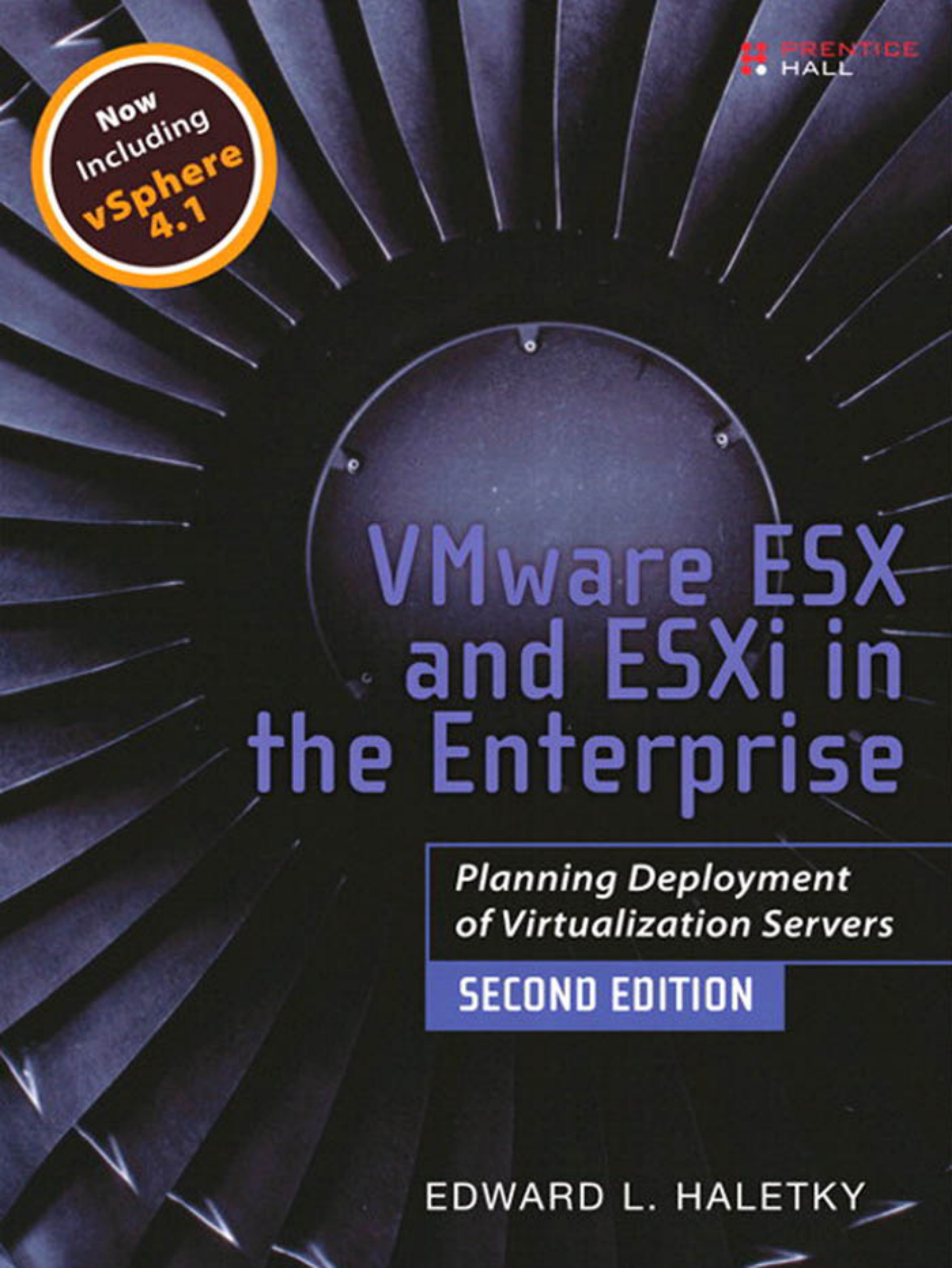
Now Including vSphere 4.1
PRENTICE HALL
VMware ESX and ESXi in the Enterprise
Planning Deployment of Virtualization Servers
SECOND EDITION
EDWARD L. HALETKY

VMware ESX and ESXi in the Enterprise

Planning Deployment of Virtualization Servers

Edward L. Haletky

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

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

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 6

Effects on Operations

The introduction of virtualization using VMware ESX and ESXi creates a myriad of operational problems for administrators, specifically problems having to do with the scheduling of various operations around the use of normal tools and other everyday activities, such as deployments, antivirus and other agent and agentless operational tasks (performance gathering, and so forth), virtual machine agility (vMotion and Storage vMotion), and backups. In the past, prior to quad-core CPUs, many of these limitations were based on CPU utilization, but now the limitations are in the areas of disk and network throughput.

