Chapter 2

Version Comparison

Virtualization as a technology has been around for a very long time. VMware was founded by a group out of Stanford and was one of the early companies that brought virtualization to the x86 platform. Their initial product was a "please try this, it is cool, and tell us what to fix" version of VMware Workstation. Soon after that, VMware Workstation version 2 came out, and the world of computing changed. When version 4 of VMware Workstation came out, more and more people started to use the product, and soon after came the server versions GSX and ESX. With ESX, another change to computing took place; *virtualization* has become the buzzword and driving force behind many datacenter choices.

VMware produces four major products with varying capabilities and functionality. The products form a triangle in which VMware Workstation, Player, and Fusion are at the bottom with the broadest ranges of functionality and capability. Here VMware tries out new ideas and concepts, making it the leading edge of virtualization technology. The second tier is VMware ACE that adds to VMware Workstation the Assured Computing Environment that allows more control over the virtual machines in use. The third tier of the triangle is VMware Server (formerly GSX Server), which could be said to be VMware ESX-light because it is a middle ground between VMware Workstation and ESX, providing VM Workstation-style functionality while running on another operating system: Windows or Linux. VMware Server is a collection of programs that includes a management interface that has its own SDK, and other programs to launch VMs in the background. The pinnacle tier is ESX and ESXi, which are their own operating systems and the version comparison covered within this chapter.

ESX v3 and ESX v4 differ in many small ways, but both differ greatly from ESX v2.These differences revolve around how the system boots and how the functionality of the earlier version was implemented inside the new version. Between ESX v3 and ESX v4 there are changes in just about every subsystem, and all for the better. This is the release of ESX that brings many of the VMware Workstation cutting-edge technologies into the server environment, including the new virtual hardware and disk file format and functionality (including thin provisioning).

Because so many subsystems have had modifications and enhancements, they need to be broken out in more detail. It is easy to say it is a new operating system, but in essence ESX v4 is an enhancement to ESX v3 that simplifies administration, increases functionality and performance, and incorporates common customer-requested improvements.

The version comparison of ESX in this chapter looks at the following:

- The vmkernel (the nuts and bolts of the virtualization hypervisor)
- The boot process and tools of the console operating system or service console (SC)
- The changes to virtual networking (vNetwork)
- VMFS datastores
- Availability
- Backup methods
- Licensing methods
- Virtual hardware functionality
- VM management
- Server and VM security
- Installation differences
- VMware Certified Professional changes

Although the text of these sections will discuss the differences between ESX v3 and ESX v4, the author has left in the tables the data referring to ESX v2 so that the reader can see the growth of the product. In addition, the tables also included VMware ESXi where necessary. In many ways, ESXi is identical to ESX. The major differences are in how ESXi boots and the lack of a full-blown service console. Unless stated in the table or discussion, the differences also refer to ESXi as well as ESX.

VMware ESX/ESXi Architecture Overview

VMware ESX and ESXi is a multilayer architecture comprising multiple types of software. In Figure 2.1, we see that the top of the software stack is the application that runs within each guest operating system, which in turn runs within the virtual machine. The virtual machine is composed of the stack that contains the Application (APP) and the Guest Operating System (OS). Below the Guest OS is the virtual machine manager (VMM). Each VM talks to its own VMM, which is composed, among other things, with the virtual hardware in use by the VM.

The VMM is a software layer that provides interaction between the Guest OS and the kernel layer. The kernel layer is referred to as the vmkernel, which provides a layer that coordinates VMM interactions with the physical hardware and schedules the VMs to run on their associated physical CPUs. The vmkernel is the guts of the VMware ESX and ESXi. The vmkernel coordination includes the virtual network components, such as virtual switches that can connect the virtual machines to each other as well as to physical NICs. The vmkernel provides VMs access to the physical resources of the host. The vmkernel breaks down all resources into CPU, memory, network, and disk and provides these to the VMM layer for use by the VMs.

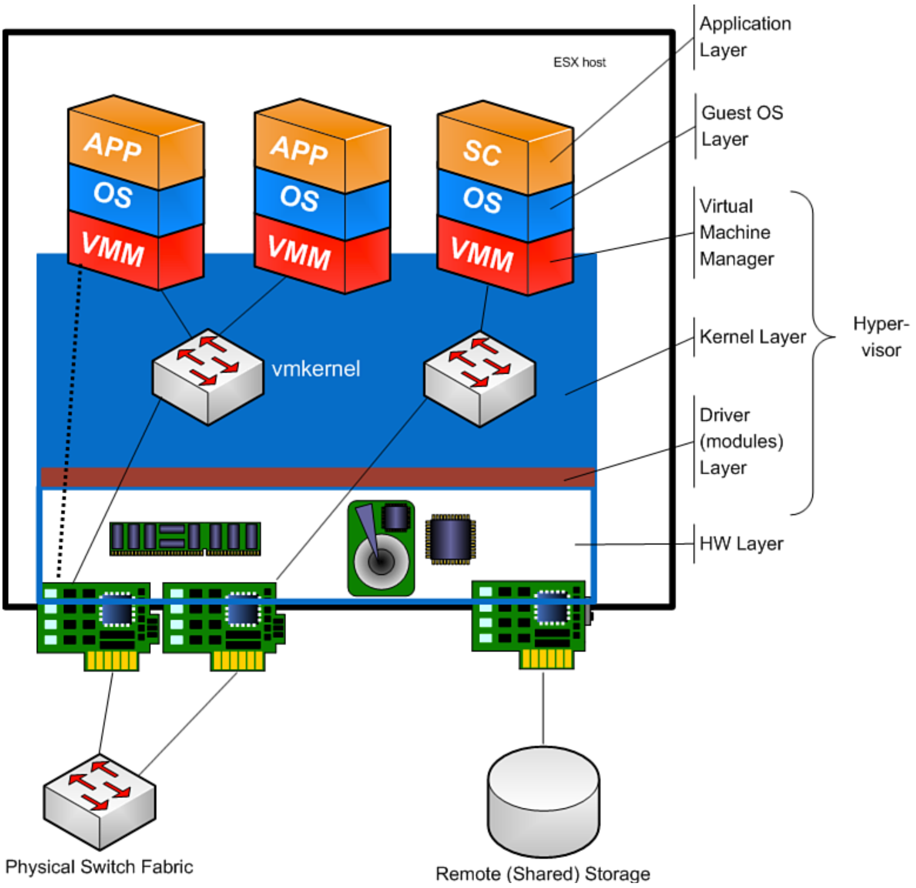

Figure 2.1 *ESX/ESXi architecture in a nutshell*

The vmkernel manages the physical devices within its core code but talks to them using drivers or modules that speak the language of the physical devices. Even though this is the case, each VM could possibly talk to one of the devices directly using a pass through mode supported by the vmkernel, which literally maps the device directly to the VM, bypassing many of the vmkernel layers (dashed line in Figure 2.1). The VMM, Kernel, and Driver or Modules layer compose what is referred to as a hypervisor.

VMs run within the hypervisor that coordinates and schedules access to the physical hardware required by the VMs. Hosts can be combined into clusters of hypervisors. Hypervisors can also communicate directly with each other via management and other networks that have physical and virtual components. VMs can communicate with each other and physical machines via virtual and physical networks.

In short, a hypervisor runs VMs and interacts with hardware.

vmkernel Differences

The heart of ESX is the vmkernel, and future enhancements all stem from improvements to this all-important subsystem. The new vmkernel supports new and different guest operating systems and upgrades to support the latest service console version and driver interactions. The vmkernel looks similar to a Linux kernel, but it is not a Linux kernel. The most interesting similarity is the way modules are loaded, and the list of supported modules has changed. Table 2.1 shows the standard modules loaded by each version of ESX.

