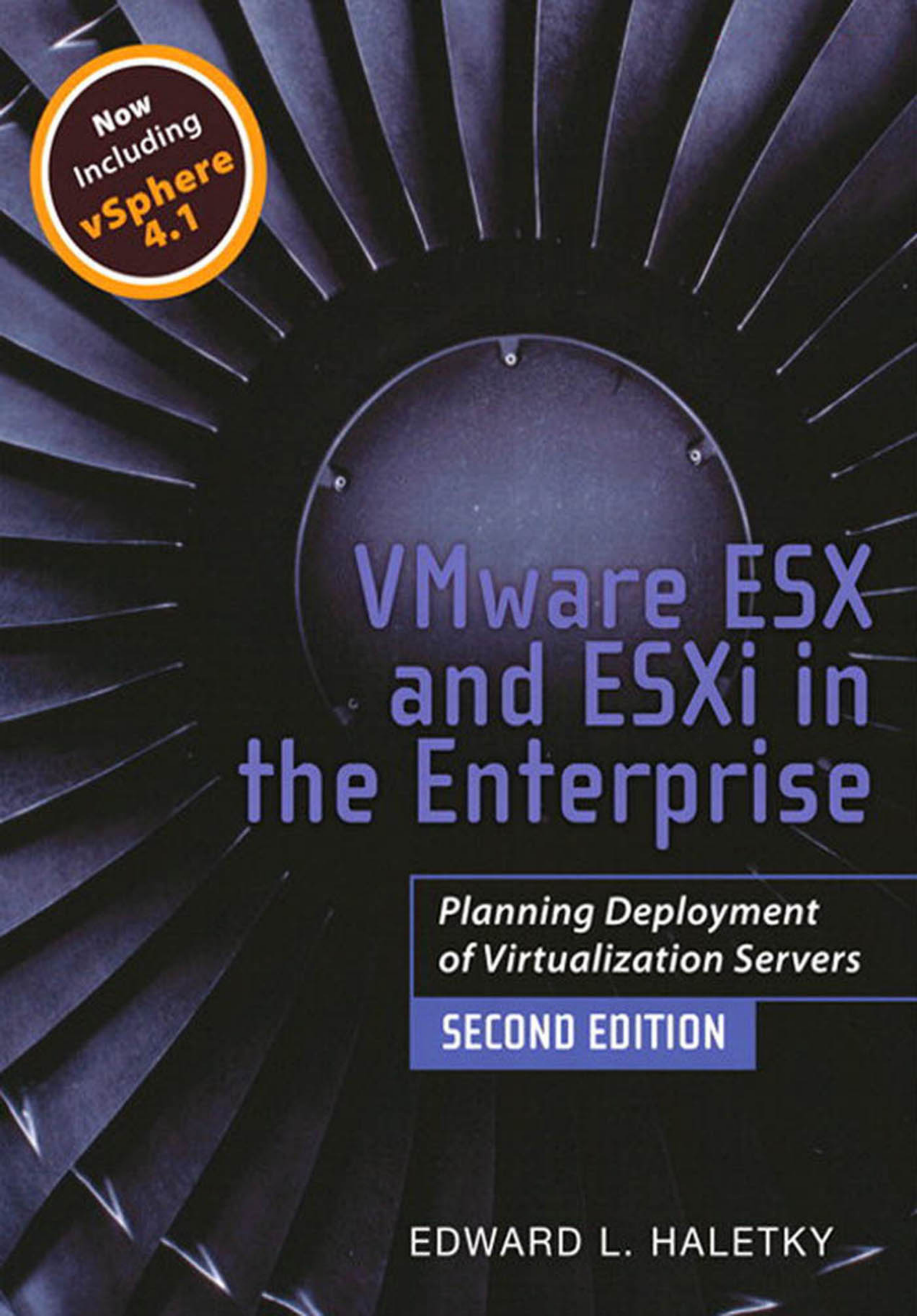


Now Including vSphere 4.1

VMware ESX and ESXi in the Enterprise

Planning Deployment of Virtualization Servers

SECOND EDITION

EDWARD L. HALETKY

VMware ESX and ESXi in the Enterprise

This page intentionally left blank

VMware ESX and ESXi in the Enterprise

Planning Deployment of Virtualization Servers

Edward L. Haletky

Pearson

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Cape Town • Sydney • Tokyo • Singapore • Mexico City

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The author and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U.S. Corporate and Government Sales
(800) 382-3419
corpsales@pearsontechgroup.com

For sales outside the United States, please contact:

International Sales
international@pearson.com

Visit us on the Web: informit.com/aw

Library of Congress Cataloging-in-Publication Data:

Haletky, Edward.

Vmware ESX and ESXI in the enterprise : planning deployment of virtualization servers / Edward Haletky. -- 2nd ed.

p. cm.

ISBN 978-0-13-705897-6 (pbk. : alk. paper) 1. Virtual computer systems. 2. Virtual computer systems--Security measures. 3. VMware. 4. Operating systems (Computers) I. Title.

QA76.9.V5H35 2010

006.8--dc22

2010042916

Pearson Education, Inc.
Rights and Contracts Department
501 Boylston Street, Suite 900
Boston, MA 02116
Fax (617) 671-3447

ISBN-13: 978-0-137-05897-6
ISBN-10: 0-137-05897-7

Text printed in the United States on recycled paper at Edwards Brothers in Ann Arbor, Michigan.

First printing January 2011

Editor-in-Chief
Mark Taub

Executive Editor
Chris Guzikowski

Senior Development Editor
Chris Zahn

Managing Editor
Kristy Hart

Project Editor
Jovana San Nicolas-Shirley

Copy Editor
Barbara Hacha

Indexer
Tim Wright

Proofreaders
Michael Henry
Sheri Cain

Publishing Coordinator
Raina Chrobak

Cover Designer
Chuti Prasertsith

Compositor
Gloria Schurick

To my mother, who always told me to read to my walls.

This page intentionally left blank

Contents

Preface

How often have you heard this kind of marketing hype around the use of VMware vSphere 4?

The latest version of ESX does everything for you!

Virtualize Everything!

It is cloud ready!

VMware ESX and ESXi, specifically the latest incarnation, VMware vSphere 4, does offer amazing functionality with virtualization: fault tolerance, dynamic resource load balancing, better virtual machine hardware, virtual networking, and failover. However, you still need to hire a consultant to share the mysteries of choosing hardware, good candidates for virtualization, choosing installation methods, installing, configuring, using, and even migrating machines. It is time for a reference that goes over all this information in simple language and in detail so that readers with different backgrounds can begin to use this extremely powerful tool.

Therefore, this book explains and comments on VMware ESX and ESXi versions 3.5.x and 4.x. I have endeavored to put together a "soup to nuts" description of the best practices for ESX and ESXi that can also be applied in general to the other tools available in the Virtual Infrastructure family inside and outside of VMware. To this end, I use real-world examples wherever possible and do not limit the discussions to only those products developed by VMware, but instead expand the discussion to virtualization tools developed by Quest, Veeam, HyTrust, and other third parties. I have endeavored to present all the methods available to achieve best practices, including the use of graphical and command-line tools.

Important Note

Although VMware has stated that the command-line is disappearing, the commands we will discuss exist in their VMware Management Appliance (vMA), which provides similar functionality of the service console. In essence, most of the command-line tools are still useful and are generally necessary when you have to debug an ESX or ESXi host. Required knowledge of these tools does not disappear with the service console.

As you read, keep in mind the big picture that virtualization provides: better utilization of hardware and resource sharing. In many ways, virtualization takes us back to the days of yore when developers had to do more with a lot less than we have available now. Remember the Commodore 64 and its predecessors, where we thought 64KB of memory was huge? Now we are back in a realm where we have to make do with fewer resources than perhaps desired. By keeping the big picture in mind, we can make the necessary choices that create a strong and viable virtual environment. Because we are doing more with less, this thought must be in the back of our mind as we move forward; it helps to explain many of the concerns raised within this tome.

As you will discover, I believe that you need to acquire quite a bit of knowledge and make numerous decisions before you even insert a CD-ROM to begin the installation. How these questions are answered will guide the installation, because you need to first understand the capabilities and limitations of the ESX or ESXi environment and the application mix to be placed in the environment. Keeping in mind the big picture and your application mix is a good idea as you read through each chapter of this book. Throughout this book we will refer to ESX as the combination of VMware ESX and VMware ESXi products.

Who Should Read This Book?

This book delves into many aspects of virtualization and is designed for the beginning administrator as well as the advanced administrator.

How Is This Book Organized?

Here is a listing, in brief, of what each chapter brings to the table.

Chapter 1: System Considerations

By endeavoring to bring you "soup to nuts" coverage, we start at the beginning of all projects: the requirements. These requirements will quickly move into discussions of hardware and capabilities of hardware required by ESX, as is often the case when I talk to customers. This section is critical, because understanding your hardware limitations and capabilities will point you in a direction that you can take to design your virtual datacenter and infrastructure. As a simple example, consider whether you will need to run 23 or 123 virtual machines on a set of blades. Understanding hardware capabilities will let you pick and choose the appropriate blades for your use and how many blades should make up the set. In addition, understanding your storage and virtual machine (VM) requirements can lead you down different paths for management, configuration, and installation. Checklists that lead to each chapter come out of this discussion. In particular, look for discussions on cache capabilities, the best practice for networking, mutual exclusiveness when dealing with storage area networks (SANs), hardware requirements for backup and disaster recovery, and a checklist when comparing hardware. This chapter is a good place to start when you need to find out where else in the book to go look for concept coverage.

Chapter 2: Version Comparison

Before we proceed down the installation paths and into further discussion, best practices, and explorations into ESX, we need to discuss the differences between ESX version 3.5.x and ESX version 4.x. This chapter opens with a broad stroke of the brush and clearly states that they *are* different. Okay, everyone knows that, but the chapter then delves into the major and minor differences that are highlighted in further chapters of the book. This chapter creates another guide to the book similar to the hardware guide that will lead you down different paths as you review the differences. The chapter covers hypervisor, driver, installation, VM, licensing, and management differences. After these are clearly laid out and explained, the details are left to the individual chapters that follow. Why is this not before the hardware chapter? Because hardware may not change, but the software running on it has, with a possible upgrade to ESX or ESXi 4, so this chapter treats the hardware as relatively static when compared to the major differences between ESX/ESXi 4 and ESX/ESXi 3.5.