The performance-gathering issues dictate which tools to use to gather performance data and how to use the tools that gather this data. A certain level of understanding is required to interpret the results, and this knowledge will assist in balancing the VMs across multiple ESX or ESXi hosts.

The disk throughput issues are based on the limited pipe between the virtualization host and the remote storage, as well as reservation or locking issues. Locking issues dictate quite a bit how ESX should be managed. As discussed in Chapter 5, "Storage with ESX," SCSI reservations occur whenever the metadata of the VMFS is changed and the reservation happens for the whole LUN and not just an extent of the VMFS. This also dictates the layout of VMFS on each LUN; specifically, a VMFS should take up a whole LUN and not a part of the LUN. Disk throughput is becoming much more of an issue and will continue to be. Which is why with vSphere 4.1, Storage IO Control (SIOC) was introduced to traffic shape egress from the ESX host to Fibre Channel arrays. SIOC comes into play if the LUN latency is greater than 20ms. SIOC should improve overall throughput for those VMs marked as needing more of the limited pipe between the host and remote storage.

The network throughput issues are based on the limited pipes between the virtual machines and the outside physical network. Because these pipes are shared among many VMs, and most likely networks, via the use of VLANs, network I/O issues come to the forefront. This is especially true when discussing

operational issues such as when to run network intensive tasks: VM backups, antivirus scans, and queries against other agents within VMs.

Virtual machine agility has its own operational and security concerns. Basically, the question is, "Can you ever be sure where your data is at any time?" Outside of the traditional operational concerns, virtual machine agility adds complexity to your environment.

Note that some of the solutions discussed within this chapter are utopian and not easy to implement within large-scale ESX environments. These are documented for completeness and to provide information that will aid in debugging these common problems. In addition, in this chapter unless otherwise mentioned we use the term ESX to also imply ESXi.

SCSI-2 Reservation Issues

With the possibility of drastic failures during crucial operations, we need to understand how we can alleviate the possibility of SCSI Reservation conflicts. We can eliminate SCSI Reservations by changing our operational behaviors to cover the possibility of failure. But what is a SCSI Reservation?

SCSI Reservations occur when an ESX host attempts to write to a LUN on a remote storage array. Because the VMFS is a clustered file system, there needs to be a way to ensure that when a write is made, that all previous writes have finished. In the simplest sense, SCSI Reservation is a lock that allows one write to finish before the next. We discuss this in detail in Chapter 5.

Although the changes to operational practices are generally simple, they are nonetheless fairly difficult to implement unless all the operators and administrators know how to tell whether an operation is occurring and whether the new operation would cause a SCSI Reservation conflict if it were implemented. This is where monitoring tools make the biggest impact.

VMware has made two major changes within the VMFS v3.31 to alleviate SCSI-2 Reservation issues. The first change was to raise the number of SCSI-2 Reservation retries that occur before a failure is reported. The second change was to allocate to each ESX host within a cluster a section of a VMFS so that simple updates do not always require a SCSI-2 Reservation. Even with these changes, SCSI-2 Reservations still occur, and we need to consider how to alleviate them.

The easiest way to alleviate SCSI-2 Reservations is to manage your ESX hosts using a common interface such as VMware vCenter Server, because vCenter has the capability to limit some actions that impact the number of simultaneous LUN actions. However, with the proliferation of PowerShell scripts, other vCenter management entities, and direct to host actions, this becomes much more difficult. Therefore, as we discussed in Chapter 4, "Auditing and Monitoring,"

it behooves you to perform adequate logging so that you can determine what caused the SCSI-2 Reservation, and then work to alleviate this from an operational perspective.

Best Practice

Verify that any other operation has first completed on a given file, LUN, or set of LUNs before proceeding with the next operation.

The primary way to avoid SCSI-2 Reservations is to verify in your management tool that all operations upon a given LUN or set of LUNs have been completed before proceeding with the next operation. In other words, serialize your actions per LUN or set of LUNs. In addition to checking your management tools, check the state of your backups and whether any current open service console operations have also completed. If a VMDK backup is running, let that take precedence and proceed with the next operation after the backup has completed. The easiest way to determine if a backup is running is to look on your backup tool's management console. However, you can also check for a snapshot that is created by your backup software using the snapshot manager that is part of the vSphere client or one of the snapshot hunter tools available. Most snapshots created by backup tools will have a very specific snapshot name. For example, if you use VCB, the snapshot will be named "_VCB-BACKUP_."

