MODULE – 3

VIRTUALIZED DATA CENTER – COMPUTE

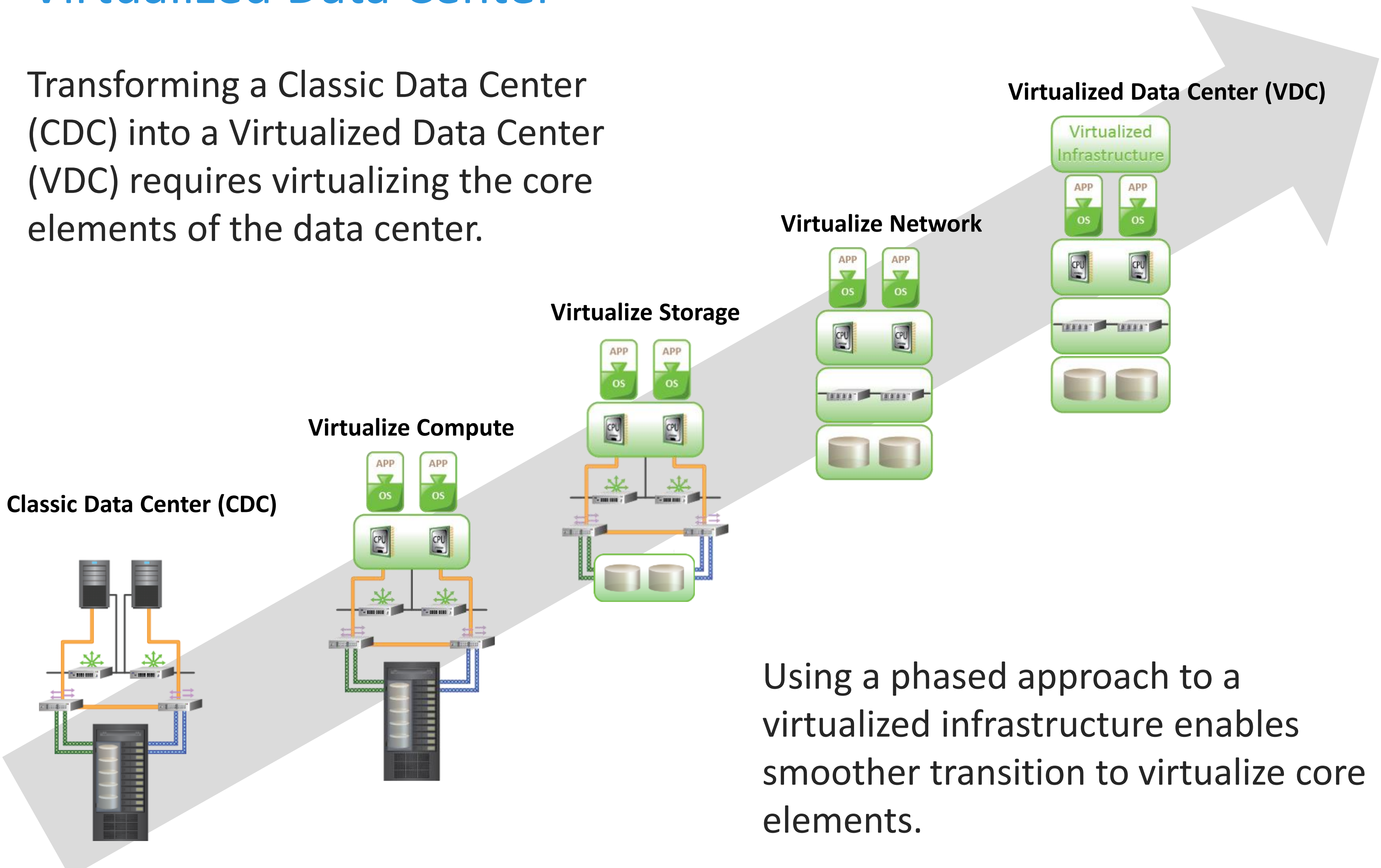
Module 3: Virtualized Data Center – Compute

Upon completion of this module, you should be able to:

- Describe compute virtualization
- Discuss the compute virtualization techniques
- Explain the virtual machine (VM) components
- Describe resource management and resource optimization techniques
- Describe the process to convert physical machine to VM

Virtualized Data Center

Transforming a Classic Data Center (CDC) into a Virtualized Data Center (VDC) requires virtualizing the core elements of the data center.



