Integrating Virtual Machines into the Cisco Data Center Architecture

This document describes how to deploy VMware ESX Server 2.5 into the Cisco data center architecture. It provides details about how the ESX server is designed and how it works with Cisco network devices.

The document is intended for network engineers and server administrators interested in deploying VMware ESX Server 2.5 hosts in a Cisco data center environment.

Contents

Corporate Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Copyright © 2006 Cisco Systems, Inc. All rights reserved.

Introduction

Presently, the enterprise data center hardware and software platforms are being consolidated and standardized to provide improved resource utilization and management. As a result, the data center, servers, and network devices must be considered as pools of available resources rather than dedicated assets "siloed" when solving specific business requirements. Virtualization is a technique that allows the abstraction of these shared resources to provide distinct services on a standardized infrastructure. As a result, the data center applications are no longer bound to specific physical resources. The application is unaware, but depends on the pool of CPUs, memory, and network infrastructure services made available through virtualization.

The one-rack unit and blade server technologies of the x86 platforms are the results of enterprise consolidation requirements. However, the ability to abstract physical server hardware (such as CPU, memory, and disk) from an application provides new opportunities to consolidate beyond the physical and to optimize server resource utilization and application performance. Expediting this revolution is the advent of more powerful x86 platforms built to support a virtual environment that provides the following:

- Multi-core CPUs
- 64-bit computing (with memory/throughput implications)
- Multiple CPU platforms
- I/O improvements (PCI-E)
- Increased memory
- Power sensitive hardware

Software products such as VMware ESX Server, Microsoft Virtual Server, and the open source project known as XEN take advantage of these advancements and allow for the virtualization of the x86 platforms to varying degrees.

VMware ESX Architecture

This section discusses the architecture of VMware ESX Server version 2.5, including the following topics:

- ESX host overview
- ESX console
- ESX virtual machines
- ESX networking

- ESX storage
- ESX management

Note This section provides an overview of ESX server technology. For more information on ESX Server 2.5.x releases, see the VMware Technology Network website at the following URL: http://www.vmware.com/support/pubs/esx_pubs.html

ESX Host Overview

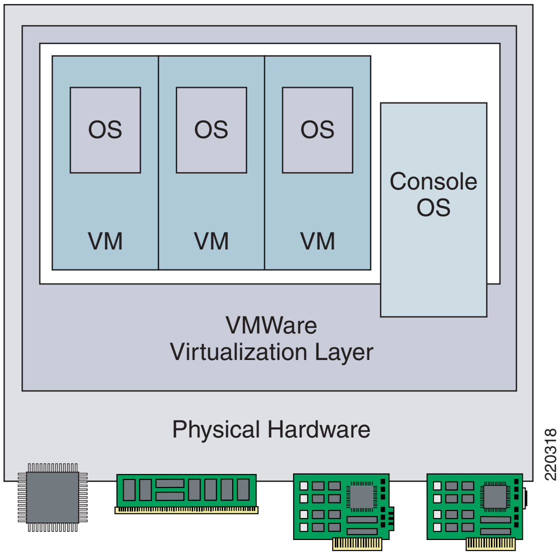
VMware ESX Server is a host operating system dedicated to the support of virtual servers or virtual machines (VMs). The ESX host system kernel (vmkernel) controls access to the physical resources of the server shared by the VMs. The ESX host system ensures that the following four primary hardware resources are available to guest VMs:

- Memory
- Processors
- Storage (local or remote)
- Network adapters

The ESX host virtualizes this physical hardware and presents it to the individual VMs and their associated operating system for use, a technique commonly referred to as *full virtualization*. A hypervisor achieves full virtualization by allowing VMs to be unaware and indifferent to the underlying physical hardware of the ESX server platform. A standard virtual hardware is presented to all VMs.

The vmkernel is a hypervisor whose primary function is to schedule and manage VM access to the physical resources of the ESX server. This task is fundamental to the reliability and performance of the ESX virtualized machines. As shown in Figure 1, the ESX vmkernel creates this virtualization layer and provides the VM containers where traditional operating systems such as Windows and Linux are installed.

Figure 1 ESX 2.5 Architecture Overview



Note Hardware restrictions require that the ESX Server runs on platforms certified by VMware. The complete list of compatible guest operating systems and server platforms can be found in the *System Compatibility Guide for ESX 2.x* document at the following URL:
http://www.vmware.com/pdf/esx_systems_guide.pdf

ESX Console

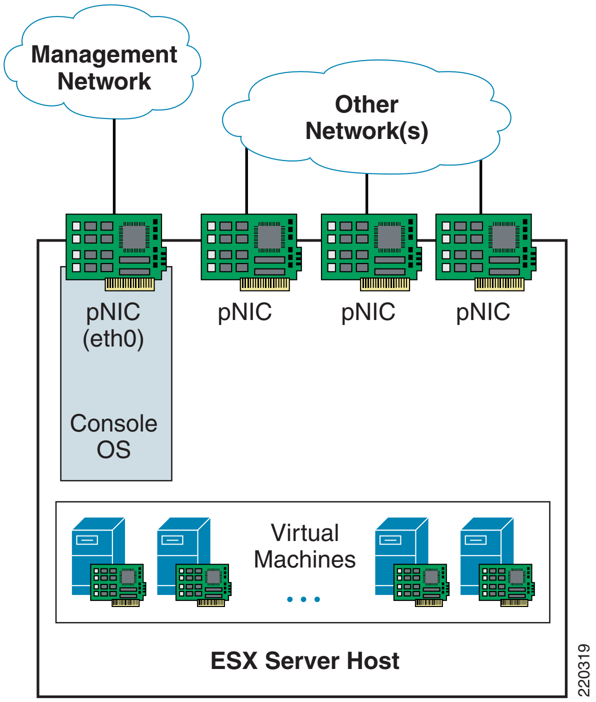
Figure 1 also shows the VM console or management interface to the ESX server system. Fundamentally, the console is based on the Red Hat Linux 7.2 server, with unique privileges for and responsibilities to the ESX system. The console provides access to the ESX host via SSH, Telnet, HTTP, and FTP. In addition, the console provides authentication and system monitoring services.

Note VMware VirtualCenter also uses the console to interact with its local ESX server agents (see ESX Host Management, page 22 for details).

The console requires hardware resources. Physical resources used by the console are either dedicated to the console itself or shared with the vmkernel; that is, VMs. Note that the use of hardware resources in a virtualized environment is a significant decision and must be approached not only from a basic resource utilization perspective but also from a network design standpoint.

Figure 2 shows the ESX console using a dedicated physical network interface card (pNIC) for connectivity to the management network.

Figure 2 ESX Console with Dedicated Network Connectivity



The ESX server always labels the console interface as eth0, which defaults to auto-negotiation of speed and duplex settings. However, these setting are manually configurable via the console using Linux command line tools or the Multilingual User Interface (MUI).

ESX Virtual Machines

VMware defines a VM as a virtualized x86 PC environment on which a guest operating system and associated application software can run. This allows multiple VMs to operate concurrently on the same host machine, providing server consolidation benefits and optimization of server resources. As previously mentioned, CPU, disk, memory, and network connections (SAN/LAN) used by the VM guest operating system are virtual devices. Therefore, it is important to understand the configuration of the virtual hardware of the VM.

Note ESX Server may host up to 80 active VMs, with a maximum of 200 VMs registered to a single host.

Virtual Processor

A VM uses discrete portions of one or more of the physical CPUs present on the ESX host to achieve virtual independence from the other VMs sharing the ESX host resources. Each VM maintains its own registers, buffers, and control structures. The ESX kernel conceals some of the CPU resources and provides scheduling. As a result, the majority of the physical CPU of the ESX host is directly available to the VM, providing compute power comparable to a non-virtualized server platform.