Chapter 3: Installation

After delving into hardware considerations and ESX version differences, we head down the installation path, but before this happens, another checklist helps us to best plan the installation. Just doing an install will get ESX running for perhaps a test environment, but the best practices will fall out from planning your installation. You would not take off in a plane without running down the preflight checklist. ESX is very similar, and it is easy to get into trouble. For example, I had one customer who decided on an installation without first understanding the functionality required for clustering VMs together. This need to cluster the machines led to a major change and resulted in the reinstallation of all ESX servers in many locations. A little planning would have alleviated all the rework. The goal is to make the readers aware of these gotchas before they bite. After a review of planning, the chapter moves on to various installations of ESX and ESXi with a discussion on where paths diverge and why they would. For example, installing boot from SAN is quite different from a simple installation, at least in the setup, and because of this there is a discussion of the setup of the hardware prior to installation for each installation path. When the installations are completed, there are post-configuration and special considerations when using different SANs or multiple SANs. Limitations on VMFS with respect to sizing a LUN, spanning a LUN, and even the choice of a standard disk size could be a major concern. This chapter even delves into possible vendor and Linux software that could be added after ESX is fully installed. Also, this chapter suggests noting the divergent paths so that you can better install and configure ESX. We even discuss any additional software requirements for your virtual environment.

This chapter is about planning your installation, providing the 20 or so steps required for installation, with only one of these steps being the actual installation procedure. There is more to planning your installation than the actual installation process.

Chapter 4: Auditing and Monitoring

Because the preceding chapter discussed additional software, it is now time to discuss even more software to install that aids in the auditing and monitoring of ESX. There is nothing like having to read through several thousands of lines of errors just to determine when a problem started. Using good monitoring tools will simplify this task and even enable better software support. That is indeed a bonus! Yet knowing when a problem occurred is only part of monitoring and auditing; you also need to know who did the deed and where they did it, and hopefully why. This leads to auditing. More and more government intervention (Sarbanes-Oxley) requires better auditing of what is happening and when. This chapter launches into automating this as much as possible. Why would I need to sit and read log files when the simple application can e-mail me when there is a problem? How do I get these tools to page me or even self-repair? I suggest you take special note of how these concepts, tools, and implementations fit with your overall auditing and monitoring requirements.

Chapter 5: Storage with ESX

There are many issues dealing with storage within ESX. Some are simple, such as “Is my storage device supported?” and “Why not?” Others are more complex, such as “Will this storage device, switch, or Fibre Channel host bus adapter provide the functionality and performance I desire?" Because SAN and NAS devices are generally required to share VMs between ESX hosts, we discuss them in depth. This chapter lets you in on the not-so-good and the good things about each SAN and NAS, as well as the best practices for use, support, and configuration. With storage devices, there is good, bad, and the downright ugly. For example, if you do not have the proper firmware version on some storage devices, things can get ugly very quickly! Although the chapter does not discuss the configuration of your SAN or NAS for use outside of ESX, it does discuss presentation in general terms and how to get the most out of hardware and, to a certain extent, software multipath capabilities. This chapter suggests you pay close attention to how SAN and NAS devices interoperate with ESX. We will also look at some real-world customer issues with storage, such as growing virtual machine file systems, changing storage settings for best performance, load balancing, aggregation, and failover.

Chapter 6: Effects on Operations

Before proceeding to the other aspects of ESX, including the creation of a VM, it is important to review some operational constraints associated with the management of ESX and the running of VMs. Operation issues directly affect VMs. These issues are as basic as maintaining lists of IPs and netmasks, when to schedule services to run through the complexities imposed when using remote storage devices, and its impact on how and when certain virtualization tasks can take place.

Chapter 7: Networking

This chapter discusses the networking possibilities within ESX and the requirements placed on the external environment if any. A good example is mentioned under the hardware discussion, where we discuss hardware redundancy with respect to networking. In ESX terms, this discussion is all about network interface card (NIC) teaming, or in more general terms, the bonding of multiple NICs into one bigger pipe for the purpose of increasing bandwidth and failover. However, the checklist is not limited to the hardware but also includes the application of best practices for the creation of various virtual switches (vSwitches) within ESX, such as the Distributed Virtual Switch, the standard virtual switch, and the Cisco Nexus 1000V. In addition we will look at best practices for what network interfaces are virtualized, and when to use one over the other. The flexibility of networking inside ESX implies that the system and network administrators also have to be flexible, because the best practices dictated by a network switch company may lead to major performance problems when applied to ESX. The possible exception is the usage of the Cisco 1000V virtual switch. Out of this chapter comes a list of changes that may need to be applied to the networking infrastructure, with the necessary data to back up these practices so that discussions with network administrators do not lead toward one-sided conversations. Using real-world examples, this chapter runs through a series of procedures that can be applied to common problems that occur when networking within ESX.

This chapter also outlines the latest thoughts on virtual network security and concepts that include converged network adapters, other higher bandwidth solutions, and their use within the virtual environment. As such, we deep dive into the virtual networking stack within an ESX host.

Chapters 8 and 9: Configuring ESX from a Host Connection and Configuring ESX from a Virtual Center or Host

These chapters tie it all together; we have installed, configured, and attached storage to ESX. Now what? We need to manage ESX. There are five ways to manage ESX: the use of the web-based webAccess; the use of vCenter (VC), with its .NET client; the use of the remote CLI, which is mostly a collection of VI SDK applications; the use of the VI SDK; and the use of the command-line interface (CLI). These chapters delve into configuration and use of these interfaces. Out of these chapters will come tools that can be used as part of a scripted installation of ESX.

Chapter 10: Virtual Machines

This chapter goes into the creation, modification, and management of your virtual machines. In essence, the chapter discusses everything you need to know before you start installing VMs, specifically what makes up a VM. Then it is possible to launch into installation of VMs using all the standard interfaces. We install Windows, Linux, and NetWare VMs, pointing out where things diverge on the creation of a VM and what has to be done post install. This chapter looks at specific solutions to VM problems posed to me by customers: the use of eDirectory, private labs, firewalls, clusters, growing Virtual Machine Disks, and other customer issues. This chapter is an opportunity to see how VMs are created and how VMs differ from one another and why. Also, the solutions shown are those from real-world customers; they should guide you down your installation paths.

Chapter 11: Dynamic Resource Load Balancing

With vSphere, Dynamic Resource Load Balancing (DRLB) is very close to being here now. As we have seen in Chapter 10, virtual machines now contain capabilities to hot add/remove memory and CPUs, as well as the capability to affect the performance of egress and ingress network and storage traffic. ESX v4.1 introduces even newer concepts of Storage IO Control and Network IO Control. Tie these new functions with Dynamic Resource Scheduling, Fault-Tolerance, and Resource management and we now have a working model for DRLB that is more than just Dynamic Resource Scheduling. This chapter shows you the best practices for the application of all the ESX clustering techniques technologies and how they enhance your virtual environment. We also discuss how to apply alarms to various monitoring tools to give you a heads up when something needs to happen either by hand or has happened dynamically. I suggest paying close attention to the makeup of DLRB to understand the limitations of all the tools.

Chapter 12: Disaster Recovery, Business Continuity, and Backup

A subset of DLRB can apply to Disaster Recovery (DR). DR is a huge subject, so it is limited to just ESX and its environment that lends itself well to redundancy, and in so doing aids in DR planning. But, before you plan, you need to understand the limitations of the technology and tools. DR planning on ESX is not more difficult than a plan for a single physical machine. The use of a VM actually makes things easier if the VM is set up properly. A key component of DR is the making of safe, secure, and proper backups of the VMs and system. What to back up and when is a critical concern that fits into your current backup directives, which may not apply directly to ESX and which could be made faster. The chapter presents several real-world examples around backup and DR, including the use of redundant systems, how this is affected by ESX and VM clusters, the use of locally attached tape, the use of network storage, and some helpful scripts to make it all work. In addition, this chapter discusses some third-party tools to make your backup and restoration tasks simpler. The key to DR is a good plan, and the checklist in this chapter will aid in developing a plan that encompasses ESX and can be applied to all the vSphere and virtual infrastructure products. Some solutions require more hardware (spare disks, perhaps other SANs), more software (Veeam Backup, Quest's vRanger, Power Management, and so on).

Epilogue: The Future of the Virtual Environment

After all this, the book concludes with a discussion of the future of virtualization.

References

This element suggests possible further reading.

Reading

Please sit down in your favorite comfy chair, with a cup of your favorite hot drink, and prepare to enjoy the chapters in this book. Read it from cover to cover, or use as it a reference. The best practices of ESX sprinkled throughout the book will entice and enlighten, and spark further conversation and possibly well-considered changes to your current environments.

Acknowledgments

I would like to acknowledge my reviewers: Pat and Ken; they provided great feedback. I would like to also thank Bob, once a manager, who was the person who started me on this journey by asking one day, “Have you ever heard of this VMware stuff?” I had. I would also like to acknowledge my editors for putting up with my writing style.

This book is the result of many a discussion I had with customers and those within the virtualization community who have given me a technical home within the ever changing world of virtualization.

This edition of the book would not have happened without the support of my wife and family, who understood my need to work long hours writing.

Thank you one and all.

About the Author

Edward L. Haletky is the author of *VMware vSphere and Virtual Infrastructure Security: Securing the Virtual Environment* as well as the first edition of this book, *VMware ESX Server in the Enterprise: Planning and Securing Virtualization Servers*. Edward owns AstroArch Consulting, Inc., providing virtualization, security, network consulting, and development, and The Virtualization Practice, where he is also an analyst. Edward is the moderator and host of the Virtualization Security Podcast, as well as a guru and moderator for the VMware Communities Forums, providing answers to security and configuration questions. Edward is working on new books on virtualization.

Chapter 1

System Considerations

At VMworld 2009 in San Francisco, VMware presented to the world the VMworld Data Center (see Figure 1.1). There existed within this conference data center close to 40,000 virtual machines (VMs) running within 512 Cisco Unified Computing System (UCS) blades within 64 USC chassis. Included in this data center were eight racks of disks, as well as several racks of HP blades and Dell 1U servers, all connected to a Cisco Nexus 7000 switch. Granted, this design was clearly to show off UCS, but it showed that with only 32 racks of servers that it is possible to run up to 40,000 VMs.

Figure 1.1 *Where the Virtual Infrastructure touches the physical world*

The massive example at VMworld 2009 showed us all what is possible, but how do you get there? The first consideration is the design and architecture of the VMware vSphere$^{\text{TM}}$ environment. This depends on quite a few things, ranging from the types of applications and operating systems to virtualize, to how many physical machines are desired to virtualize, to determining on what hardware to place the virtual environments. Quite quickly, any discussion about the virtual infrastructure soon evolves to a discussion of the hardware to use in the environment. Experience shows that before designing a virtual datacenter, it's important to understand what makes a good virtual machine host and the limitations of current hardware platforms. In this chapter, customer examples illustrate various architectures based on limitations and desired results. These examples are not exhaustive, just a good introduction to understanding the impact of various hardware choices on the design of the virtual infrastructure. An understanding of potential hardware use will increase the chance of virtualization success. The architecture potentially derived from this understanding will benefit not just a single VMware vSphere™ ESX host, but also the tens to thousands that may be deployed throughout a single or multiple datacenters. Therefore, the goal here is to develop a basis for enterprisewide VMware vSphere™ ESX host deployment. The first step is to understand the hardware involved.