Using a phased approach to a virtualized infrastructure enables smoother transition to virtualize core elements.

Module 3: Virtualized Data Center – Compute

Lesson 1: Compute Virtualization Overview

Topics covered in this lesson:

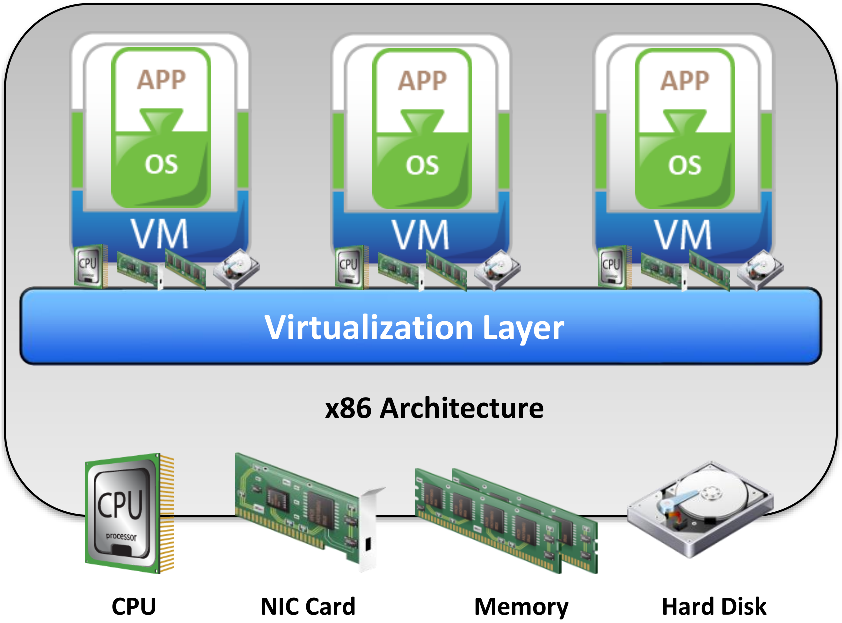
- Drivers for compute virtualization
- Types of hypervisor
- Benefits of compute virtualization