ESX Server version 2.5 has the following hardware requirements/limitations:

- Minimum of 2 physical processors per host
- Maximum of 16 physical processors per host
- Maximum of 80 virtual CPUs per host

By default, VMs share the processor resources available on the ESX host server equally. To provide better VM performance, the ESX kernel dynamically adjusts the system processor utilization to temporarily allow VMs, requiring more CPU to consume for performing their tasks. This may or may not be to the detriment of other VMs located on the ESX system. To address this issue, ESX provides processor resource controls that allow the administrator to define processor usage boundaries. The following tools are available in the current version of ESX Server:

- Shares
- Minimum/maximum percentages
- Combination of share and minimum/maximum percentages

Shares allow the administrator to define the processor usage of each VM in relation to the other VMs hosted by the system. Minimum and maximum percentages describe the lowest and highest CPU a particular VM may consume or require to power up. Used independently or together, these ESX features allow for greater processor regulation by the server administrator.

VMware Virtual SMP is an add-on module that allows guest operating systems to be configured as multi-processor systems. A Virtual SMP-enabled VM is able to use multiple physical processors on the ESX host. Virtual SMP functionality, however, contributes to the virtualization overhead on the ESX server. Deploy Virtual SMP-capable guests only where the operating systems and applications can benefit from using multiple processors. In addition, the ESX kernel enforces the concept of processor affinity. Affinity scheduling defines which processors a certain VM is permitted to use. Affinity is available only on a multiprocessor host.

Note The allocation of the ESX processor is as much art as science. A complete understanding of the hosted VMs performance requirements and behavior is recommended before any modification to the default utilization scheme is made.

Virtual Memory

The memory resources of the ESX host are divided among multiple consumers: the kernel, the service console, and the VMs. The ESX virtualization layer uses approximately 24 MB of memory, which is allocated to the system at startup and is not configurable. In addition to the memory required by the virtualization layer, the service console must be addressed by the ESX administrator to properly configure the system for VM support.

As described previously, the service console is a management interface to the ESX host that calls for memory based on the number of VMs operating concurrently on the ESX host. VMware provides the general guidelines shown in Table 1 to follow when defining the startup profile of an ESX system.

Table 1 Service Console Memory Guidelines

Number of Virtual Machines	RAM
8	192 MB
16	272 MB
32	384 MB
>32	512 MB
Maximum	800 MB

The ESX kernel grants access to the remaining memory on the host that is not used by the virtualization layer or service console to the VMs. The ESX administrator may set minimum and maximum memory allocations to each VM on the system. In general, maximum and minimum limits are placed on the system to guarantee performance levels of the VMs individually and the ESX system as a whole. Minimum memory allowances provide VM performance with minimum memory paging, where maximum settings allow VMs to benefit from memory resources underutilized on the ESX system. To meet the urgent memory demands of active VMs, ESX Server maintains 6 percent of the available memory pool free for immediate allocation.

Note Approximately 54 MB of memory is used per virtual CPU. In the case of dual virtual CPUs, this number increases to 64 MB of virtual CPU.

Even in a virtual environment, memory is a finite resource. The ESX Server uses the following methods to provide optimal memory utilization by the hosted VMs:

- Transparent page sharing
- Idle memory tax
- Ballooning
- Paging

Each of these techniques allows the ESX administrator to oversubscribe the memory of the system among the VMs, allowing ESX Server to optimize the performance of each.

Note For more information about the memory optimization techniques employed by the ESX server system, see *Memory Resource Management in VMware ESX Server* by Carl A. Waldspurger at the following URL: http://www.vmware.com/pdf/usenix_resource_mgmt.pdf

Virtual Disks

VMs use virtual disks for storage. The actual physical storage may be a local hard drive on the ESX host system or a remote storage device located in the SAN. The virtual disk is actually not a disk but a VM disk image file (VMDK). This file exists with the VM file system (VMFS), which is a flat file system created for better performance. Therefore, the guest operating system and its associated applications are installed into a **.VMDK** file residing in a VMFS on the local drive or SAN.

Typically, the disk image (VMDK) file is stored in the SAN, which is a requirement for VMware VMotion support and for environments seeking boot support from a SAN, such as blade servers. The local hard drive of the ESX host system normally houses the ESX console and VMFS swap files. This provides improved performance for VMs when memory utilization is great.

VMDK files may employ one of the following four modes:

- Persistent
- Nonpersistent
- Undoable
- Append

The VMDK file modes have a direct effect on the behavior of the VM. Persistent mode allows permanent writes to the disk image. This mode is comparable to the behavior of a normal disk drive on a server. Nonpersistent mode disregards all modifications to the VMDK file after a reboot.

Undoable mode uses REDO logs that allow administrators to choose whether modifications made to the VM should be accepted, discarded, or appended to the REDO log. Building on this functionality, the append mode adds changes to the REDO log automatically and does not commit the changes unless committed by the administrator. Each of these modes requires administrator input at the time the VM is powered down.

Note ESX Server does not support IDE drives beyond a CD-ROM mount.

Virtual Adapters

VMs provide virtual network interface cards (vNICs) for connectivity to guest operating systems. A VM supports up to four vNICs. Each vNIC has a unique MAC address, either manually assigned or dynamically generated by the ESX platform.

Generated vNIC MAC addresses use the Organizationally Unique Identifiers (OUI) assigned by VMware and the Universal Unique Identifier (UUID) of the VM to create a vNIC MAC. ESX Server verifies that each generated vNIC MAC address is unique to the local ESX system. Table 2 lists the OUIs assigned to VMware.

Table 2 OUIs Assigned to VMware

OID	Note
00:0C:29	Generated address range
00:50:56	Generated address range Reserved for manually configured MACs: 00:50:56:00:00:00 ◊ 00:50:56:3F:FF:FF Reserved for VirtualCenter-assigned MACs: 00:50:56:80:00:00 ◊ 00:50:56:BF:FF:FF
00:05:69	Used by ESX versions prior to Release 1.5

Note The vNIC MAC addresses are present in the MAC address table on the physical switch.

The vNICs are virtual adapters that must be supported by the guest operating system (OS) of the VM. Two device drivers are available for the guest OS to use when communicating with the vNICs; the vlance and vmxnet drivers. The vlance driver provides universal compatibility across all guest operating systems by emulating an AMD PCNet device.

Note The vlance driver always indicates a 100 Mbps link speed on the guest OS, even though it is capable of using the full bandwidth (>100 Mbps) available on the physical adapter.

The vmxnet driver provides better vNIC performance because it is optimized for the virtual environment and utilization of ESX resources. The vmxnet driver must be installed on all guest operating systems via the VMware tools software package.

Note For more detailed information about ESX Server 2.5.x VM specifications, see the *ESX Server 2 Installation Guide* at http://www.vmware.com/pdf/esx25_install.pdf and the *ESX Server 2 Administration Guide* at http://www.vmware.com/pdf/esx25_admin.pdf

ESX Networking Components

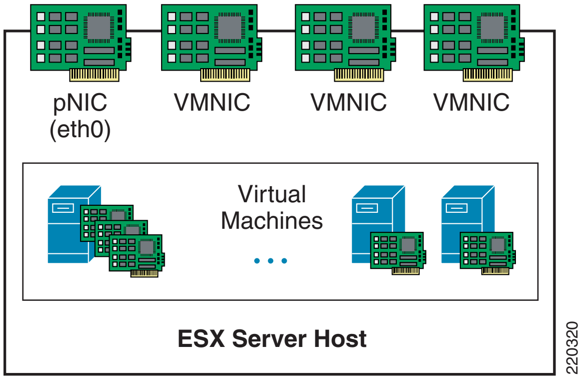
pNICs and VMNICs

Beyond providing an emulated hardware platform for VMs, ESX Server offers connectivity to the external “physical” enterprise network and other VMs local to the host. The following ESX networking components provide this internal and external access:

- Physical Network Interface Cards (pNICs)
- Virtual machine Network Interface Cards (VMNICs)
- Virtual switches

Figure 3 shows the provisioning of physical and VM adapters in an ESX host.

Figure 3 ESX Server Interfaces



In this example, four pNICs are present on the ESX server platform. The server administrator designates which NICs support VM traffic, virtual machine NICs (VMNICs), and those allocated for use by the ESX management console, pNICs. The vmkernel labels the management interface as eth0.

