5-6 on page 234) or just the Console is installed.


IBM Virtual Machine Manager 2.1 - InstallShield Wizard

Setup Type

Select the installations you wish to perform

☑ Install VMM Server and Console

□ Install VMM Console only

□ Install VMM Subagent for Microsoft Virtual Server

□ Install VMM Subagent for VMware VirtualCenter

□ Install VMM Subagent for GSX host

Figure 5-6 Install VMM Server and Console on IBM Director server

5.5 Using Virtual Machine Manager

VMM offers the ability to manage systems running ESX Server and Microsoft Virtual Server from within IBM Director. This section describes what functions can be performed on the virtual servers using VMM.

The tasks are associated with each object, such as Virtual Machine Manager, Coordinator, VMM farm, hosts and virtual machines. These tasks are described more in detail here:

- Virtual Machine Manager tasks
  - Create VMM Farm: This task is used to create a VMM farm.
  - Help: Provides online help for VMM.
  - Migrate All Virtual Machines: This task is used to create a IBM Director schedulable tasks for migrating all virtual machines from a single host to a different host.
  - Migrate Single Virtual Machine: This task is used to create the IBM Director schedulable tasks for migrating a single virtual machine from one host to a different host.
  - Start Vendor Software: This task is used to start the virtualization vendor application for the targeted VMM object.

- Coordinator Tasks
  - Discover VMM Farms discovers all farms that are defined on a system that is running VMware VirtualCenter server and creates VMM farm objects for the coordinator as necessary.
  - Revoke Credentials revokes the credentials for a coordinator.
- VMM Farm management Tasks
  - Delete From Coordinator (VMware VirtualCenter only) deletes the farm from VMware VirtualCenter and deletes the corresponding managed object for the VMM farm from IBM Director.

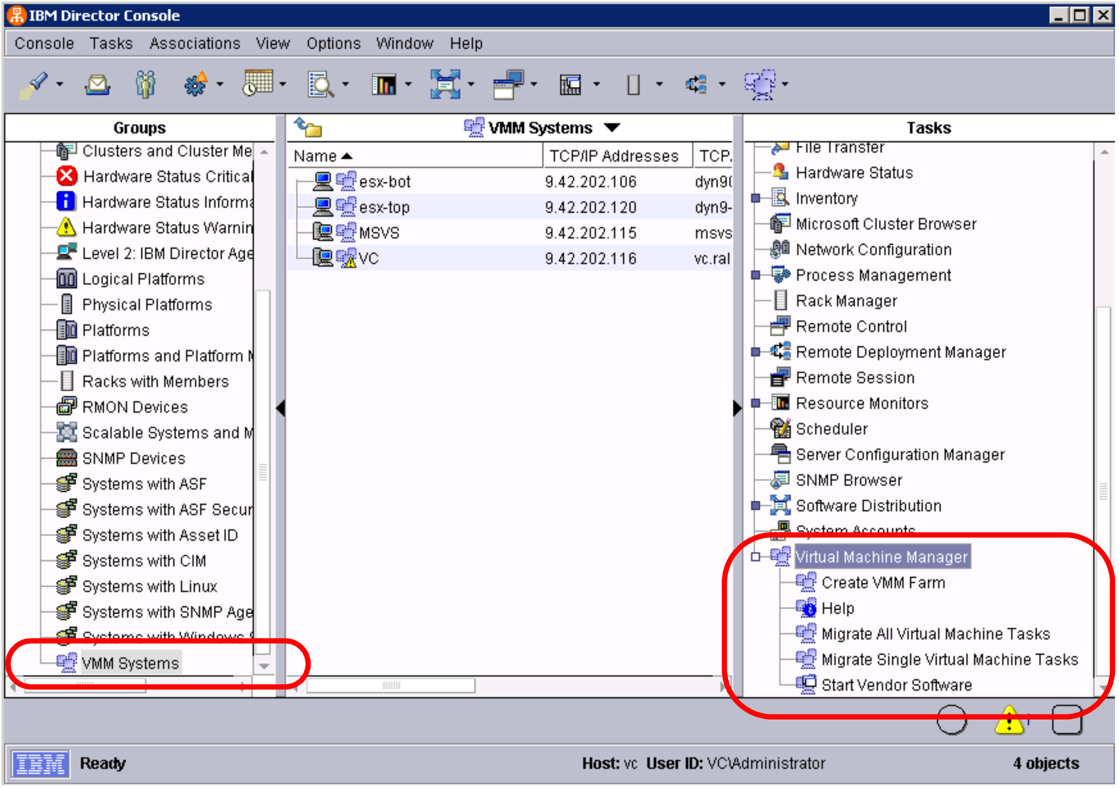
    > **Note:** If you do this task, the VMM farm cannot be rediscovered and instead must be recreated.

  - Start (Microsoft Virtual Server only) starts all hosts that are associated with the targeted VMM farm.
  - Stop (Microsoft Virtual Server only) stops all hosts that are associated with the targeted VMM farm.
- Host Management Tasks
  - Create Virtual Machine creates Virtual Machines within the Virtual Server.
  - Discover Virtual Machines discovers all virtual machines that are associated with a host.
  - Force Power Off All Running Virtual Machines powers-off all running virtual machines that are associated with a host without an orderly shut down of any guest operating systems.
  - Power On All Stopped Virtual Machines powers-on all stopped virtual machines that are associated with a host.
  - Register Virtual Machine
  - Remove Host From VMM Farm removes the managed object for the host from the VMM farm object in IBM Director Console.
  - Resume All Suspended Virtual Machines resumes all suspended virtual machines that are associated with a host.
  - Start (Hosts that are running Microsoft Virtual Server only) starts the host that is represented by the managed object. You can create scheduled jobs that use this task only for hosts that are currently stopped.
  - Stop (Hosts that are running Microsoft Virtual Server only) stops the host that is represented by the managed object. You can create scheduled jobs that use this task only for hosts that are currently started.

  - Suspend All Running Virtual Machines suspends all running virtual machines that are associated with a host.

- Virtual Machine Tasks
  - Delete From Disk deletes the virtual machine from its virtualization application.
  - Migrate All Virtual Machine Tasks migrates all virtual machines on a single host to a different host.
  - Migrate Single Virtual Machine Tasks migrates a single virtual machine from one host to a different host.

In the IBM Director Console, as shown in Figure 5-7, the group **VMM Systems** contains all systems that have the VMM agent installed. Tasks available are a combination of those tasks in the Tasks pane (Figure 5-7) and those available in the right-click context menu, Figure 5-8 on page 237.



Name ▲	TCP/IP Addresses	TCP.
esx-bot	9.42.202.106	dyn9(
esx-top	9.42.202.120	dyn9-
MSVS	9.42.202.115	msvs
VC	9.42.202.116	vc.ral

Ready   Host: vc User ID: VC\Administrator   4 objects

Figure 5-7 IBM Director Panel

As an example of a task creating a virtual machine on a system running Microsoft Virtual Server, follow these steps:

1. Right-click the system running MSVS and click **Host Management** → **Create Virtual Machine**.

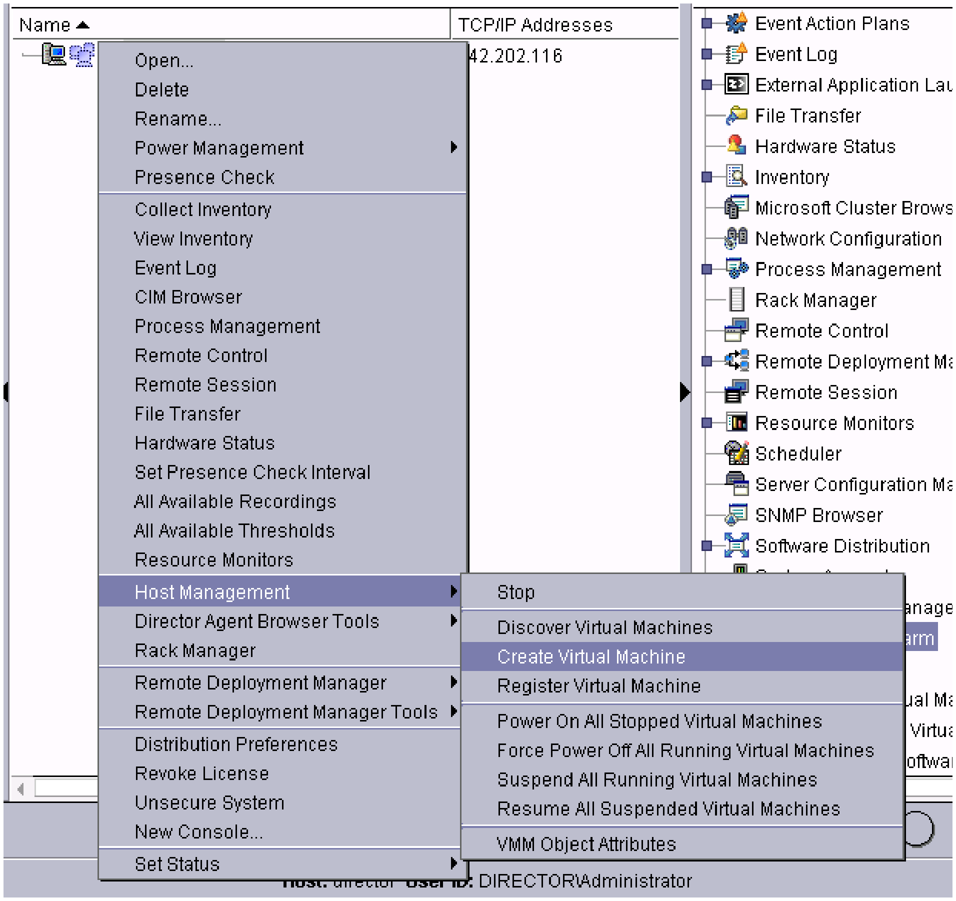
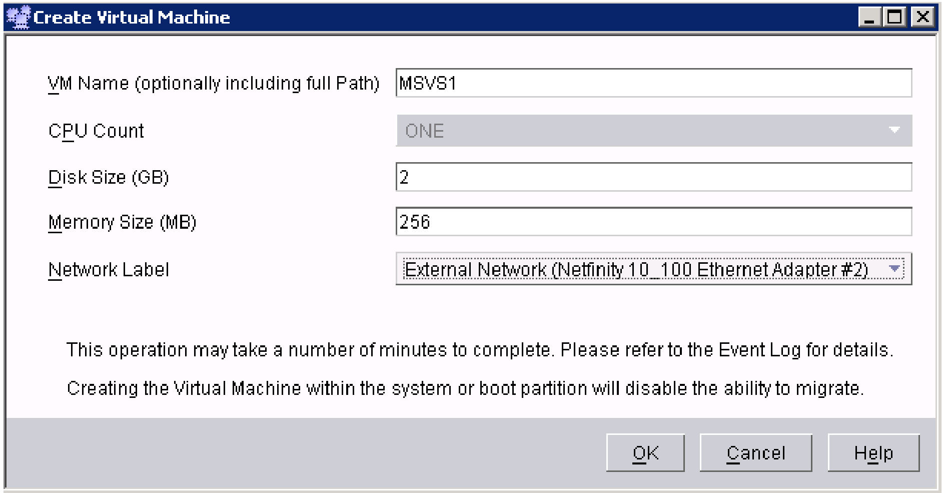


Figure 5-8 Creating a virtual machine in MSVS using VMM

The Create Virtual Machine window opens, as shown in Figure 5-9 on page 238.