Compute Virtualization

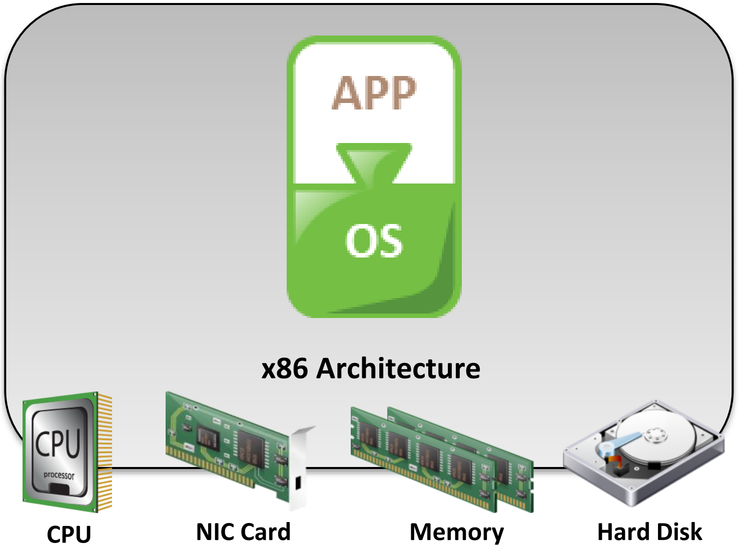
Compute Virtualization

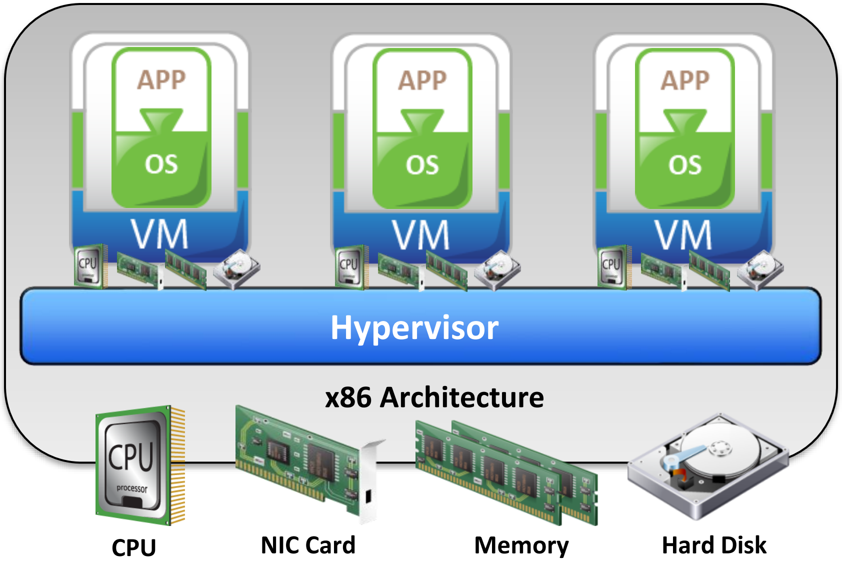
It is a technique of masking or abstracting the physical compute hardware and enabling multiple operating systems (OSs) to run concurrently on a single or clustered physical machine(s).

- Enables creation of multiple virtual machines (VMs), each running an OS and application
  - VM is a logical entity that looks and behaves like physical machine
- Virtualization layer resides between hardware and VMs
  - Also known as hypervisor
- VMs are provided with standardized hardware resources



Need for Compute Virtualization





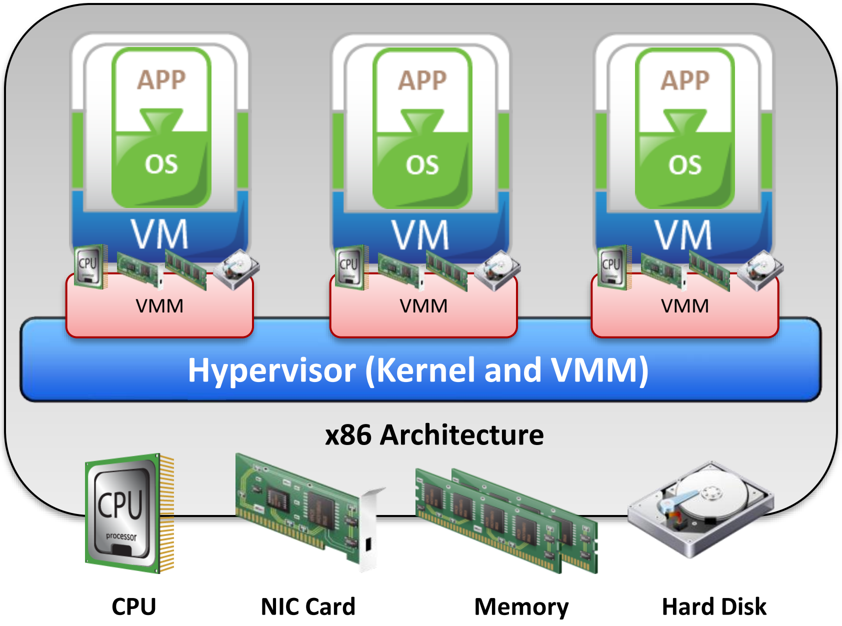
Before Virtualization	After Virtualization
• Runs single operating system (OS) per machine at a time • Couples s/w and h/w tightly • May create conflicts when multiple applications run on the same machine • Underutilizes resources • Is inflexible and expensive	• Runs multiple operating systems (OSs) per machine concurrently • Makes OS and applications h/w independent • Isolates VM from each other, hence no conflict • Improves resource utilization • Offers flexible infrastructure at low cost