Table 2.1 *Module Version Differences (Proliant DL380; Found Using vmkload –b Command)*

ESX v4	ESX v3.5	ESX v3.0	ESX 2.5.x	Comments
vmklinux	vmklinux	vmklinux	vmklinux	Linux interface
random				Random numbers
ehci-hcd usb-uhci usb-storage pclassify cbt				USB support
hid				Human interface devices

ESX v4	ESX v3.5	ESX v3.0	ESX 2.5.x	Comments
ipmi_msghandler				IPMI drivers
ipmi_sr_drv				
ipmi_devintf				
dm				
cosShadow	cosShadow			
vmci				VMCI (VM to VM Communication Interface)
vmkstatelogger				vmkernel state logging
libata	ata_piix	ata_piix	IDE CDROM	IDE CDROM
ata_piix				
aic79xx	aic7xxx aic79xx	aic7xxx	aic7xxx	Adaptec SCSI HBA for local tape device
bnx2	e1000	e1000	e 1000	Intel and Broadcom pNIC drivers
e1000e	e100	e100	e100	
e1000*	tg3	tg3e	bcm5700	
tg3*	forcedeth*			
forcedeth*	bnx2*			
cdp	tcpip	bond	bond	Cisco Discovery Protocol, vSwitch drivers, and network drivers
etherswitch	etherswitch			
hub	netflow			
tcpip2				
tcpip2v6*				
dvsdev				Distributed
dvfilter				Virtual Switch
deltadisk	deltadisk	deltadisk		Snapshots

continues

Table 2.1 *(Continued)*

ESX v4	ESX v3.5	ESX v3.0	ESX 2.5.x	Comments
lpfc820*	lpfc_740*	lpfcdd_7xx	lpfcdd_2xx	Emulex and Qlogic FC-HBA
qla2xxx	qla2300_707	qla2300_7xx*	qla2[23]00_xxx*	
cciss|vmw_satp_local	cciss	cciss	cpqarray*	RAID HBA
vmw_satp_default_aa	ips*	ips*	cciss	
vmw_satp_alua	aic...*	aic...*	ips*	
vmw_satp_cx	mptscsi_2xx*	aic...*		
vmw_satp_default_ap	sata_nv*			
vmw_satp_eva	sata_promise*			
vmw_satp_lsi	sata_svw*			
vmw_satp_symm	sata_vsc*			
vmw_satp_inv				
vmw_satp_eql				
vmw_satp_msa				
vmw_satp_svc				
vmw_satp_alua_cx				
vmw_psp_rr				
vmw_psp_mru				
nmp				Native multipath driver
		vmkapimod		vmkernel API module
vmfs2	vmfs2	vmfs2		VMFS-2
vmfs3	vmfs3	vmfs3		VMFS-3 and VMFS LVM-Driver
lvmdriver				
multiextent				extents
nfsclient	nfsclient	nfsclient		vmkernel NFS client

ESX v4	ESX v3.5	ESX v3.0	ESX 2.5.x	Comments
iscsi_trans	iscsi_mod	iscsi_mod		vmkernel iSCSI support
iscsi_linux	qla4010	qla4010*		
iscsi_vmk*	qla4022			
qla4xxx*				
shaper	shaper	shaper	nfshaper	vSwitch traffic shaper
migrate	migration	migration	migration	vmkernel vMotion
filedriver	filedriver			
vmkibft*				Fault Tolerance
tpm_tis*				Trusted Platform Module
fsaux*	fsaux*	fsaux		VMFS filesystem utilities

*Not loaded by default

In ESX v3, the split between the service console and the vmkernel was physical, but there was quite a bit of bleed-through nonetheless. As of ESX v3.5, this bleed-through had been nearly eliminated. This bleed-through was in the need for third-party management agents to properly control some aspects of the hardware; however, with ESX v4 VMware has pretty much done away with the need for third-party management agents by including new drivers to handle these needs. Such agents include the Dell Openmanage and HP Insight Management agents, which are now handled by the improved IPMI support.

With the introduction of ESX v4, VMware deprecated some modules that were present in earlier versions of ESX. If the devices that these modules support are a requirement for your ESX installation, you will not be able to upgrade to ESX v4. Table 2.2 lists the devices in ESX v3 that are missing from ESX v4, whereas Table 2.3 includes the differences between ESX 2.5 and ESX v3 for historical purposes. The developers of ESX v4 preferred to settle on modern hardware, and much of the older PCI or PCI-X hardware is obsolete. From a stability point of view, this is a very good thing. Minimizing the number and type of devices that must be supported enables the development team to focus their attention on building quality support for the devices that are supported.

Table 2.2 *ESX v3 Devices Obsolete in ESX v4*

Driver	Device
3c90x	3Com Etherlink 10/100 PCI NIC
e100	Intel 10/100 NIC
aic7xxx	Superseded by the aic79xx driver
sata_vsc	Vitesse VSC7174 4 port DPA SATA
mptscsi_2xx	Superseded by mptspi driver
DAC960	Mylex DAC960 RAID

Table 2.3 *ESX 2.5 Devices Obsolete in ESX v3*

Driver	Device
cpqarray	Compaq SmartArray devices earlier than the SmartArray 5300
gdth	GDT SCSI disk array controller
3c990	3Com EtherLink 10/100 PCI NIC with 3XP Processor
acenic	AceNIC/3C985/GA620 Gigabit Ethernet
ncr53c8xx	NCR53C SCSI
sym53c8xx	SYM53C SCSI
dpt_i20	Adaptec I20 RAID Driver
nfshaper	Traffic Shaper

Several other vmkernel features should be noted that are different between ESX v3 and ESX v4. The first and foremost change is the exposure of internal vmkernel constructs via well-defined APIs that allow third parties to add elements into the vmkernel. These APIs are vNetwork, vStorage, vCompute, and VMsafe, which are discussed throughout the rest of this book.

vStorage is a new marketing name for the virtual disk development kit (vDDK) that was available for ESX v3. The other APIs are all brand new and add major functionality.

In addition to these changes, with ESX v4, the vmkernel is now 64-bit and supports up to 1TB of memory and 320 VMs utilizing up to 512 virtual CPUs.

ESX Boot Differences

Simply put, the service console has been upgraded from being based on a variant of 32-bit Red Hat Enterprise Linux Enterprise Server 3 Update 8 to being based on a variant of 64-bit Red Hat Enterprise Linux Enterprise Server 5.1. ESX is in no way a complete distribution of GNU/Linux. Technically, it is not Linux at all, because the vmkernel is what is interacting with the hardware, and the service console (SC) is running within a VM. Legally, the vmkernel is not Linux either, because it is proprietary. Although the SC is a variant of GNU/Linux, it is a management appliance and not the operating system of ESX.

Even with the change in SC version, the rule that "no Red Hat updates should be used" has *not* been changed. All updates to the SC should come only from VMware. This is crucial. Consider the following: ESX consists of a single CD-ROM, whereas the official version of RHEL5 takes up five CD-ROMs. Therefore, they are not the same and should never be considered the same. For RHEL5, the method to configure any part of the system is to use the supplied system-config- scripts. These are not a part of ESX. Instead, there are a series of esxcfg- scripts that do not map one-to-one to the original Red Hat scripts.

The esxcfg- scripts, however, outlined in a later chapter, do map pretty well to the new management tool: vSphere Client (vSC). This allows for either the client to be used to configure an ESX host directly or through the use of a VMware vCenter server. Although there continues to be a web-based interface, it does not present a method to configure the ESX host or a way to create VMs.

ESX v4 has a kernel that is proprietary, the vmkernel, as well as a modified from the stock RHEL5 kernel that runs within the service console and therefore cannot be Linux. ESXi v4 has a kernel that is proprietary, the vmkernel, only. The modifications to the stock kernel enable the SC to manage the ESX hypervisor. The SC in ESX v4 sees only devices presented or passed through from the vmkernel and does not interact directly with the hardware unless using a pass-through device. Granted, the modifications for ESX are limited in scope to controlling the addition and removal of device drivers to the vmkernel and the ability to control virtual machine and virtual switch objects running within the vmkernel.

In ESX versions earlier than version 3, the vmkernel would load after the SC had fully booted, and the vmkernel would usurp all the PCI devices set to be controlled by the kernel options. In ESX version 3, this changed. The vmkernel loads first, and then the SC, which runs within a specialized VM with more privileges than a standard VM. In ESX v3 the SC was installed onto a local disk, and the VM accessed the local disk through a RAW pass-through SCSI device. In ESX v4, this has changed so that the RAW pass-through SCSI device is no longer used. Instead, the GNU/Linux environment lives within a virtual machine disk file (VMDK). This change further punctuates the difference between the hypervisor and GNU/Linux. So to repeat: The hypervisor is *not* Linux.

Table 2.4 lists the boot steps for ESX versions 4, 3.x, and 2.5.x, as well as the boot steps for ESXi. This documents the changes in the boot sequences for each version. Note that other than a few changes to the daemons used, such as sfbc instead of Pegasus for CIM, and the order of some startups (wsman), there are not many changes between ESX v3.5 and ESX v4. The most significant change is that IPMI has been moved to a vmkernel driver and is no longer part of the service console.