VM Name (optionally including full Path) MSVS1

CPU Count ONE

Disk Size (GB) 2

Memory Size (MB) 256

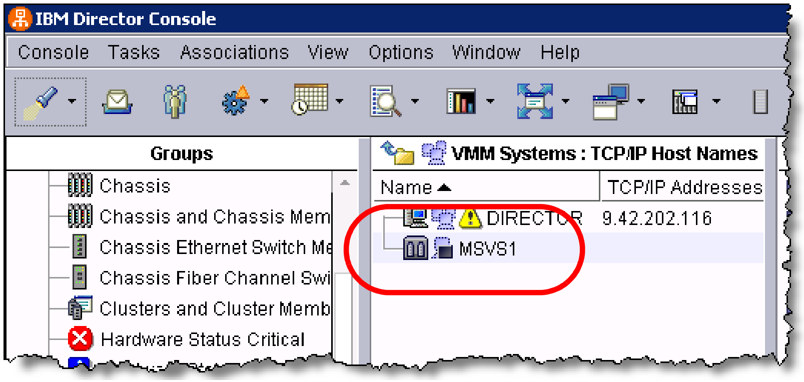
Network Label External Network (Netfinity 10_100 Ethernet Adapter #2)

This operation may take a number of minutes to complete. Please refer to the Event Log for details.

Creating the Virtual Machine within the system or boot partition will disable the ability to migrate.

Figure 5-9 Creating a virtual machine

2. Enter the name of the virtual machine, assign processor, disk, memory, and the network adapter. then click **OK**. When the VM is created, the Director console will be updated with the new virtual machine (Figure 5-10).



Name ▲	TCP/IP Addresses
DIRECTOR	9.42.202.116
MSVS1	

Figure 5-10 The new VM created now appears in the list of VMM systems

In the next example, we show how to perform the same task on a ESX Server.

1. Right-click the Microsoft Virtual Server within the VMM Systems area and click **Create Virtual Machine** as shown in Figure 5-11 on page 239.

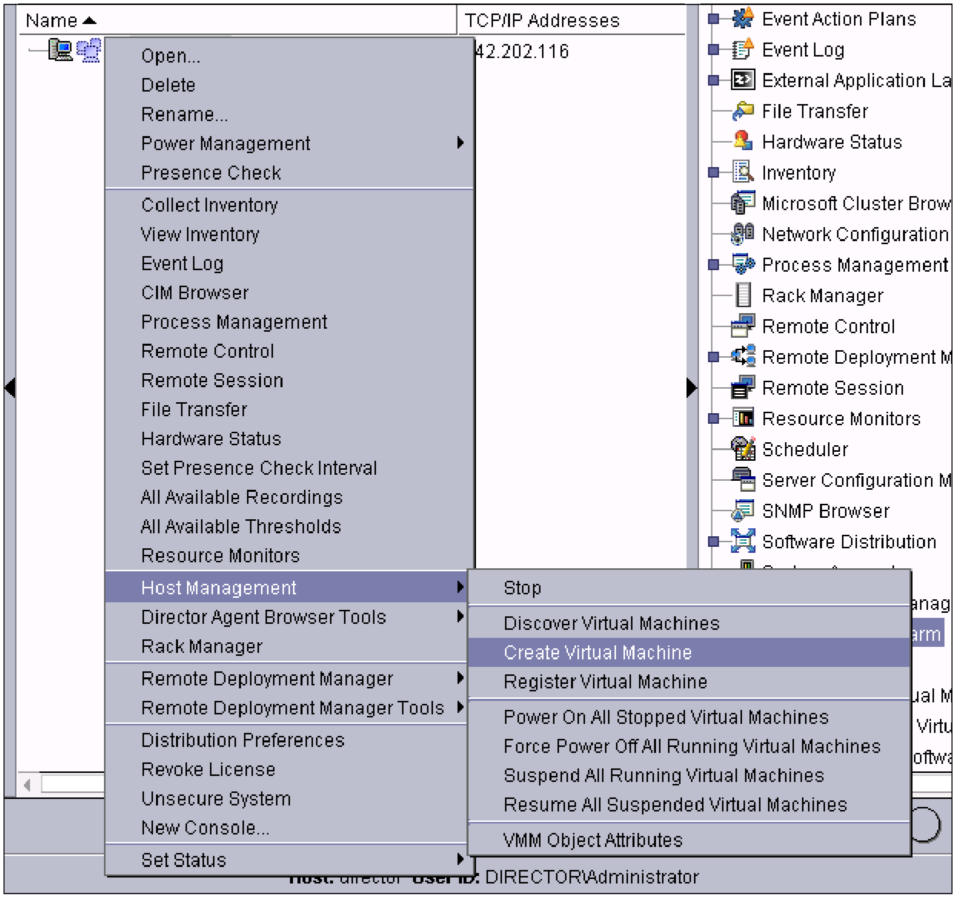
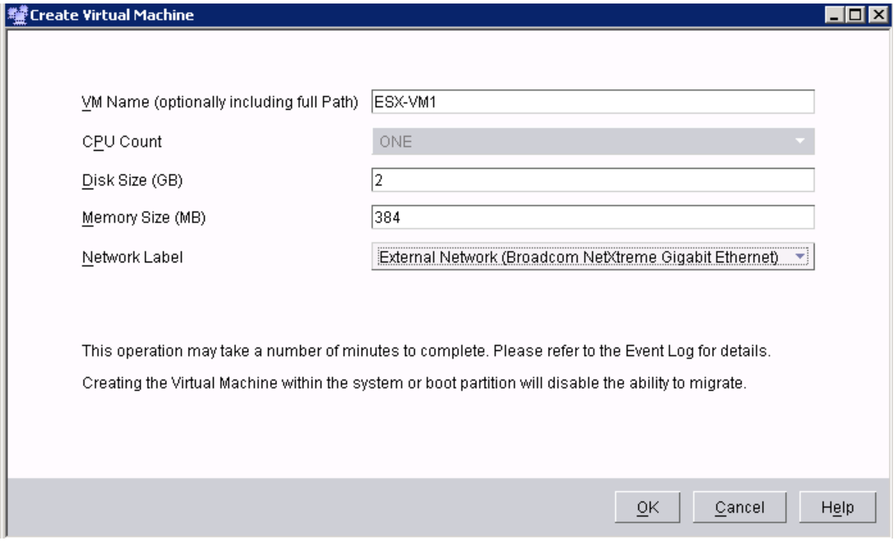


Figure 5-11 Create ESX virtual machine using VMM

2. The Create Virtual Machine window opens, as shown in Figure 5-12 on page 240. Type in the name of the virtual machine, assign processor, disk and memory size and the network adapter then click **OK**.



VM Name (optionally including full Path): ESX-VM1

CPU Count: ONE

Disk Size (GB): 2

Memory Size (MB): 384

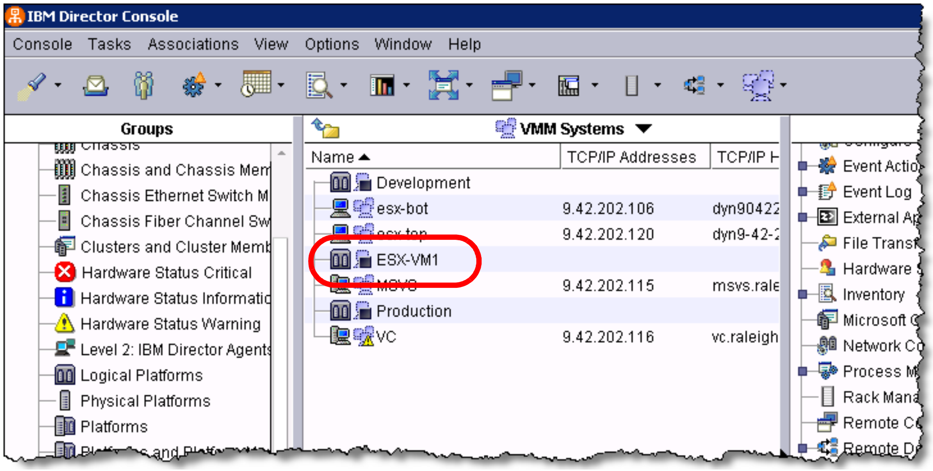
Network Label: External Network (Broadcom NetXtreme Gigabit Ethernet)

This operation may take a number of minutes to complete. Please refer to the Event Log for details.

Creating the Virtual Machine within the system or boot partition will disable the ability to migrate.

Figure 5-12 Create Virtual Machine window

3. As shown in the Figure 5-13, we have now created the ESX1 virtual machine on ESX Server using VMM.



Name ▲	TCP/IP Addresses	TCP/IP H
Development		
esx-bot	9.42.202.106	dyn90422
esx-top	9.42.202.120	dyn9-42-2
ESX-VM1		
MSVS	9.42.202.115	msvs.rale
Production		
VC	9.42.202.116	vc.raleigh

Figure 5-13 Virtual Machine created using VMM

The user interface is the same no matter if you are using virtualization technology from either Microsoft, VMware, or both at the same time.