Note Physical NICs map to VMNICs, which are not equivalent to the virtual NICs used by each VM and defined in the previous section.

The server administrator may also choose to share the physical NIC resources between the ESX management console and VMs present on the host. Sharing resources in this manner is effectively a form of inband management. VMware does not recommend sharing in this manner unless it is necessary. For more information about assigning adapters, refer to Interface Assignment, page 18.

Virtual Switches

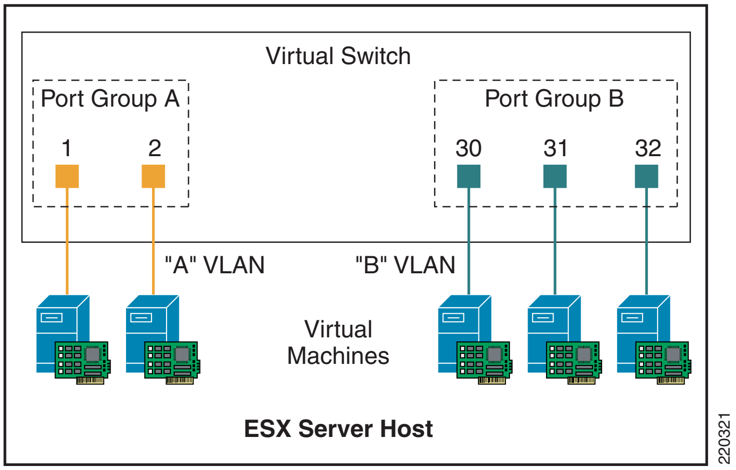
The ESX host links local VMs to each other and to the external enterprise network via a software construct named a virtual switch (vswitch). The vswitch emulates a traditional physical Ethernet network switch to the extent that it forwards frames at the data link layer. ESX Server may contain multiple vswitches, each providing 32 internal virtual ports for VM use. Each vNIC assigned to the vswitch uses one internal virtual port, which implies that no more than 32 VMs can be used per virtual switch.

The virtual switch connects to the enterprise network via outbound VMNIC adapters. A maximum of eight Gigabit Ethernet ports or ten 10/100 Ethernet ports may be used by the virtual switch for external connectivity. The vswitch is capable of binding multiple VMNICs together, much like NIC teaming on a traditional server. This provides greater availability and bandwidth to the VMs using the vswitch. A public virtual switch employs outbound adapters, while a private vswitch does not, offering a completely virtualized network for VMs local to the ESX host. ESX internal networks are commonly referred to as VMnets.

Note Naming or labeling virtual switches is an important standard to develop and maintain in an ESX environment. It is recommended to indicate the public or private status of the vswitch or VLANs it supports via the vswitch name.

Virtual switches support VLAN tagging and take advantage of this capability with the port group construct. One or more port groups may exist on a single virtual switch. Virtual machines then assign their virtual NICs (vNICs) to these port group. Figure 4 shows two port groups defined on a virtual switch: port groups A and B that are associated with VLANs A and B. The server administrator then assigns the VMNIC vNIC to one of the port groups.

Figure 4 Virtual Switch with Port Groups



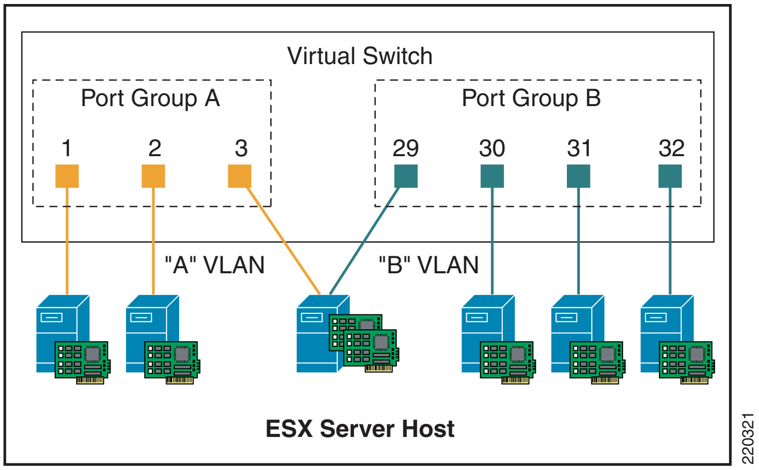
Internal Networking (VMnets)

VMnets are internal networks of VMs local to the ESX host. VMnets use the virtual switch to link VMs on the same VLAN. The system bus provides the transport and the CPU manages the traffic. VMnets are generally used in test and development environments.

In its simplest form, a VMnet architecture requires that the VMs have Layer 2 adjacency, meaning they are part of the same port group on the vswitch. For example, in Figure 4 above, the two machines in VLAN A may communicate; however, these VMs are isolated from the VMs comprising port group B, which uses another VLAN.

Figure 5 shows a more complex VMnet design. In this example, a VM is a member of both port groups A and B, requiring the use of two vNICs on the VM. This VM may be configured to forward IP traffic between the two VLANs, allowing the VMs on port groups A and B to communicate.

Figure 5 ESX Server VMnets (Private Virtual Switches)



Note Performance issues may occur when using VMnets. When considering the use of VMnets, it is important to understand the throughput requirements of the guest operating system applications. Typically, the external network provides greater throughput and frees bus and CPU resources on the ESX host. VMware has acknowledged issues with TCP flow control and throughput on VMnets within the ESX platform. For more information, refer ti VMTN Answer ID 1428.

External Networking (Public Vswitch)

A public virtual switch uses at least one of the physical adapters, or VMNICs, on the server to link VMs to the external network. The vmkernel allows the public vswitch software construct to use some of the hardware acceleration features available on the physical NICs, including the following:

- TCP segmentation offload
- VLAN tagging
- Checksum calculations

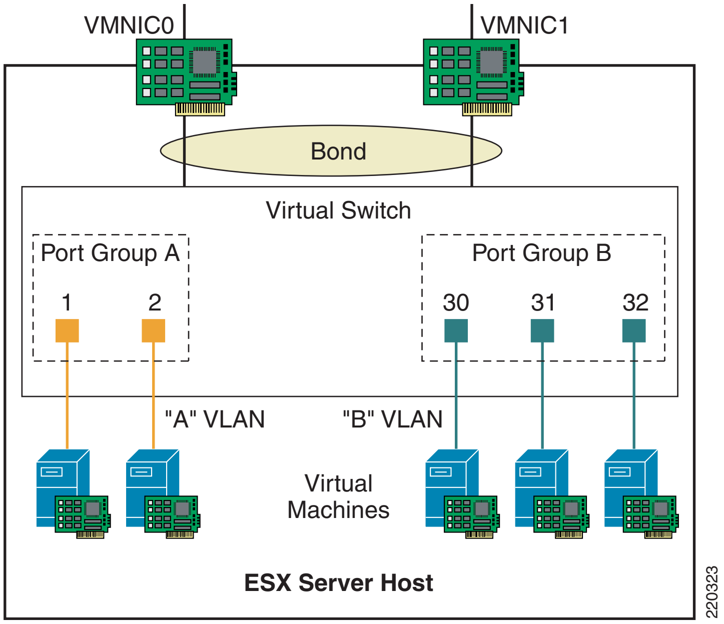
Bonding

To provide increased availability and greater bandwidth to the VMs, the public vswitch may create adapter bonds. Bonding is a method to group multiple VMNICs together to supply greater bandwidth and higher availability services to the VMs associated with the public switch. A vswitch adapter bond is comparable to NIC teaming on a traditional server. Because of the high availability afforded to the VMs on the ESX host via bonding, VMs do not need to be dual-homed to virtual switches, and typically use a single vNIC for connectivity.

Note VMware recommends using identical physical adapters on an ESX platform. The basis for this recommendation is that the vswitch can use only hardware features common to all bonded adapters on the server.

Figure 6 shows a logical view of a public vswitch configured for bonding.