For example, a customer wanted a 40:1 compression ratio for virtualization of their physical machines. However, they also had networking goals to compress their network requirements. At the same time, the customer was limited by what hardware they could use. Going just by the hardware specifications and the limits within VMware vSphere™, the customer's hardware could do what was required, so the customer proceeded down that path. However, what the specification and limits state is not necessarily the best practice for VMware vSphere™, which led to quite a bit of hardship as the customer worked through the issues with its chosen environment. The customer could have alleviated certain hardships early on with a better understanding of the impact of VMware vSphere™ on the various pieces of hardware and that hardware's impact on VMware vSphere™ ESX v4 (ESXi v4) or VMware Virtual Infrastructure ESX v3 (ESXi v3). (Whereas most, if not all, of the diagrams and notes use Hewlett-Packard hardware, these are just examples; similar hardware is available from Dell, IBM, Sun, Cisco, and many other vendors.)

Basic Hardware Considerations

An understanding of basic hardware aspects and their impact on ESX v4 can greatly increase your chances of virtualization success. To begin, let's look at the components that make up modern systems.

When designing for the enterprise, one of the key considerations is the processor to use: specifically the type, cache available, and memory configurations. All these factors affect how ESX works in major ways. The wrong choices may make the system seem sluggish and will reduce the number of virtual machines that can run, so it is best to pay close attention to the processor and system architecture when designing the virtual environment.

Before picking any hardware, always refer to the VMware Hardware Compatibility Lists (HCLs), which you can find as a searchable database from which you can export PDFs for your specific hardware. This is located at www.vmware.com/support/pubs/vi_pubs.html.

Best Practice

Never purchase or reuse hardware unless you have first verified it exists on the VMware Hardware Compatibility Lists.

Although it is always possible to try to use commodity hardware that is not within the VMware hardware compatibility database, this could lead to a critical system that may not be in a supportable form. VMware support will do the best it can, but may end up pointing to the HCL and providing only advisory support and no real troubleshooting. To ensure this is never an issue, it is best to purchase only equipment VMware has blessed via the HCL database. Some claim that commodity hardware is fine for a lab or test environment; however, I am a firm believer that the best way to test something is 12 inches to 1 foot; in other words, use exactly what you have in production and not something you do not have—otherwise, your test could be faulty. Therefore, always stick to the hardware listed within the HCL.

Before we look at all the components of modern systems, we need to examine the current features of the ESX or ESXi systems. Without an understanding of these features at a high level, you will not be able to properly understand the impact the hardware has on the features and the impact the features have on choosing hardware.

Feature Considerations

Several features that constitute VMware vSphere have an impact on the hardware you will use. In later chapters, we will look at these in detail, but they are mentioned here so you have some basis for understanding the rest of the discussions within this chapter.

High Availability (HA)

VMware HA detects when a host or individual VM fails. Failed individual VMs are restarted on the same host. Yet if a host fails, VMware HA will by default boot the failed host's VMs on another running host. This is the most common use of a VMware Cluster, and it protects against unexpected node failures. No major hardware considerations exist for the use of HA, except that there should be enough CPU and memory to start the virtual machines. Finally, to have network connectivity, there needs to be the proper number of portgroups with the appropriate labels.

vMotion

vMotion enables the movement of a running VM from host to host by using a specialized network connection. vMotion creates a second running VM on the target host, hooks this VM up to the existing disks, and finally momentarily freezes a VM while it copies the memory and register footprint of the VM from host to host. Afterward, the VM on the old host is shut down cleanly, and the new one will start where the newly copied registers say to start. This often requires that the CPUs between hosts be of the same family at the very least.

Storage vMotion

Storage vMotion enables the movement of a running VM from datastore to datastore that is accessible via the VMware vSphere management appliance (ESXi) or service console (ESX). The datastore can be any NFS Server or local disk, disk array, remote Fibre Channel SAN, iSCSI Server, or remote disk array employing a SAN-style controller on which there exists the virtual machine file system (VMFS) developed by VMware.

Dynamic Resource Scheduling (DRS)

VMware DRS is another part of a VMware Cluster that will alleviate CPU and memory contention on your hosts by automatically vMotioning VMs between nodes within a cluster. If there is contention for CPU and memory resources on one node, any VM can automatically be moved to another underutilized node using vMotion. This often requires that the CPUs between hosts be of the same family at the very least.

Distributed Power Management (DPM)

VMware DPM will enable nodes within a VMware Cluster to evacuate their VMs (using vMotion) to other hosts and power down the evacuated host during off hours. Then during peak hours, the standby hosts can be powered on and again become active members of the VMware cluster when they are needed. DPM requires Wake on LAN (WoL) or IMPI functionality on the VMware ESX service console pNIC (VMware ESXi management pNIC) in order to be used; or it requires the use of IPMI or an HP ILO device within the host. WoL is the least desirable method to implement DPM. DPM is a feature of DRS.

Enhanced vMotion Capability (EVC)

VMware EVC ties into the Intel FlexMigration and AMD-V Extended Migration capabilities to present to the VMware Cluster members a common CPU feature set. Each CPU in use on a system contains a set of enhanced features; Intel-VT is one of these. In addition, there are instructions available to one chipset that may be interpreted differently on another chipset. For vMotion to work, these feature sets must match. To do this, there is a per VM set of CPU masks that can be set to match up feature sets between disparate CPUs and chipsets. EVC does this at the host level, instead of the per VM level. Unfortunately, EVC will work only between Intel CPUs that support Intel Flex Migration or between

AMD CPUs that support Extended Migration. You cannot use EVC to move VMs between the AMD and Intel families of processors. EVC requires either the No eXecute (NX) or eXecute Disable (XD) flags to be set within the CPU, as well as Intel-VT or AMD RVI to be enabled.

Virtual SMP (vSMP)

VMware vSMP enables a VM to have more than one virtual CPU so as to make it possible to run Symmetric Multiprocessing (SMP) applications that are either threaded or have many processes (if the OS involved supports SMP).

Fault Tolerance (FT)

VMware Fault Tolerance creates a shadow copy of a VM in which the virtual CPUs are kept in lockstep with the master CPU employing the VMware vLock-Step functionality. VMware FT depends on the VM residing on storage that VMware ESX or ESXi hosts can access, as well as other restrictions on the type and components of the VM (for example, there is only support for one vCPU VM). When FT is in use, vMotion is not available.

Multipath Plug-In (MPP)

The VMware Multipath Plug-in enables a third-party storage company such as EMC, HP, Hitachi, and the like to add their own multipath driver into the VMware hypervisor kernel.

VMDirectPath

VMDirectPath bypasses the hypervisor and connects a VM directly to a physical NIC card, not a specific port on a physical NIC card. This implies that VMDirectPath takes ownership of an entire PCIe or mezzanine adapter regardless of port count.

Virtual Distributed Switch (vDS)

The VMware vDS provides a mechanism to manage virtual switches across all VMware vSphere hosts. vDS switches also have the capability to set up private VLANs using their built-in dvFilter capability. This is a limited capability port security mechanism. The implementation of vDS enabled the capability to add in third-party virtual switches, such as the Cisco Nexus 1000V. The enabling technology does not require the vDS to use third-party virtual switches.

Host Profiles

Host Profiles provide a single way to maintain a common profile or configuration across all VMware vSphere hosts within the virtual environment. In the case of the VMworld 2009 conference data center, host profiles enable one configuration to be used across all 512 UCS blades within the VMworld 2009 Data

Center. Host Profiles eliminate small spelling differences that could cause networking and other items to not work properly across all hosts.

Storage IO Control

Storage IO Control allows for storage QoS on block level storage request exiting the host using cluster-wide storage latency values.

Network IO Control

Network IO Control allows for QoS on egress from the ESX host instead of on entry to the VMs.

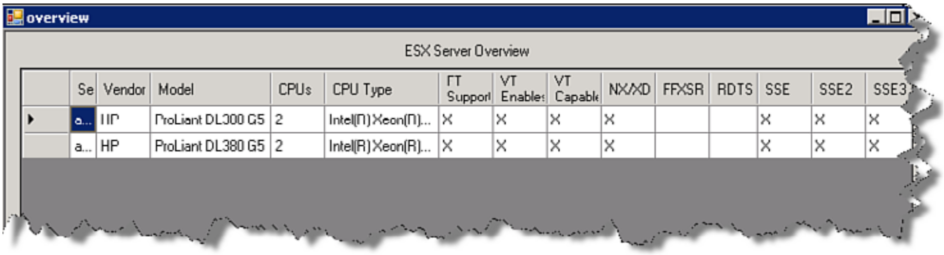
Load-Based Teaming

When VMs boot, they are associated with a physical NIC attached to a vSwitch. Load-Based Teaming allows for this association to be modified based on network latency.

Processor Considerations

Processor family, which is not a huge consideration in the scheme of things, is a consideration when picking multiple machines for the enterprise because the different types of processor architectures impact the availability of vSphere features. Specifically, mismatched processor types will prevent the use of vMotion DRS, EVC, and Fault Tolerance (FT). If everything works appropriately when vMotion is used or FT enabled, the VM does not notice anything but a slight hiccup that can be absorbed with no issues. However, because vMotion and FT copy the register and memory footprint from host to host, the processor architecture and chipset in use need to match. It is not possible without proper masking of processor features to vMotion from a Xeon to an AMD processor or from a dual-core processor to a single-core processor, but it is possible to go from a single-core to a dual-core processor. Nor is it possible to enable FT between Xeon and AMD processors for the same reason. If the VM to be moved is a 64-bit VM, the processors must match exactly because no method is available to mask processor features. Therefore, the processor architecture and chipset (or the instruction set) are extremely important, and because this can change from generation to generation of the machines, it is best to introduce two machines into the virtual enterprise at the same time to ensure that vMotion and FT actually work. When introducing new hardware into the mix of ESX hosts, test to confirm that vMotion and FT will work. VMware EVC has gone a long way to alleviate much of the needs of vMotion so that exact processor matches may no longer be required, but testing is still the best practice going forward.

VMware FT, however, adds a new monkey wrench into the selection of processors for each virtualization host because there is a strict limitation on which processors can be used with FT, and in general all machines should share the same processors and chipsets across all participating hosts. The availability of VMware FT can be determined by using the VMware SiteSurvey tool (www.vmware.com/download/shared_utilities.html). VMware SiteSurvey connects to your VMware vCenter Server and generates a report based on all nodes registered within a specific cluster. The SiteSurvey Tool, however, could give errors and not work if the build levels on your hosts within your cluster are different. In that case, use the VMware CPU Host Info Tool from www.run-virtual.com to retrieve the same information, as shown in Figure 1.2. Within this tool, the important features are FT Support, VT Enabled, VT Capable, and NX/XD status. All these should have an X in them. If they do not exist, you need to refer to VMware technical resources on Fault Tolerance (www.vmware.com/resources/techresources/1094). The best practices refer to all hosts within a given VMware cluster.