Next, we show how to add a host to a VMM farm that is running VirtualCenter. When adding a host to a VMM farm that represents a farm defined in VMware VirtualCenter, complete the following steps in the Add Host window:

1. In the Host field, type the IP address or name of the host that is to be added to the VMM farm.

   Note: Before you type a host name, make sure you can resolve the IP address from the host name (for example using **`ping`** ***`hostname`***). Otherwise, the operation to add a host will fail. If you cannot resolve the host name to an IP address, enter the IP address instead.

2. In the User ID field, type the user name for the administrator of the system.

   Generally, this is **`root`** for hosts that are running VMware ESX Server. This user name is used by VMware VirtualCenter server to communicate with the host that is running VMware ESX Server or VMware GSX Server.

3. In the Password field, type the password for the user name that you used.

4. If necessary, in the Port field, type the address of the port that VMware VirtualCenter server will use for communication with VMware ESX Server.

   By default, VMM uses port 902 for this communication. If the system that is running VMware ESX Server is configured to use a port address other than 902, type that port address in this field. Hosts that are running VMware GSX Server do not use the port number.

5. Click **OK**.

> **Note:** The VMM Agent does not enable VirtualCenter VMotion for a newly added ESX Server host. If you want to migrate virtual machines dynamically to or from this host, you must use VirtualCenter to enable VMotion for the host. For information about VirtualCenter VMotion requirements, see the documentation that comes with VirtualCenter.

5.5.1 Migration

VMM supports two types of migration based on the VMM Agent associated with the virtual machines that are being migrated:

- When using the VMM Agent for VirtualCenter with ESX Server hosts, VMM uses dynamic migration.

> **Note:** The VMM Agent for VirtualCenter does not support dynamic migration for GSX hosts.

- When using the VMM Agent for VirtualCenter with GSX Server hosts or the VMM Agents for ESX Server, GSX Server, or Virtual Server, VMM uses static migration.

General migration requirements

There are several migration requirements that apply to both types of migration. Irrespective of the migration type involved, these cautions must be observed when migrating virtual machines:

- Migration of virtual machines is only possible between hosts within the same VMM farm.
- Both the source and destination host must have access to a shared SAN storage. This restriction does not apply when migrating virtual machines on GSX Server hosts with the VMM Agent for VirtualCenter.
- Both the source and destination host must have access to a shared communications network.
- The destination host must have enough memory to support the virtual machine.
- The destination host must support the configuration version of the virtual machine.
- Migration of clustered virtual machines is not supported.
- Migration of virtual machines that are suspended or in a transition state is not supported.
- Source and destination hosts must have a virtual network device with the same label.
- Virtual machines to be migrated cannot be connected to a removable device such as a CD or diskette drive.
- The version of a configuration file for a virtual machine must be supported by the virtualization application (ESX Server or MSVS) that VMM Agent communicates with. Otherwise, the virtual machine cannot be migrated.

Dynamic migration

VMM supports dynamic migration (sometimes referred to as *live migration* or simply *migration* in the VMotion documentation) of virtual machines when using the VMM Agent for VirtualCenter. Dynamic migration is supported only for hosts that are running ESX Server in a VirtualCenter environment. It is not supported for hosts that are running GSX Server in a VirtualCenter environment.

The guest operating systems on migrated virtual machines remain available for use; they are not shut down. VirtualCenter VMotion must be enabled on both the source host and destination host between which you want to migrate virtual machines dynamically. You can use VirtualCenter client to enable VMotion for the applicable hosts and then configure migration using VMM. For information about VirtualCenter VMotion requirements, see Chapter 3, “VMware ESX Server” on page 79 and the documentation that comes with VirtualCenter.

Static migration

VMM supports static migration of virtual machines when using the VMM Agent for VirtualCenter with GSX Server hosts or the VMM Agents for ESX Server, GSX Server, or Virtual Server.

Creating migration tasks

Irrespective of the migration type involved, you can create tasks to migrate virtual machines in one of the following ways:

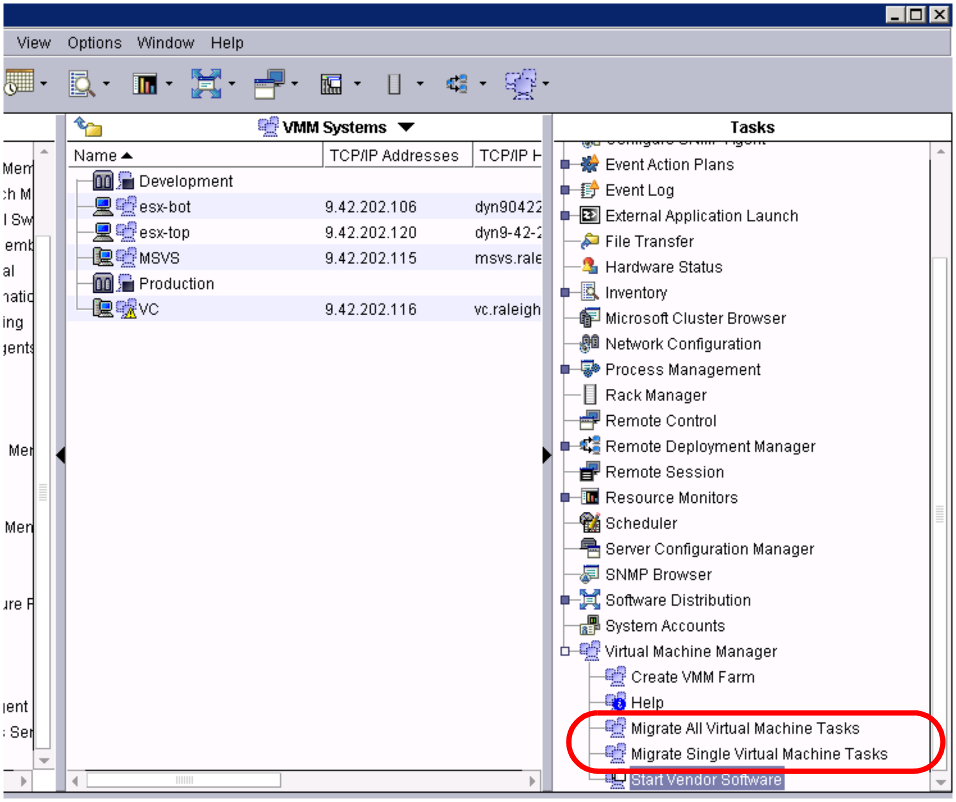
- Create a task to migrate a single virtual machine.

  Use the **Migrate All Virtual Machine Tasks** subtask to create IBM Director schedulable tasks for migrating all virtual machines from a single host to a different host.

- Create a task to migrate all virtual machines on a host.

  Use the **Migrate Single Virtual Machine Tasks** subtask to create IBM Director schedulable tasks for migrating a single virtual machine from one host to a different host.

These tasks are shown in Figure 5-14 on page 244.



Name	TCP/IP Addresses	TCP/IP H
Development		
esx-bot	9.42.202.106	dyn90422
esx-top	9.42.202.120	dyn9-42-2
MSVS	9.42.202.115	msvs.rale
Production		
VC	9.42.202.116	vc.raleigh

Figure 5-14 Migration tasks

5.6 Alerts and event action plans

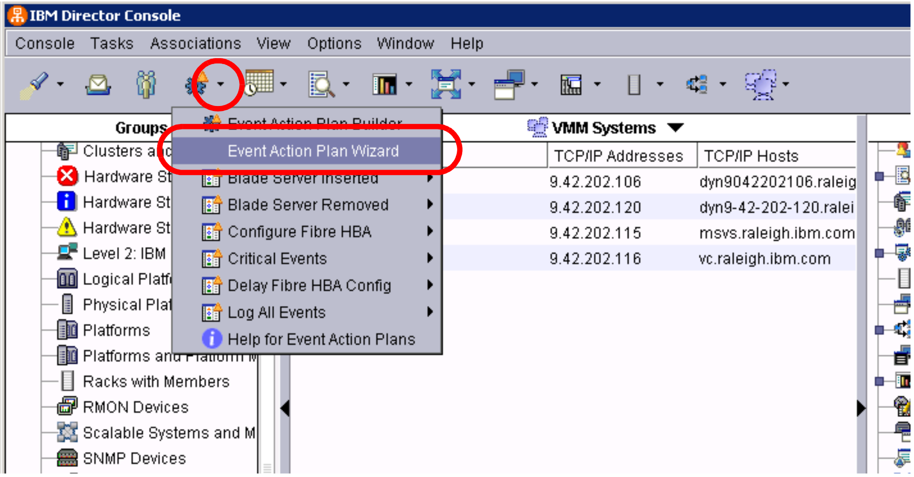
You can use IBM Director to create event action plans based on certain events. This can be used to perform actions on the VMM objects in the event of any hardware failures on the servers that are running the virtual servers. For example, a PFA error on a processor can trigger and event action plan to automatically move the virtual machines from that virtual server to another.

One of the first steps towards a true self-managing system is establishing what actions are to be taken at the first indications of trouble. You are probably already familiar with IBM Director Event Action Plans which enable you to automate responses to alerts, with notification or actions to minimize outages.

5.6.1 Event Action Plan Wizard

Let us see how you would create an Event Action Plan with IBM Director to monitor Predictive Failure Analysis (PFA) events on one of your systems. We use the Event Action Plan Wizard to create the action plan. The Event Action Plan Wizard provides an easy-to-use, step-by-step guide to let IBM Director know what you want to monitor and what automated action should be taken in response to an alert.

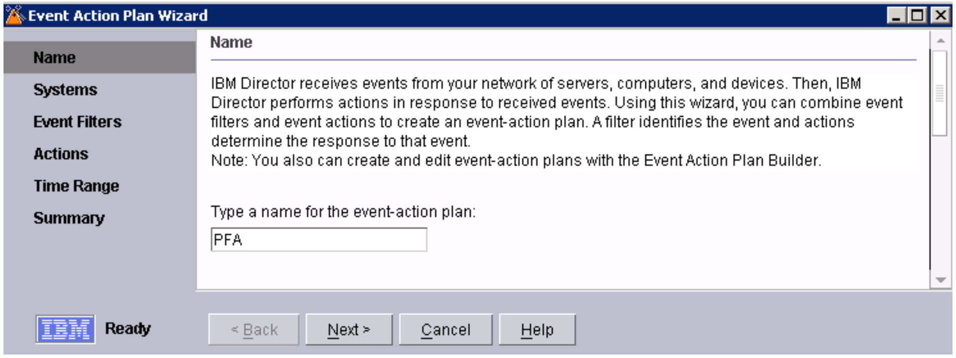
1. Launch the Wizard by clicking the triangle next to the Event Action Plans icon in the Console and click **Event Action Plan Wizard** as shown in Figure 5-15.