Figure 6 ESX Server Public Virtual Switch with Bonding



In this example, VMs in VLANs A and B benefit from the VMNIC0 and VMNIC1 bond. The virtual switch load balances egress traffic across the bonded VMNICs via the source vNIC MAC address or a hash of the source and destination IP addresses. The virtual switch uses all VMNICs in the bond. If a link failure occurs, the vswitch reassigns VM traffic to the remaining functional interfaces defined in the bond.

It is important to remember that the IP-based load balancing method may result in the vNIC MAC address of a single VM being known on multiple physical ports on the external network. This may have adverse effects on the performance of the external switch or switches, depending on the network design. ESX Server supports IEEE 802.3ad link aggregation, and the use of this feature is recommended when using IP-based load balancing. For details, refer to Integrating ESX Hosts into the Cisco Data Center Architecture, page 24.

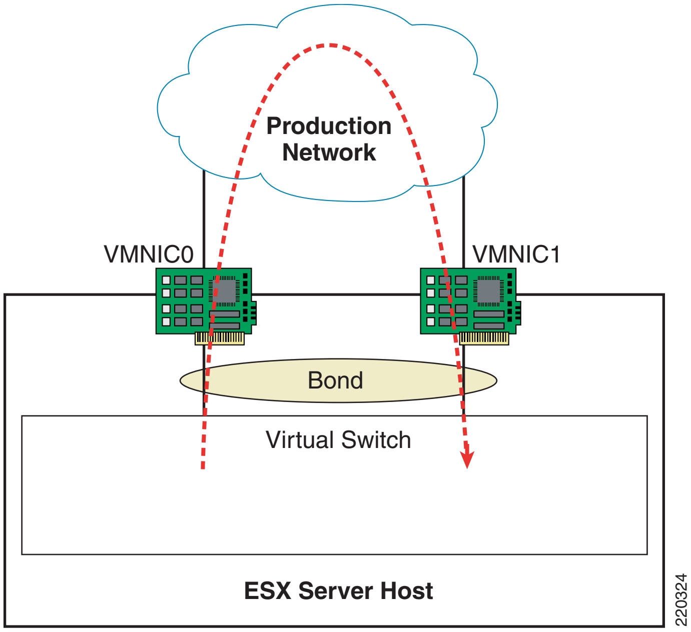
The source vNIC MAC address is the default load-balancing method used for bonded VMNICs. In addition to the MAC and IP-based load balancing methods, a VMNIC bond is configurable in an active/standby or failover-based scenario. Standby mode assigns a single VMNIC as primary. This means that all traffic from the VMs use this adapter unless a link failure occurs, in which case the standby VMNIC assumes the traffic load of the VMs.

Note Cisco recommends using standby mode to provide a highly available vswitch design that allows for rapid and predictable network convergence.

Beaconing

Beaconing is a probing function that allows the ESX host to monitor the availability of VMNICs within a bond. Beaconing requires that the VMNICs reside in the same broadcast domain. Beacons are intended for use with bonds connected to more than one external switch. The ESX server monitors the loss of beacon probes to determine failures in the external network. If a failure condition exists, meaning that a VMNIC has not reported receiving x number of beacons from the beacon initiator, the ESX server switches adapters and declares the primary adapter down. Figure 7 shows a logical representation of a virtual switch using beaconing on a production network.

Figure 7 Beaconing with the Virtual Switch



Note The beacon probe interval and failure thresholds are global parameters that are manually configurable on the ESX host and applied to every virtual switch on the server.

It is not recommended to use beaconing as a form of external network failure detection because of the possibility of false positives. To provide a highly available external network infrastructure, use redundant paths and/or protocols to achieve high availability. For more information on high availability designs for VMs in the data center, refer to Integrating ESX Hosts into the Cisco Data Center Architecture, page 24.

VLAN Tagging

Historically, the physical access switches in the data center provide the VLAN tagging functionality, allowing a single network infrastructure to support multiple virtual LANs and respective business needs. With the introduction of ESX Server into the data center, the traditional method of VLAN tagging is no longer the only option. ESX server provides the following three methods to extend VLANs in the enterprise data center:

- Virtual guest tagging (VGT)
- External switch tagging (EST)
- Virtual switch tagging (VST)

This section discusses the benefits and drawbacks of each of these approaches.

Virtual Guest Tagging

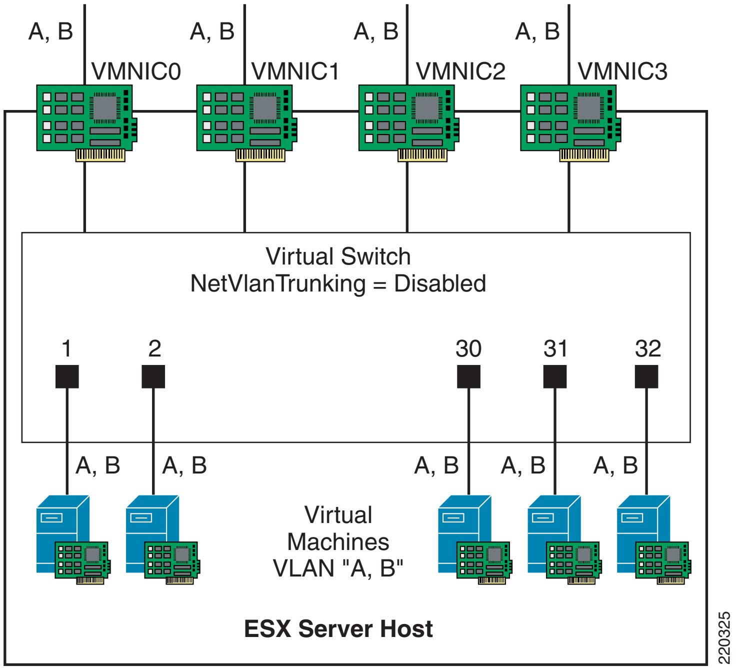
VGT requires that each VM guest operating system support and manage 802.1q tags. The VM manages the vNIC, removing all tagging responsibilities from the virtual switch. Disabling 802.1q tag support on the ESX host is a global configuration and applies to all virtual switches on the system.

Figure 8 shows a VGT scenario where each VM processes the VLAN tags. A VGT configuration requires more processing power from each VM, reducing the efficiency of the VM and overall ESX host. VGT deployments are uncommon but are necessary if a single VM must support more than four VLANs.

Note Each VM has four independent virtual interfaces that reside on separate VLANs.

Figure 8 Virtual Guest Tagging

\



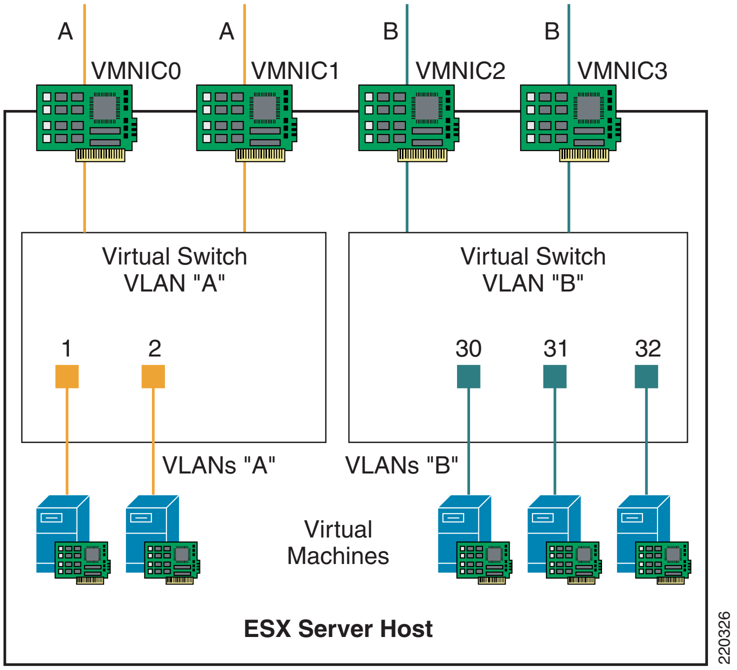
Note Currently, there is limited support for VLAN tag processing within operating systems. For example, newer Linux kernels support 802.1q tags while Windows systems do not.