ESX Server Overview

	Se	Vendor	Model	CPUs	CPU Type	FT Suppor	VT Enable	VT Capabl	NX/XD	FFXSR	RDTS	SSE	SSE2	SSE3
▸	a...	HP	ProLiant DL380 G5	2	Intel(R)Xeon(R)...	X	X	X	X			X	X	X
	a...	HP	ProLiant DL380 G5	2	Intel(R)Xeon(R)...	X	X	X	X			X	X	X

Figure 1.2 *Output of Run-Virtual's CPU Host Info tool*

Best Practice

Standardize on a single processor and chipset architecture. If this is not possible because of the age of existing machines, test to ensure vMotion still works, or introduce hosts in pairs to guarantee successful vMotion and FT. Different firmware revisions can also affect vMotion and FT functionality.

Ensure that all processors support VMware FT capability.

Ensure that all the processor speed or stepping parameters in a system match, too.

Note that many companies support mismatched processor speeds or stepping in a system. ESX would really rather have all the processors at the same speed and stepping. In the case where the stepping for a processor is different, each vendor provides different instructions for processor placement. For example, Hewlett-Packard (HP) requires that the slowest processor be in the first processor slot and all the others in any remaining slots. To alleviate any type of issue, it is a best practice that the processor speeds or stepping match within the system.

Before proceeding to the next phase, a brief comment on eight-core (8C), six-core (6C), quad-core (QC), dual-core (DC), and single-core (SC) processors is warranted. ESX Server does not differentiate in its licensing scheme between 6C, QC, DC, and SC processors, so the difference between them becomes a matter of cost versus performance gain of the processors. However, with 8C and above you may need to change your ESX license level. The 8C processor will handle more VMs than a 6C, which can handle more VMs than a QC, which can handle more than a DC, which can handle more than an SC processor. If performance is the issue, 6C or QC is the way to go. Nevertheless, for now, the choice is a balance of cost versus performance. It is not recommended that any DC or SC processors be used for virtualization. These CPUs do not support the density of VMs required by today's datacenters. Granted, if that is all you have, it is still better than nothing. Even SMBs should stick to using quad-core CPUs if running more than two VMs.

Cache Considerations

Like matching processor architectures and chipsets, it is also important to match the L2 Cache between multiple hosts if you are going to use FT. A mismatch will not prevent vMotion from working. However, L2 Cache is most likely to be more important when it comes to performance because it controls how often main memory is accessed. The larger the L2 Cache, the better ESX host will run. Consider Figure 1.3 in terms of VMs being a complete process and the access path of memory. Although ESX tries to limit memory usage as much as possible through content-based page sharing and other techniques discussed later, even so the amount of L2 Cache plays a significant part in how VMs perform.

As more VMs are added to a host of similar operating system (OS) type and version, ESX will start to share memory pages between VMs; this is referred to as Transparent Page Sharing (TPS) or Content Based Page Sharing (CBPS). During idle moments, ESX will collapse identical 4KB (but not 8KB) pages of memory (as determined by a hash lookup then a bit by bit comparison) and leave pointers to original memory location within each VM's memory image. This method of overcommitting memory does not have any special processor requirements; during a vMotion or FT the VM has no idea this is taking place because it happens outside the VM and does not impact the guest OS directly. Let's look at Figure 1.3 again. When a processor needs to ask the system for memory, it first goes to the L1 Cache (up to a megabyte usually) and sees whether the memory region requested is already on the processor die. This action is extremely fast, and although different for most processors, we can assume it is an instruction or two (measured in nanoseconds). However, if the memory region is not in the L1 Cache, the next step is to go to the L2 Cache. L2 Cache is generally off the die, over an extremely fast channel (light arrow) usually running at processor speeds. Even so, accessing L2 Cache takes more time and instructions than L1 Cache access. If the memory region you desire is not in L2 Cache, it is possibly in L3 Cache (if one exists, dotted arrow) or in main memory (dashed arrow). L3 Cache or main memory takes an order of magnitude above processor speeds to access. Usually, a cache line is copied from main memory, which is the desired memory region and some of the adjacent data, to speed up future memory access. When we are dealing with nonuniform memory access (NUMA) architecture, which is the case with Intel Nahelem and AMD processors, there is yet another step to memory access. The memory necessary could be sitting on a processor board elsewhere in the system. The farther away it is, the slower the access time (darker lines), and this access over the CPU interconnect will add another order of magnitude to the memory access time.

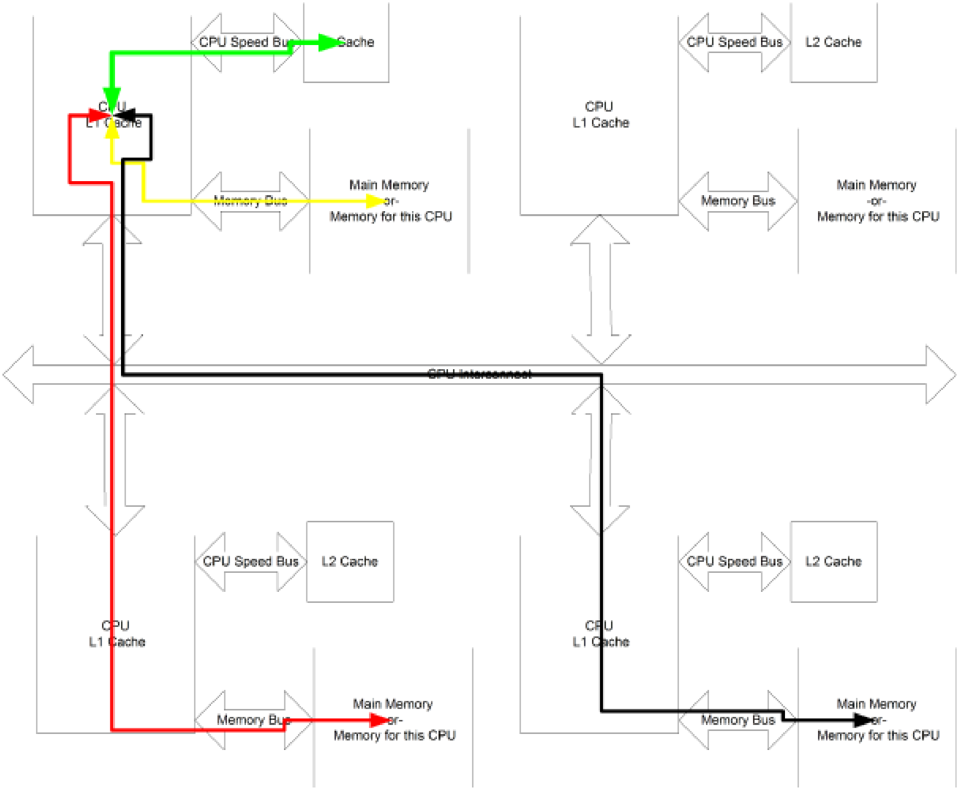


Figure 1.3 *Memory access paths*

What does this mean in real times? Assuming that we are using a 3.06GHz processor without L3 Cache, the times could be as follows:

- L1 Cache, one cycle (~0.33ns).
- L2 Cache, two cycles, the first one to get a cache miss from L1 Cache and another to access L2 Cache (~0.66ns), which runs at CPU speeds (light arrow).
- Main memory is running at 333MHz, which is an order of magnitude slower than L2 Cache (~3.0ns access time) (dashed arrow).
- Access to main memory on another processor board (NUMA) is an order of magnitude slower than accessing main memory on the same processor board (~30–45ns access time, depending on distance) (darker lines).

Now let's take the same calculation using L3 Cache:

- L1 Cache, one cycle (~0.33ns).
- L2 Cache, two cycles, the first one to get a cache miss from L1 Cache and another to access L2 Cache (~0.66ns), which runs at CPU speeds (light arrow).
- L3 Cache, two cycles, the first one to get a cache miss from L2 Cache and another to access L3 Cache (~0.66ns), which runs at CPU speeds (light arrow).
- Main memory is running at 333MHz, which is an order of magnitude slower than L3 Cache (~3.0ns access time) (dashed arrow).
- Access to main memory on another processor board (NUMA) is an order of magnitude slower than accessing main memory on the same processor board (~30–45ns access time, depending on distance) (darker lines).

This implies that large L2 and L3 Cache sizes will benefit the system more than small L2 and L3 Cache sizes: the larger the better. If the processor has access to larger chunks of contiguous memory, because the memory to be swapped in will be on the larger size, this will benefit the performance of the VMs. This discussion does not state that NUMA-based architectures are inherently slower than regular-style architectures, because most NUMA-based architectures running ESX host do not need to go out to other processor boards very often to gain access to memory. However, when using VMs making use of vSMP, it is possible that one CPU could be on an entirely different processor board within a NUMA architecture, and this could cause serious performance issues depending on whether quite a bit of data is being shared between the multiple threads and processes within the application. We will discuss this more in Chapter 11, "Dynamic Resource Load Balancing." One solution to this problem is to use CPU affinity settings to ensure that the vCPUs run on the same processor board. The other is to limit the number of vCPUs to what will fit within a single processor. In other words, for quad-core processors, you would use at most four vCPUs per VM.

Best Practice

Invest in the largest amount of L2 and L3 Cache available for your chosen architecture.

If using NUMA architectures, ensure that you do not use more vCPUs than there are cores per processor.

Memory Considerations

After L2 and L3 Cache comes the speed of the memory, as the preceding bulleted list suggests. Higher-speed memory is suggested, and lots of it! The amount of memory and the number of processors govern how many VMs can run simultaneously without overcommitting this vital resource. Obviously, there are trade-offs in the number of VMs and how you populate memory, but generally the best practice is high speed and a high quantity. Consider that the maximum number of vCPUs per core is 20 when using vSphere™. On a 4-QC processor box, that could be 320 single vCPU VMs. If each of these VMs is 1GB, we need 339GB of memory to run the VMs. Why 339GB? Because 339GB gives both the service console (SC) and the hypervisor up to 2GB of memory to run the VMs and accounts for the ~55MBs per GB of memory management overhead. Because 339GB of memory is a weird number for most computers these days, we would need to overcommit memory. When we start overcommitting memory in this way, the performance of ESX can degrade. In this case, it might be better to move to 348GB of memory instead. However, that same box with 8C processors can, theoretically, run up to 640 VMs, which implies that we take the VM load to the logical conclusion, and we are once more overcommitting memory.

Important Note

vSphere^{TM} can only run 320 VMs per host regardless of theoretical limits and only supports 512 vCPUs per host.

Even so, 20 VMs per processor is a theoretical limit, and it's hard to achieve. (It is not possible to run VMs with more vCPUs than available physical cores, but there is still a theoretical limit of 20 vCPUs per core.) Although 20 is the theoretical limit, 512 vCPUs is the maximum allowed per host, which implies that 16 vCPUs per core on an 8C four-processor box is not unreasonable. Remember that the vmkernel and SC (management appliance within ESXi) also use memory and need to be considered as part of any memory analysis.