TCP/IP Addresses	TCP/IP Hosts
9.42.202.106	dyn9042202106.raleig
9.42.202.120	dyn9-42-202-120.ralei
9.42.202.115	msvs.raleigh.ibm.com
9.42.202.116	vc.raleigh.ibm.com

Figure 5-15 Start the Event Action Plan Wizard

2. In the first window of the Wizard, enter a name for the plan and click **Next**. We entered `PFA`.



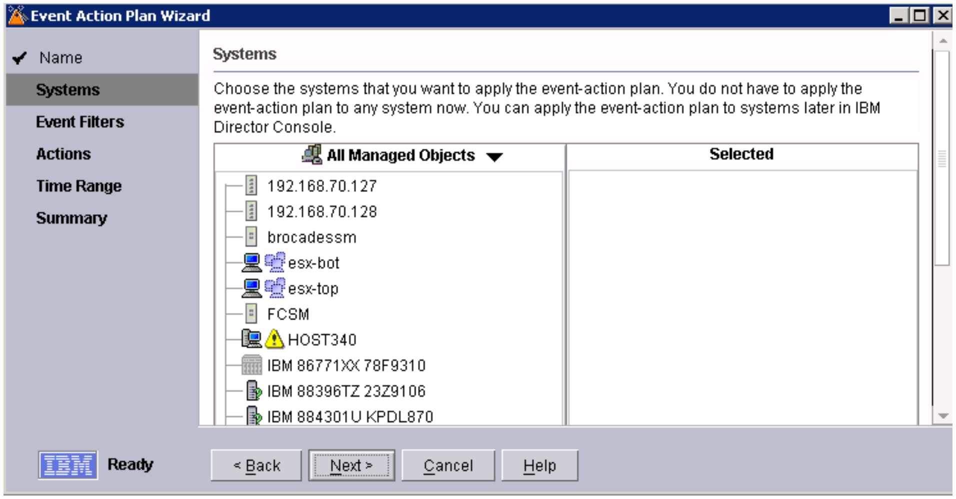
Name

IBM Director receives events from your network of servers, computers, and devices. Then, IBM Director performs actions in response to received events. Using this wizard, you can combine event filters and event actions to create an event-action plan. A filter identifies the event and actions determine the response to that event.
Note: You also can create and edit event-action plans with the Event Action Plan Builder.

Type a name for the event-action plan:
PFA

Figure 5-16 Name the EAP

3. Optionally, select the systems you want to apply the EAP to and click **Add**. (This button is not shown in Figure 5-17; scroll the window down to see it). Click **Next** when you are finished. In this case, we are creating a template for PFA events and will be selecting systems later on, so we have left the list of systems selected blank for now.

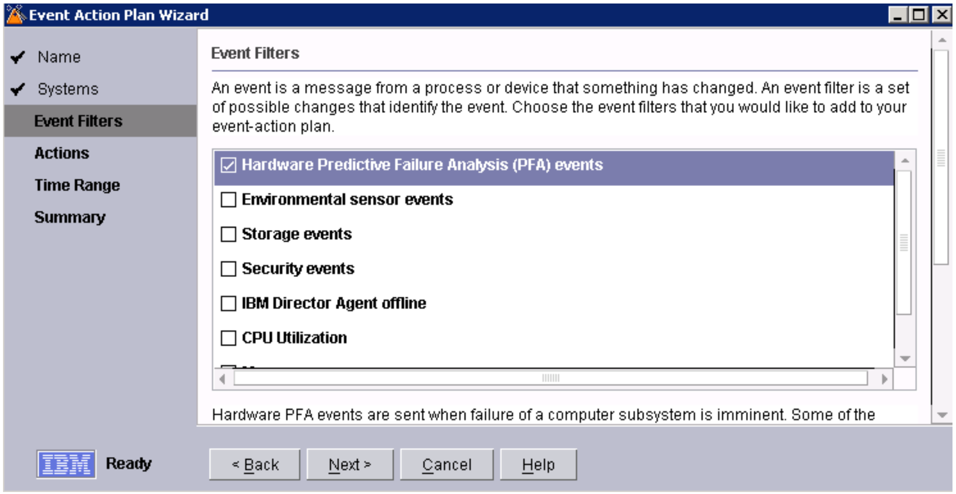


Choose the systems that you want to apply the event-action plan. You do not have to apply the event-action plan to any system now. You can apply the event-action plan to systems later in IBM Director Console.

Figure 5-17   EAP Wizard: Select System

4. In the Event Filters page of the Wizard, select the check boxes adjacent to the types of events you want to monitor. In this example, we are going to trigger on PFA events, shown in Figure 5-18 on page 247. The following event filters are available. The one we chose is in bold:

   - **Hardware Predictive Failure Analysis**
   - (PFA) events
   - Environmental sensor events
   - Storage events
   - Security events
   - IBM Director Agent offline
   - CPU Utilization
   - Memory use



Event Filters

An event is a message from a process or device that something has changed. An event filter is a set of possible changes that identify the event. Choose the event filters that you would like to add to your event-action plan.

☑ Hardware Predictive Failure Analysis (PFA) events
☐ Environmental sensor events
☐ Storage events
☐ Security events
☐ IBM Director Agent offline
☐ CPU Utilization

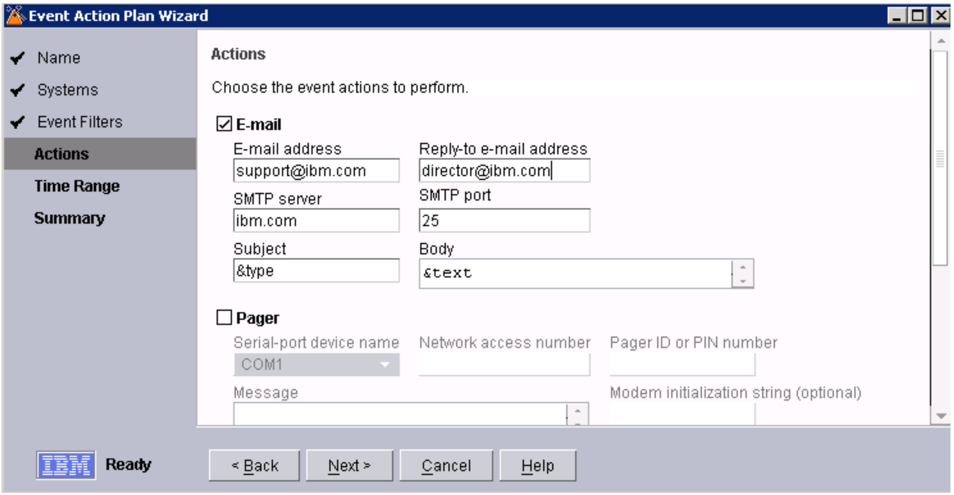
Hardware PFA events are sent when failure of a computer subsystem is imminent. Some of the

Figure 5-18 EAP Wizard: Event Filters

5. Select the action to be taken after the event has occurred and been received by the IBM Director Server. The available options using the Wizard

   – Send an e-mail or
   – Send a page
   – Start a program on the server, the event system, or any other system.

   There are many other actions available in the regular Event Action Plan Builder, but the Wizard only offers these two for the sake of simplicity. In our example we chose **E-mail**, as shown in Figure 5-19.



Actions

Choose the event actions to perform.

☑ E-mail

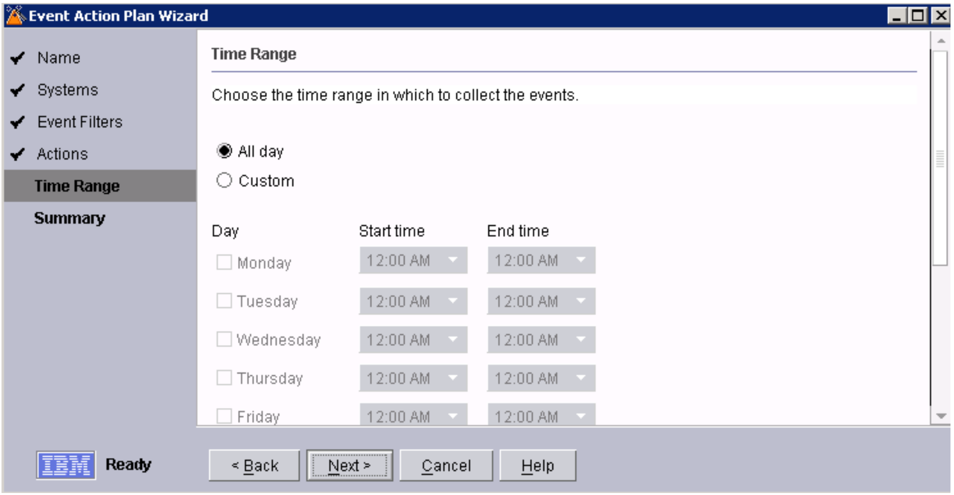
E-mail address: support@ibm.com
Reply-to e-mail address: director@ibm.com
SMTP server: ibm.com
SMTP port: 25
Subject: &type
Body: &text

☐ Pager

Serial-port device name: COM1
Network access number
Pager ID or PIN number
Message
Modem initialization string (optional)

Figure 5-19 EAP Wizard: Actions

6. You are also able to specify time ranges for the EAP, as shown in the Figure 5-20. In this example we use the default, **All day**.