Table 2.4 *ESX Server Boot Steps*

ESX v4	ESX v3.x	ESXi	ESX 2.5.x	Comments
GRUB	GRUB	GRUB	LILO	Boot Loader (Boot strap process that loads the kernel).
N/A	N/A	N/A	kernel-2.4-9vmnix	Loaded by the boot loader.
vmkernel	vmkernel	vmkernel	N/A	vmkernel is loaded from the RAM disk associated with the kernel during the first phase of the kernel boot.
vmkernel devices	vmkernel devices	vmkernel devices	Linux devices	Devices loaded from the RAM disk associated with the kernel during the first phase of the kernel boot.
VM Booted	VM Booted	Start BusyBox	N/A	Boot the VM that runs the service console.
2nd phase kernel loaded	2nd phase kernel loaded	N/A	2nd phase kernel load	The second phase of the kernel is loaded into the administrative VM created when the vmkernel was loaded and becomes the kernel for the COS.
init	init	init	init	Process that loads all other processes and that is started by the second phase of the kernel boot.
S00vmkstart	S00vmkstart	N/A	S00vmkstart	S00 represents the first set of usermode programs to run. S00vmstart ensures that there is no other vmkernel process running; if there is, the vmkernel process is halted.

ESX v4	ESX v3.x	ESXi	ESX 2.5.x	Comments
S01vmware	S01vmware	hostd	N/A	S01 represents the second level of user mode programs to run on boot. In this case, the vmkernel network and storage modules are started.
S09firewall	S09firewall	N/A	N/A	S09 represents the tenth level of user mode programs to run on boot. In this case, the ESX firewall is started using esxcfg-firewall.
S10network	S10network	N/A	S10network	S10 represents the eleventh level of user mode programs to run on boot. In this case, the COS network is started.
S12syslog	S12syslog	N/A	S12syslog	S12 represents the thirteenth level of user mode programs to run on boot. In this case, the logging daemon syslog is started.
Moved to vmkernel driver	S14ipmi	N/A	N/A	S14 represents the fourteenth level of user mode programs to run on boot. In this case, the IPMI service is loaded.
S19slpd	N/A	N/A	N/A	S19 represents the twentieth level of user mode programs to run on boot. In this case, the Service Locator process loaded.
S21wsman	N/A	N/A	N/A	S21 represents the twenty-second level of user mode programs to run on boot. In this case, the VMware Web Services are started.

continues

Table 2.4 *(Continued)*

ESX v4	ESX v3.x	ESXi	ESX 2.5.x	Comments
N/A	S32vmware-aam	N/A	N/A	S32 represents the thirty-second level of user mode programs to run on boot. In this case, the Legato AAM service is loaded to support VMware HA and DRS.
N/A	S55vmware-late	N/A	N/A	S55 represents the fifty-sixth level of user mode programs to run on boot. In this case, the NAS and iSCSI vmkernel devices are initialized using esxcfg-nas and esxcfg-swiscsi tools.
S56xinetd	S56xinetd	N/A	S56xinetd	S56 represents the fifty-seventh level of user mode programs to run on boot. In this case, the Internet super-daemon xinetd is started. The vmware-authd server is now running inside the COS.
N/A	N/A	hostd	N/A	Starting Hostd on ESXi only.
S58ntpd	S58ntpd	ntpd	N/A	S58 represents the fifty-ninth level of user mode programs to run on boot. In this case, the Network Time Protocol Daemon is started.
S62vmware-late	N/A	N/A	N/A	S62 represents the sixty-third level of user mode programs to run on boot. In this case, the NAS and iSCSI vmkernel devices are initialized using esxcfg-nas and esxcfg-swiscsi tools.

ESX v4	ESX v3.x	ESXi	ESX 2.5.x	Comments
N/A	S85vmware-webAccess	N/A	N/A	S85 represents the eighty-sixth level of user mode programs to run on boot. In this case, the Web Based MUI is started.
N/A	S90pegasus (ESX v3.0 Only)	N/A	N/A	S90 is the ninety-first level of user mode programs to run on boot. The OpenPegasus Common Interface Model/ Web Based Enterprise Management server used for managing the ESX server.
N/A	N/A	N/A	S90vmware	S90 is the ninety-first level of user mode programs to run on boot. The vmkernel starts here and vmkernel devices are loaded after the vmkernel usurps PCI devices; in addition the vmware-serverd, and vmware-authd processes start.
N/A	N/A	N/A	S91httpd. vmware	S91 represents the ninety-second level of user mode program to run on boot. The MUI is now running.
S97vmware-vmkauthd	S97vmware-vmkauthd	N/A	N/A	S97 represents the ninety-eighth level of user mode programs to run on boot. The vmkernel authorization server is initialized inside the vmkernel.
S98mgmt-vmware	S98mgmt-vmware	N/A	N/A	S98 represents the ninety-ninth level of user mode programs to run on boot. The vmware host agent is now running. The host agent replaces the vmware-serverd server.

continues

Table 2.4 *(Continued)*

ESX v4	ESX v3.x	ESXi	ESX 2.5.x	Comments
N/A	N/A	sfcbd	N/A	Start Small Footprint CIM Broker Daemon
S98sfcbd-watchdog	N/A	sfcbd-watch-dog	N/A	S99 represents the hundredth level of user mode programs run on boot. The Small Footprint CIM Broker Daemon is launched.
N/A	N/A	slpd	N/A	Launch Service Location Protocol ESXi
N/A	N/A	vobd	N/A	Launch vobd service
N/A	S99pegasus (ESX v3.5 Only)	N/A	N/A	S99 is the one-hundredth level of user mode programs to run on boot. The OpenPegasus Common Interface Model/Web Based Enterprise Management server used for managing the ESX server.
S99vmware-autostart	S99vmware-autostart	N/A	N/A	S99 represents the one-hundredth level of user mode programs run on boot. The VMs that need to be autostarted are at this time.
S99vmware-vpxa	S99vmware-vpxa	N/A	N/A	S99 represents the one-hundredth level of user mode programs run on boot. The vCenter Agents are started if vCenter manages this node.
N/A	S99wsman	wsman	N/A	S99 represents the one-hundredth level of user mode programs to run on boot. The Web Services Management is started.
Login enabled	Login enabled	Login enabled	Login enabled	After the startup processes are run, the system is fully operational and login is enabled on the console.

In Table 2.4, we can also see the boot process for ESXi. ESXi boots entirely differently than ESX does. Specifically, ESXi does not launch a VM; instead it launches a Posix environment called BusyBox. BusyBox then launches very few user mode processes: just enough to support the necessary management daemons.

This is a significant change, because the BusyBox processes are running within the context of the vmkernel. This implies the vmkernel can run arbitrary code. In ESX, a clear distinction exists between the management console and the vmkernel. In ESXi this distinction is blurred.

Tool Differences

In addition to all the boot changes and OS changes, the ESX-specific commands have changed, forcing many custom scripts to need some form of rewriting. Table 2.5 shows the ESX v4 specific commands and what has been deprecated. Many information tools have been replaced; even so, more information is now available within ESX v4 using the new tools.

Table 2.5 *New and Deprecated ESX Commands*

ESX Command	Status	Functionality
esxcfg-addons	New	List all add-ons within the vmkernel, such as VMsafe drivers.
esxcfg-linuxnet	Deprecated	
esxcfg-vmhbadevs	Deprecated	Replaced by esxcfg-scsidevs
esxcfg-volume	New	Interact with SAN snapshot/replica volumes
esxnet-support	Deprecated	
esxcli	New	Tool to configure/query specific devices like corestorage, Native Multipath Plug-in, and iSCSI
vmfsqhtool, vmfsqueuetool	Deprecated	
vmkiscsi-device, vmkiscsi-ls, vmkiscsi-util	Deprecated	Replaced by vmkiscsiadm, vmkiscsi-tool
vmkperf	New	Command to look at performance information by event type (which is really clock-related data)
vmkvsitools	New	Command to interact with the vmkernel to grab useful information about processes, kernel options, and so on

continues

Table 2.5 *(Continued)*

ESX Command	Status	Functionality
vmsnap.pl, vmsnap_all, vmres.pl	Deprecated	ESX v2 style REDO mode. Removed from ESX v4.
vm-support	Changed	Different output
vmkload_mod		This command has not changed.