Note that VMware ESX hosts have quite a few features to enable the amount of memory overcommit that will occur. The primary feature is Transparent Page Sharing or Content Based Page Sharing (CBPS). This mechanism collapses identical pages of memory used by any number of VMs down to just one memory page as an idle time process. If your ESX host runs VMs that use the same operating system and patch level, the gain from CBPS can be quite large—large enough to run at least one or maybe even two more VMs. The other prominent memory overcommit prevention tool is the virtual machine balloon driver. We will discuss both of these further in Chapter 11.

Best Practice

High-speed memory and lots of it! However, be aware of the possible trade-offs involved in choosing the highest-speed memory. More VMs may necessitate the use of slightly slower memory, depending on server manufacturer.

What is the recommended memory configuration? The proper choice for a size of a system depends on a balancing act of the four major elements—CPU, memory, disk, and network—of a virtualization host. This subject is covered when we cover VMs in detail, because it really pertains to this question; but the strong recommendation is to put in the maximum memory the hardware will support that is not above the memory limit set by ESX as one of the ways the system overcommits memory is to swap to disk, which can be helped by moving to SSD style disks, but this is still 100 times slower than memory. When swapping occurs, the entire system's performance will be impacted. However, redundancy needs to be considered with any implementation of ESX; it is therefore beneficial to cut down on the per-machine memory requirements to afford redundant systems. Although we theoretically could run 320 VMs (maximum allowed by VMware vSphere^{TM}) on a four-processor 8C box, other aspects of the server come into play that will limit the number of VMs. These aspects are disk and network IO, as well as VM CPU loads. It also depends on the need for local and remote redundancy for disaster recovery and business continuity, which are covered in Chapter 12, “Disaster Recovery, Business Continuity, and Backup.”

I/O Card Considerations

The next consideration when selecting your virtualization hosts is which I/O cards are supported. Unlike other operating systems, ESX has a finite list of supported I/O cards. There are limitations on the redundant array of inexpensive drives (RAID) controllers; Small Computer System Interface (SCSI) adapters for external devices including tape libraries; network interface cards (NICs); and Fibre Channel host bus adapters. Although the list changes frequently, it boils down to a few types of supported devices limited by the set of device drivers that are a part of ESX. Table 1.1 covers the devices and the associated drivers.

Table 1.1 *Devices and Drivers*

Device Type	Device Driver Vendor	Device Driver Name	Notes
Network	Broadcom	bnx2	NetXtreme II Gigabit
	Broadcom	bnx2x	NetXtreme II 5771x 10Gigabit
	Broadcom	tg3	
	3Com	3c90x	ESX v3 Only
	Intel	e1000e	PRO/1000
	Intel	e1000	PRO/1000
	Intel	e100	ESX v3 Only
	Intel	igb	Gigabit
	Intel	ixgbe	10 Gigabit PCIe
	Cisco	enic	10G/ESX v4 Only
	Qlogic	nx_nic	10G/ESX v4 Only
	Nvidia	forcedeth	
Fibre Channel	Emulex	lpfc820	Dual/Single ports
	Cisco	fnic	FCoE/ESX v4 Only
	Qlogic	qla2xxx	Dual/Single ports

continues

Table 1.1 *(Continued)*

Device Type	Device Driver Vendor	Device Driver Name	Notes
SCSI/SAS/SATA	Adaptec	aic79xx	Supported for External Devices
	Adaptec	adp94xx	Supported for External Devices
	Adaptec	aic7xxx	ESX v3 Only
	Intel	ata_piix	PATA/SATA
	Silicon Image	sata_sil	SATA
	Promise	sata_promise	SATA TX2/TX4
	ServerWorks	sata_svw	Frodo/Apple K2 SATA
	NVidia	sata_nv	SATA
	Vitesse	sata_vsc	SATA ESX v3 Only
	LSI Logic	megaraid_sas	SAS
	LSI Logic	mptscsi_2xx	ESX v3 Only
	LSI Logic	mptsas	SAS/ESX v4 Only
	LSI Logic	mptspi	LSI53C*/ESX v4 Only
Raid Array	HP	cciss	External SCSI is for Disk Arrays only
	Dell	aacraid	
	Dell	megaraid	
	IBM/Adaptec	ips	
	IBM/Adaptec	aacraid	
	Mylex	DAC960	ESX v3 Only
	LSI	megaraid	
iSCSI	Qlogic	qla4xxx	

If the driver in question supports a device, in most cases it will work in ESX. However, if the device requires a modern device driver, do not expect it to be part of ESX, because ESX by its very nature does not support the most current devices. ESX is designed to be stable, and that often precludes modern devices. For example, not all Serial Advanced Technology Attachment (SATA) devices are a part of ESX, and many drivers are dropped from support when you move to ESX v4, as shown in Table 1.1. Also noted in Table 1.1, various SCSI adapters have limitations. A key limitation is that an Adaptec non-RAID card is required for external tape drives or libraries, whereas any SCSI or RAID card is usable with external disk arrays.

Table 1.1 refers particularly to those devices that the vmkernel can access, and not necessarily the devices that the SC installs. There are quite a few devices for which the SC has a driver, but the host cannot use them. Two examples of this come to mind. The first are NICs (not listed in Table 1.1) that actually have a SC driver; Kingston or old Digital NICs fall into this category. For ESX to run, it needs at a minimum two NICs (yes, it is possible to use one NIC, but this is never a recommendation for production servers) and one supported storage device. One NIC is for the service console (management appliance for ESXi) and the other for the VMs. Although it is possible to share these so that only one NIC is required, VMware does not recommend this except in extreme cases (and it leads to possible performance and security issues). The best practice for ESX is to provide redundancy for everything so that all your VMs stay running even if network or a Fibre Channel path is lost. To do this, there needs to be some considerations around network and Fibre configurations and perhaps more I/O devices.

Best Practice Regarding I/O Cards

If the card you desire to use is *not* on the HCL, do not use it. The HCL is definitive from a support perspective. Although a vendor may produce a card and self-check it, if it is not on the HCL, VMware will not support the configuration.

Best Practice

The best practice is to have redundant NICs for each network trust zone in use for performance, security, and redundancy.

If adding more networks for use by the VMs, either use 802.1q VLAN tagging to run over the existing pair of NICs associated with the VMs or add a new pair of NICs for the VMs.

When using iSCSI with ESX or ESXi v3, the service console or management appliance must participate in the iSCSI network for CHAP authentication. For ESX or ESXi v4 this limitation no longer exists.

When using iSCSI, add at least another pair of NIC ports to provide performance and redundancy.

When using Network File System (NFS) via network-attached storage (NAS) with ESX or ESXi, add another pair of NIC ports to give performance and redundancy.

If you are using locally attached tape drives or libraries, use an Adaptec non-RAID SCSI adapter. No other adapter will work properly. However, the best practice for tape drives or libraries is to use a remote archive server.

With ESX v4, there is now support for Single Root I/O Virtualization (SR-IOV) devices (generally 10G Ethernet adapters) that make use of VMDirectPath. VMDirectPath requires Intel VT-d or AMD IOMMU support within the hardware. VMDirectPath may grant greater IO capability within a VM, but because it bypasses the hypervisor there is no built-in redundancy or security. The VM would need to provide this functionality, much like a normal physical server using 802.3ad.

iSCSI and NAS support is available, and iSCSI support differs distinctly from ESX v3 to ESX v4 because the need for the service console or management appliance to participate within the iSCSI network has been removed. iSCSI and NFS-based NAS are accessed using their own network connection assigned to the vmkernel, similar to the way vMotion and FT work or how a standard VMFS-3 is accessed via Fibre. Although NAS and iSCSI access can share bandwidth with other networks, keeping them separate could be better for performance. For ESX v3.x the iSCSI vmkernel device must share the subnet as the SC for authentication reasons, regardless of whether Challenge Handshake Authentication Protocol (CHAP) is enabled, although an NFS-based NAS would be on its own network. Chapter 8, "Configuring ESX from a Host Connection," discusses this new networking possibility in detail.

10Gb Ethernet

10Gb Ethernet is sweeping through data centers and ESX has kept up to date with this wave by supporting various 10Gb network adapters. Use of 10Gb has increased the use of VLANs within the virtual environment and is in the midst of redefining the standard trust zones associated with networking. We cover this in Chapter 7, "Networking." Use of 10Gb should be considered for high-performance networking requirements such as storage or virtual machine networks. It can also be used to combine networks; however, this depends on whether VLANs are used to segregate networks or if physical separation is required.

Converged Network Adapters

Converged Network Adapters (CNAs) are a set of networking adapters that combine FC SAN and networking features onto one device, thereby eliminating the need for excessive cabling within the virtual environment.

Disk Drive Space Considerations

The next item to discuss is what is required for drive space. In essence, the disk subsystem assigned to the system needs to be big enough to contain the SC for

ESX, the swap file for the SC, storage space for the per VM virtual swap files (used to overcommit memory in ESX), VM disk files, local ISO images, and backups of the Virtual Machine Disk Format (VMDK) files for Disaster Recovery reasons. If Fibre Channel, NFS, or iSCSI is available, you should offload the VM disk files to these systems. Putting temporary storage (SC swap) onto expensive SAN or iSCSI storage is not a best practice; the recommendation is that there be some form of local disk space to host the OS and the SC swap files. It is a requirement for vMotion, FT, and HA that the VM configuration and VMDK files live on the remote storage device. For FT and HA, the per VM vmkernel swap file should live on the remote storage device. However, for vMotion the per VM vmkernel swap file could live on local storage because vMotion copies the non-zero pages over to the target host during a vMotion. The minimal recommendation is roughly 12GB of available space in a RAID 1 or RAID 10 configuration for the operating system (ESX) and minimally 2GBs for ESXi and its necessary file systems. Use NFS for ISO files and other items as necessary, while using FC or iSCSI SAN storage for all VMDKs.

The most common recommendation for any VMFS that contains VMs is to use a RAID 5 configuration over as many spindles as possible for the best protection of data, while using RAID 1 for all high disk performance virtual machines. For the highest performance, you may want to consider solid state drives as well. For archival and low disk performance VMs, RAID 5 or Advanced Data Guard (ADG/RAID-6) should be used. Chapter 12, covers the disk configuration in much more detail as it investigates the needs of the local disk from a Disaster Recovery (DR) point of view. The general DR point of view is to have enough local space to run critical VMs from the host without the need for a SAN or iSCSI device.

Best Practice for Disk

Use RAID-1 on FC or iSCSI SAN Storage for those VMs with high disk IO requirements.

Use RAID-5 for average use VMs.

Use anything you want for low disk IO or archived VMs.

Use NFS for storage of ISO Images.

Have as much local disk necessary to hold the OS, local backups of critical VMs, and perhaps some local VMs.

Basic Hardware Considerations Summary

Table 1.2 conveniently summarizes the hardware considerations discussed in this section.