Hypervisor

Hypervisor

It is a software that allows multiple operating systems (OSs) to run concurrently on a physical machine and to interact directly with the physical hardware.

- Has two components
  - Kernel
  - Virtual Machine Monitor (VMM)

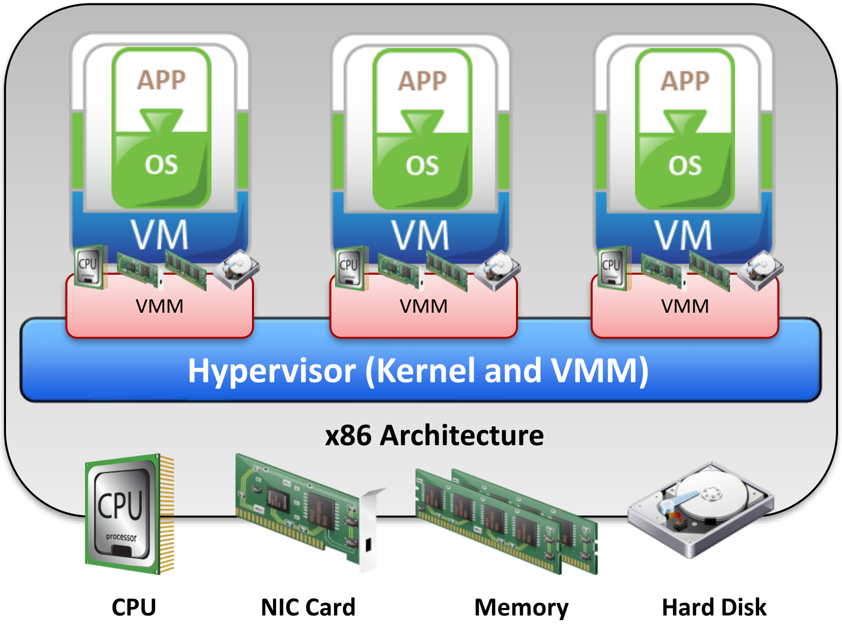


Hypervisor

Hypervisor

It is a software that allows multiple operating systems (OSs) to run concurrently on a physical machine and to interact directly with the physical hardware.

- Has two components
  - Kernel
    - Provides functionalities like an OS
    - E.g. process creation and scheduling, file management, I/O processing