Virtual Networking

There are six major changes to virtual networking between ESX v4 and ESX v3.x:

- Introduction of the vNetwork Distributed Switch (vDS)
- Capability to add in third-party virtual switches, such as the Cisco Nexus 1000V
- Introduction of another vmkernel network for Fault Tolerance Logging
- Support for Jumbo Frames for NFS and iSCSI
- IPV6 implementation
- Introduction of Network IO Control (NetIOC) within the vNetwork Distributed Switch (ESX v4.1)
- Introduction of PVLANs when using vDS
- Introduction of Network vMotion when using vDS, in essence the ability to allow the network state to also move when a VM is vMotioned
- Introduction of Load Based Teaming (LBT), which allows the assignment of VM to pNIC to be changed based on the outbound load requirements within the vDS. (ESX 4.1)
- A subtle change in how iSCSI now works within the vNetwork

One other change will be discussed later, and that is the introduction of the VMsafe-net API that allows third-party security tools to work from within the hypervisor.

vNetwork Distributed Switch

The vNetwork Distributed Switch (vDS), available when you have an Enterprise Plus license, allows you to manage per ESX host virtual networks from the cluster level as well allowing for more switch functionality. The vDS is a datacenter or cluster construct seen within VMware vCenter but not necessarily at each host. With previous versions of ESX, you were required to maintain identical virtual network labels across all your hosts. vDS takes care of this for you (as do host profiles, which we will talk about later).

The vDS also supports the concept of Private VLANs (PVLAN), which are Access Control List (ACL) protected VLANs. PVLANs can span more than one ESX v4 host. PVLANs are implemented using the internal functionality called dvFilter. At this time, however, dvFilter has not been exposed to an API for use.

vDS adds the concept of Network vMotion to allow vMotion to also include the network state of a VM as it moves around the virtual network controlled by vDS. This state information is required to allow dvFilter to be used after a vMotion occurs.

vDS works for every type of virtual switch port, including vmkernel ports, which includes the ability to create PVLANs for the vMotion network and other interesting enhanced controls of the virtual networks within the virtual environment.

As of ESX v4.1, vDS now includes Network IO Control (NetIOC) which is a form of traffic shaping where you can split the pNICs connected to a vDS by shares for all outbound traffic. In addition, you can set limits for all outbound connectivity. While traffic shaping does exist for ports attached to VMs, this is the only traffic shaping that exists for outbound traffic on given pNICs. NetIOC is by pNIC on each host. The other new feature is Load Based Teaming, which allows for the mapping of VM to pNIC to be modified based on outbound load congestion but no more often than every 30 seconds.

Third-Party Virtual Switches

ESX v4 exposes the vNetwork API, which allows third parties to create virtual switches. One such virtual switch is the Cisco Nexus 1000V, which you can add into the system if you have the Enterprise Plus licensing level. The Cisco Nexus 1000V works quite a bit differently than normal VMware virtual switches. Specifically, there is only a single Cisco Nexus 1000V allowed per host. In essence, the Cisco Nexus 1000V becomes a new edge switch sitting before the virtual machines.

Even if the Cisco 1000V is in use, it is possible to use normal VMware vSwitches for vmkernel devices and other virtual network needs.

Fault Tolerance (FT) Logging

There is another vmkernel network available when you add virtual portgroups within ESX v4—the Fault Tolerance Logging (FT Logging) network. Like vMotion, the FT Logging network passes critical information between two hosts participating in FT. This information is critical to keeping the virtual machine in lockstep, so it must utilize a high-speed network.

iSCSI Participation

In ESX v3, the service console was required to participate in all forms of iSCSI networking. The service console was used for authentication, whether you were using CHAP or not. In ESX v4, the service console is no longer required to participate in the iSCSI network and therefore will lead to better isolation.

IPv6 Support

IPv6 support now exists for the service console and vmkernel devices within an ESX v4 host. This allows ESX to be used within IPv6 only and mixed IPv6/IPv4 networks. However, iSCSI support using IPv6 is experimental.

VMsafe-Net

VMware has exposed another set of APIs to allow third parties to manage security within the vmkernel which can be authoritative about what is connected to the ESX host. VMsafe-net is one of these APIs. VMsafe-net sits just before the vNIC attached to a VM and not within any virtual switches. This functionality does not exist within ESX v3 and is entirely new for vSphere.

Summary

Table 2.6 summarizes the virtual network functional differences between ESX v4, 3, and 2.5.x.

Table 2.6 *Virtual Network Functional Comparison*

vNetwork Functionality	ESX v4	ESX v3	ESX 2.5.x
Logical ports	Settable to 8, 24, 56, 120, 248, 504, or 1016 ports	Settable to 8, 24, 56, 120, 248, 504, or 1016 ports	Fixed 32 Ports
Portgroups	Many per vSwitch	Many per vSwitch	Many per vSwitch
802.1q	Supported	Supported	Supported

vNetwork Functionality	ESX v4	ESX v3	ESX 2.5.x
802.3ad	Supported	Supported	Supported
NIC Teaming	Supported	Supported	Supported
Source Port ID Load Balancing	Supported	Supported	N/A
Source MAC Address Load Balancing	Supported	Supported	Supported
Source IP Address Load Balancing	Supported	Supported	Supported
GUI Settable standby pNICs	Supported	Supported	N/A
Rolling mode for standby pNICs	Supported	Supported	N/A
vMotion through router/gateway	Supported	Supported	N/A
SC required to participate in iSCSI network	No	Yes	N/A
VMCI	Supported	Supported (ESX v3.5 only)	N/A
IPv6	Supported	N/A	N/A
Virtual Distributed Switch	Supported	N/A	N/A
Network vMotion	Supported with vDS	N/A	N/A
Network IO Control (NetIOC)	Supported with vDS and ESX 4.1	N/A	N/A
Load-Based Teaming	Supported with vDS and ESX 4.1	N/A	N/A
Third-Party vSwitch	Cisco Nexus 1000V	N/A	N/A
Private VLAN	Supported within vDS & Cisco Nexus 1000v	N/A	N/A
Jumbo Frames Support	Supported for iSCSI and NFS	Supported but NOT for iSCSI or NFS	N/A
VMsafe NET	Supported	N/A	N/A
iSCSI vmkernel	Supported	Supported	N/A

continues

Table 2.6 *(Continued)*

vNetwork Functionality	ESX v4	ESX v3	ESX 2.5.x
vMotion vmkernel	Supported	Supported	Supported
NFS vmkernel	Supported	Supported	N/A
Service Console vSwif	Supported	Supported	N/A
Management Console vmkernel (ESXi only)	Supported	Supported	N/A
FT vmkernel	Supported	N/A	N/A
Multilayer vSwitch	Only with VM between each vSwitch	Only with VM between each vSwitch	Only with VM between each vSwitch
Beacon Monitoring	Supported	Supported	Supported

Storage

The biggest change from ESX v3 to ESX v4 occurred within the storage subsystem. Although the VMFS-3 major version number did not change, there are many changes to storage for the better, which included a minor version number change from VMFS 3.31 to VMFS 3.32. However one item has already bitten some people performing upgrades to vSphere; the VMFS-3 file system supports up to 2TB LUNs whereas ESX v4 supports up to (2TB minus 512 bytes) of space. This small change of not supporting a full 2TBs will create a little havoc for those doing upgrades that have LUNs sized exactly 2TBs.They will have to find a way to re-create the LUNs.

Important Note

ESX v4 supports LUNs up to 2TB – 512bytes only.

Migration to ESX v4 from v3 with LUN sizes exactly 2TBs will require LUN modifications.

Other changes to storage introduced with ESX v4 include the capability to grow a VMFS volume, Multipath Plug-in support, extension of virtual disks on-the-fly, storage path load balancing, Jumbo Frames support for iSCSI and NFS (as well as other improvements), FCoE support, and the capability to better use storage device LUN snapshots.

In addition to all these other changes, ESX v4 has simplified the creation of thin provisioned VMs. Although not a new concept, thin provisioning support has been improved within all the management tools.