Multiple concurrent vMotions or Storage vMotions are a common cause for SCSI-2 Reservations, and this is why VMware has limited the number of simultaneous vMotions and Storage vMotions that can take place to six (increased to 8 in vSphere 4.1). Note that although vSphere will allow this number of migrations take place concurrently, it is not recommended for all arrays. For high-end arrays, the maximum can be performed simultaneously.

To check to see whether service console operations that could affect a LUN or set of LUNs have completed, judicious use of `sudo` is recommended. sudo can log all your operations to a file called `/var/log/secure` that you can peruse for file manipulation commands (`cp`, `rm`, `tar`, `mv`, and so on). Hopefully, this is being redirected to your log server, which has a script written to tell you if any LUN operations are taking place. Additionally, as the administrator, you can check the process lists for all servers for similar operations. No VMware user interface combines backups, vMotion, and service console actions. However, the HyTrust appliance is one such device that does provide a central place to audit for LUN requests (but not the completion of such requests).

When you work with ESXi, filesystem actions can still take place via Tech Support Mode, VMware Management Appliance (vMA) and the use of the vifs command. Even for ESXi, logging will be required.

For example, let's look at a system of three ESX hosts with five identical LUNs presented to the servers via Hitachi storage. Because each of the servers shares LUNs we need, we should limit our LUN activity to one operation per LUN at any given time. In this case, we could perform five operations simultaneously as long as those operations were LUN specific. After LUN boundaries are crossed, the number of simultaneous operations drops. To illustrate the second case, consider a VM with two disk files, one for the C: drive and one for the D: drive. Normally in ESX, we would place the C: and D: drives on separate LUNs to improve performance, among other things. In this case, because the C: and D: drives live on separate LUNs, manipulation of this VM, say with vMotion, counts as four simultaneous VM operations. This count is due to one operation affecting two LUNs, and the locks need to be set up on both the source and target of the vMotion. Therefore, five LUN operations could equate to fewer VM operations.

This leads to a set of operational behaviors with respect to SCSI Reservations.

Using the preceding examples as a basis, the suggested operational behaviors are as follows:

- Simplify deployments so that a VM does not span more than one LUN. In this way, operations on a VM are operations on a single LUN. This may not be possible because of performance requirements of the LUNs.
- Determine whether any operation is happening on the LUN you want to operate on. If your VM spans multiple LUNs, check the full set of LUNs by visiting the management tools in use and making sure that no other operation is happening on the LUN in question.
- Choose one ESX host as your deployment server. In this way, it is easy to limit deployment operations, imports, or template creations to only one host and LUN at a time.
- Use a naming convention for VMs that also tells what LUN or LUNs are in use for the VM. This way it is easy to tell what LUN could be affected by VM operation. This is an idealistic solution to the problem, given the possible use of Storage vMotion, but at least label VMs as spanning LUNs.
- Inside vCenter or any other management tool, limit access to the administrative operations so that only those who know the process can enact an operation. In the case of vCenter, only the administrative users should have any form of administrative privileges. All others should have only VM user or read-only privileges.
- Only administrators should be allowed to power on or off a VM. A power-off and power-on are considered separate operations unrelated to a reboot

or reset from within the Guest OS. Power on and off operations open and close files on the LUN. However, more than just SCSI Reservation concerns exist with this case—there are performance concerns. For example, if you have 80 VMs across 4 hosts, rebooting all 80 at the same time would create a performance issue called a boot storm, and some of the VMs could fail to boot. The standard boot process for an ESX host is to boot the next VM only after VMware Tools is started, guaranteeing that there is no initial performance issue. However, this does not happen if VMware Tools is not installed or does not start. The necessary time of the lock for a power-on or -off operation is less than 7 microseconds, so many can be done in the span of a minute. However, this is not recommended, because the increase in load on ESX could adversely affect your other VMs. Limiting this is a wise move from a performance viewpoint.

