Types of virtualization

Hardware

Hardware virtualization or *platform virtualization* refers to the creation of a virtual machine that acts like a real computer with an operating system. Software executed on these virtual machines is separated from the underlying hardware resources. For example, a computer that is running Microsoft Windows may host a virtual machine that looks like a computer with the Ubuntu Linux operating system; Ubuntu-based software can be run on the virtual machine.

In hardware virtualization, the *host machine* is the actual machine on which the virtualization takes place, and the *guest machine* is the virtual machine. The words *host* and *guest* are used to distinguish the software that runs on the physical machine from the software that runs on the virtual machine. The software or firmware that creates a virtual machine on the host hardware is called a *hypervisor* or *Virtual Machine Manager*.

Different types of hardware virtualization include:

- **Full virtualization**: Almost complete simulation of the actual hardware to allow software, which typically consists of a guest operating system, to run unmodified.
- **Partial virtualization**: Some but not all of the target environment is simulated. Some guest programs, therefore, may need modifications to run in this virtual environment.
- **Paravirtualization**: A hardware environment is not simulated; however, the guest programs are executed in their own isolated domains, as if they are running on a separate system. Guest programs need to be specifically modified to run in this environment. components that help improve the performance of a guest environment.

Note : Hardware virtualization is not the same as hardware emulation. In hardware emulation, a piece of hardware imitates another, while in hardware virtualization, a hypervisor (a piece of software) imitates a particular piece of computer hardware or the entire computer. Furthermore, a hypervisor is not the same as an emulator; both are computer programs that imitate hardware, but their domain of use in language differs.

- *Full virtualization* causes the hypervisor to “trap” the machine operations the OS uses to read or modify the system’s status or perform input/output (I/O) operations. After it has trapped them, the hypervisor emulates these operations in software and returns status codes consistent with what the real hardware would deliver.

- *Paravirtualization (sometimes known as partial virtualization)* eliminates much of the trapping-and-emulation overhead associated with software implemented virtualization by requiring that the guest OS cooperates in creating the virtualizing illusion — it essentially agrees to be fooled by the hypervisor. Paravirtualization can be faster than the other forms of virtualization

Hardware-Assisted Virtualization

- Hardware-assisted virtualization relies on hardware extensions to the x86 system architecture to eliminate much of the hypervisor overhead associated with trapping and emulating I/O operations and status instructions executed within a guest OS.

- AMD Virtualization™ (AMD-V™) is the collective name for AMD's hardware-based virtualization features. With the recent Quad-Core AMD Opteron processor introduction, AMD-V receives an enhancement called Rapid Virtualization Indexing. Rapid Virtualization Indexing provides the Virtualized Page Tables.

- As a result, Quad-Core AMD Opteron processors are expected to provide a significant performance advantage for many virtualized workloads.

- Key hypervisor suppliers (Microsoft, VMware, Virtual Iron, and XenSource) all support elements of AMD-V in their software.

Software

- **Operating system-level virtualization**, hosting of multiple virtualized environments within a single OS instance.

- **Application virtualization**, the hosting of individual applications in an environment separated from the underlying OS. Application virtualization is closely associated with the concept of portable applications.

Memory

Memory virtualization is aggregating Random Access Memory resources from networked systems into a single memory pool.

Memory virtualization decouples volatile RAM resources from individual systems in the data center, and then aggregates those resources into a virtualized memory pool available to any computer in the cluster.

The memory pool is accessed by the operating system or applications running on top of the operating system. The distributed memory pool can then be utilized as a high-speed cache, a messaging layer, or a large, shared memory resource for a CPU application.

Benefits

- Improves memory utilization via the sharing of scarce resources
- Increases efficiency and decreases run time for data intensive and I/O bound applications
- Allows applications on multiple servers to share data without replication, decreasing total memory needs
- Lowers latency and provides faster access than other solutions such as SAN.

Storage virtualization

- Storage virtualization is a concept wherein storage systems may use virtualization concepts as a tool to enable better functionality and more advanced features within and across storage systems.
- Broadly speaking, a 'storage system' is also known as a storage array or Disk array or a filer. Storage systems typically use special hardware and software along with disk drives in order to provide very fast and reliable storage for computing and data processing. Storage systems are complex, and may be thought of as a special purpose computer designed to provide storage capacity along with advanced data protection features. Disk drives are only one element within a storage system, along with hardware and special purpose embedded software within the system.

Within the context of a storage system, there are **two** primary types of virtualization that can occur:

- **Block virtualization** used in this context refers to the abstraction (separation) of logical storage (partition) from physical storage so that it may be accessed without regard to physical storage or heterogeneous structure. This separation allows the administrators of the storage system greater flexibility in how they manage storage for end users.
- **File virtualization** addresses the challenges by eliminating the dependencies between the data accessed at the file level and the location where the files are physically stored. This provides opportunities to optimize storage use and server consolidation and to perform non-disruptive file migrations.