Grow a VMFS Volume

One of the most significant changes to ESX v4 is the long-awaited capability to manage the size of a VMFS by growing the LUN underneath the VMFS, and then being able to grow the VMFS on top of the LUN. This capability alleviates the need to use extents for smaller VMFS to gain more space or the need to destroy and re-create LUNs bigger than originally planned. Even though you can grow a LUN, the LUN size limits still apply, which are 2TBs – 512 bytes.

Storage IO Control (SIOC)

ESX v4.1 introduces the concept of per virtual disk SIOC for Fibre Channel and iSCSI connectivity. SIOC allows you to set how to split up the storage controller outbound traffic per virtual disk. In other words, SIOC provides Fibre Channel and iSCSI traffic shaping based on a number of shares assigned to each virtual disk on a given LUN presented over Fibre Channel and iSCSI.

Multipath Plug-in (MPP)

The standard multipath capability for ESX has now been modularized within the kernel. ESX has had support for multipath failover since ESX 2.x days. In ESX v3.5, VMware added experimental support for multipath load balancing. These two features exist within the Native Multipath Plug-in (NMP).

Multipath in the storage world has always implied failover, link aggregation, and link load balancing. Multipath makes it possible to improve overall throughput within a server by adding more links (FC-HBAs or Ethernet) to the SAN through multiple controllers.

Now a third party can create its own multipath plug-in and add link aggregation and improvements to load balancing and failover into an ESX v4 host. One such MPP available as of this writing is EMC's PowerPath module written specifically for ESX v4. To use MPP, however, you need to have an Enterprise Plus license.

iSCSI and NFS Improvements

ESX v4 improves overall iSCSI and NFS performance by introducing the capability to use Jumbo Frames with these storage protocols. Jumbo Frames allow more data to be transferred per Ethernet frame (packet) than the default size.

This one change improves overall performance because fewer packets will be transferred, which implies less TCP overhead.

Furthermore, with iSCSI, there has been one other major change mentioned previously. This change is the removal of the requirement that the management console participate within the iSCSI network for authentication reasons. This improves the overall security of using iSCSI within the virtual environment.

FCoE

Fibre Channel over Ethernet (FCoE) using Converged Network adapters was experimentally supported in ESX v3.5. It is fully now supported in ESX v4 using several different 10GB FCoE converged network adapters. FCoE is really a networking solution, but because the vast majority of all bandwidth belongs to the storage channel, it is often considered a storage protocol.

FCoE can reduce the overall cabling required by an ESX host, yet could impact the overall security of the virtual environment. It is best to consider all options when using FCoE. In many cases, the reduction in cabling without loss of functionality is the way to go.

Storage Summary

The ESX Storage subsystem has seen many improvements in ESX v4. The key enhancements include the capability to grow VMFS volumes and the introduction of the modularized multipath capabilities, which allow third parties to add multipath drivers into ESX v4. Table 2.7 presents a summary of the datastore options by ESX version.

Table 2.7 *Datastore Functional Comparison*

Datastore Functionality	ESX v4	ESX v3	ESX v2.5.x
VMFS on SAN	Supported	Supported	Supported
VMFS on SCSI	Supported	Supported	Supported
VMFS on IDE	N/A	N/A	N/A
VMFS on iSCSI	Supported	Supported using iSCSI Initiator within COS. Requires vmkernel device be attached as a port to the COS vSwitch.	N/A

Datastore Functionality	ESX v4	ESX v3	ESX v2.5.x
VMFS on SAS	Supported	Supported	N/A
VMFS on SATA	Supported	Supported (ESX v3.5 only)	N/A
NFS	Supported	Supported for 2GB sparse files only, which is the old style Workstation, GSX, or template file formats	N/A
VMFS-1	R/O	R/O	R/O
VMFS-2	R/O	R/O	R/W
VMFS-3	R/W	R/W	N/A
LUN Size	2TB – 512 bytes	2TB	2TB
LUN Count	256 (but only 128 at a time)	256 (but only 128 at a time)	128
Thin Provisioning	Supported (through Management tools)	Supported	N/A
vStorage API	Supported	Supported as VDDK	N/A
FCoE	Supported	Supported (ESX v3.5 only)	N/A
Load Balancing	Supported	Experimental	N/A
Multipath Plug-in	Supported	N/A	N/A
VMFS Volume Grow	Supported	N/A	N/A
LUN Resignature	Supported	Supported	Supported
Storage IO Control (SIOC)	Supported w/ESX or ESXi v4.1 only	N/A	N/A
Default Disk. MaxLUN Setting	128	128	8
Access to COS File Systems	None	Limited to JUST /vmimages	Full Access
COS Access to Datastores	Supported	Supported	Supported

Availability

ESX version 4 does not introduce any new virtual resources into ESX. The basic four still exist: CPU, memory, network, and storage. However, ESX version 4 adds new availability constructs. The new constructs are Host Profiles, Fault Tolerance, Dynamic Power Management, VMware High Availability improvements, and official support for Storage vMotion.

Host Profiles

Host Profiles does not sound much like an availability component of the service because it enables you to manage the configurations of your ESX hosts from within vCenter. However, it is an availability control mechanism because it will alleviate the common issues of misnamed or misconfigured networking controls that currently plague large installations with respect to virtual networking. As we will see in a later chapter, virtual networking requires that the vSwitch and Portgroup names be identical (case sensitive) across all hosts in order for vMotion, High Availability, and other migrations to be seamless. Host Profiles can maintain these configurations across a multitude of ESX hosts and even apply an existing configuration to newly installed systems.

With ESX v3, this was achieved by the use of scripts written by the administrators. These scripts would configure virtual networking, storage, high availability, and other ESX specific options. Host Profiles moves toward eliminating the need for such scripts, other than one to apply a chosen host profile with the appropriate networking options (vmkernel IP addresses for example).

Host Profiles is available only with an Enterprise Plus license of VMware ESX and, as we will see in Chapter 3, "Installation," can be used in evaluation mode to aid in the upgrade from ESX v3 to ESX v4 for more than one ESX host. Given the license level availability of Host Profiles, there will still be a need for scripts to achieve the same features and configure many more. Host Profiles can only modify those things within ESX exposed by the vSphere Software Development Kit (SDK). Scripts that perform security hardening, for example, will still be necessary.

Host Profiles enables the quick deployment and recovery of an ESX v4 host, which improves the overall availability.

Fault Tolerance

Fault Tolerance (FT) provides a mechanism to keep a VM alive even if the host on which it is running fails. Unlike VMware HA, which will reboot a VM on a new host, with FT the VM keeps running with no interruption. FT does this by keeping a shadow copy of a single vCPU virtual machine using VMDKs and/or virtual RDMs on shared storage in lockstep with the running virtual machine. The shadow copy would run on another ESX v4 host that shares the same storage used by the ESX v4 host where the VM is already running (primary VM). VMware achieves this using its vLockStep technology, which uses a private fault tolerance logging network to keep the primary and shadow copy VMs in sync. In the case of a host or VM failure the FT shadow copy VM would be promoted to the primary VM as soon as such a failure is detected (almost immediately).

When FT is enabled, the shadow copy of a VM is created on a new host by first performing some parts of a vMotion, namely starting a VM linked to the same datastore on the new host, performing a memory copy, and then copying the CPU state. After that is achieved, the vLockStep technology kicks in and the FT logging network is used to communicate nondeterministic events (keystrokes, mouse movements, I/O events, and so on) from the primary VM to the shadow copy. Each VM will process the events independently; however, everything between the nondeterministic events is deterministic, so you know that the vCPUs are in a consistent state between the VMs. This technology builds on the shadow copy VM created normally during a vMotion.

So why is FT for multiple vCPUs a difficult problem to solve?

Keeping one vCPU in vLockStep with another just requires the transfer of nondeterministic events from one VM to its shadow copy. However, when you deal with more than one vCPU, you also deal with timing issues between the vCPUs, which implies that the previously understood deterministic events between nondeterministic events are no longer deterministic. These timing issues deal with when instructions on each of the vCPUs are issued. If even one instruction in one vCPU is issued before the appropriate one within another vCPU, the operation could manipulate the wrong chunk of memory, producing incorrect results. which implies the VM's applications would behave poorly on the shadow copy and at best case keep running but at worst case crash the application as well as the VM with some sort of random violation. This is actually the classic multithreaded application race condition. If you consider each vCPU as its own thread (which most likely they are) you can see that locking and timing information is all important when discussing FT over multiple vCPUs. This has yet to be solved by the world's threads developers. Thread debuggers solve this problem by almost serializing threaded programs, which greatly impacts performance. Although useful for debuggers, this approach does not work for virtual machines.