Time Range

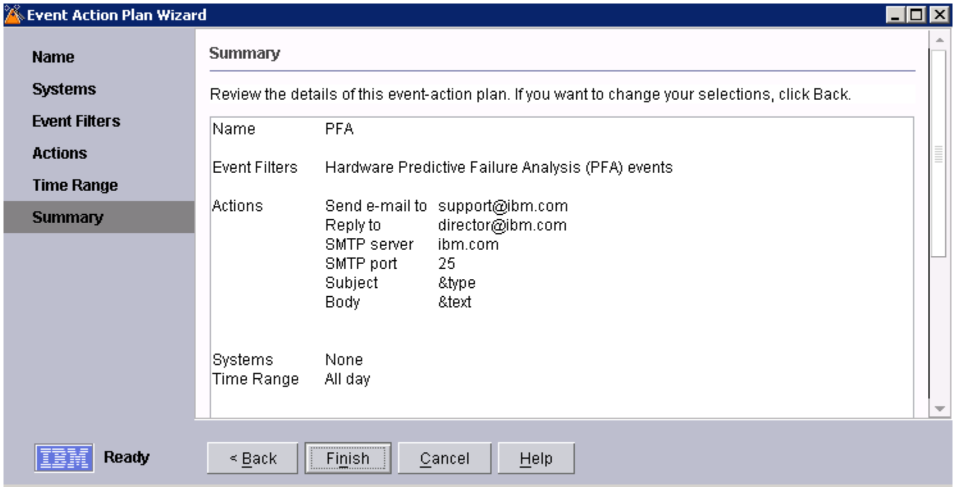
Choose the time range in which to collect the events.

● All day
○ Custom

Day	Start time	End time
□ Monday	12:00 AM	12:00 AM
□ Tuesday	12:00 AM	12:00 AM
□ Wednesday	12:00 AM	12:00 AM
□ Thursday	12:00 AM	12:00 AM
□ Friday	12:00 AM	12:00 AM

Figure 5-20 EAP Wizard: Time ranges

7. Review the summary and select **Finish** when finished, as shown in the Figure 5-21.



Summary

Review the details of this event-action plan. If you want to change your selections, click Back.

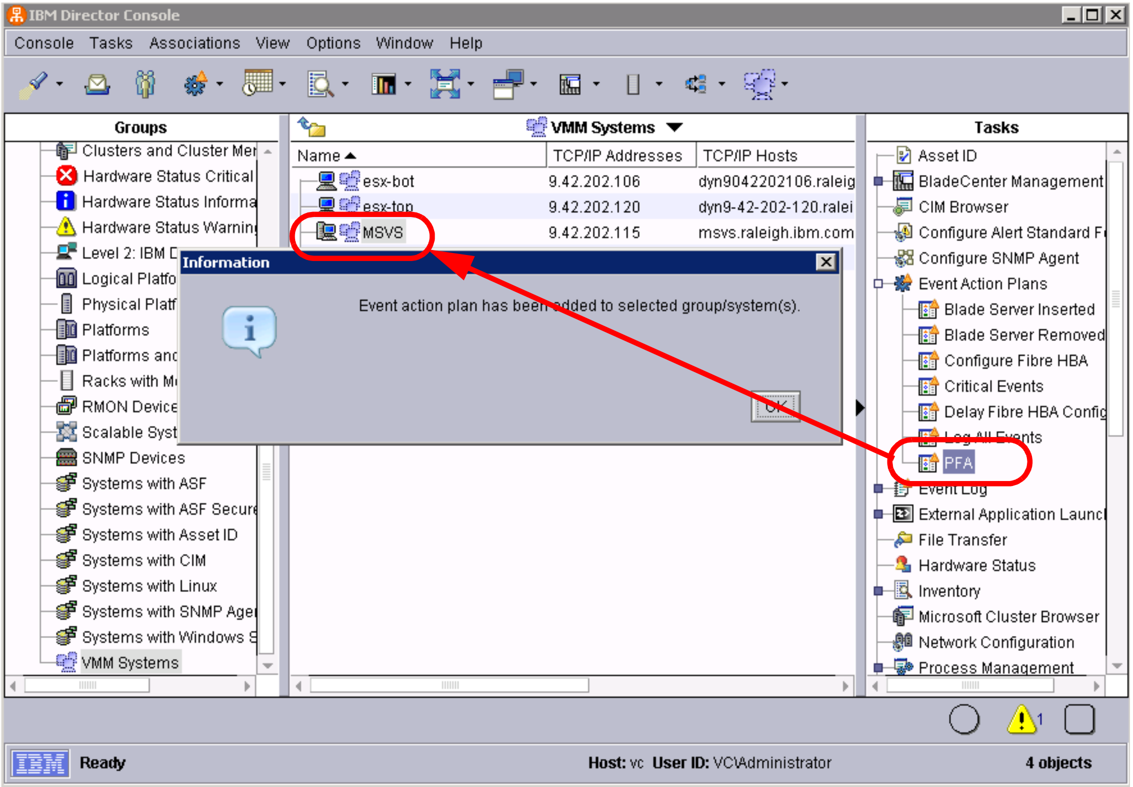
Name	PFA	
Event Filters	Hardware Predictive Failure Analysis (PFA) events	
Actions	Send e-mail to	support@ibm.com
	Reply to	director@ibm.com
	SMTP server	ibm.com
	SMTP port	25
	Subject	&type
	Body	&text
Systems	None	
Time Range	All day	

Figure 5-21 EAP Wizard: Summary

8. You are now able to apply this to a single server or a group of servers (managed objects).

   If you need to edit this EAP at a later stage, open the Event Action Plan Builder, right-click the EAP you want to edit and click **Edit with Event Action Plan Wizard**. Only EAPs built with the EAP Wizard can be edited using the EAP Wizard.

   In this example, we apply this event action plan to the Microsoft Virtual Server. To do this, drag and drop the PFA event action plan (from the right-hand Tasks side to the MSVS server on the middle pane, as shown in Figure 5-22.



VMM Systems

Name	TCP/IP Addresses	TCP/IP Hosts
esx-bot	9.42.202.106	dyn9042202106.raleig
esx-top	9.42.202.120	dyn9-42-202-120.ralei
MSVS	9.42.202.115	msvs.raleigh.ibm.com

Event action plan has been added to selected group/system(s).

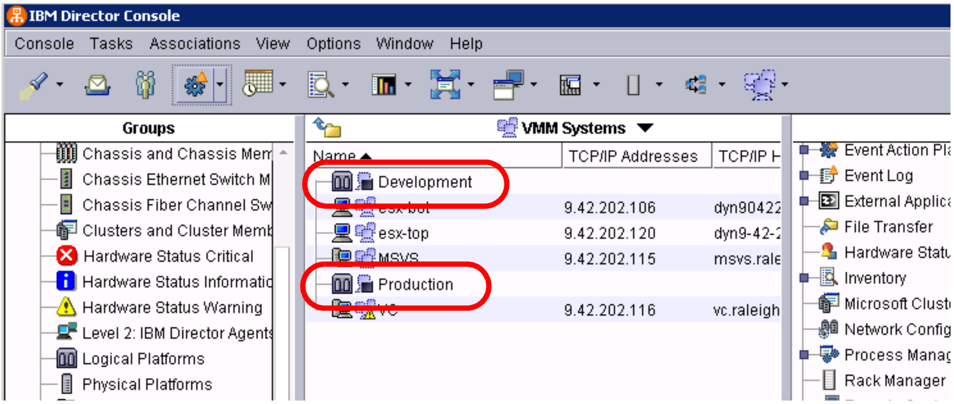
Ready **Host:** vc **User ID:** VC\Administrator **4 objects**

Figure 5-22 Trigger MSVS on PFA alerts

In this example, we have now configured IBM Director to send an e-mail if there is a PFA event on the Microsoft Virtual Server.

5.6.2 Event Action Plan Builder

The Event Action Plan Builder is the primary full-function tool to create event action plans. As an example, we are running Microsoft Virtual Server with two virtual machines configured, Development and Production, as shown in Figure 5-23.

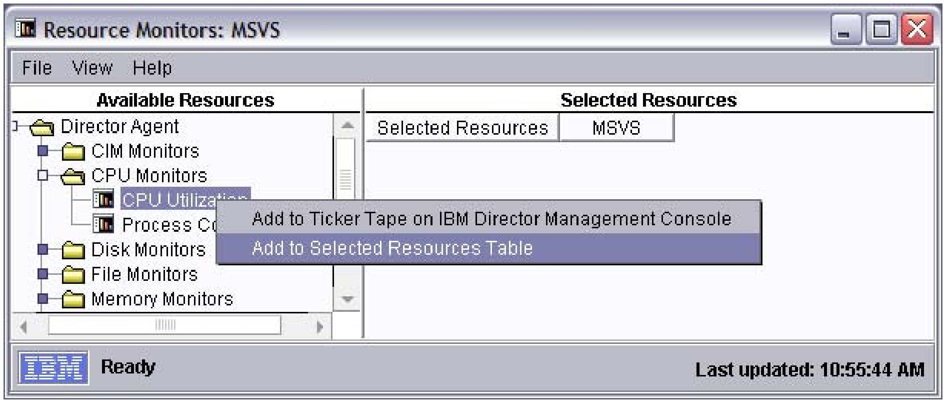


Name	TCP/IP Addresses	TCP/IP H
Development		
esx-bot	9.42.202.106	dyn90422
esx-top	9.42.202.120	dyn9-42-2
MSVS	9.42.202.115	msvs.rale
Production		
vc	9.42.202.116	vc.raleigh

Figure 5-23 Production and Development VMs

In our example, we want to suspend the Development virtual machine if CPU utilization of the virtual server is above 80%. The steps to configure this are:

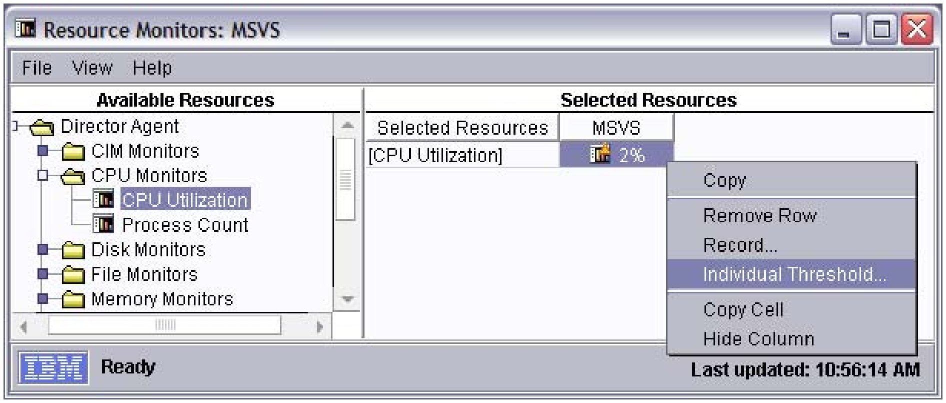
1. Create a resource monitor on the Virtual Server. To do this, right-click the Virtual Server system and click **Resource Monitors**.
2. Under Available Resources expand the tree, **Director** → **Director Agent** → **CPU Monitors**. Then right-click **CPU Utilization** and click **Add to Select Resources Table**, as shown in Figure 5-24.