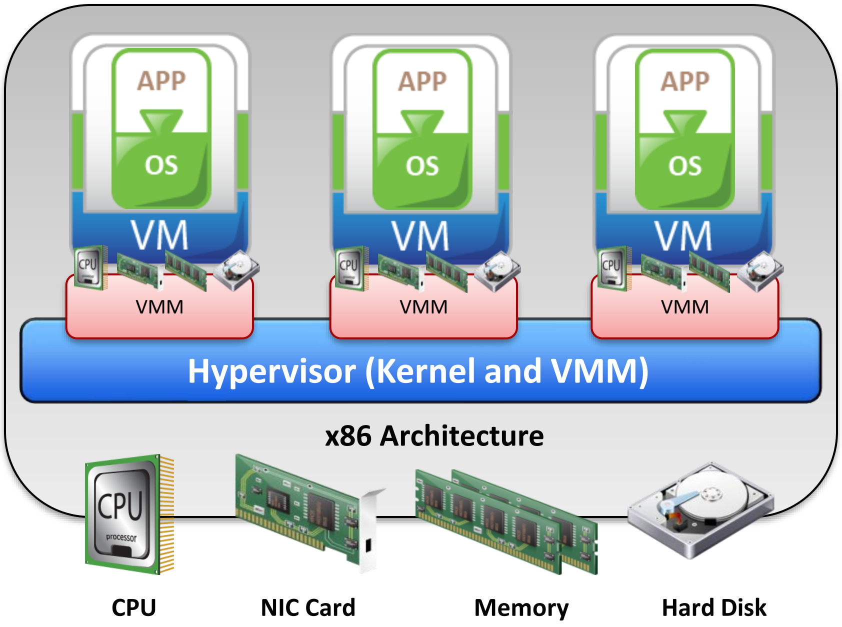


Hypervisor

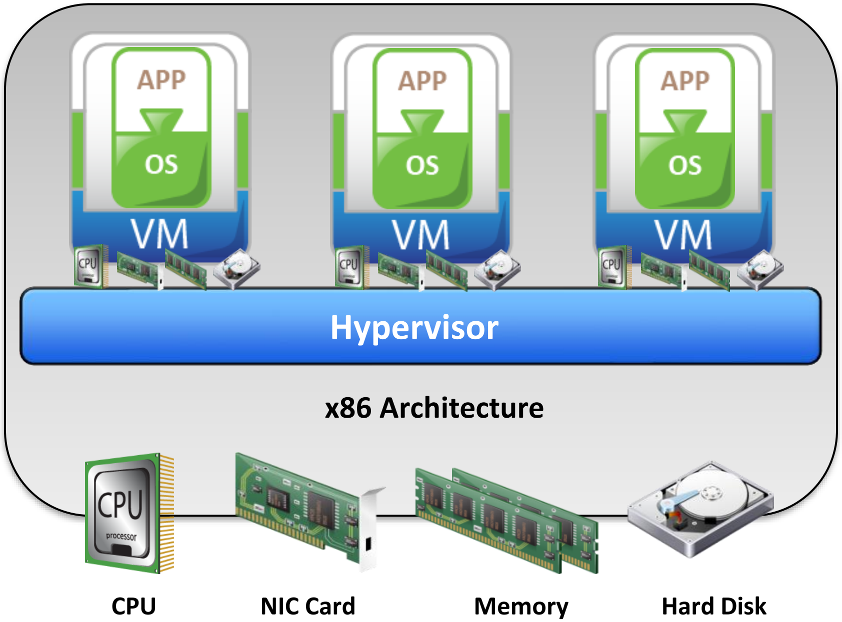
Hypervisor

It is a software that allows multiple operating systems (OSs) to run concurrently on a physical machine and to interact directly with the physical hardware.

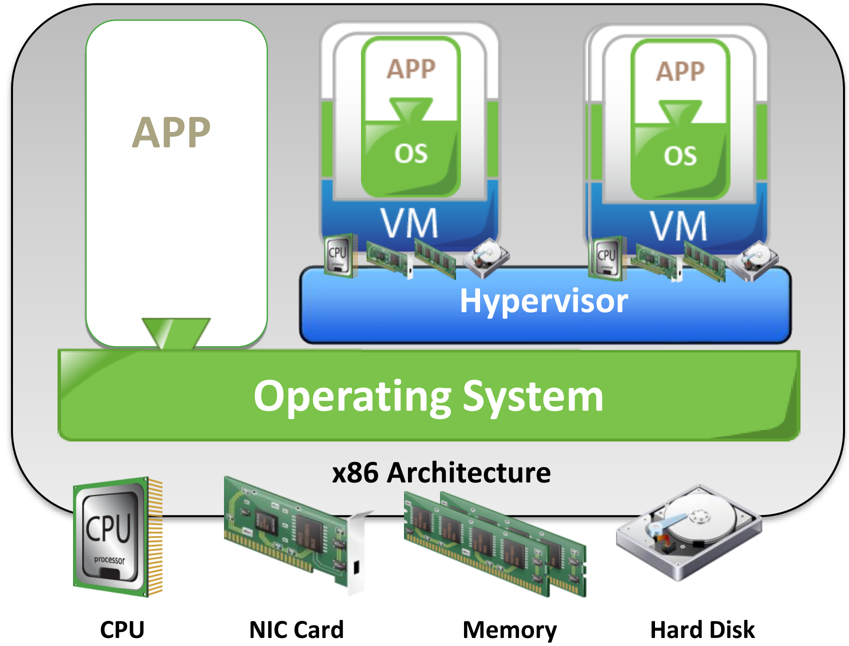
- Has two components
  - Virtual Machine Monitor (VMM)
    - Responsible for actual execution of the commands on the CPUs and performing Binary Translation (BT)
    - abstracts the HD for the VM