Because FT has a limit of currently supporting only a single vCPU, the utilization of this capability is limited in scope. For example, a vCenter Server running within a VM often requires two vCPUs; given this, it is not usually a candidate for FT, however much we would like it to be. Unfortunately, FT for multiple vCPUs is a very difficult problem to solve. Eventually it will be, and FT will get widespread use as one more availability tool.

Dynamic Power Management

Dynamic Power Management (DPM) was experimental within ESX v3. DPM allowed vCenter Server to move VMs off an ESX host to other hosts using vMotion, then would power off the ESX host to save on electricity use. For significantly underutilized ESX clusters, this feature allows lull times to use fewer hosts. An ESX host would be powered on using wake-on-lan (WoL), IPMI, or HP ILO technology when the lull times were over, and VMs would automatically be vMotioned back to the waiting host using Dynamic Resource Scheduling. WoL is the least reliable method to wake a host.

ESX v4 moves DPM from experimental to fully supported and adds other mechanisms to use instead of WoL. These additions are use of IPMI and the HP ILO cards to power on and power off hosts. These other technologies are much more specific and directed than WoL, which is still supported. These technologies alleviate false WoL starts due to existing network traffic such as AD broadcasts and other directed requests.

High Availability (HA) Improvements

HA has been improved so that it is now aware of hosts in maintenance mode. There are also improvements in the admission control of nodes within the cluster with many more controllable options. In ESX v3, there were many HA-related problems because of nodes suddenly dropping out of an HA cluster or HA failing for relatively unknown reasons. Many of these issues have now been fixed.

vMotion

vMotion is an availability tool that allows you to move VMs from one host to another host without downtime, as long as the VM used shared storage and both hosts (target and source) could see the storage). ESX v4 introduces some new capability into vMotion as well. Whereas vMotion is used to move VMs from host to host when combined with a vDS, it can also move the network state from host to host (Network vMotion). With ESX v4.1, EVC has also been improved to add another EVC mode. Currently there are EVC modes for AMD CPUs and one for Intel CPUs. ESX v4.1 changes the AMD mode to be a mode with 3Dnow! support and one without 3Dnow! support, because AMD has dropped 3Dnow! from some of its processors.

Storage vMotion

Storage vMotion allows the virtual disk of a VM to be moved from datastore to datastore within a single ESX or ESXi host with no downtime to the VM.

Storage vMotion has moved from the experimental state in ESX v3 to fully supported, with its own vSphere Client integrations. You no longer need to use a third-party plug-in to gain this level of integration. In addition, Storage vMotion is no longer bundled with vMotion, which implies you need at least an Advanced license to use this technology.

Availability Summary

VMware ESX and ESXi are all about redundancy and availability. Many of the major features added to each version of ESX are to improve overall availability and redundancy. When planning ESX, nearly all plans will include many, if not all, of the availability tools and functions.

Table 2.8 provides a comparison of the how the versions support virtual resources.

Table 2.8 *Virtual Resource Functional Comparison*

Functionality	ESX v4	ESX v3	ESX v2.5.x
CPU	Supported	Supported	Supported
Memory	Supported	Supported	Supported
Disk	Supported	Supported	Supported
Network	Supported	Supported	Supported
Resource Pools	Based on CPU/ Memory Resources Utilization Only	Based on CPU/ Memory Resources Utilization Only	N/A
Clusters	Supported	Supported	N/A
Distributed Resource Scheduling	Based on CPU/ Memory Resource Utilization Only, requiring an ESX Cluster	Based on CPU/ Memory Resource Utilization Only, requiring an ESX Cluster	Extremely limited via HPSIM Virtual Machine Manager plug-in
Dynamic Power Management	Based on CPU/ Memory Resource Utilization Only, requiring an ESX Cluster	Based on CPU/ Memory Resource Utilization Only, requiring an ESX Cluster (experimental)	N/A
Fault Tolerance	Requiring an ESX Cluster	N/A	N/A
Host Profiles	Requiring an ESX Cluster	N/A	N/A

continues

Table 2.8 *(Continued)*

Functionality	ESX v4	ESX v3	ESX v2.5.x
Storage vMotion	Supported	Experimental but widely used	N/A
High Availability	Full Support via Legato Automated Availability Management requiring an ESX Cluster	Full Support via Legato Automated Availability Management requiring an ESX Cluster	By hand/poor man's cluster

Disaster Recovery and Business Continuity Differences

VMware ESX and ESXi implementations are all about redundancy, but what is redundancy without the capability to improve uptime via Business Continuity (BC) or disaster avoidance mechanisms? Hand in hand with BC is Disaster Recovery (DR). DR has also been improved within the environment, either by improving backup capabilities or data replication. With ESX or ESXi v3, the only VMware backup mechanism was to use VMware Consolidated Backup and for data replication was the Site Recover Manager (SRM). These tools worked quite differently from each other.

VMware Consolidated Backup made use of a backup server that mounted the VM datastores directly from the SAN, NAS, or iSCSI Server. The backup server is used to offload backups from the actual ESX or ESXi hosts, using the backup server would, working with VMware vCenter, copy the virtual disk blocks direct from SAN after a snapshot was created through vCenter. This resulted in faster backups.

Site Recovery Manager, on the other hand, would aid in LUN to LUN duplication by allowing the storage device to communicate with vCenter to create snapshots and quiesce disks in order to allow the data for the VMs to be copied with great accuracy.

The idea behind both VCB and SRM was to end up with non-crash-consistent backups or replication of VMs. A crash-consistent backup leads to the possibility of quite a bit of testing and repair when problems exist. If the resultant VM is not crash consistent, the recovery time is shortened, and there is less overall risk of failure when there is a critical recovery required. VMware achieves this by having the quiesce scripts tie directly into VSS on Windows machines and use of the sync command to ensure all disk writes have completed before backup or replication can continue.

VMware Consolidated Backup is still supported within ESX v4 but is basically replaced by VMware Data Recovery (VDR) to centralize the management of backups. VDR makes use of the vStorage API to make backups of VMs using a virtual appliance. Similar to VCB, VDR supports using SAN as well as network transfers. Unlike VCB, VDR performs data de-duplication to save on disk space. Many of the third-party backup tools, such as PhD Virtual Backup for VMware ESX, Veeam Backup, and Vizioncore vRanger DPP, make use of the vStorage change block tracking (CBT) functionality. CBT makes use of the block map made when a snapshot is created to know exactly which blocks of a virtual disk have changed since the last time a snapshot has been made. This block map represents every block within a VMDK or a virtual RDM. Every time a block has changed, the block map gets updated that the corresponding block of data has changed. If a backup tool tracks the block map, it can be used to reduce the overall amount of data to be backed up on incremental backups. It also implies you do not need to rely on the Guest OS within the VM, but on the vStorage API only. When you have terabytes of data, backup times need to be reduced and CBT allows this to happen.

SRM allows the SAN, NAS, or iSCSI servers to make use of the vStorage API as well to increase their overall replication throughput. SRM is the glue between the virtual environment management and the hardware LUN replication mechanisms.

Table 2.9 summarizes the backup functions by ESX version.

Table 2.9 *Backup Functional Comparison*

Functionality	ESX v4	ESXi	ESX v3	ESX v2.5.x
vmsnap.pl	N/A	N/A	Deprecated	Supported
Snapshots	Supported	Supported	Supported	N/A
VMware Consolidated Backup	Supported	Supported (paid version only)	Supported	N/A
VMware Data Recovery	Supported	Supported (ESXi v4 and paid version only)	N/A	N/A
VDR De-Duplication	Supported w/VDR only	Supported (ESXi v4 and paid version only)	N/A	N/A
Site Recovery Manager	Supported	Supported (paid version only)	Supported	N/A

continues

Table 2.9 *(Continued)*

Functionality	ESX v4	ESXi	ESX v3	ESX v2.5.x
Third-Party Backup	vRanger, Veeam, PhD Virtual Backup	Paid version only	vRanger,Veeam, PhD Virtual Backup	vRanger,esXpress
LUN Mirroring	Supported only by datastore appliance	Supported only by datastore appliance	Supported only by datastore appliance	Supported only by SAN appliance

Virtual Hardware

ESX v4 has some interesting improvements to the virtual hardware presented to a VM as compared to ESX v3. There is now support for Paravirtualized SCSI (PVSCSI) to improve overall disk IO; an improved third-generation paravirtualized network driver (VMXNET 3); the capability to hot add memory and vCPUs for specific guest operating systems; the capability to implement eight-way vSMP VMs, which requires an Advanced or Enterprise Plus license; support for 255GBs of memory per VM; additional VMDK disk types (SAS and IDE); and the capability to use VMDirectPath to bypass the ESX v4 virtual network layer completely or to directly access any PCI device if the host has Intel VT-d or AMD IOMMU support and the PCI device supports Single Root IO Virtualization.