External Switch Tagging

As stated earlier, external switch tagging (EST) is a standard procedure within enterprise data centers. EST is the practice of VLAN tagging at the access port of the server. There is a one-to-one relationship between the number of physical NICs on the server dedicated to VMs (VMNICs) and the maximum number of VLANs supported on a single ESX host. This limitation is often addressed by using larger server platforms for ESX. An added benefit of these platforms is the general increase in processor and memory available to the system, providing a robust environment for VMs.

Figure 9 shows the logical view of an EST deployment.

Figure 9 External Switch Tagging



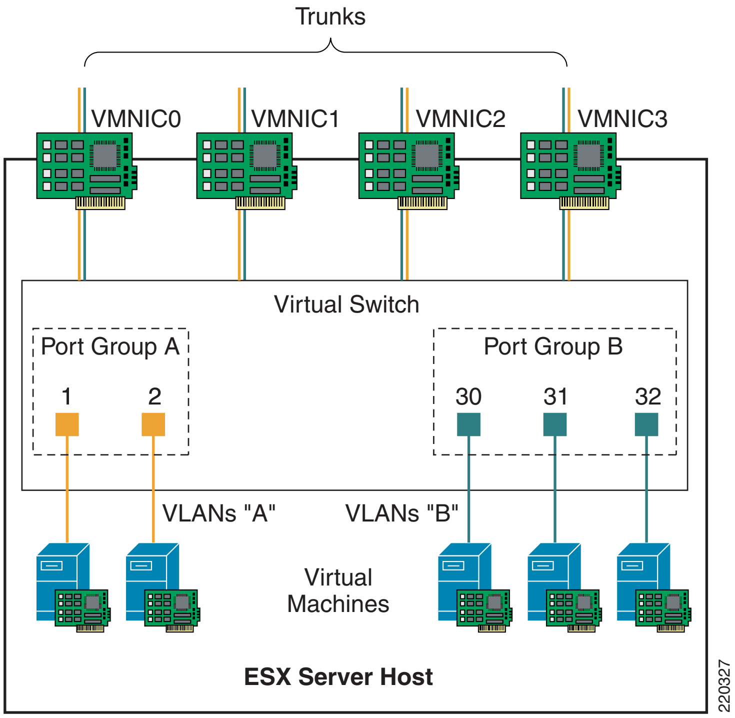
In this example, each virtual switch is associated with a single VLAN: VLANs A and B. The external network defines the VMNIC links to the virtual switches as access ports supporting a single VLAN per port. The vswitch does not perform any VLAN tag functions.

Virtual Switch Tagging

Virtual switch tagging (VST) allows the virtual switch to perform the 802.1q tag process. The vmkernel actually allows the physical adapters to carry out the VLAN tag operations, relieving the vmkernel of the work and improving overall system performance. VST requires that the VMNICs connected to a VST-enabled switch be 802.1q trunks, which requires that the external network ports be 802.1q trunks as well.

Figure 10 shows a logical view of VST.

Figure 10 Virtual Switch Tagging



The vNICs of the VM are assigned to a port group that is associated with a specific VLAN, in this case VLANs A and B. The virtual switch defines the VMNICs as ports supporting all of the VLANs within the switch; that is, as trunks.

Note Dynamic Trunking Protocol (DTP) is not supported by ESX virtual switches.

In VST mode, the vswitch may support numerous VLANs over a limited number of ports, which allows the server administrator to define more VLANs than physical adapters. This is an obvious difference from EST mode. VST allows more flexibility on the number of VLANs and therefore types of VMs a single ESX server may host, while reducing the number of physical adapters required on the platform.

Note Configuring trunk ports to the server is not a generally accepted practice. If using VST mode, it is important to define specific ownership roles and implementation protocols between server administrators and network administrators with regards to the virtual switch.

VMotion Networking

VMotion is the method used by ESX Server to migrate active VMs within an ESX server farm from one physical ESX host to another. VMotion is perhaps the most powerful feature of an ESX virtual environment, allowing the movement of active VMs with minimal downtime. Server administrators may schedule or initiate the VMotion process manually through the VMware VirtualCenter management tool.

The VMotion process occurs in the following steps:

Step 1 VirtualCenter verifies the state of the VM and target ESX host. VirtualCenter determines the availability of resources necessary to support the VM on the target host.

Step 2 If the target host is acceptable, a copy of the active VMs state is sent from the source ESX host to the target ESX host. The state information includes memory, registers, network connections, and configuration information. This is an ongoing process until the delta between the source and target state information is nominal.

Step 3 The source ESX server VM is suspended.

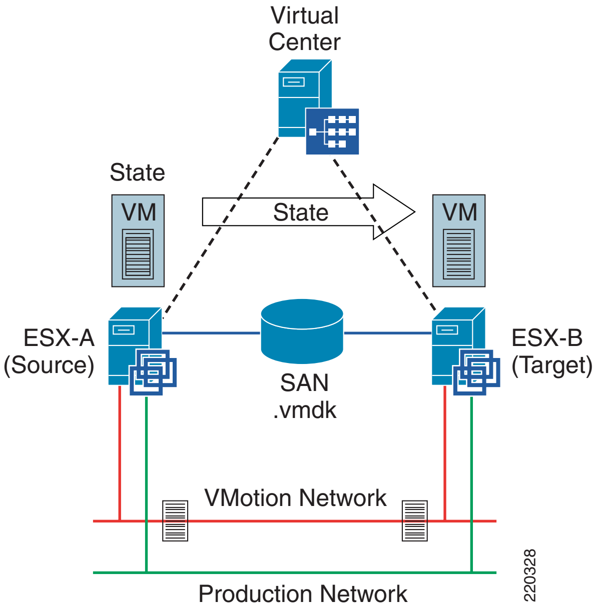
Step 4 The **.vmdk** file (virtual disk) lock is released by the source ESX host.

Step 5 The remaining copy of state information is sent to the target ESX host.

Step 6 The target ESX host activates the new resident VM and simultaneously locks its associated **.vmdk** file.

Figure 11 shows the VMotion process and the key components in the system.

Figure 11 VMotion Process



VMotion is not a full copy of a virtual disk from one ESX host to another but rather a copy of State. The **.vmdk** file resides in the SAN on a VMFS partition and is stationary; the ESX source and target servers simply swap control of the file lock after the VM state information synchronizes.

Note For some heavily-used VMs, VMotion may not be able to copy state information at an acceptable rate to the target ESX. The suspension of the VM (Step 3) does not occur. In these cases, schedule the VMotion during low use times of the day.

Deploying a VMotion-enabled ESX server farm requires the following:

- VirtualCenter management software with the VMotion module.
- ESX server farm (VMotion only works with ESX hosts). Each host in the farm should have identical hardware processors to avoid errors after migration.
- Shared SAN, granting access to the same VMFS volumes (**.vmdk** file) for source and target ESX hosts.
- Volume names used when referencing VMFS volumes to avoid world wide name (WWM) issues between ESX hosts.
- Use of dedicated Gigabit Ethernet Network for state information exchange.

Note VMotion is not supported using local storage.

For more information about VMotion, refer to *VMware VirtualCenter User Guide* at the following URL: http://www.vmware.com/pdf/vc_users13.pdf.