Selected Resources	MSVS

Figure 5-24 Add to Selected Resources Table

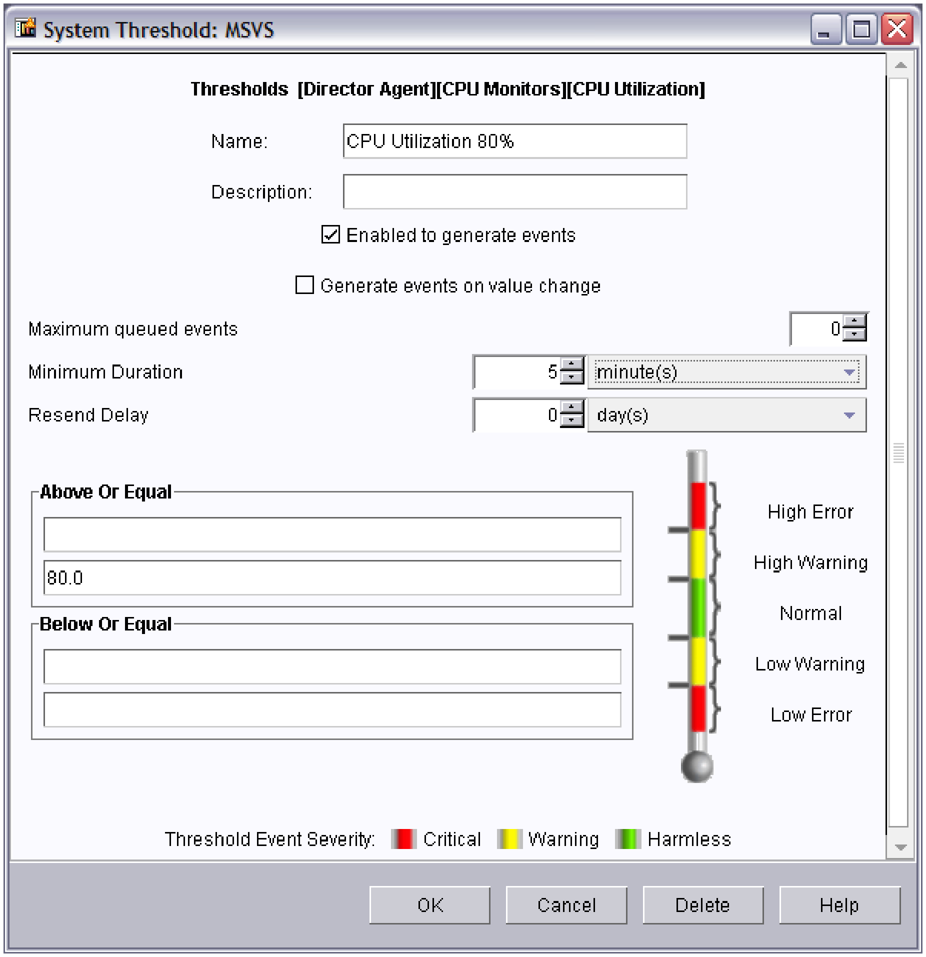
3. Right-click the selected resource in the right pane and click **Individual Threshold**, as shown in Figure 5-25.



Selected Resources	MSVS
[CPU Utilization]	2%

Figure 5-25 Individual Threshold

4. Next, you must configure the threshold. In this example we gave the threshold an appropriate name, configured the threshold to generate an event if CPU Utilization is sustained at 80% during a **5**-minute(s) Minimum Duration, as shown in Figure 5-26 on page 252.



System Threshold: MSVS

Thresholds [Director Agent][CPU Monitors][CPU Utilization]

Name: CPU Utilization 80%

Description:

☑ Enabled to generate events

□ Generate events on value change

Maximum queued events: 0

Minimum Duration: 5 minute(s)

Resend Delay: 0 day(s)

Above Or Equal

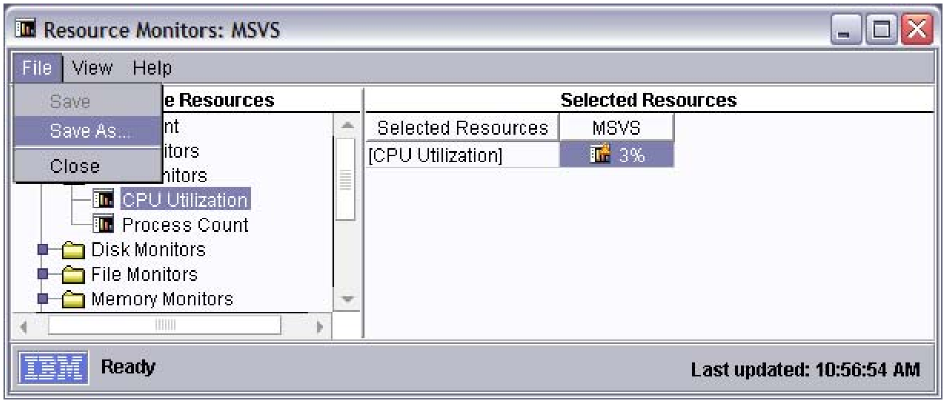
80.0

Below Or Equal

Threshold Event Severity: 🟥 Critical 🟨 Warning 🟩 Harmless

Figure 5-26 Configure System Threshold

5. Click **OK** to save the changes, then click **File** →> **Save As** to save the resource monitor, as shown in Figure 5-27 on page 253.



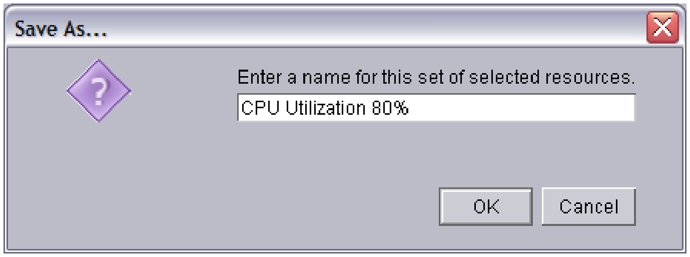
Selected Resources	MSVS
[CPU Utilization]	3%

Ready

Last updated: 10:56:54 AM

Figure 5-27 Save Resource Monitor

6. Give the Resource Monitor an appropriate name and click **OK**, as shown in Figure 5-28. Click **File** → **Close** to exit.



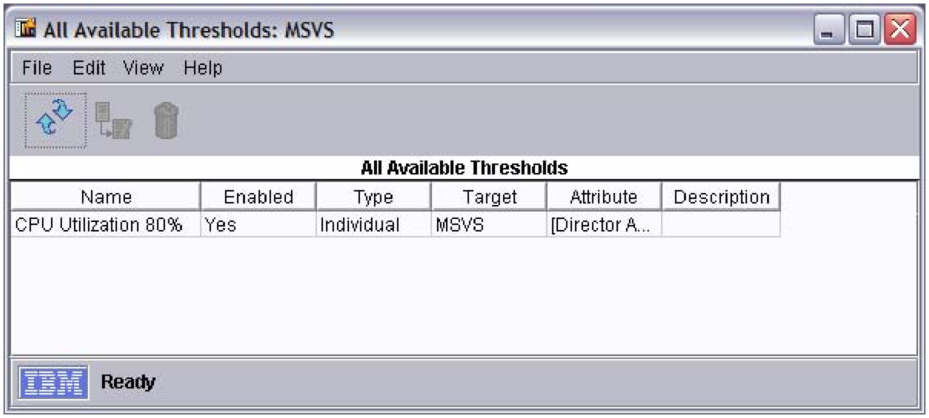
Enter a name for this set of selected resources.

CPU Utilization 80%

Figure 5-28 Save Resource Monitor continued

7. In the next step we are going to verify that the resource monitor got created OK. To do this, right-click the Virtual Server system again and click **All Available Thresholds**.

   The thresholds defined on this system will now display, as shown in Figure 5-29 on page 254.

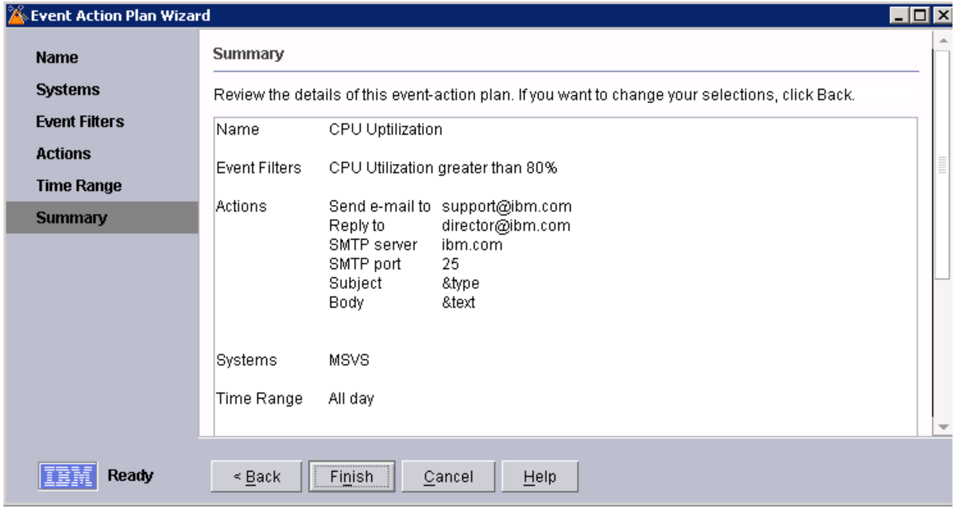


All Available Thresholds

Name	Enabled	Type	Target	Attribute	Description
CPU Utilization 80%	Yes	Individual	MSVS	[Director A...	