- Use care when scheduling VMDK-level backups. It is best to have one host schedule all backups and to have one script to start backups on all other hosts. In this way, backups can be serialized per LUN. The serialization problem is solved by using the VMware Consolidated Backup, VMware Data Recovery, and many third-party tools such as Veeam Backup, Vizioncore vRangerPro, and Symantec BackupExpress. It is better for performance reasons to have each ESX host doing backups on a different LUN at any given time. For example, our three machines can each do a backup using a separate LUN. Even so, the activity is still controlled by only *one* host or tool so that there is no mix up or issue with timing so that each per LUN operation is serialized for a given LUN. Let the backup process limit and tell you what it is doing. Find tools that will
- Never start a backup on a LUN while another is still running.
- Signal the administrators that backups have finished either via email, message board, or pager(s). This way there is less to check per operation.
- Limit vMotion (hot migrations), fast migrates, cold migrations, and Storage vMotions to one per LUN. If you must do a huge number of vMotion migrations at the same time, limit this to one per LUN. With our example, there are five LUNs, so there is the possibility of five simultaneous vMotions, each on its own LUN, at any time. This assumes the VMs do not cross LUN boundaries.
- vMotion needs to be fast, and the more you attempt to do vMotions at the same time, the slower all will become. The slower the vMotion process, the higher the chance of the Guest OS having issues such as a blue screen of death for Windows. Using vMotion on 10 VMs at the same time could be a serious issue for the performance and health of the VM regardless of

SCSI Reservations. Make sure the VM has no active backup snapshots before invoking vMotion.

- Use only the default VM disk modes. The nondefault persistent disk modes lead to not being able to perform snapshots and use the consolidated backup tools. Nonpersistent modes such as read-only create snapshot files on LUNs during runtime and remove them on VM power-off so as to not affect the master disk file.
- Do not suspend VMs, because this also creates a file and therefore requires a SCSI Reservation.
- Do not run vm-support requests unless all other operations have completed.
- Do not use the `vdf` service console tool when any other modification operation is being performed. Although vdf does not normally force a reservation, it could experience one if another host, because of a metadata modification, locked the LUN.
- Do not rescan storage subsystems unless all other operations have completed.
- Limit use of `vmkmultipath`, `vmkfstools`, and other VMware-specific service console and remote CLI commands until all other operations have completed.
- Create, modify, or delete a VMFS only when all other operations have completed.
- Be sure no third-party agents are accessing your storage subsystem via vdf, or direct access to the `/vmfs` directory.
- Do not run scripts that modify VMFS ownership, permissions, access times, or modification times from more than one host. Localize such scripts to a single host. It is suggested that you use the deployment server as the host for such scripts.
- Run all scripts that affect LUNs from a management node that can control when actions can occur.
- Stagger the running of disk-intensive tools within a VM, such as virus scan. The extra load on your SAN could cause results similar to those that occur with SCSI Reservations but which are instead queue-full or unavailable-target errors.

- Use only one file system per LUN.
- Do not mix file systems on the same LUN.

What this all boils down to is ensuring that any possible operation that could somehow affect a LUN is limited to only one operation per LUN at any given time. The biggest hitters of this are automated power operations, backups, vMotion, Storage vMotion, and deployments. A little careful monitoring and changes to operational procedures can limit the possibility of SCSI Reservation conflicts and failures to various operations.

A case in point follows: One company under review because constant, debilitating SCSI Reservation conflicts reviewed the list of 23 items and fixed one or two possible items but missed the most critical item. This customer had an automated tool that ran simultaneously on all hosts at the same time to modify the owner and group of every file on every VMFS attached to the host. The resultant metadata updates caused hundreds of SCSI-2 Reservations to occur. The solution was to run this script from a single ESX host for all LUNs. By limiting the run of the script to a single host, all the reservations disappeared, because no two hosts were attempting to manipulate the file systems at the same time, and the single host, in effect, serialized the actions.

Hot and cold migrations of VMs can change the behavior of automatic boot methodologies, which can affect LUN locking. Setting a dependency on one VM or a time for a boot to occur deals with a single ESX host where you can start VMs at boot of ESX, after VMware Tools starts in the previous VM, after a certain amount of time, or not at all. This gets much more difficult with more than one ESX host, so a new method has to be used. Although starting a VM after a certain amount of time is extremely useful, what happens when three VMs start almost simultaneously on the same LUN? Remember, we want to limit operations to just one per LUN at any time. We have a few options:

- Stagger the boot or reboot of your ESX host and ensure that your VMs start only after the previous VMs' VMware Tools start, to ensure that all the disk activity associated with the boot sequence finishes before the next VM boots, thereby helping with boot performance and eliminating conflicts. VM boots are naturally staggered by ESX when it reboots anyway if the VM is auto-started.
- Similar to doing backups, have one ESX host that controls the boot of all VMs, guaranteeing that you can boot multiple VMs but only one VM per LUN at any time. If you have multiple ESX hosts, more than one VM can start at any time on each LUN, one per LUN. In essence, we use the VMware vSphere SDK to gather information about each VM from each ESX host and correlate the VMs to a LUN and create a list of VMs that

can start simultaneously; that is, each VM is to start on a separate LUN. Then we wait a set length of time before starting the next batch of VMs. This method is not needed when VMware Fault Tolerance fires because the shadow VM is already running. Also, VMware HA uses its own rules for starting VMs in specific orders.