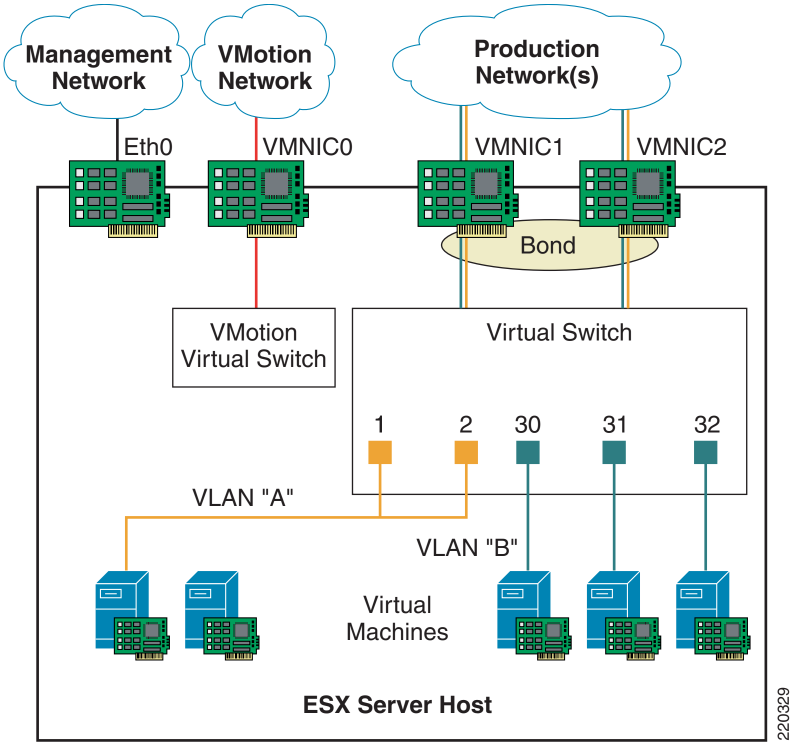
Interface Assignment

One of the benefits virtualization offers is the efficient use of physical resources, including network adapters. VMware recommends at a minimum that an ESX server use two Ethernet adapters to support the following traffic types:

- Management traffic (ESX Console traffic)
- Virtual machine traffic (production traffic)

In addition to these two basic traffic categories, the server administrator may wish to use VMotion. VMware recommends enabling VMotion on its own dedicated network segment using its own Gigabit Ethernet network interface. Figure 12 shows the suggested ESX interface assignments in an environment employing VMotion.

Figure 12 ESX Interface Assignment



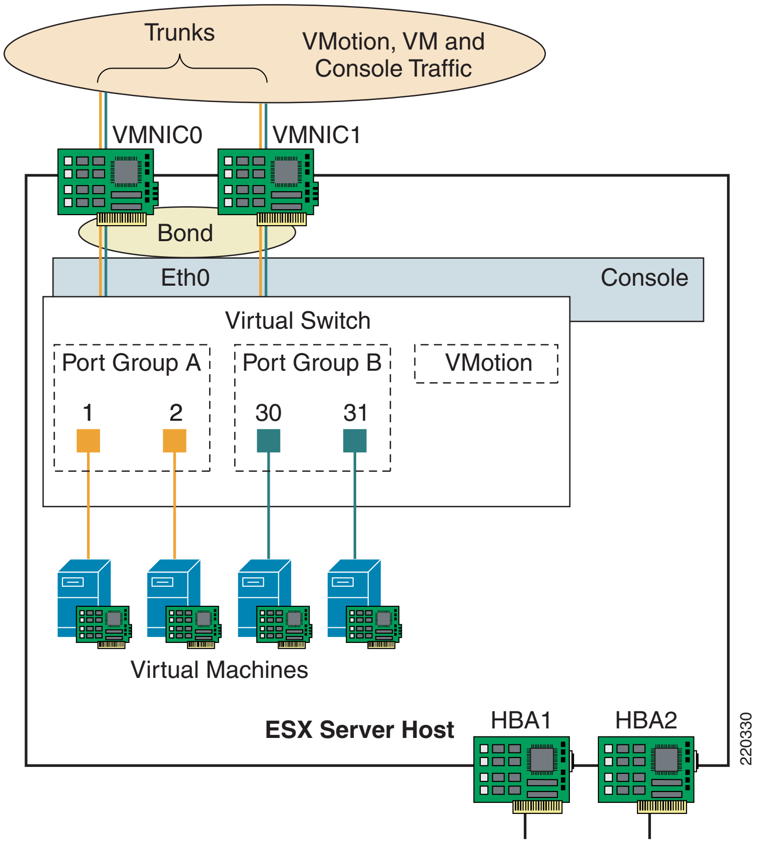
Note Figure 12 does not include the Host-Based Adapters (HBA) necessary to connect to the SAN for VMotion functionality, but these must be present on the ESX platform.

Each traffic type has a dedicated physical resource, pNIC, on the server. For high availability purposes, VM production traffic employs two adapters in a bonded configuration. Additionally, note that the VMotion network has a dedicated virtual switch to optimize performance of the VMotion process. The VMotion vswitch does not support VM traffic.

Figure 12 above shows an optimal or at least a favorable ESX interface configuration; however, it is possible to configure ESX hosts on a server platform with fewer physical adapters available. Blade servers are one such server platform where interface constraints exist. Depending on the blade server vendor, only two or three Ethernet adapters may be available. Therefore, ESX supports blade server deployments through the sharing of physical resources via trunks.

To share limited network resources in a blade server environment, the ESX administrator must make some modifications to the host configuration. Figure 13 shows the logical design for a blade server with two Ethernet interfaces.

Figure 13 Sharing Adapter Resources



The virtual switch supports VMotion transport in addition to the traffic originating from the VMs. The ESX console uses the vmxnet_console driver to connect to the same set of VMNICs. The console labels this bond as eth0 and benefits from the added bandwidth and availability of two VMNICs.

Sharing network resources may influence the performance of all traffic types because competition for a finite resource may occur. For example, the VMotion process may take longer to complete when interfaces are allocated for production and management traffic.

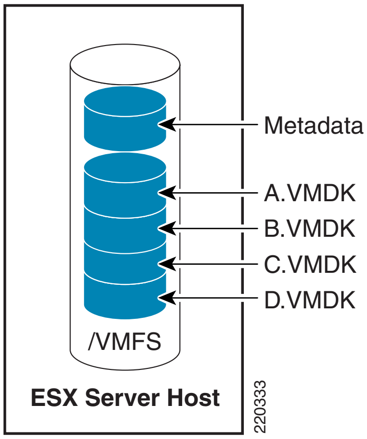
ESX Storage

The VMware ESX file system (VMFS) is a flat file system that is optimized for VM access. The VMFS stores VM virtual disks (VMDK), REDO logs, and memory snapshots of VMs. The ESX console mounts the VMFS volumes under the /vmfs directory on booting. The VMFS volume spans multiple partitions, consisting of a maximum of 32 physical disks or logical unit numbers (LUNs). The VMFS supports file level locking to allow multiple ESX servers to access files on a single VMFS volume.

Note Each ESX host can support a maximum of 128 VMFS volumes.

Figure 14 shows an ESX host using the local SCSI drive for storage, containing numerous virtual disks in the file system. VMDK files require a SCSI disk and are not permitted to increase in size. This reduces the probability of disk fragmentation and improves system performance.

Figure 14 ESX Local Storage



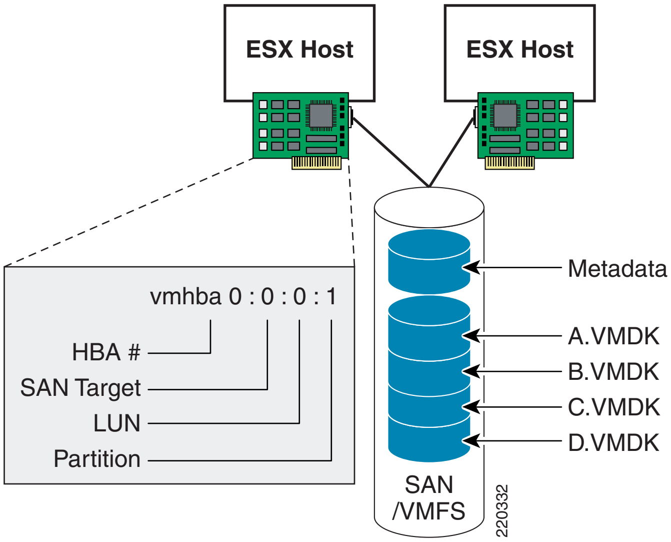
Local storage has its limitations. An enterprise ESX deployment should consider the use of remote storage, a SAN, for the following reasons:

- Scalability
- High availability
- ESX server advanced functionality such as VMotion

Remote storage moves the VMFS file system beyond a single host, creating data accessible to all ESX servers. This allows the virtual server solution to scale with shared data beyond the confines of a single box. VMotion, for example, requires that the source and destination ESX hosts use common storage for access to the same **.vmdk** file.