Types of Hypervisor



Type 1: Bare-Metal Hypervisor



Type 2: Hosted Hypervisor

Type 1: Bare-Metal Hypervisor

- It is an operating system (OS)
- It installs and runs on x86 bare-metal hardware
- It requires certified hardware

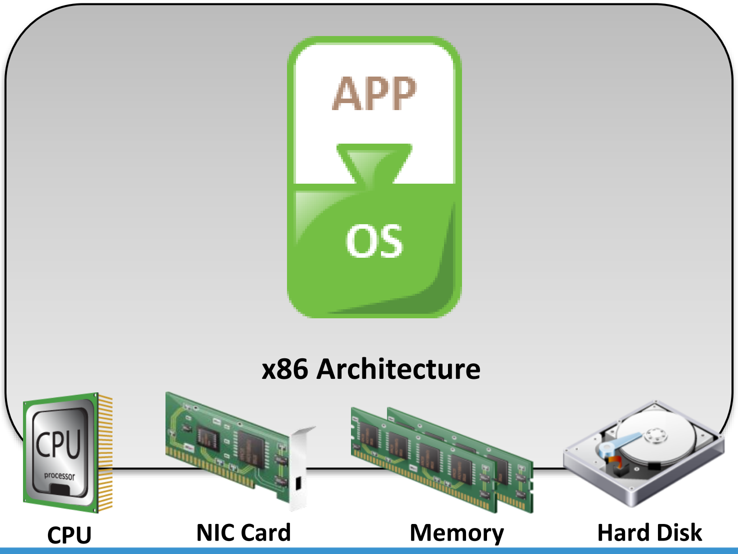
Type 2: Hosted Hypervisor

- It installs and runs as an application
- It relies on operating system (OS) running on physical machine for device support and physical resource management

Hypervisors

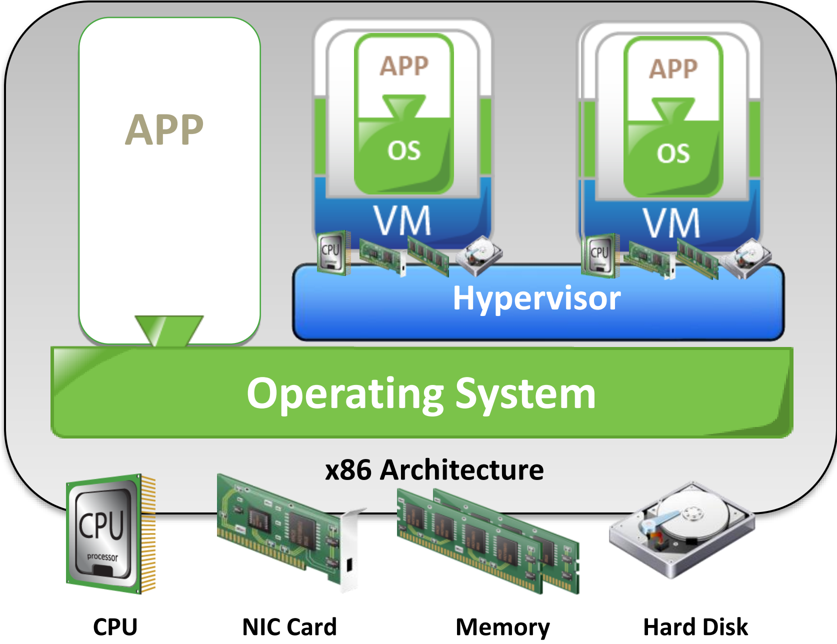
- Hypervisors are categorized into two types: hosted hypervisor and bare-metal hypervisor.