Figure 5-29 Display Available Thresholds on MSVS

8. Use the Event Action Plan Wizard to create a simple EAP. In this example, we create a simple EAP with the characteristics shown in Figure 5-30.



Summary

Review the details of this event-action plan. If you want to change your selections, click Back.

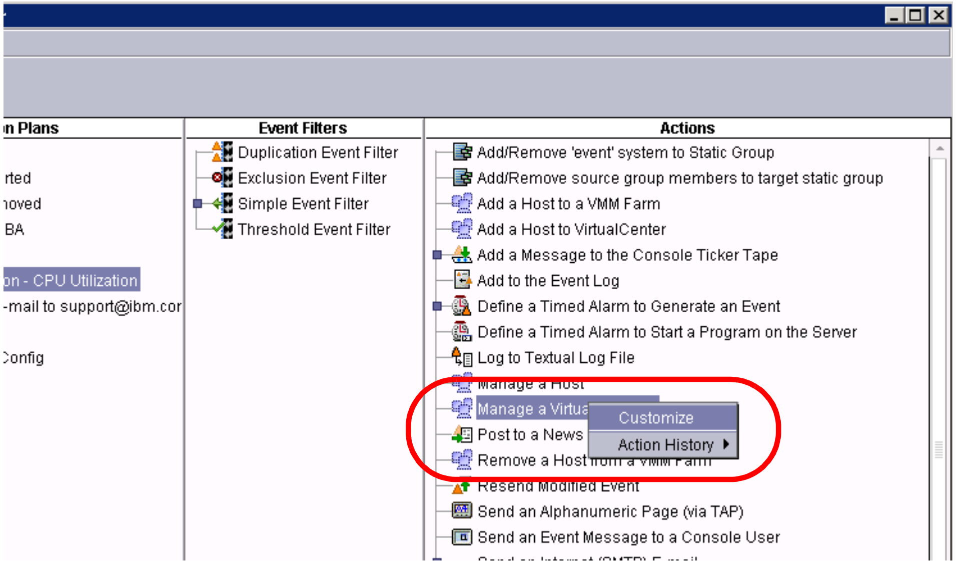
Name	CPU Uptilization	
Event Filters	CPU Utilization greater than 80%	
Actions	Send e-mail to	support@ibm.com
	Reply to	director@ibm.com
	SMTP server	ibm.com
	SMTP port	25
	Subject	&type
	Body	&text
Systems	MSVS	
Time Range	All day	

Figure 5-30 CPU Utilization EAP Summary

9. Associate this event action plan to the server running Microsoft Virtual Server either in the Wizard or by dragging and dropping it in the IBM Director Console.

10. Now we need to create a new event action to suspend the Development VM. Launch the **Event Action Plan Builder**.

11. In the Actions pane, right-click **Manage a Virtual Machine** and click **Customize** as shown in Figure 5-31 on page 255.



n Plans

rted
noved
BA

on - CPU Utilization
-mail to support@ibm.cor

Config

Event Filters

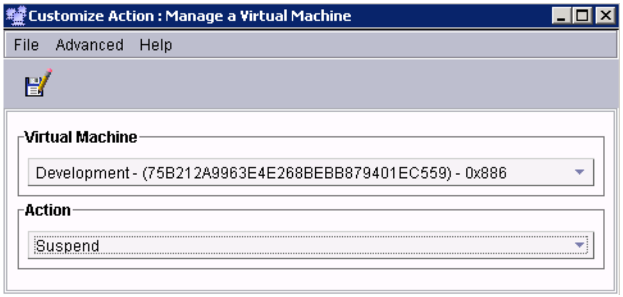
- Duplication Event Filter
- Exclusion Event Filter
- Simple Event Filter
- Threshold Event Filter

Actions

- Add/Remove 'event' system to Static Group
- Add/Remove source group members to target static group
- Add a Host to a VMM Farm
- Add a Host to VirtualCenter
- Add a Message to the Console Ticker Tape
- Add to the Event Log
- Define a Timed Alarm to Generate an Event
- Define a Timed Alarm to Start a Program on the Server
- Log to Textual Log File
- Manage a Host
- Manage a Virtua
- Post to a News
- Remove a Host from a VMM Farm
- Resend Modified Event
- Send an Alphanumeric Page (via TAP)
- Send an Event Message to a Console User

Figure 5-31 Customizing the Manage a Virtual Machine action

12. Figure 5-32 opens. Select the Development virtual machine from the drop-down list and select **Suspend** in the Action drop-down list.



Customize Action : Manage a Virtual Machine

Virtual Machine: Development - (75B212A9963E4E268BEBB879401EC559) - 0x886

Action: Suspend

Figure 5-32 Customization of Manage a Virtual Machine

13. Select **File** → **Save As**, type an appropriate name, and click **OK**, as shown in the Figure 5-33 on page 256.

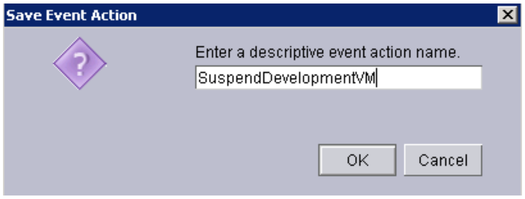


Figure 5-33 Save the customized action

14.Associate the new customized action to the CPU Utilization event action plan by dragging and dropping it as shown in Figure 5-34 .

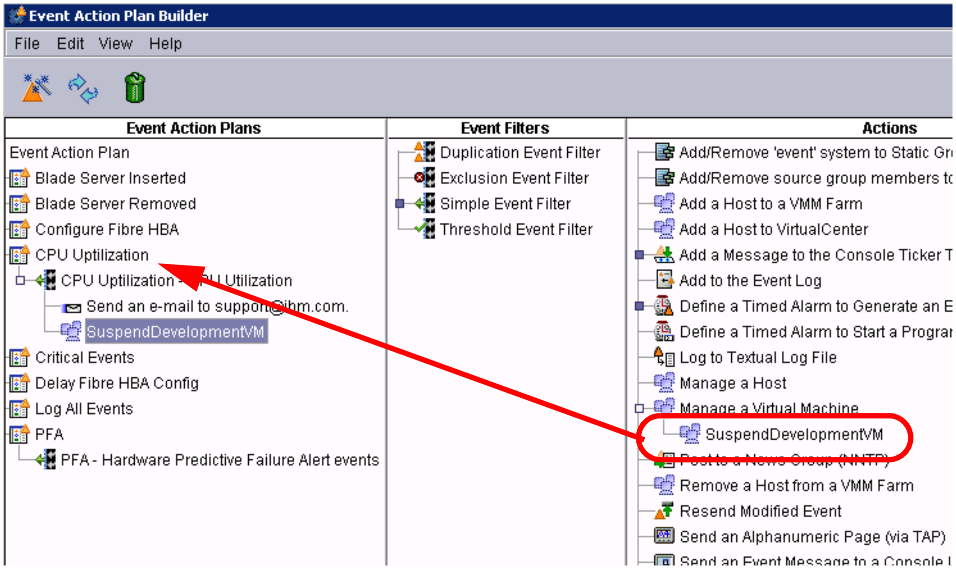

Event Action Plan Builder
File Edit View Help
Event Action Plans
Event Action Plan
Blade Server Inserted
Blade Server Removed
Configure Fibre HBA
CPU Uptilization
CPU Uptilization - CPU Utilization
Send an e-mail to support@ibm.com.
SuspendDevelopmentVM
Critical Events
Delay Fibre HBA Config
Log All Events
PFA
PFA - Hardware Predictive Failure Alert events
Event Filters
Duplication Event Filter
Exclusion Event Filter
Simple Event Filter
Threshold Event Filter
Actions
Add/Remove 'event' system to Static Gr
Add/Remove source group members to
Add a Host to a VMM Farm
Add a Host to VirtualCenter
Add a Message to the Console Ticker T
Add to the Event Log
Define a Timed Alarm to Generate an E
Define a Timed Alarm to Start a Prograr
Log to Textual Log File
Manage a Host
Manage a Virtual Machine
SuspendDevelopmentVM
Post to a News Group (NNTP)
Remove a Host from a VMM Farm
Resend Modified Event
Send an Alphanumeric Page (via TAP)
Send an Event Message to a Console U

Figure 5-34 Apply the Event Action to Action Plan

You have now successfully configured IBM Director to suspend the Development VM if CPU Utilization is above 80% on the Microsoft Virtual Server. Other examples of Event Action Plans includes trigger migrations based on hardware based events.

Abbreviations and acronyms

AC	alternating current
AMD	Advanced Micro Devices
APC	American Power Conversion
APIC	Advanced Programmable Interrupt Controller
AS	Australian Standards
BIOS	basic input output system
BMC	baseboard management controller
CD	compact disk
CD-ROM	compact disc read only memory
CIMOM	Common Information Model
COM	Component Object Model
CPU	central processing unit
CRM	Customer Relationships Management
CSG	Chip Select Group
DAS	Distributed Availability Services
DDR	Double Data Rate
DHCP	Dynamic Host Configuration Protoco
DIMM	dual inline memory module
DMZ	demilitarized zone
DNS	Domain Name System
DR	disaster recovery
DRAM	dynamic random access memory
DRS	Distributed Resource Scheduler
DSA	Digital Signature Algorithm
EAP	event action plan
ECC	error checking and correcting
EMEA	Europe, Middle East, Africa
ERP	enterprise resource planning
EU	European Union
FAMM	full array memory mirroring
FQDN	fully qualified domain names
GB	gigabyte
GUI	graphical user interface
HA	High Availability
HAM	hot-add memory
HBA	host bus adapter
HCL	Hardware Compatibility List
HMC	Hardware Management Console
HPC	high performance computing
HPMA	high performance memory array
HT	Hyper-Threading
HTML	Hypertext Markup Language
HW	hardware
I/O	input/output
IBM	International Business Machines Corporation
ID	identifier
IIS	Internet Information Server
IP	Internet Protocol
ISA	industry standard architecture
ISO	International Organization for Standardization
ISV	independent software vendor
IT	information technology
ITSO	International Technical Support Organization
IXA	Integrated xSeries Adapter