All the listed operational changes will limit the number of SCSI subsystem errors that will be experienced. Although it is possible to implement more than one operation per LUN at any given time, we cannot guarantee success with more than one operation. This depends on the type of operation, the SAN, settings, and most of all, timings for operations.

Yet you may ask yourself, "Wouldn't using ESXi solve many of these issues because there is no service console?" The answer is, "Partially." Many of the "scripting" issues that occur within a service console are no longer a concern. Scripting issues can come up using the new VMware Virtual Management Appliance (vMA) or by using the remote CLI directly if there is not a single control mechanism for when these scripts run against all LUNs in question. So the problems can still occur even with ESXi. On top of this, it is still possible to run scripts directly within the ESXi Posix environment that comprises the ESXi management console. Granted, it is much harder, but not impossible.

There are several other considerations, too. Most people want to perform multiple operations simultaneously, and this is possible as long as the operations are on separate LUNs or the storage array supports the number of simultaneous operations. Because many simultaneous operations are storage array specific, it behooves you to run a simple test with the array in question to determine how many simultaneous operations can happen per LUN. As ESX improves, arrays improve, vStorage API for Array Integration is used within arrays, and transports improve in performance the number of simultaneous operations per LUN will increase.

With vSphere, the number of SCSI Reservations have dropped drastically but they still occur; when they do, this section will help you to track down the reasons and provide you the necessary information to test your arrays. You should also test to determine how many hosts can be added to a given cluster before SCSI Reservations start occurring. On low-end switches, this value may just be 2, whereas on others it could be 4.

Best Practice

Verify the number of simultaneous LUN activities that can occur on the remote storage arrays chosen for use.

Performance-Gathering and Hardware Agents Within a VM

Performance and other types of monitoring are important from an operational point of view. Many customers monitor the health of their hardware and servers by monitoring hardware and performance agents. Although hardware agents monitor the health of the ESX host, they should not monitor the health of a VM, because the virtual hardware is truly dependent on the physical hardware. In addition, most agents are talking to specific chips, and these do not exist inside a VM. So using hardware agents will often slow down your VM.

Best Practice for Hardware Agents

Do *not* install hardware agents into a VM; they will cause noticeable performance issues.

Measuring performance now is a very important tool for the Virtual Environment; it will tell you when to invest in a new ESX host and how to balance the load among the ESX hosts. Although there are automated ways to balance the load among ESX hosts (they are covered in Chapter 11, "Dynamic Resource Load Balancing"), most if not all balancing of VM load across hosts is performed by hand, because there are more than just a few markers to review when moving VMs from host to host.

There is an argument that Dynamic Resource Scheduling (DRS) will balance VMs across all hosts, but DRS does balancing only when CPU contention exists. If you never have contention, you may still want to balance your loads by hand, regardless of DRS settings.

The first item to understand is that the addition of a VM to a host will impact the performance of the ESX host—sometimes in small ways, and sometimes in other ways that are more noticeable. The second item to understand is how performance tools that run within a VM, for example Windows, calculates utilization. It does this by incrementing a tic counter in its idle loop and then subtracts that amount of time from the system clock time interval. Because the VM gets put to sleep when idle, the idle time counter is skewed, which results in a higher utilization representation than typical. Because there are often more VMs than CPUs or cores, a VM will share a CPU with others, and as more VMs are added the slice of time the VM gets to run on a CPU is reduced even further. Therefore, a greater time lag exists between each usage of the CPU and thus a longer CPU cycle. Because performance tools use the CPU cycle to measure performance and to keep time, the data received is relatively inaccurate. When the system

is loaded to the desired level, a set of baseline data should be discovered using VMware vCenter or other Performance Management tools.

After a set of baseline data is available, internal to the VM performance tools can determine whether a change in performance has occurred, but it cannot give you raw numbers, just a ratio of change from the baseline. For example, if the baseline for CPU utilization is roughly 20% measured from within the VM and suddenly shows 40%, we know that there was a 2x change from the original value. The original value is not really 20%, but some other number. However, even though this shows 2x more CPU utilization for the VM, it does not imply a 2x change to the actual server utilization. Therefore, to gain performance data for a VM, other tools need to be used that do not run from within the VM. VMware vCenter, a third-party tool such as Vizioncore vFoglight, or the use of esxtop from the command line or resxtop from the remote CLI are the tools to use because these all measure the VM and ESX host performance from outside the VM. In addition, they all give a clearer picture of the entire ESX host. The key item to realize is that when there is a sustained over 80% utilization of CPU for an ESX host as measured by vCenter or one of the tools, a new ESX host is warranted and the load on the ESX host needs to be rebalanced. This same mechanism can be used to determine whether more network and storage bandwidth is warranted.