Type 1 (Bare-metal hypervisor): In this type, the hypervisor is directly installed on the x86 based hardware. Bare-metal hypervisor has direct access to the hardware resources. Hence, it is more efficient than a hosted hypervisor.



Hypervisors

Type 2 (Hosted hypervisor): In this type, the hypervisor is installed and run as an application on top of an operating system. Since it is running on an operating system, it supports the broadest range of hardware configurations.



Type 2: Hosted Hypervisor

- A hypervisor is the primary component of virtualization that enables compute system partitioning (i.e. partitioning of CPU and memory). In this course, we will focus on type 1 hypervisors because it is most predominantly used within Virtualized Data Center (VDC).

Benefits of Compute Virtualization

- Hardware independence
  - A virtual machine is configured with virtual components such as CPU, memory, network card, and SCSI controller that are completely independent of the underlying physical hardware.
  - This gives the freedom to move a virtual machine from one x86 machine to another without making any change to the device drivers, operating system, or applications.
- Reduced cost
  - Compute virtualization reduces the following direct costs:
  - Space (leased or owned) for physical machines, power and cooling, Hardware (including switches and Fibre Channel HBA), and annual maintenance

Benefits of Compute Virtualization

- Server consolidation
  - enables running multiple virtual machines on a physical server
- Isolation
  - While virtual machines can share the physical resources of a physical machine, they remain completely isolated from each other as if they were separate physical machines.
  - If, for example, there are four virtual machines on a single physical machine and one of the virtual machines crashes, the other three virtual machines remain unaffected.
- Encapsulation
  - A virtual machine is a package that contains a complete set of virtual hardware resources, an operating system, and applications. Encapsulation makes virtual machines portable and easy to manage.
  - For example, a virtual machine can be moved and copied from one location to another just like a file.

Module 3: Virtualized Data Center – Compute

Lesson 2: Compute Virtualization Techniques

Topics covered in this lesson:

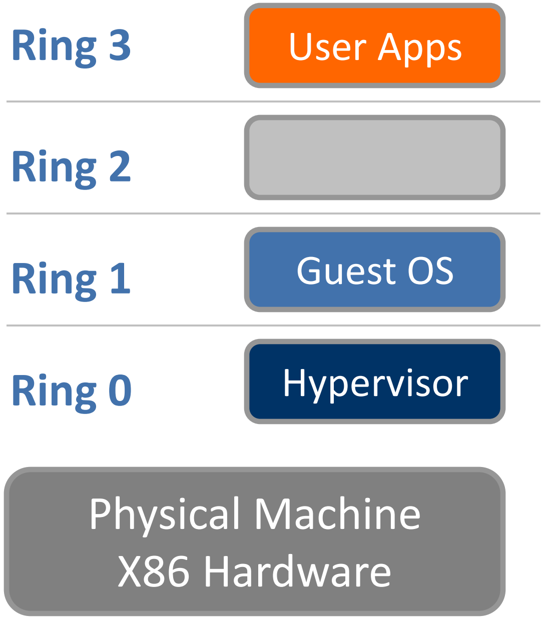
- Requirements of x86 hardware virtualization
- Compute virtualization techniques