Table 1.2 *Best Practices for Hardware*

Item	ESX v3	ESX v4	Chapter to Visit for More Information
CPU	4C Intel VT/AMD RVI No eXecute/eXecute Disable	4C or greater Intel VT/AMD RVI No eXecute/eXecute Disable Intel VT-d for SRIOV	
Fibre Ports	Minimally Two 4Gbps	Minimally Two 4Gbps	Chapter 5
Network Ports	Two per Trust Zone depending on physical network from 4–10 ports	Two per Trust Zone depending on physical network from 4–12 ports	Chapter 8
Local disks	SCSI/SAS/SATA RAID Enough to keep a copy of the most important VMs Consider SSD	SCSI/SAS/SATA RAID Enough to keep a copy of the most important VMs Consider SSD	
iSCSI	Two 1GB network ports via vmkernel or iSCSI HBA	Two 1GB network ports via vmkernel or iSCSI HBA	Chapter 8
SAN	Enterprise class	Enterprise class	Chapter 5
Tape	Remote	Remote	Chapter 11
NFS-based NAS	Two 1GB network ports via vmkernel	Two 1GB network ports via vmkernel	Chapter 8
Memory	Up to max allowed	Up to max allowed	
Networks	Four to five Admin/iSCSI network VM network vMotion network NFS network iSCSI network	Five to six Admin network VM network vMotion network NFS network iSCSI network FT network	Chapter 8

Specific Hardware Considerations

Now we need to look at the hardware currently available and decide how to use it to meet the best practices listed previously. All hardware will have some issues to consider, and applying the comments from the first section of this chapter will help show the good, bad, and ugly about the possible hardware currently used as a virtual infrastructure node. The primary goal is to help the reader understand the necessary design choices when choosing various forms of hardware for an enterprise-level ESX host farm. Note that the number of VMs mentioned is based on an average machine that does not do very much network, disk, or other I/O and has average processor utilization. This number varies too much based on the utilization of the current infrastructure, and these numbers are a measure of what each server is capable of and are not intended as maximums or minimums. A proper analysis will yield the best use of your ESX hosts and is part of the design for any virtual infrastructure.

With modern systems that support quad-core and greater processors as well as increased memory density, memory and CPU are no longer limiting factors in virtualization. We are in the phase of hardware development where IO becomes an issue. However, this is fairly cyclic and as workloads increase once more, CPU may become an issue again. Therefore it is best to consider all aspects of your hardware and not just concentrate on any specific issue. Well-rounded systems that provide security, redundancy, and performance are the necessities.

Blade Server Systems

Because blade systems (see Figure 1.4) virtualize hardware, it is a logical choice for ESX, which further virtualizes a blade investment by running more servers on each blade. Although blades generally lack the capability to add in PCIe cards, they do support several mezzanine structures. Most modern blades can handle the port density necessary for high performance, redundant, and secure networking and storage connectivity. Even so, a limit exists to how many ports can be placed within a blade and how these are handled within the enclosure itself. The solution from some vendors' enclosures is to aggregate all networking to several 1G or 10G links to limit the number of cables in use. HP's Flex10 and Cisco Unified Computing System are two examples of this type of networking.

In addition to possible port density issues and issues with aggregation, issues could exist with local disk requirements. Whether you need local disk depends entirely on your current design; however, many companies use local disk as a mechanism to back up their most important virtual machines (see Chapter 12 for details regarding backup and Disaster Recovery) in case their remote iSCSI, NAS, or SAN devices fail. Some vendors, HP for example, have created storage blades that can be used with any number of compute blades within an enclosure.

Blades generally have several issues; port density issues are solved by the use of CNAs and other blade class aggregators (HP Flex10 and so on) as they provide a way to combine network and Fibre Channel into one adapter. However, given that blades have a dearth of expansion slots, these systems have some distinct limitations if you do not or cannot use CNAs. Other issues include a limited number of expansion ports, shared backplanes, which increase density but limit full redundancy, and limited PCI device support (usually to vendor-specific riser cards).

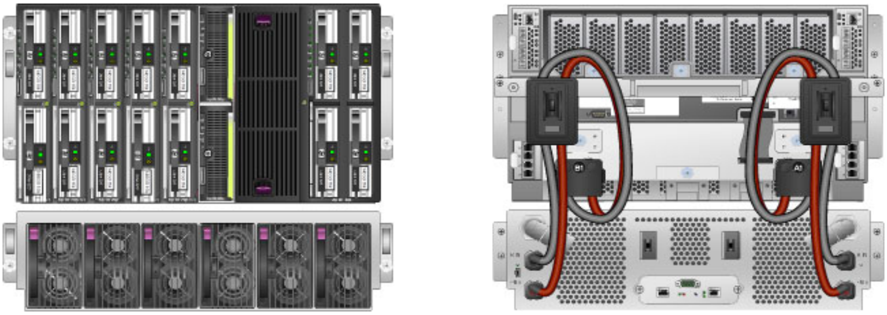


Figure 1.4 *Front and back of blade enclosure*

Best Practice with Blades

Pick blades that offer full NIC, Fibre redundancy, and sufficient local disk space.

1U Server Systems

The next device of interest is the 1U server (see Figure 1.5), which offers in most cases two onboard NICs and sometimes four, generally no onboard Fibre, perhaps two PCI slots, and perhaps two to four SCSI/SAS disks. This is perfect for adding a quad-port NIC and a dual-port Fibre controller; but if you need a SCSI card for a local tape device, which is sometimes necessary but never recommended, there is no chance to put one in unless there is a way to get more onboard NIC or Fibre ports. In addition to the need to add more hardware into these units, there is a chance that PCI card redundancy would be lost, too. Consider the HP DL360 G6 as a possible ESX host, which is a 1U device with up to eight SAS or SATA drives, two onboard NICs, and two PCIe expansion slots. In this case, we would want to add at least a quad-port NIC card to get to the six NICs that make up the best practice and gain more redundancy for ESX. In some cases, there is a SCSI port on the back of the device, so access to a disk array will increase space dramatically, yet often driver deficiencies affect its usage with tape devices.

1U servers can have up to two sockets with many cores and up to 128GBs of memory.

Figure 1.5 *1U server front and back*

In the case of SAN redundancy, if there were no mezzanine Fibre Channel adapter, the second PCI slot would host a dual- or quad-port Fibre Channel adapter, which would round out and fill all available slots. With the advent of quad-port NIC support, adding an additional pair of NIC ports for another network requires the replacement of the additional dual-port NIC with the new PCI card. There are some trade-offs when choosing this platform, just as there are for blades. The small number of available expansion slots limits the quantity of ports and adapters available for security, redundancy, and performance.

Best Practice for 1U Boxes

Pick a box that has on-board Fibre Channel adapters so that there are free slots for more network and any other necessary I/O cards. Also, choose large disk drives when possible. There should be at least two on-board network ports. Add quad-port network or dual-port 10Gbe and dual-port Fibre Channel cards as necessary to get port density.

2U Server Systems

The next server considered is the 2U server (see Figure 1.6), similar to the HP DL380. This type of server usually has two or four onboard Ethernet ports, perhaps two onboard Fibre Channel ports, and usually an external SCSI port for use with external drive arrays. In addition to all this, there are up to five PCIe slots, up to eight SAS or SATA disks, and possibly twice as much memory as a 1U machine. The extra PCIe slots add quite a bit of functionality; they can either host an Adaptec SCSI card to support a local tape drive or library, which is sometimes necessary but never recommended, or it can host more network (10Gbe) or storage capability. At the bare minimum, two to four more NIC ports are required and perhaps a dual- or quad-port Fibre Channel adapter if a pair of ports is not already in the server. Because this class of server can host eight SAS or SATA disks, they can be loaded up with more than 2TB of storage, which makes the 2U server an excellent stand-alone ESX host. Introduce six-core processors and this box has the power to run many VMs. This class of server makes an excellent virtualization host and generally provides the most for your dollar.

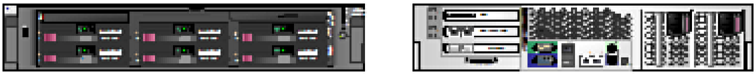


Figure 1.6 *Front and back of 2U server*

2U servers generally have up to two sockets with many cores but a limited amount of memory sockets. 128GBs of memory is not an uncommon maximum for 2U servers.

Pairing a 2U server with a small tape library to become an office in a box, which ships to a remote location, does not require a SAN or another form of remote storage because it has plenty of local disk space and the capability to connect to a disk array to provide even more storage.

Best Practice for 2U Servers

Pick a server that has at least two on-board NIC ports, two on-board Fibre Channel ports, plenty of disk, and as much memory as possible. Add a quad-port network card to gain port density and, if necessary, two single-port Fibre Channel adapters to add more redundancy.

Large Server-Class Systems

The next discussion combines multiple classes of servers (see Figure 1.7). The class combines the 4-, 8-, and 16-processor machines. Independent of the processor count, all these servers have many of the same hardware features. Generally, they have at least four SCSI/SAS/SATA drives and sometimes up to eight or sixteen for 8TBs of internal storage, at least six PCI slots, two onboard NICs, RAID memory, and very large memory footprints ranging from 32GB to 512GB. Some of these servers can even include 1TB of memory. The RAID memory is just one technology that allows for the replacement of various components while the machine is still running, which can alleviate hardware-based downtime unless it's one of the critical components. RAID memory is extremely nice to have, but it is just a fraction of the total memory in the server and does not count as available memory to the server. For example, it is possible to put a full 512GB of memory into an HP DL785, but the OS will see only 256GB of memory in mirrored mode, 504GB using an online spare, or the full 512GBs. The mirrored memory or online space, which comes into use only if there is a bad memory stick discovered by the hardware, alleviates crashes when DIMMs go bad. Historically, the larger machines have fewer disks than the 2U servers do, but it makes up for that by having an abundance of PCI buses and slots enabling multiple Fibre Channel adapters and dual-port NICs for the highest level of redundancy. Many of the larger system now can house more drives than the 2U servers offering up to 16TBs of disk. In these servers, the multiple Fibre Channel ports suggested by the general best practice would each be placed on different PCI buses, as would the NIC cards to get better performance and redundancy in PCI cards, SAN fabric, and networking. These types of servers can host a huge number of VMs, usually up to the maximum supported by ESX itself.