Figure 15 shows the use of a VMFS volume residing in the SAN. Each ESX host accesses the file system via a local HBA. The volume known via the HBA numbering may be referenced by a more readable name by labeling it. For example, vmhba0:0:0:1 can be renamed to VOL1.

Figure 15 ESX Server Using the SAN



The ESX server supports the following SAN features:

- LUN masking (security/administrative feature)
- Multi-pathing (preferred path and most recently used)
- Raw disk mapping

Note For more information about ESX storage, refer to the *VMware ESX Server SAN Configuration Guide* at the following URL: http://www.vmware.com/pdf/esx25_san_cfg.pdf

ESX Host Management

The following are three approaches to managing the ESX server:

- Console
- Multilingual User Interface (MUI)
- Management application such as VMware VirtualCenter

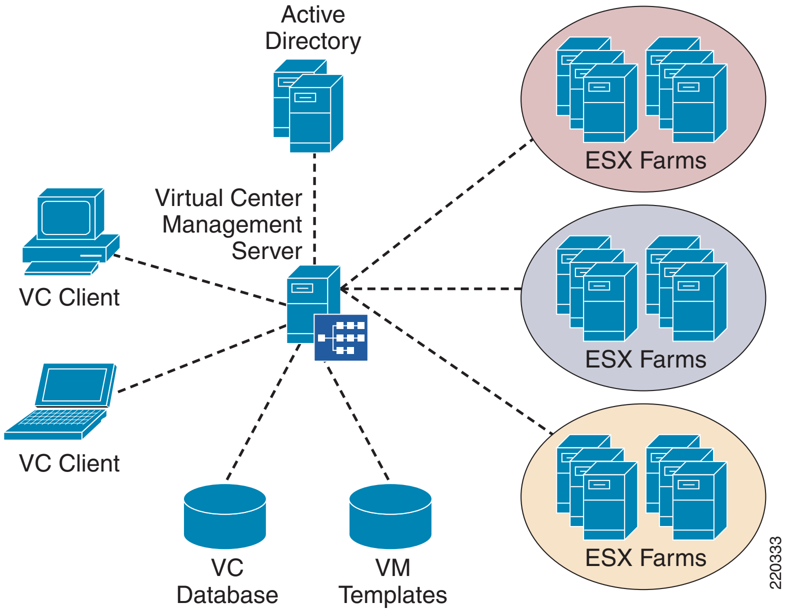
As discussed in ESX Console, page 4, the console provides access to the ESX host via SSH, Telnet, HTTP, and FTP. In addition, the console supplies authentication and system monitoring services. For more information on the console, refer to ESX Console, page 4.

The MUI is accessible through an Internet Explorer 6.0 browser and provides a web interface to manage the ESX host. Every ESX host provides a console and MUI interface.

The console and MUI are sufficient for managing a single ESX host; however, each of these solutions are limited to a single platform. VirtualCenter is a central management solution that, depending on the VC platform, scales to support numerous clients, ESX hosts, and VMs.

Figure 16 shows the major components of a virtual management infrastructure using VMware VirtualCenter.

Figure 16 VirtualCenter Management Infrastructure



VirtualCenter clients access the management server and are able to administer the numerous ESX server farms. VirtualCenter provides the following capabilities:

- Role-based permissions (RO, VM User, VM Admin, and VC Admin)
- Creation and/or modification of VMs
- Monitoring of performance of ESX host and VMs
- Task scheduler

ESX server farms usually consist of hosts with similar hardware and resources. This best practice readily permits the enterprise to employ VMotion through VirtualCenter. Note that an optimized VMotion infrastructure provides identically-configured hardware hosts and virtual platforms.

Note Increasing the hardware requirement to dual CPUs and 3 GB RAM can scale the VirtualCenter Management Server to support up to 50 concurrent client connections, 100 ESX servers, and 2000 VMs.

For details about VirtualCenter, the virtual infrastructure management software from VMware, refer to the following URL:

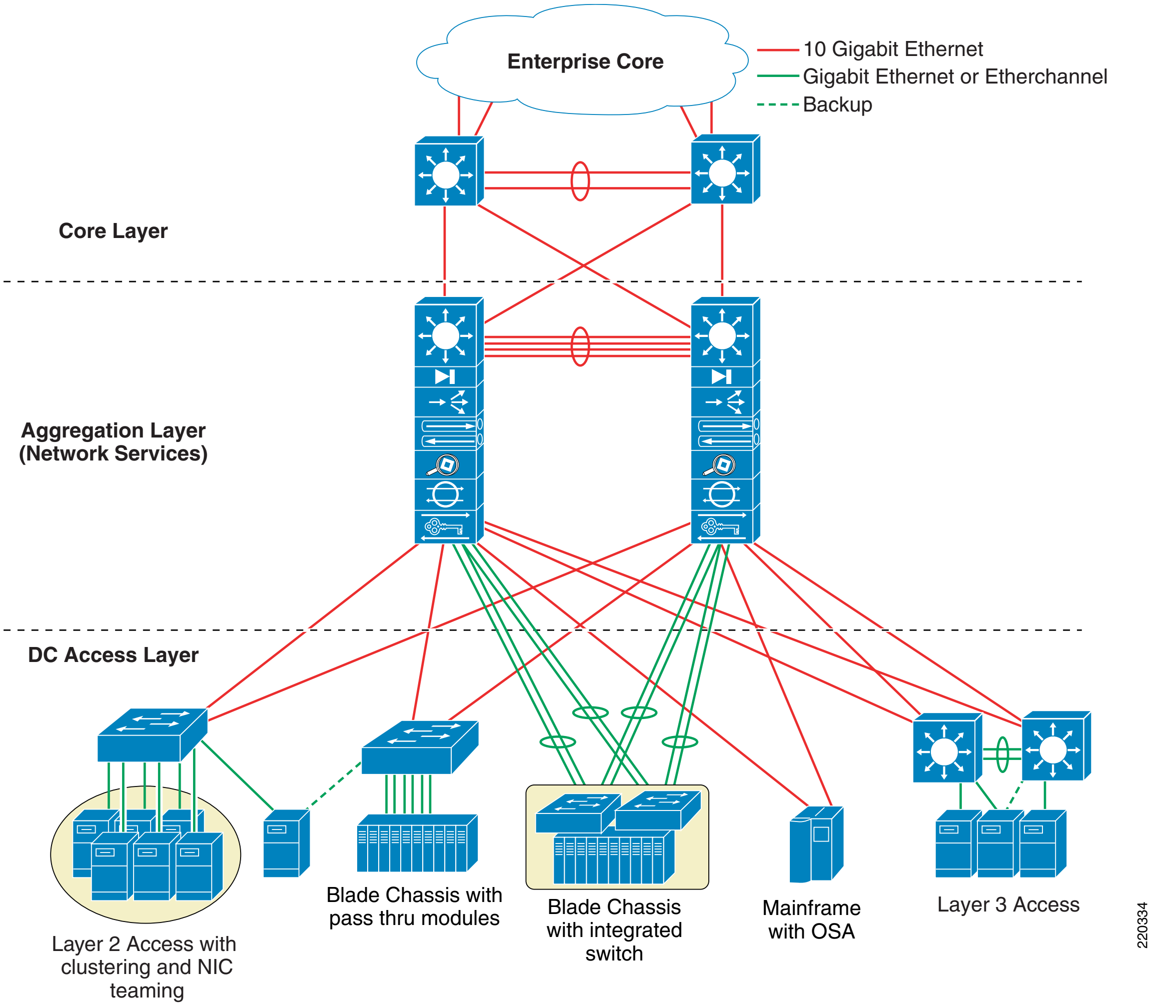
http://www.vmware.com/products/vc/.