Balancing ESX hosts can happen daily or even periodically during the day by using the vMotion technology to migrate running VMs from host to host with zero downtime. Although this can be dynamic (see Chapter 11), using vMotion and Storage vMotion by hand can give a better view of the system and the capability to rebalance as necessary. For example, if an ESX host's CPU utilization goes to 95%, the VM that is the culprit needs to be found using one of the tools; once found, the VM can be moved to an unused or lightly used ESX host using vMotion. If this movement becomes a normal behavior, it might be best to place the VM on a lesser-used machine permanently. This is often the major reason an N+1 host configuration is recommended.

Deployment of VMs can increase CPU utilization. Deployment is discussed in detail in a later chapter, but the recommendation is to create a deployment server that can see all LUNs. This server would be responsible for deploying any new VM, which allows the VM to be tested on the deployment server until it is ready to be migrated to a true production server using vMotion.

For example, a customer wanted to measure the performance of all VMs to determine how loaded the ESX host could become with the current networking configuration. To do so, we explained the CPU cycle issues and developed a plan of action. We employed two tools in this example, VMware vCenter, and `esxtop` running from the service console or from the vMA in batch mode (`esxtop -b`). For performance-problem resolution, `esxtop` is the best tool to use, but it spits out reams of data for later graphing. vCenter averages things

over 5-minute or larger increments for historical data, but its real-time stats are collected every 20 seconds. `esxtop` uses real and not averaged data gathered as low as every 2 seconds with a default of 5 seconds. The plan was to measure performance using each tool as each VM was running its application. Performance of ESX truly depends on the application within each VM. It is extremely important to realize this, and when discussing performance issues to not localize to just a single VM, but to look at the host as a whole. This is why VMware generally does not allow performance numbers to be published, as the numbers are workload dependent. It is best to do your own analysis using your applications, because one company's virtualized application suite has nothing to do with another company's; therefore, there can be dramatic variations in workload even with the same application set.

If you do want to measure performance of your ESX hosts for purposes of comparison to others, VMware has developed VMmark, which provides a common workload for comparison across multiple servers and hypervisors. Unfortunately, VMmark is not a standard yet. There also exists SPECvirt_sc2010 from the Standards Performance Evaluation Corporation located at www.spec.org/virt_sc2010/.

Network Utilization

Network utilization or I/O is a constant operational concern within the physical data center, and this does not change within the virtual environment. What does change is that network concerns are now affected by ALL virtual machines using the link in question and that the virtual switches are tied to CPU utilization of the ESX host in question. Many people claim that no one VM would ever saturate a single gigabit connection, and now with 10 gigabit connections this is impossible. Neither of these are impossible; there is more than enough capability in modern hypervisors to saturate any link. However, as with the discussion of disk I/O and CPU performance, we must remember that many VMs are sharing those same network links and that the bandwidth used by one VM will affect all other VMs using the same link.

Even when you use VLANs, the traffic for all those VLANs is running over a single wire, perhaps a few wires if you are using the built in ESX load-balancing methods. Even so, it is possible for all VMs to adversely affect overall network utilization. Now when we throw into the mix VMsafe-Net and other network and security virtual appliances, we can throttle down bandwidth even more. At the very least we are adding to the overall CPU requirements for networking.

In a recent class, I was asked, "Why is this the case when virtual switches are 100% in memory?" The problem is that while the data and virtual switch code is in memory, that code must still run within the CPU as the vmkernel. So as you add more VMs, virtual switches, and snapshots, there is an increase in overall

CPU utilization because now the vmkernel has to do more to handle virtual networking (an increase in snapshots implies that the CPU has to do more work to handle disk blocks, which includes change block tracking modes in vSphere because there is now more to do within the CPU for snapshots).

Anything happening within the vmkernel will impact CPU requirements just as VMsafe will impact virtual switch performance and therefore directly impact a host's virtual networking. What happens within a given VM can impact a host's virtual networking. Solutions like Intel-VT and AMD RVI reduce overall CPU overhead as they offload what the vmkernel needs to do. vStorage API for Array Integration (VAAI) will also decrease overall vmkernel needs because repetitive actions for storage will be placed into the arrays.

Virtual Machine Mobility

Virtual machine mobility is becoming an increasing concern for operations and compliance tracking because we often need to answer the question: "Where is our data?"