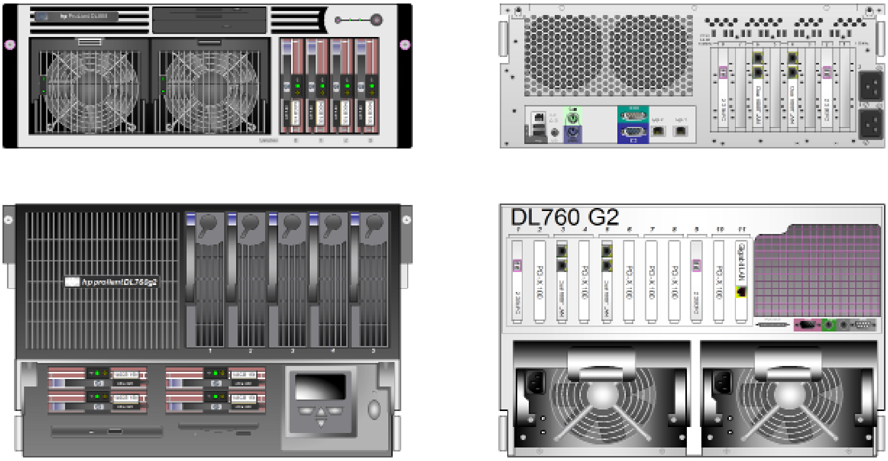


Figure 1.7 *Back and front of large server-class machines*

The Effects of External Storage

There are many different external storage devices, ranging from simple external drives, to disk arrays, shared disk arrays, active/passive SAN, active/active SAN, SCSI tape drives, to libraries and Fibre-attached tape libraries. The list is extensive, but we will be looking at the most common devices in use today and those most likely to be used in the future. We'll start with the simplest device and move on to the more complex devices. As we did with servers, this discussion points out the limitations or benefits in the technology so that all the facts are available when starting or modifying virtual infrastructure architecture.

For local disks, it is strongly recommended that you use SCSI/SAS RAID devices; although IDE is supported for running ESX, it does not have the capability to host a VMFS, so some form of external storage is required. Since ESX v3, there has been support for local SATA devices, but they share the same limitations as IDE. In some servers you can hand SATA drives off a SAS controller to gain SCSI-like functionality, but not the performance. In addition, if you are running any form of shared disk cluster, such as Microsoft Cluster servers across hosts, a local VMFS is required for the boot drives, yet remote storage is required for all shared volumes using raw disk maps. If remote storage is not available, the shared disk cluster will fail with major locking issues. We cover clusters in detail in Chapter 10, "Virtual Machines."

Best Practice for Local Disks

Use SCSI or SAS disks.

Outside of local disks, the external disk tray or disk array (see Figure 1.8) is a common attachment and usually does not require more hardware outside of the disk array and the proper SCSI cable. However, like standalone servers, the local disk array does not enable the use of vMotion to hot migrate a VM. However, when vMotion is not required, this is a simple way to get more storage attached to a server. If the disk array is using a SATA controller, it is probably better to go to a SAS controller instead, because you gain all the supported benefits of using SCSI/SAS controllers. Unfortunately, you do not gain the performance of SCSI/SAS when using SATA drives.

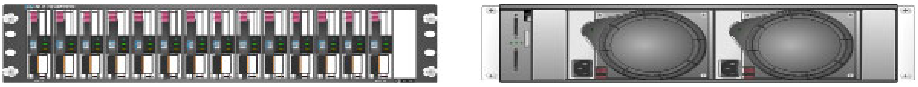


Figure 1.8 *Front and back of an external disk array*

The next type of device is the shared disk array (see Figure 1.9), which has its own controllers and can be attached to a set of servers instead of only one. The onboard controller allows logical unit numbers (LUNs) to be carved out and to be presented to the appropriate server or shared among the servers. It is possible to use this type of device to share only VMFS-formatted LUNs between at most four ESX hosts, because that is generally the limit on how many SCSI interfaces are available on each shared disk array. It is a very inexpensive way to create multimachine redundancy. However, using this method limits the cluster of ESX Servers to exactly the number of SCSI ports that are available, and limits the methods for accessing raw LUNs from within VMs. With ESX v3 it is possible to use local raw LUNs directly within a VM but with ESX v4, this is no longer possible and all local virtual disks must live on a VMFS. We cover this in more detail in Chapter 10.

This type of device should not be confused with a standard disk array; the differentiation is that the shared disk array has its own built in controllers.

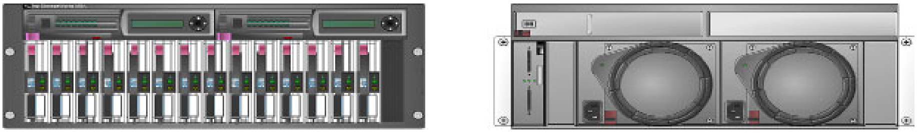


Figure 1.9 *Front and back of a shared SCSI array*

A SAN is one of the devices that will allow vMotion to be used; it generally comes in entry-level (see Figure 1.10) and enterprise-level (see Figure 1.11) styles. Each has its uses with ESX and all allow the sharing of data between multiple ESX hosts, which is the prime ingredient for the use of vMotion. SAN information is covered in detail in Chapter 5, "Storage with ESX."

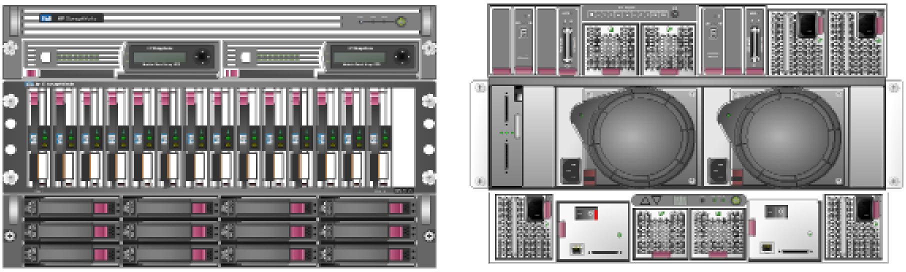


Figure 1.10 *Front and back of an entry-level SAN with SATA drives*

Although SATA controllers are supported directly by ESX or within a SAN, they are generally slower than using SCSI or SAS controllers. Because of poor performance numbers, SATA may not be a good choice for primary VMDK storage, but would make a good temporary backup location. The best solution is to avoid non-SCSI, SAS controllers as much as possible. Although the entry-level SAN is very good for small installations, enterprise-class installations really require an enterprise-level SAN (refer to Figure 1.11). The enterprise-level SAN provides a higher degree of redundancy, storage, and flexibility for ESX than an entry-level version. Both have their place in possible architectures. For example, if you are deploying ESX to a small office with a pair of servers, it is less expensive to deploy using an entry-level SAN than a full-sized enterprise-class SAN.

Best Practice for SAN Storage

Use SCSI- or SAS-based SAN storage systems. For two to four hosts, entry-level systems may be best. However, for anything else, it is best to use enterprise SAN systems for improved availability, performance, and redundancy.

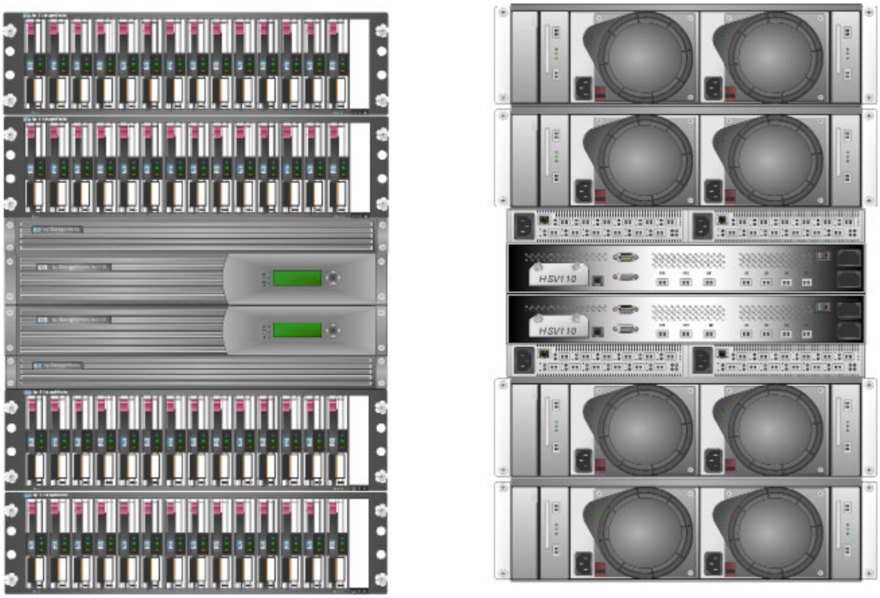


Figure 1.11 *Front and back of an enterprise-level SAN*

The last physical entry in the storage realm is that of NAS devices (see Figure 1.12), which present file systems using various protocols, including Network File System (NFS), Internet SCSI (iSCSI), and Common Internet File System (CIFS). VMware ESX does not support CIFS as a data store but does support NFS, iSCSI, and FC SAN. iSCSI and NFS have a throughput similar to that of FC SAN on ESX v4 now that you can use jumbo frames with each of these protocols. Granted, to achieve this level of performance, you must either use iSCSI HBAs, CNAs, or 10G ethernet adapters. With NAS, there is no need for Fibre Channel adapters, only more NICs to support the iSCSI and NFS protocols while providing redundancy.

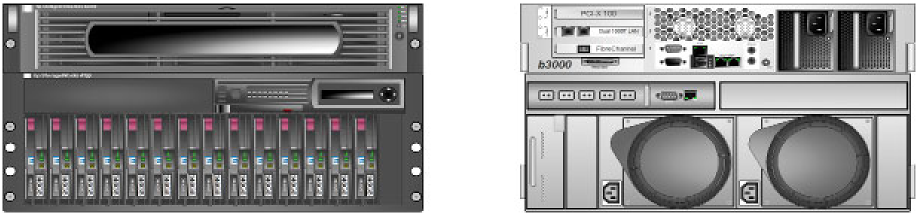


Figure 1.12 *NAS device*

There is a new class of storage devices called virtual storage appliances (VSAs) that should be considered. These devices make use of locally available disk space and make it available to a cluster of ESX or ESXi hosts. Some VSAs serve up only NFS and others serve up NFS and iSCSI. VSAs can replicate data from host to host for redundancy and work with multiple LUNs, or only one LUN. In either of these cases, VSAs could be a viable option for small-scale deployments that have lots of local disk space available that now need cluster functionality such as vMotion and shared disk clusters between multiple hosts.

Examples

Now it is time to review what customers have done in relation to the comments in the previous sections. The following six examples are from real customers, not from our imagination. The solutions proposed use the best practices previously discussed and a little imagination.

Example 1: Using Motherboard X and ESXi Will Not Install

This is a very common issue because many think that ESXi, being free, will run on nearly all hardware. Unlike VMware Workstation, which requires a host operating system, ESXi is a bare metal install. Like ESX, ESXi has a fairly large official and unofficial hardware compatibility list (HCL). In general, the reasons for ESXi not installing relate to the hardware involved, or the firmware for said hardware. It is best to first peruse the HCLs and look for the hardware that composes your system, if you do not find your hardware or anything similar, there is a good chance ESXi will not install nor work properly. When using existing hardware, it is best to look at the HCL for the storage and networking devices that may be onboard your motherboard. You may need to disable the onboard hardware and add in supported IO devices.