Network

- Network virtualization is creation of a virtualized network addressing space within or across network subnets.
- In computing, **network virtualization** is the process of combining hardware and software network resources and network functionality into a single, software-based administrative entity, a virtual network. Network virtualization involves platform virtualization, often combined with resource virtualization.
- Network virtualization is categorized as either *external*, combining many networks, or parts of networks, into a virtual unit, or *internal*, providing network-like functionality to the software containers on a single system.
- In software testing, network virtualization is a pre-production process for recreating network conditions from the production environment within the test environment. A component of application performance engineering, network virtualization enables connections between applications, services, dependencies and end users to be accurately emulated in the test environment.

CPU Virtualization

- CPU virtualization emphasizes performance and runs directly on the processor whenever possible. The underlying physical resources are used whenever possible and the virtualization layer runs instructions only as needed to make virtual machines operate as if they were running directly on a physical machine.

- CPU virtualization is **not** the same thing as emulation. With emulation, all operations are run in software by an emulator. A software emulator allows programs to run on a computer system other than the one for which they were originally written. The emulator does this by emulating, or reproducing, the original computer's behavior by accepting the same data or inputs and achieving the same results. Emulation provides portability and runs software designed for one platform across several platforms.

VMFS

VMware VMFS (*Virtual Machine File System*) is VMware, Inc.'s cluster file system. It is used by VMware ESX Server and the company's flagship server virtualization suite, vSphere.

It was developed and is used to store virtual machine disk images, including snapshots. Multiple servers can read/write the same filesystem simultaneously. VMFS volumes can be logically "grown" (non-destructively increased in size) by spanning multiple VMFS volumes together. It is not mandatory to use VMFS with VMware; an alternative is NFS.

Features

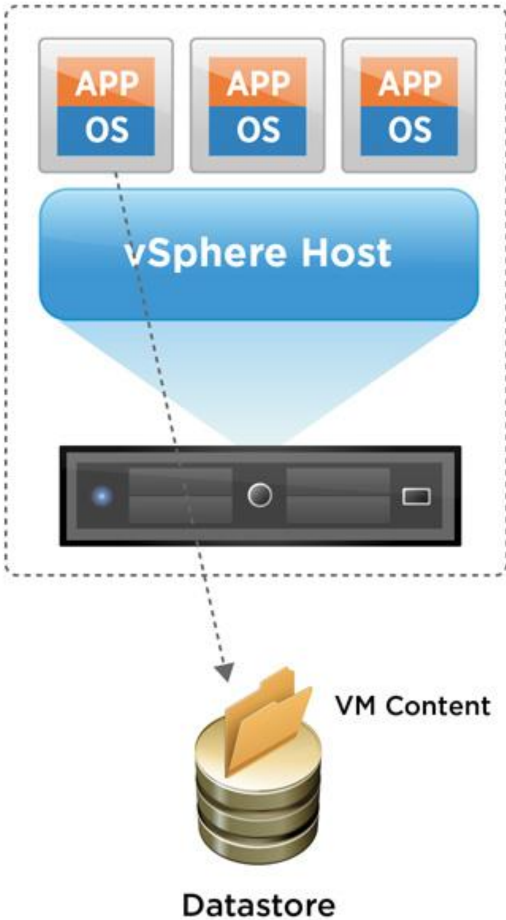
- Simplify virtual machine provisioning and administration with a high performance cluster file system optimized for virtual machines.
- Central storage of virtual machines with VMFS provides more control, flexibility and performance in managing your virtualized IT environment.
- Deliver high service levels.
- Seamlessly manage virtual machine storage.
- Deliver high performance and scalability
- Allows access by multiple ESX servers at the same.
- Optimize your virtual machine I/O with adjustable volume, disk, file and block sizes.
- Recover virtual machines faster and more reliably in the event of server failure.

Technical Details

VMFS is a high-performance file system optimized for storing virtual machines. You can deploy a VMFS datastore on any SCSI-based local or networked storage device, including Fibre Channel, Fiber Channel over Ethernet (FCoE) and iSCSI SAN equipment.

A virtual machine is stored as a set of files in its own directory in a datastore. Datastores are logical containers, analogous to file systems, which hide specifics of each storage device and provide a uniform model for storing virtual machine files. Datastores can also be used for storing ISO images, virtual machine templates, and floppy images.