Guest OS customization continues to support sysprep, so it now includes Windows 2008. The open source customization tools now support more Linux operating systems, such as Debian and Debian-based distributions such as Ubuntu.

All this new functionality requires the use of the virtual hardware version 7, yet virtual hardware version 4 is still supported. This enables the migration of VMs from ESX v3 to ESX v4 without requiring any changes, which improves upgradeability from ESX v3 to ESX v4. Virtual hardware version 7 is also the same virtual hardware used by the new VMware Workstation 7.

With ESX v3.5, VMware added VIX, VMCI, and vProbe functionality to virtual machines. This functionality has not significantly changed with ESX v4. VIX enables the administrator to run commands within a VM from a remote location, and VMCI is an inter-VM communication channel that bypasses the virtual network layer. vProbe enables better debugging of virtual machines and to determine exactly what the VM is doing.

Table 2.10 summarizes the virtual hardware comparison between ESX v4 and ESX v3.

Table 2.10 *Virtual Hardware Functional Comparison*

Functionality	ESX v4	ESX v3	ESX 2.5.x
vCPU (vSMP)	1-8 including odd numbers	1, 2, or 4	1 or 2
USB	Yes (with Update 1)	No	No
MSCS	Up to 8 nodes	Up to 8 nodes	2 node
Configuration File Location	With VMDK	With VMDK	In /home
Memory	255GB	64 GB	3.6 GB
SCSI HBA	PVSCSI/SAS/ LSI|BUSLogic	LSI|BUSLogic	LSI|BUSLogic
IDE VMDK	Supported	N/A	N/A
SAS VMDK	Supported	N/A	N/A
vNIC	Flexible	Flexible	Selectable pcnet32
VMXNET	Version 1,2 or 3	Version 1 or 2	Version 1
PVSCSI	Supported for non-boot and boot in ESX v4.1	N/A	N/A
Hot Add Disk	Supported[1]	Supported[1]	N/A
Hot Add CPU	Supported[1]	N/A	N/A
Functionality	**ESX v4**	**ESX v3**	**ESX 2.5.x**
Hot Add Memory	Supported[1]	N/A	N/A
NPIV	Supported (disk only)	Supported (disk only)	N/A
VMDirectPath	Supported[1]	Supported[1]	N/A
64-Bit Guests	Only on 64-bit hardware	Only on 64-bit hardware	N/A
vHardware Level	> Workstation 5.5	> Workstation 5.5 < 7.0	<Workstation 5.5
Snapshots	Supported	Supported	N/A
vProbes	Supported	N/A	N/A
VIX	Supported	Supported (ESX v3.5)	N/A
VMCI	Supported	Supported (ESX v3.5)	N/A

continues

Table 2.10 *(Continued)*

Functionality	ESX v4	ESX v3	ESX 2.5.x
vSWP	With VMDK	With VMDK	Global

[1] With proper support within guest operating system

Virtual Machine and Server Management

There are only a few changes to the way VMs are managed when moving to ESX v4. VMware has rewritten the Virtual Infrastructure Client user interface and named it the vSphere Client. The look and feel is quite a bit different, but the most important things are the same, such as editing VM settings and moving around the different host and VM views.

In addition to the changes to the management client, which are numerous, VMware has improved the vSphere SDK to expose even more capability. There is also a new version of the vSphere SDK in Beta named Onyx. Onyx exposes even more of the functionality via powershell cmdlets to interact with the virtual distributed switch and other new features. The vSphere SDK has language bindings for nearly every language available, most notably PowerShell, Perl, .NET, and Java.

At the time of ESX v3, an appliance was introduced named Virtual Infrastructure Management Appliance (VIMA). VIMA has been replaced by the Virtual Management Appliance (vMA), which is a Just Enough Operating System (JeOS) version of Linux with the Perl vSphere SDK language bindings installed and scripts written to make use of them. Also included in this is the Remote CLI commands. The goal of the vSphere SDK, vMA, and Remote CLI is to alleviate the need to ever log in to the management console to perform any work.

All the improvements imply that ESX v4 is highly automatable.

Table 2.11 summarizes the virtual management differences across versions.

Table 2.11 *Virtual Management Functional Comparison*

Functionality	ESX v4	ESXi	ESX v3	ESX 2.5.x
webAccess	Supported for access to VMs ONLY	Supported for access to VMs ONLY	Supported for access to VMs ONLY	Supported, independent of MUI or CLI
vSphere Client	**Supported**	**Supported**	**Supported**	**R/O**
Virtual Infrastructure Client	N/A	Supported (ESXi v3 Only)	Supported	R/O

Functionality	ESX v4	ESXi	ESX v3	ESX 2.5.x
Remote CLI	Supported	Supported	Supported	N/A
vSphere (VI) SDK	Supported	Supported	Supported	N/A
vMA	Supported	Supported	Supported	N/A
Command Line (CLI)	Supported, Integrated with vSphere Client	Available in unsupported mode	Supported, Integrated with VIC	Supported, independent of VC
vCenter	Supported, integrated with CLI and webAccess	Supported, integrated with CLI and webAccess	Supported, integrated with CLI and webAccess	Supported, independent of MUI and CLI
VM Creation	No special requirements for 2.6 Kernels; in addition, Virtual Floppy image for Windows XP SCSI driver now a part of ESX Install	No special requirements for 2.6 Kernels; in addition, Virtual Floppy image for Windows XP SCSI driver now a part of ESX Install	No special requirements for 2.6 Kernels; in addition, Virtual Floppy image for Windows XP SCSI driver now a part of ESX Install	2.6 Kernel Versions of Linux require Custom VM creation modes

Security Differences

There are a few security differences between ESX v4 and ESX v3.

The first is the availability of VMware vShield Zones. vShield Zones implements an inline firewall appliance that sits between two virtual switches within your virtual network. vShield Zones requires an Advanced or higher license to use. In addition, vShield Zones is a Zone firewall based on iptables without NAT or port redirection support as such is not designed for edge firewall use. After v4.1, there is a new suite of VMware vShield security tools, of which Zones is one such tool. There is now vShield Edge (Edge Firewall), vShield App (VMsafe Firewall), and vShield Endpoint (used by Antivirus vendors).

The second is VMware VMsafe. VMsafe is a set of APIs that splits functionality between slowpath and fastpath. The fastpath implies the use of a third-party driver within your hypervisor, whereas the slowpath uses a virtual appliance that the fastpath driver talks to for management purposes. With VMsafe a certain amount of intelligence can live within the fastpath driver, but any heavy lifting must be performed within the slowpath appliance. The first use of VMsafe is the VMsafe-net APIs, which allow third-party vendors to implement firewalls.

Antivirus vendors are looking at using VMsafe-memory APIs to do per ESX v4 host virus checking instead of per VM.

Another security difference is the capability to use a trusted platform module (TPM) to do disk integrity checks to ensure that the vmkernel has not been modified. This does require a TPM device within your server to enable. In addition, for ESX v4.1, the TXT extensions for TPM are supported.

With the introduction of the Cisco Nexus 1000V and Virtual Distributed Switch, there is also now the concept of Private VLAN support or VLANs that have an access control list to control who can use these VLANs.

The security differences between versions are summarized in Table 2.12.