Given VMware vMotion, DRS, DPM, FT, HA, and Storage vMotion, we could surmise that our virtual machines are always in motion and therefore the data within the VM is never actually at rest. Given this, it is becoming more of an issue to know exactly where that VM is at all times. Did the VM end up on a host where it should not be running because of compliance, networking, or other concerns?

The current guidance with respect to compliance and virtualization security is to silo VMs within security zones contained within specific clusters. One way to enforce this is to tag the VMs, hosts, virtual switches, and other virtualization host objects with security zone tags (such as how the HyTrust Appliance and Reflex Systems approaches compliance) so that a VM cannot be placed on a host, virtual switch, and so forth that does not share the same tag.

Without these types of tags, a VM could end up on a host that does not have the proper virtual trust zones configured. Host Profiles can help to solve many of these configuration issues for the same cluster, but does not solve the problem for a different cluster, enclave, datacenter, and so on. However, tags apply to manual operations. The automatic operations from HA, DRS, and DPM are limited to only those hosts within the cluster. Hence, we see the oft-required security zone silo per cluster.

If tags are not in use, and they are not in use for the vast majority of virtualization systems today, there is an increasing risk that VMs end up on misconfigured hosts, and therefore application availability is impacted. In addition, a VM could end up somewhere else within your virtual environment. Perhaps it ends up on a single development host that is part of the virtual environment but shares the same LUNs as your production hosts.

Operationally, it is important to know where a virtual machine is at all times. To aid in this there are tools such as the HyTrust Appliance, VMware vCenter, and Hyper9, as well as any other virtualization search tools. For large environments, you may need to search for the location of your critical virtual machines or have canned reports that report on anomalies caused by vMotion and Storage vMotion. Anomalies to look for could be VMs for one trust zone ending up on hosts not vetted for that trust zone (DMZ VMs are a good point). These anomalies can happen to hosts outside a given cluster but within the same datacenter, as defined by VMware vCenter.

Data Store Performance or Bandwidth Issues

Because bandwidth is an issue, it is important to make sure that all your data stores have as much bandwidth as possible and to use this bandwidth sparingly for each data store.

"As much bandwidth as possible" and "use sparingly" may sound counter intuitive, but they are not from an operational perspective. Normal operational behavior of a VM often includes such things as full disk virus scans, backups, spyware scans, and other items that are extremely disk-intensive activities. Although none of these activities will require any form of locking of the data store on which the VMDK resides, they all take a serious amount of bandwidth to accomplish. The bandwidth requirements for a single VM are not very large compared to an ESX host with more VMs. All activities are fairly additive in nature. What you do within one VM, from a disk perspective, affects all other VMs on the same datastore and, depending on the storage solution, all VMs on other data stores. How is this possible? Think about the networks involved, with traditional iSCSI over the network and NFS; your ultimate bandwidth is limited to the speed of the links used, so a single gigabit ethernet link is much more limited than links that use Fibre Channel host bus adapters (FC or iSCSI). This is why it is important to have as much bandwidth as possible, including using load balancing of your storage links for each data store in use. If you have access to a multipath plug-in driver, you may also be able to aggregate your storage links to form one larger trunk of pipes to your storage device and at the same time increase your overall storage bandwidth. Even with MPP and bandwidth aggregation, load balancing, either by hand or automatically, is a step in the proper direction.

Staggering storage-intensive activities in time will greatly reduce the strain on the storage environment, but remember that staggering across ESX hosts is a good idea as long as different data stores are in use on each ESX host. For example, it would cause locking issues for VMs that reside on the same LUN but different ESX hosts to be backed up at the same time, unless you are using

in-VM agents; in that case, no locking issues would exist. Locking should be avoided. However, virus scans will not cause many issues when done from multiple VMs on the same LUN from multiple ESX hosts, because operations on the VMDK do not cause locks at the LUN level. By running backup and vStorage based antivirus tasks on different ESX hosts, you are using different links to the SAN and therefore are spreading your overall bandwidth usage across multiple links and hopefully using less of each link than running everything on a single ESX host.

It is possible that running of disk-intensive tools within a VM could cause results similar to those that occur with SCSI reservations, such as overloaded links that return errors instead of completing the operational task. These types of failures are not SCSI reservations. Instead, they are load issues that cause the SAN or NAS to be overworked and therefore present failures similar to SCSI-2 reservations.

Best Practice for Internal VM Disk Operations

Stagger all disk-intensive operations internal to the VM over time and ESX hosts to reduce strain on the storage network.
Spread the load across multiple ESX hosts and storage links.