In addition to hardware issues, you need to be cognizant of the BIOS used on the motherboard. To run vSphere ESX or ESXi 4, you must be able to enable Intel-VT or AMD-V and enable the No eXecute (NX) or eXecute Disabled (XD) bits. Without these BIOS settings, vSphere ESX or ESXi 4 will not run properly and perhaps at all. These settings only affect EVC within VMware Virtual Infrastructure ESX or ESXi 3.x.

Example 2: Installing ESX and Expecting a Graphical Console

Another interesting issue that comes up within the VMware Communities Forums (http://communities.vmware.com) is installing ESX or ESXi on a laptop. Although this is possible, and in some cases could be useful for demos and development, in general it will not install. On modern laptops with at least Intel Nahalem or similar AMD support, it may be possible to install ESXi even though this is not listed on the HCL, which limits your support options. Even more interesting is that no GUI or method exists to access the VMs within the ESX or ESXi host directly from the ESX or ESXi host. A command line has limited functionality to manage and create VMs, but there is no self-contained method to access a VM after it is running.

This is by design. VMware wants ESX and ESXi to be treated as appliances; therefore, management tools require at least one external system in order to access the VMs.

Example 3: Existing Datacenter

A customer was in the midst of a hardware-upgrade cycle and decided to pursue alternatives to purchasing quite a bit of hardware. The customer wanted to avoid buying 300+ systems at a high cost and decided to pursue ESX host. Furthermore, the customer conducted an exhaustive internal process to determine the need to upgrade the 300+ systems and believes all of them could be migrated to ESX, because they meet or exceed the documented constraints. The existing machine mix includes several newer machines from the last machine refresh (around 20), but is primarily made up of machines that are at least two to three generations old, running on processors no faster than 900MHz. The new ones range from 1.4GHz to 3.06GHz 2U machines (see Figure 1.6). The customer would also like to either make use of existing hardware or purchase very few machines to make up the necessary difference, because the price for ESX to run 300+ machines approaches the customer's complete hardware budget. In addition, a last bit of information was also provided, and it really throws a monkey wrench into a good solution: The customer has five datacenters, each with its own SAN infrastructure.

Following best practices, we could immediately state that we could use the 3.06GHz hosts. Then we could determine whether there were enough to run everything. However, this example shows the need for something even more fundamental than just hardware to run 300+ virtual machines. It shows the need for an appropriate analysis of the running environment to first determine whether the 300+ servers are good candidates for migration, followed by a determination of which servers are best fit to be the hosts of the 300+ VMs. The tool used most often to perform this analysis is the VMware Capacity Planner. This tool will gather up various utilization and performance numbers for each server over a one- to two-month period. This information is then used to determine which servers make good candidates to run as VMs.

Instead of VMware Capacity Planner, which is available only to VMware Authorized Consultants (VAC), you can use the lighter-weight guided consolidation tool built in to VMware vCenter Server to determine what systems would be good candidates for virtualization. The reporting is lighter weight and the tests are not as exhaustive, but it does not require anything more than an evaluation license of VMware vCenter Server.

Best Practice

Use a capacity planner or something similar to get utilization and performance information about servers.

When the assessment is finished, you can better judge which machines could be migrated and which could not. Luckily, the customer had a strict “one application per machine” rule, which was enforced, and which removes possible application conflicts and migration concerns. With the details released about their current infrastructure, it was possible to determine that the necessary hardware was already in use and could be reused with minor hardware upgrades. Each machine would require an additional quad-port NIC and Fibre Channel cards, as well as an increase in memory and local disk space. To run the number of VMs required and to enable the use of vMotion, all hosts were paired up at each site at the very least, with a further recommendation to purchase another host machine per site (because there were no more hosts to reuse) at the earliest convenience so that they could alleviate possible host failures in the future. To perform the first migrations, some seed units would be borrowed from the manufacturer and LUNs carved from their own SANs, allowing migration from physical to virtual using the seed units. Then the physical host would be converted to a ESX and the just-migrated VM vMotioned off the borrowed seed host. This host would be sent to the other sites as their seed unit when the time came to migrate their hosts. This initial plan would be revised after the capacity planner was run and analyzed.

Example 4: Office in a Box

One of the author’s earliest questions was from a company that wanted to use ESX to condense hundreds of remote locations into one easy-to-use and administer package of a single host running ESX with the remote office servers running as VMs. Because the remote offices currently used outdated hardware, this customer also felt that ESX would provide better remote management capability. The customer also believed that the hardware should be upgraded at these remote offices all over the world. The goal was to ship a box to the remote location, have it plugged in, powered up, and then remotely manage the server. If there were a machine failure of some sort, the customer would ship out a new box. The concern the customer had was the initial configuration of the box and how to perform backups appropriately.

They set up their eight-drive dual-quad core processor hosts with a full complement of memory and disks, an extra quad-port Ethernet card, an external tape device via an Adaptec card (see Figure 1.13), and enough file system space for a possible shared virtual machine disk file, which should be on its on virtual machine file system, due to locking concerns. We discussed a SAN and the use of vMotion, but the customer thought that this would be overkill for the remote offices. For the datacenter, this was a necessity, but not for a remote office.

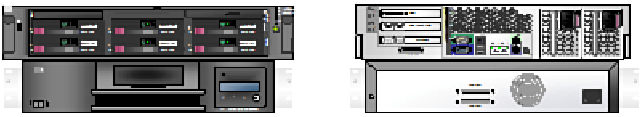


Figure 1.13 *Office in a box server with tape library*
Visio templates for image courtesy of Hewlett-Packard.

However, the best-laid plan was implemented incorrectly, and a year after the initial confirmation of the design, the customer needed to implement Microsoft Clustering as a cluster in a box. Because of this oversight, the customer had to reinstall all the ESX host to allocate a small VMFS for a shared virtual machine disk file. The customer chose to reinstall the machines, but first set up the operating system disk as a RAID 1, making using of hardware mirroring between disks 1 and 2, leaving four disks to make a RAID 5 + 1 spare configuration of 146GB disks. Then another LUN was carved from the remaining two disks of RAID 1 just for the shared VMDK for a cluster of VMs.

One other solution was to initially build the units with only six drives, leaving the last two slots for drives to be added if there was a need to add more VMFS space, or special use VMFS space such as for a cluster. Although for a Cluster in a Box the clustered VMDK can use the same VMFS as the general use drives, separating out the cluster VMDKs allows for ESX to handle the locking more efficiently.

Best Practice

When using clusters of machines, place the cluster virtual disk volumes on separate LUNs.

Example 5: The Latest and Greatest

One of our opportunities dealt with the need for the customer to use the latest and greatest hardware with ESX, and in doing so to plan for the next release of the OS at the same time. The customer decided to go with a full blade enclosure using dual CPU quad-core blades with no disk, and iSCSI TOE cards in order to boot the ESX host via iSCSI from a NAS (see Figure 1.12). The customer also required an easier and automated way to deploy the ESX hosts.

This presented several challenges up front. The first challenge was that the next release of the OS was not ready at the time, and the HCL for the current release *and* the first release of the next version of ESX showed that some of the customer's desired options would *not* be available. So, to use ESX, the hardware mix needed to be modified to support the common set of devices between ESX v3.5 and v4.0. The customer therefore traded in the iSCSI TOE cards for supported 10G cards.

The main concern here is that the customer wanting the latest and greatest instead got a mixed bag of goodies that were not compatible with the current release, and the prelist of the HCL for the next release did not list the customer's desired hardware either. In essence, if it is not on the HCL now, most likely it will not be on the list in the future; if you can get a prerelease HCL, this can be verified. In essence, this customer had to change plans based on the release schedules, and it made for quite a few headaches for the customer and required a redesign to get started, including the use of 10G mezzanine cards and 10G network switches. In essence, always check the HCL on the VMware website before purchasing anything.

As for the deployment of ESX, the onboard remote management cards and the multiple methods to deploy ESX made life much easier. Because these concepts are covered elsewhere, we will not go into a lot of detail. ESX provides its own method for scripted installations just for blades. Many vendors also provide mechanisms to script the installations of operating systems onto their blades. It should be noted that with vSphere ESX v4, host profiles make installations simpler. The key to scripted installations is adding in all the extra bits often required that are outside of ESX, such as hardware agents.

Example 6: The SAN

Our sixth example is a customer who brought in consulting to do a bake-off between competing products using vendor-supplied small SANs. The customer made a choice and implemented the results of the bake-off in a production environment that used a completely different SAN and SAN layout than used in the bake-off. Although this information was available during the bake-off, it was pretty much a footnote. This, in turn, led to issues with how the customer was implementing ESX in production that had to be reengineered.

The customer wanted fully compatible multipath features, such as load balancing, failover, and path aggregation (which is not available by default) in order to place within their hosts more than two Fibre ports to increase overall SAN throughput. Unfortunately, the customer's existing licensing did not consider this option and the SAN provider chosen did not have the proper capability at the time of the bake-off.

The solution is to increase the customer's license level to Enterprise Plus, as well as look into using a vendor supplied Multi-Path Plug-in for vSphere ESX v4. Unfortunately, very few vendors have multipath drivers for vSphere ESX v4.

It is crucial to look at all aspects of your hardware needs during any bake-offs and testing and to measure everything 12 inches to the foot, or exactly what you will use in production.

Example 7: Secure Environment

It is increasingly common for ESX to be placed into secure environments as long as the security specialist understands how ESX works and why it is safe to do so. However, in this case, the security specialist assumed that because the VMs share the same air within the host, they are therefore at risk. Although we could prove this was not the case, the design of the secure environment had to work within this scope. The initial hardware was two-socket quad-core CPU machines and a small SAN that would later be removed when they proved everything worked and their large corporate SANs took over. The customer also wanted secure data not to be visible to anyone but the people in the teams using the information.

This presented several concerns. The first is that the administrators of the ESX box must also be part of the secure teams, have the proper corporate clearances, or be given an exception, because anyone with administrator access to an ESX host also has access to all the VMDKs available on the ESX Server. Chapter 4, "Auditing and Monitoring," goes into auditing your ESX environment in detail. Because the customer wanted to secure the data completely, it is important to keep the service console, vMotion, iSCSI, NFS, FT Logging, and the VM networks all on their own secure networks. Why should we secure vMotion and everything? Because vMotion will pass the memory footprint of the server across an Ethernet cable and, combined with access to the service console, will give a hacker everything a VM is doing. If not properly secured, this is quite a frightening situation.

Whereas the company had a rule governing use of SANs to present secure data LUNs, it had no such policy concerning ESX. In essence, it was important to create an architecture that kept all the secure VMs to their own set of ESX hosts and place on another set of ESX hosts those things not belonging to the secure environment. This kept all the networking separated by external firewalls and kept the data from being accessed by those not part of the secure team. If a new secure environment were necessary, another pair of ESX hosts (so we can vMotion VMs) would be added with their own firewall.

The preceding could have easily been performed on a single ESX cluster, yet require the administrators to have the proper corporate clearances to be allowed to manipulate secured files. Given this and the appropriate network