Table 2.12 *Security Functional Comparison*

Functionality	ESX v4	ESXi	ESX v3	ESX v2.5.x
iptables/Firewall	Installed and configured	N/A	Installed and configured	On media not installed by default
vShield tools	Supported	Supported (ESXi v4)	N/A	N/A
Third-Party Inline Firewalls	Supported	Supported	Supported	Supported
VMsafe	Supported	Supported (ESXi v4)	N/A	N/A
TPM	Supported	Supported (ESXi v4)	N/A	N/A
Private VLANS	Supported with Virtual Distributed Switch or Cisco Nexus 1000V	Supported with Virtual Distributed Switch or Cisco Nexus 1000V (ESXi v4)	N/A	N/A
Data-Link Layer Security	On vSwitch and Portgroup and Port with dVSwitch	On vSwitch and Portgroup	On vSwitch and Portgroup	With VM

Installation Differences

Minor differences exist between ESX v4 and ESX v3 in the installation or upgrade routines. Although the install can work on unsupported SCSI or RAID hardware, the boot of ESX will generally fail. So it is important to use a supported SCSI or RAID adapter. It is possible to upgrade various versions of ESX earlier than version 3, and that list is fully available in Chapter 3. Not all versions of ESX support this upgrade, however.

Three noticeable differences exist between the upgrade routine for ESX v4 and ESX v3. In ESX v2.x, you were requested to add a license key during install. With ESX v3 this requirement was dropped; with ESX v4, you are again requested to enter a license key. However, this can be delayed until you use the vSphere Client for the first time, whether through vCenter or direct access. The second is the automated disk layout during upgrades. During upgrades the VMware default disk layout is assumed to be overriding your existing settings. Another significant improvement in the ESX v4 install routine is that you can now choose your initial service console NIC rather than the installer automatically choosing whatever NIC shows up first.

ESX versions earlier than version 3 install on and run from any disk media supported by Red Hat Linux version 7.2. With ESX v3, although it is possible to install onto disk media supported by RHEL3-ES, is it not possible to run from anything but one of the supported SCSI/RAID devices. With ESX v4, you can install onto media supported only by ESX now because the Red Hat based service console version no longer matters. If the drivers do not exist inside ESX for your installation device, there is no way to install onto that media. ESX v4 now installs everything onto a VMFS with a small boot volume, which is how ESXi has always been installed.

In ESX v3, there was always confusion about what was supported, because the installer ran RHEL3, yet you ran ESX. VMware rewrote the installer for ESXi so that *only* the supported devices could be installed upon. This new installer is the one used by ESX v4.

Licensing Differences

There have been major changes in licensing within ESX v4; the most important change is that there is now no need for a License Manager (except when managing ESX v3 systems from vSphere vCenter 4). Instead there is now a single license key that you either input via VMware vCenter Server or directly into the host using the vSphere Client. This single license key contains all the necessary information to enable or disable the features you have purchased.

The other major change is that VMware no longer sells features a la carte; instead they sell in bundles of functionality. If you previously purchased items a la carte and you still have service and support when you upgrade to vSphere, you can retrieve your license keys, and your a la carte functionality should still exist. If you do not have service and support, when you go to retrieve your vSphere licenses, you will not be able to do so until you purchase service and support for your older ESX v3 licenses.

All VMware vSphere licensing is done per individual socket (ESX v3 was licensed per TWO sockets), not per core. However, now there are limits based on license level for the number of cores of which you can make use. The split is at six cores. So when you upgrade CPUs to eight cores, you may also need to upgrade your license to make use of the two extra cores.

Although not exactly a licensing change, it is a cost change, VMware vSphere ESX v4 is now licensed by socket; in the past, VMware was licensed by a pair of sockets.

The license levels have also changed:

- Essentials—Equivalent to Starter
- Essentials Plus—Equivalent to Foundation
- Standard—No change
- Advanced—New, adding some new features to Standard
- Enterprise—No change
- Enterprise Plus—New, adding all new features to Enterprise

Table 2.13 provides a rundown of the licensing differences between ESX versions 2.5.x and 3.

Table 2.13 *Virtual Resource Functional Comparison*

Functionality	ESX4	ESX v3	ESX v2.5.x
ESX	Essentials, Essentials Plus, Standard, Advanced, Enterprise, Enterprise Plus	Starter, Foundation, Standard, Enterprise	Separate, VI Bundle
vCenter Server	Bundled	Separate or with a Bundle	Separate
VCB	Bundled	Bundled	N/A
Host Based Licenses	Supported	Supported	Supported
Server Based Licenses	N/A	Supported	N/A

Functionality	ESX4	ESX v3	ESX v2.5.x
Virtual SMP Support	Supported Four-way Eight-way (Enterprise Plus only)	Supported Four-way (Separate or Enterprise)	Supported Two-way (Separate)
Multiple Cores per Processor	Supported Up to 6 Up to 12 (Advanced or Enterprise Plus only)	Supported Unlimited	Supported Unlimited
Memory/Physical Server	256GB Unlimited (Enterprise Plus only)	128GB	64GB
Thin Provisioning	Supported	Supported	Supported
Update Manager	Supported	Supported	N/A
VMsafe	Supported	N/A	N/A
vStorage API	Supported	Via VDDK	N/A
Resource Pools	Based on CPU/ Memory Resources Utilization only. Requires vCenter.	Based on CPU/ Memory Resources Utilization only. Requires vCenter.	N/A
Clusters	Supported	Supported	N/A
High Availability	Full Support via Legato Automated Availability Management requiring at least Essentials Plus	Full Support via Legato Automated Availability Management separately purchasable or Enterprise	By hand/poor man's cluster
Data Recovery VDR	Not available for Essentials or Standard	N/A	N/A
Hot Add Memory	Requires at least Advanced and a Guest OS that supports functionality	N/A	N/A
Fault Tolerance	Requires at least Advanced Limited to one vCPU and requiring an ESX Cluster	N/A	N/A

continues

Table 2.13 *(Continued)*

Functionality	ESX4	ESX v3	ESX v2.5.x
vShield	Zones is free with at least Advanced. Other vShield tools are licensed separately.	N/A	N/A
vMotion	Requires at least Advanced	Separately purchasable, Standard or Enterprise	Separately purchasable
Storage vMotion	Requires at least Enterprise	Comes with vMotion	N/A
Distributed Resource Scheduling	Based on CPU/ Memory Resource Utilization only, requiring an ESX Cluster. Requires at least Enterprise.	Based on CPU/ Memory Resource Utilization only, requiring an ESX Cluster. Separate or Enterprise.	Extremely limited via HPSIM Virtual Machine Manager plug-in
Dynamic Power Management	Comes with DRS	Experimental	N/A
Enhanced vMotion Capability	Comes with DRS	Supported	N/A
Storage IO Control	Requires Enterprise Plus	N/A	N/A
Network IO Control	Requires Enterprise Plus	N/A	N/A
Load Based Teaming	Requires Enterprise Plus	N/A	N/A
Virtual Distributed Switch	Requires Enterprise Plus	N/A	N/A
Host Profiles	Requires Enterprise Plus	N/A	N/A
Multipath Plug-in	Requires Enterprise Plus	N/A	N/A

VMware Certification

The VMware Certified Professional (VCP) exam has been updated for ESX v4. Any previous VCP will still exist, *but* will not apply to ESX v4. Any ESX v3 VCP could have taken the VCP for version 4 without first having to sit a class until January 30, 2010. After this date, all VCPs are required to sit a class. Scoring of the exam has also changed—it is no longer just a percentage but a score that is a count of the questions answered correctly.

If you fail the VCP for ESX v4 exam, to retake the exam you must first take the full four-day VMware ESX v4 course. The passing grade for the VCP is 70%; to be eligible to become a VMware Certified Instructor, the passing grade is 85%.

VMware has also introduced the VMware Certified Design Expert (VCDX), which is one of the more difficult certifications to achieve; you need to take at least two more exams and defend a design you created, much like you would for a master's or PhD thesis. The steps to gain this certification are these: have a VCP3/VCP4, complete the Qualification Review, pass the Enterprise Exam, pass the Design Exam, apply for the VCDX Defense, and pass the defense of your design as judged by those who sit on the defense panel, which mainly consists of those who already have their VCDX and work for VMware.

Last, VMware has introduced the VMware Certified Advanced Professional (VCAP). VCAP has two tracks: Datacenter Administrator (DCA) and Datacenter Design (DCD). To gain either of these certifications, you are required to have a VCP4 (for vSphere 4) and pass the appropriate exams. After you have your VCP4, you are not required to sit any classes.

Conclusion

This chapter provided a review of the major differences between VMware ESX v4 and ESX v3, as well as the key differences between ESXi and ESX. As you can see, not that many differences exist between ESX and ESXi. This chapter serves as a starting point from which to plan an upgrade. Major differences were pointed out in each of the sections. The chapters following this one go into ESX and ESXi v4 in detail.