Other Operational Issues

ESX makes extensive use of memory. There are operational concerns regarding the use of memory, too. The main issue with memory is to prevent the swapping of memory during runtime of VMs. The runtime covers the memory actually used, and not always what is allocated. A VM may have 256GB of memory allocated to it. If we allocate 64Gbs to a VM, and this much memory is allocated to all the VMs, on a 64GB ESX host, only one VM could be created and the memory will be overcommitted as the ESX takes some memory. If the goal is to run 20 VMs, there is a memory requirement of 1280GB, which is quite a bit over the 1TB server memory limit inherent in ESX. Which means that if all the 64GB of memory is actually used by a VM, the ESX host will need to start swapping (or paging) memory out in large chunks to accommodate the running of another VM.

If in reality only 1GB of each VM is used, only 20GB of the available 64GB of memory is in use at any time, allowing more VMs to be created and used without swapping memory, even though there is potential for up to 1280GB of memory to be used. In this case, it is best to assign memory sparingly and to give a VM only what it needs to run. This way, memory management will allow a denser population of VMs to run.

Consider the following thought: With ESX, we are now back in time to the realm of limited resources. There are no longer gobs of memory and disk available for any particular machine, but a realm where memory and disk can be vast; but as more VMs are added, more resources are used. The goal is now to preserve memory. For example, consider programming the old Commodore 64, where no more than 360K would fit on a single floppy; to go past this, more than one floppy had to be used. Everyone programmed to the 360K limit of the floppy so that code would fit on a single disk. After another floppy was in use, the applications usage went downhill, performance suffered, and wait time increased. With ESX, we are back in this realm where we need to be cognizant of the limitations of the host, which is trying to do much, much more with less than ever before.

Best Practice for ESX

The mindset for ESX is to give out only the necessary resources to each VM, rather than give out all the resources.

All VMs affect the resource limits of the host. Therefore, resource management becomes a huge issue (as covered in another chapter). Note, however, that changes to the way resources are used, assigned, and managed can inadvertently affect all VMs on a host or in a farm.

Limiting memory assignment to VMs can allow more VMs to run in a single ESX host without impacting memory or performance limits.

Life-Cycle Management

Because it is easy to overuse resources besides memory, it is very important to have some sort of life-cycle management tool in place, so that VMs can be ordered and approved by the responsible parties. With VMware vSphere vCenter Server, now the VMware vCenter Orchestrator product can provide a limited form of life-cycle management. The VMware Life-Cycle Manager product requires its own license, and it provides a necessary life-cycle management process.

A life-cycle management tool is, however, only as good as the process backing it up. A tool is not useful without a written life-cycle process. At the very least, such a process should include the following:

- Virtual machine request with the resources required in terms of memory, CPU, disk, and network.

- Virtual machine request with justification, lifetime of VM, VM owner, application owner, and manager involved. This should also include what to do with the VM after the lifetime is over, such as archive, delete, and so on.
- Management approval of the justification.
- Architectural review of the virtual machine resource requirements with the results being whether a new ESX host is required, as well as the actual resources to allocate, which most likely will be lower than the requested amounts. This should include exactly which networks and storage devices should be used, as well as the ESX host to initially place the VM. Architectural review should consider FT, DPM, DRS, SRM, EVC, SIOC, NetIOC, and HA, as well the VM's impact on any other VM. Has this VM superseded another VM?
- Application design review for the application and operating system to be placed within the VM. This should include a review of all agents required within the virtual environment.
- Placement of the VM within a development cluster for testing.
- Staging the VM through QA to determine that all is working as expected.
- Staging the VM into production with a final review before going live.
- Final sign off on live VM by VM owner.
- VM decommission and disposition.

Throughout a VM's life cycle, the ownership of the VM may change. It is very important to track such changes. If a VM has issues, a virtualization administrator will need to go to this VM owner to assist in solving the problem. It could also be that the VM has now become obsolete and as such is at the end of its life cycle. Life-cycle management will help control your ever-growing number of VMs and limit dependency issues.

Such controls often exist for physical machines, and they should be translated into the virtual environment. Although it is very easy to create a VM, it should never happen at the request of anyone directly but should follow a life-cycle process.

The virtual machine life-cycle process is the one process that will aid in debugging operational issues caused by the politics behind virtual machine creation.

Conclusion

By paying careful attention to operational issues, it is possible to successfully manage ESX and remove some of the most common issues related to poor operational use of ESX. ESX is designed to be centrally managed and care should be taken to do so. It is also important to realize that each implementation of ESX has different operational concerns.

This page intentionally left blank

Index

C

D

E

F

G

H

I

Q

R

S

W

Y-Z