Integrating ESX Hosts into the Cisco Data Center Architecture

Overview

The Cisco data center architecture provides scalability, availability, and network services to enterprise server farms. Figure 17 shows the Cisco data center network design.

Figure 17 Cisco Data Center Architecture



The design comprises three major functional layers (core, aggregation, and access), and provides the following:

- Support for Layer 2 an 3 requirements (high availability via HSRP and STP)
- High performance multi-layer switching
- Multiple uplink options
- Consolidated physical infrastructure
- Network services (security, load balancing, application optimization)

- Scalable modular design

The data center should be flexible, scalable, and provide a highly available and predictable environment for an enterprise server farm. VMs are the most recent addition to the data center, which will benefit from this well-designed infrastructure and offer accessibility, future growth, and centralized services.

Access Layer

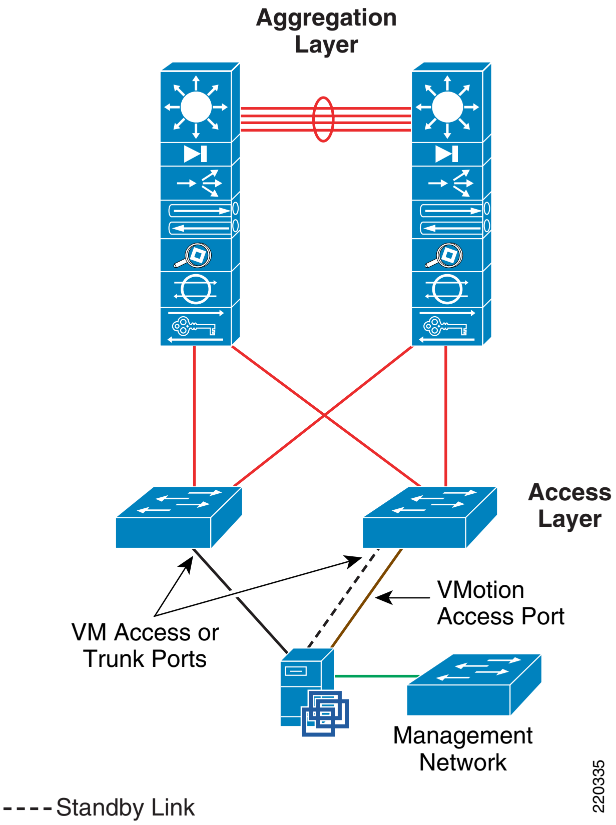
The access layer provides connectivity to the server farm. Port density is a significant consideration in any access layer design. In addition, the access switches must support many requirements including dual-homing and Layer 2 adjacency for server clustering. ESX Server benefits from the following access layer designs:

- Classic access layer design
- Single switch design (dual supervisors)

Classic Access Layer Design

Figure 18 shows a traditional access layer design providing connectivity for an ESX host.

Figure 18 Classic Access Layer Design



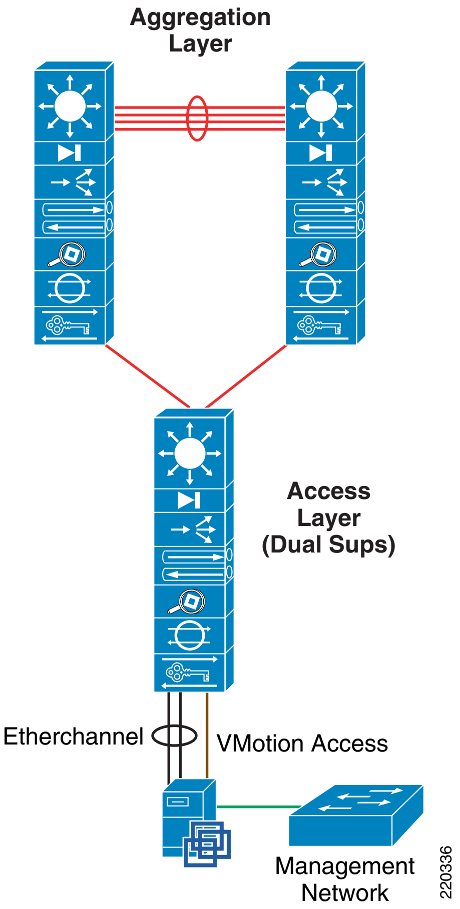
The ESX server is dual-homed to the access layer switches. The VMNIC connections are in standby mode, providing high availability and predictable failover traffic patterns. Redundancy is evident throughout this design because there is no single point of failure. Rapid PVST+ manages the traffic patterns in this design. Configuring the access ports as edge ports in relation to spanning tree accelerates network convergence when failure conditions occur. The VMotion process uses the same physical infrastructure virtualized through VLANs to allow active VM migrations.

Note The classic access layer design provides Layer 2 adjacency with raw port density, easily accommodating the NIC bonding features of an ESX server.

Single Switch Design (Dual Supervisors)

Figure 19 shows the use of a single switch for server farm connectivity.

Figure 19 Single Switch Access Layer Design (Dual Supervisors)



Redundant uplinks from the access switch increase availability of the ESX server, as well as the dual supervisors that provide rapid convergence via stateful switchover (SSO). SSO provides Layer 2 high availability using redundant supervisors in an active/standby scenario, introducing approximately 0 to 3 seconds of packet loss when a supervisor switchover occurs. Attaching the ESX server to a single access layer switch with supervisor redundancy may be an acceptable level of redundancy for some enterprises.

Aggregating the ESX server links to the access layer switch via Link Aggregation Control Protocol (LACP) allows for increased utilization of server resources. The ESX administrator may configure the bond to load balance egress traffic on the source and destination IP address information. This algorithm may improve the overall link use of the ESX system by providing a more balanced distribution across the aggregated links. A single access switch design permits IP-based load balancing because it negates the issue of identical VM MAC addresses being present on multiple switches. The 802.3ad links remove a single point of failure from the server uplink perspective, which reduces the chances that VM traffic will be black-holed.

Note An aggregated link EtherChannel configuration (for example, Cisco 3750 platform) can also be used with stackable switches. A switch stack is logically a single switch that allows the use of aggregated ports and src/dst IP load-balancing equivalent to the single switch access design.

Configurations

The following configurations are applicable to the ports on the access layer switches to support the various VLAN tagging modes available with ESX server. The following configurations use a Cisco Catalyst 6500 with Cisco native IOS Release 12.2(18)SXD5.

External Switch Tagging

External switch tagging (EST) mode allows the external access layer switch to assign all traffic on a port to a single VLAN.

```
spanning-tree portfast bpduguard default
!
interface GigabitEthernetX/Y
 description <<** EST Mode **>>
 no ip address
 switchport
 switchport access vlan <id>
 switchport mode access
 switchport port-security maximum <number>
 switchport port-security violation shutdown
 switchport port-security aging time 20
 no cdp enable
 spanning-tree portfast
```

Note The maximum number of MAC addresses is determined by the number of VMs on the ESX host using the VLAN.

Virtual Switch Tagging and Virtual Guest Tagging

Virtual switch tagging (VST) permits the vswitch to tag all egress traffic, and conversely to remove the tags from all ingress traffic. VST and virtual guest tagging (VGT) mode require the use of 802.1q trunks.

```
spanning-tree portfast bpduguard default
!
vlan dot1q tag native
!
interface GigabitEthernetX/X
 description <<** VM Port **>>
 no ip address
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk native vlan <id>
 switchport trunk allowed vlan xx,yy-zz
 switchport mode trunk
 switchport nonegotiate
 no cdp enable
 spanning-tree portfast trunk
!
```

Note The native VLAN carries switch control and management data. It is important to ensure that the native VLAN is not used by any of the VMs on the ESX server; that is, it is not supported by a port group.

Additional Resources

For more information about ESX Server 2.5.x releases, refer to the VMware Technology Network website at the following URL:

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