KB	kilobyte
KVM	keyboard video mouse
LAN	local area network
LED	light emitting diode
LPAR	logical partitions
LUN	logical unit number
MAC	media access control
MB	megabyte
MBR	Master Boot Record
MCSA	Microsoft Certified Systems Administrator
MIOC	Memory and I/O Controller
MMU	memory management unit
MOF	Managed Object Format
MOM	Microsoft Operations Manager
MP	multiprocessor
MSCS	Microsoft Cluster Server
MSVS	Microsoft Virutal Server
MUI	management user inferface
MXE	modular expansion enclosure
NAS	network addressable storage
NIC	network interface card
NTP	Network Time Protocol
NUMA	Non-Uniform Memory Access
NVRAM	non-volatile random access memory
ODBC	Open DataBase Connectivity
OLTP	online transaction processing
OS	operating system
PC	personal computer
PCI	peripheral component interconnect
PDF	Portable Document Format
PFA	Predictive Failure Analysis
POST	power on self test
PXE	Pre-boot-execution
RAC	Real Application Clusters
RAID	redundant array of independent disks
RAM	random access memory
RAS	remote access services; row address strobe
RBS	redundant bit steering
RDM	Remote Deployment Manager
RPM	Red Hat Package Manager
RSA	Remote Supervisor Adapter
SAN	storage area network
SAS	Serial Attached SCSI
SCSI	small computer system interface
SDK	Software Developers' Kit
SDRAM	static dynamic RAM
SLA	service level agreement
SMI	Structure of Management Information
SMP	symmetric multiprocessing
SNMP	Simple Network Management Protocol
SOA	Service Oriented Architecture
SPORE	ServerProven® Opportunity Request for Evaluation
SQL	structured query language
SRAT	Static Resource Allocation Table
SSL	Secure Sockets Layer
TB	terabyte
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TEC	Tivoli Enterprise™ Console
UML	Unified Modeling Language
UPS	uninterruptible power supply
URL	Uniform Resource Locator

USB	universal serial bus
VB	Visual Basic
VC	VirtualCenter
VCP	VMware Certified Professional
VIN	Virtual Infrastructure Node
VLAN	Virtual Local Area Network
VM	virtual machine
VMFS	virtual machine file system
VMM	Virtual Machine Manager Virtual Machine Monitor
VS	Virtual Server
VSMT	Virtual Server Migration Toolkit
WHDC	Windows Hardware & Driver Central
XAPIC	multi-processor interrupt communication protocol

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 265. Note that some of the documents referenced here may be available in softcopy only.

- *VMware ESX Server: Scale Up or Scale Out?*, REDP-3953
- *Introducing Windows Server x64 on IBM @server xSeries Servers*, REDP-3982
- *IBM Director 5.10*, SG24-6188
- *Tuning IBM @server xSeries Servers for Performance*, SG24-5287
- *Planning and Installing the IBM @server X3 Architecture Servers*, SG24-6797
- *Server Consolidation with VMware ESX Server*, REDP-3939
- *Implementing VMware ESX Server 2.1 with IBM TotalStorage FAStT*, SG24-6434
- *Introducing Microsoft Virtual Server 2005 on IBM @server xSeries Servers*, REDP-3912
- *IBM @server xSeries Server Consolidation: an Introduction*, REDP-3785

Other publications

These publications are also relevant as further information sources:

- Whitepaper: *Managing VMware ESX Server using IBM Director - IBM Director*

  http://www.pc.ibm.com/support?page=MIGR-59701
- IBM Director 5.10 publications

  http://www.pc.ibm.com/support?page=MIGR-61788

- *Virtual Machine Manager Installation and User's Guide*

  http://www.pc.ibm.com/support?page=MIGR-56955
- ESX Server Documentation

  http://www.vmware.com/support/pubs/esx_pubs.html
- Whitepaper *Hyper-Threading Support in VMware ESX Server 2*

  http://www.vmware.com/pdf/esx21_hyperthreading.pdf
- *Timekeeping in VMware Virtual Machines*

  http://www.vmware.com/pdf/vmware_timekeeping.pdf
- *ESX Server 2 Best Practices*

  http://www.vmware.com/pdf/esx2_best_practices.pdf
- *Virtual Server 2005 Administrator's Guide*

  http://www.microsoft.com/technet/prodtechnol/virtualserver/2005/proddocs

Online resources

These Web sites and URLs are also relevant as further information sources:

IBM Web sites

- IBM System x home page

  http://www.ibm.com/systems/x
- IBM System x3950 home page

  http://www.ibm.com/systems/x/scalable/x3950
- IBM @server x460 home page

  http://www.ibm.com/servers/eserver/xseries/x460.html
- Installing Microsoft Windows Server 2003 on the x3950

  http://www.pc.ibm.com/support?page=MIGR-61178
- Installing Microsoft Windows Server 2003 x64 Edition on the x3950

  http://www.pc.ibm.com/support?page=MIGR-60676
- Installing Microsoft Virtual Server 2005 on the x3950

  http://www.pc.ibm.com/support?page=MIGR-61438

- Whitepaper: *Managing VMware ESX Server using IBM Director - IBM Director*

  http://www.pc.ibm.com/support?page=MIGR-59701
- IBM Director downloads

  http://www.pc.ibm.com/support?page=SERV-DIRECT
- IBM Director resources page

  http://www.ibm.com/servers/eserver/xseries/systems_management/ibm_director/resources
- IBM Director 5.10 publications

  http://www.pc.ibm.com/support?page=MIGR-61788
- Virtual Machine Manager product information

  http://www.ibm.com/servers/eserver/xseries/systems_management/ibm_director/extensions/vmm.html
- IBM Virtual Machine Manager downloads

  http://www.pc.ibm.com/support?page=MIGR-56914
- *Virtual Machine Manager Installation and User's Guide*

  http://www.pc.ibm.com/support?page=MIGR-56955
- RSA II daemon for Linux on the x3950

  http://www.pc.ibm.com/support?page=MIGR-59454
- RSA II driver for Windows on the x3950

  http://www.pc.ibm.com/support?page=MIGR-62147
- UpdateXpress CD

  http://www.pc.ibm.com/support?page=MIGR-53046
- RETAIN Tip: System crash with ESX Server on an x3950 with 60+ GB RAM

  http://www.pc.ibm.com/support?page=MIGR-62310

VMware Web sites

- ESX Server Documentation

  http://www.vmware.com/support/pubs/esx_pubs.html
- Whitepaper *Hyper-Threading Support in VMware ESX Server 2*

  http://www.vmware.com/pdf/esx21_hyperthreading.pdf
- ESX Server Technical Papers

  http://www.vmware.com/vmtn/resources/esx_resources.html

- Installing and Configuring NTP on VMware ESX Server

  `http://www.vmware.com/support/kb/enduser/std_adp.php?p_faqid=1339`
- *Timekeeping in VMware Virtual Machines*

  `http://www.vmware.com/pdf/vmware_timekeeping.pdf`
- Setting Maximum Queue Depth for the QLogic qla2x00 Adapter

  `http://www.vmware.com/support/kb/enduser/std_adp.php?p_faqid=1267`
- Setting the Maximum Outstanding Disk Requests per Virtual Machine

  `http://www.vmware.com/support/kb/enduser/std_adp.php?p_faqid=1268`
- *ESX Server 2 Best Practices*

  `http://www.vmware.com/pdf/esx2_best_practices.pdf`

Microsoft Web sites

Here is a list of available resources for Microsoft Virtual Server;

- Microsoft Virtual Server Site

  `http://www.microsoft.com/virtualserver`
- Windows Hardware & Driver Central (WHDC)

  `http://www.microsoft.com/whdc`
- Technical Communities

  `http://www.microsoft.com/communities/products`
- Microsoft Public Newsgroups

  `http://www.microsoft.com/communities/newsgroups`
- Technical Chats and Webcasts

  `http://www.microsoft.com/communities/chats`
  `http://www.microsoft.com/webcasts`
- Microsoft Blogs

  `http://www.microsoft.com/communities/blogs`
- Virtual Server 2005 performance tips

  `http://support.microsoft.com/default.aspx?scid=kb;en-us;903748`
- Virtual Server 2005 R2 Technical Overview

  `http://www.microsoft.com/windowsserversystem/virtualserver/overview/vs2005tech.mspx`

- *Virtual Server 2005 Administrator's Guide*

  http://www.microsoft.com/technet/prodtechnol/virtualserver/2005/proddocs

- Virtual Server Host Clustering Step-by-Step Guide for Virtual Server 2005 R2

  http://www.microsoft.com/downloads/details.aspx?FamilyID=09cc042b-154f-4eba-a548-89282d6eb1b3&displaylang=en

- Linux Guest Support for Virtual Server 2005 R2

  http://www.microsoft.com/windowsserversystem/virtualserver/evaluation/linuxguestsupport

- Virtual Server Downloads

  http://www.microsoft.com/windowsserversystem/virtualserver/downloads

- Quick Start Guide for Server Clusters

  http://go.microsoft.com/fwlink/?LinkId=55162

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

- IBM Support and downloads

  ibm.com/support

- IBM Global Services

  ibm.com/services

Index

IBM

Redbooks

Virtualization on the IBM System x3950 Server

(0.5" spine)
0.475"<->0.875"
250 <-> 459 pages

Virtualization on the IBM System x3950 Server

Redbooks

Covers VMware ESX Server 2.5.2 & 3.0 and Microsoft Virtual Server R2

Describes how the x3950 is an ideal solution for virtualization

Includes IBM Director's Virtual Machine Manager

Virtualization is becoming more and more a key technology enabler to streamline and better operate data centers. In it's simplest form, virtualization refers to the capability of being able to run multiple OS instances, such as Linux® and Windows, on a physical server.

Usually the concept of virtualization is associated with high-end servers, such as the IBM System x3950, that are able to support and consolidate multiple heterogeneous software environments. The System x3950 is a highly scalable x86 platform capable of supporting up to 32 processors and 512 GB of memory and is aimed at customers that want to consolidate data centers.

Between the server hardware and the operating systems that will run the applications is a virtualization layer of software that manages the entire system. The two main products in this field are VMware ESX Server and Microsoft Virtual Server.

This IBM Redbook discusses the technology behind virtualization, x3950 technology, and the two virtualization software products. We also discuss how to manage the solution properly as though they all were a pool of resources with Virtual Machine Manager, a unique and consistent management interface.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7190-00 ISBN 0738497096