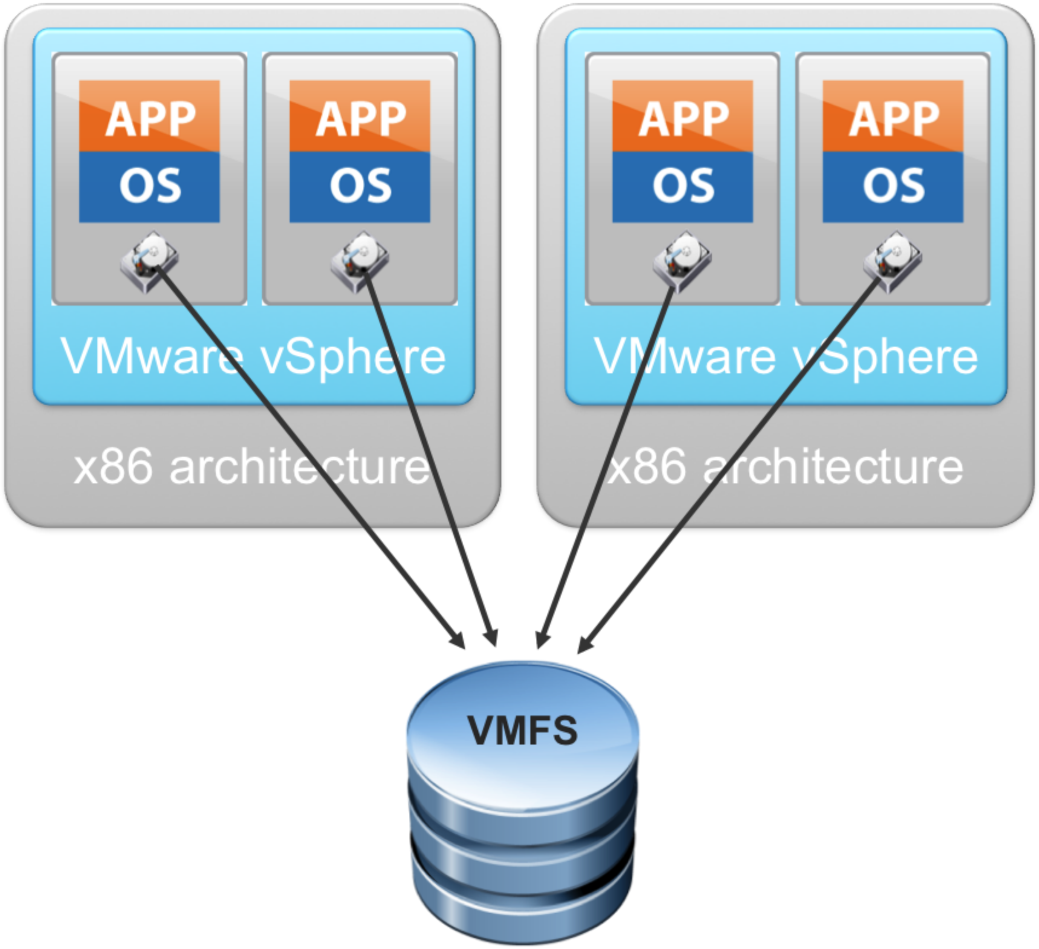
Depending on the type of storage you use, datastores can be backed by the either VMFS or NFS file system formats.


APP
OS
APP
OS
APP
OS
vSphere Host
VM Content
Datastore

- VMware vSphere Virtual Machine File System (VMFS) is a high-performance cluster file system that provides storage virtualization optimized for virtual machines. VMFS is the default storage management interface for block based disk storage.

- VMFS allows multiple instances of VMware vSphere servers to access shared virtual machine storage concurrently. It also enables virtualization-based distributed infrastructure services such as VMware DRS, vSphere HA, vMotion and Storage vMotion to operate across a cluster of vSphere servers.

- In short, VMFS provides the foundation that enables the scaling of virtualization beyond the boundaries of a single system.


APP
OS
APP
OS
VMware vSphere
x86 architecture
APP
OS
APP
OS
VMware vSphere
x86 architecture
VMFS

Interrupt Management

Nachos simulates interrupts by maintaining an event queue together with a simulated clock. As the clock ticks, the event queue is examined to find events scheduled to take place now. The clock is maintained entirely in software and ticks under the following conditions:

Every time interrupts are restored (and the restored interrupt mask has interrupts enabled), the clock advances one tick. Nachos code frequently disables and restores interrupts for mutual exclusion purposes by making explicit calls to *interrupt::SetLevel()*.

Whenever the MIPS simulator executes one instruction, the clock advances one tick.

Whenever the clock advances, the event queue is examined and any pending interrupt events are serviced by invoking the procedure associated with the timer event (e.g., the interrupt service routine). All interrupt service routines are run with interrupts disabled, and the interrupt service routine may not re-enable them.

Nachos simulates a machine that roughly approximates the MIPS architecture. The machine has registers, memory and a cpu. In addition, an event-driven simulated clock provides a mechanism to schedule interrupts and execute them at a later time. The simulated MIPS machine can execute arbitrary programs.

One simply loads instructions into the machine's memory, initializes registers (including the program counter PCReg) and then tells the machine to start executing instructions. The machine then fetches the instruction PCReg points at, decodes it, and executes it.

The process is repeated indefinitely, until an illegal operation is performed or a hardware interrupt is generated. When a trap or interrupt takes place, execution of MIPS instructions is suspended, and a Nachos interrupt service routine is invoked to deal with the condition.