Requirements: x86 Hardware Virtualization

- An operating system (OS) is designed to run on a bare-metal hardware and to fully own the hardware
  - x86 architecture offer four levels of privilege
    - Ring 0, 1, 2, and 3
    - User applications run in Ring 3
    - OS run in Ring 0 (most privileged)
- Challenges of virtualizing x86 hardware
  - Requires placing the virtualization layer below the OS layer
  - Is difficult to capture and translate privileged OS instructions at runtime
- Techniques to virtualize compute
  - **Full, Para, and hardware assisted virtualization**

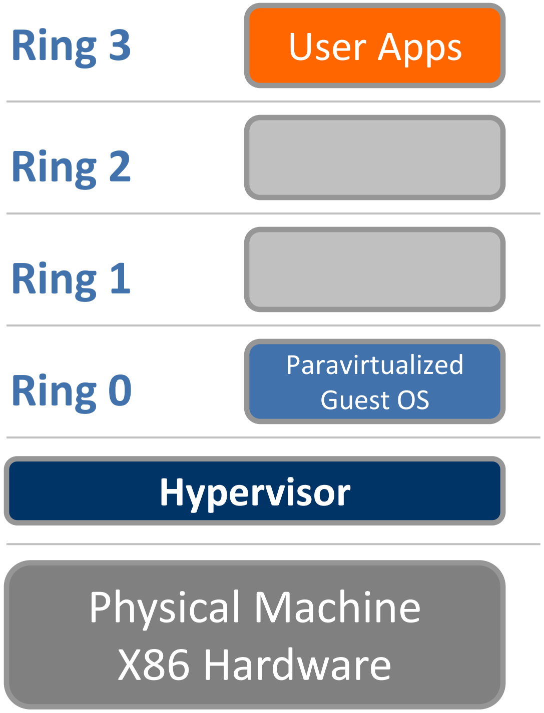
Full Virtualization

- Virtual Machine Monitor (VMM) runs in the privileged Ring 0
- Guest operating system (OS)does not know that it is virtualized
- VMM decouples guest operating system (OS) from the underlying physical hardware
- Each VM is assigned a VMM
  - Provides virtual components to each VM
  - Performs Binary Translation (BT)
  - Guest OS is not aware of being virtualized
  - **VMware ESX/ESXi (VMware vSphere)** and **Microsoft Hyper-V** are product examples that implement the full virtualization technique.



Paravirtualization (PV)

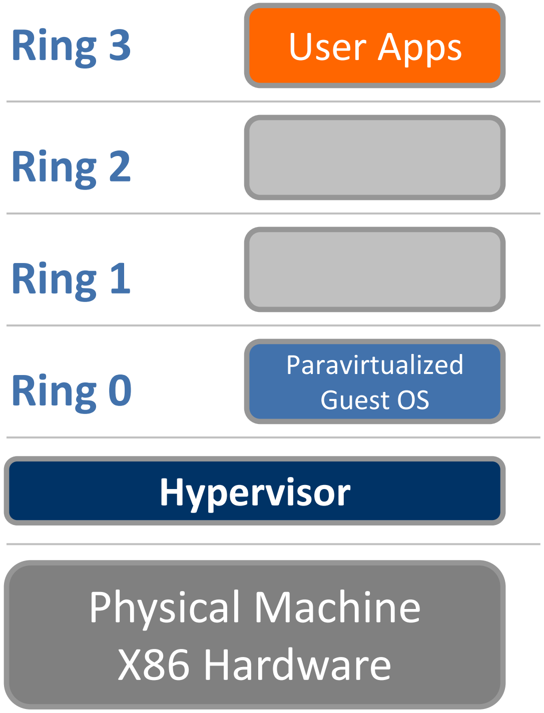
- Guest operating system (OS) knows that it is virtualized
- Guest OS runs in Ring 0
- Modified guest OS kernel is used, such as Linux and OpenBSD
- Possible in open source operating systems
- Unmodified guest OS is not supported, such as Microsoft Windows
  - Xen and KVM are product examples of paravirtualization.
  - http://wiki.xen.org/



Paravirtualization

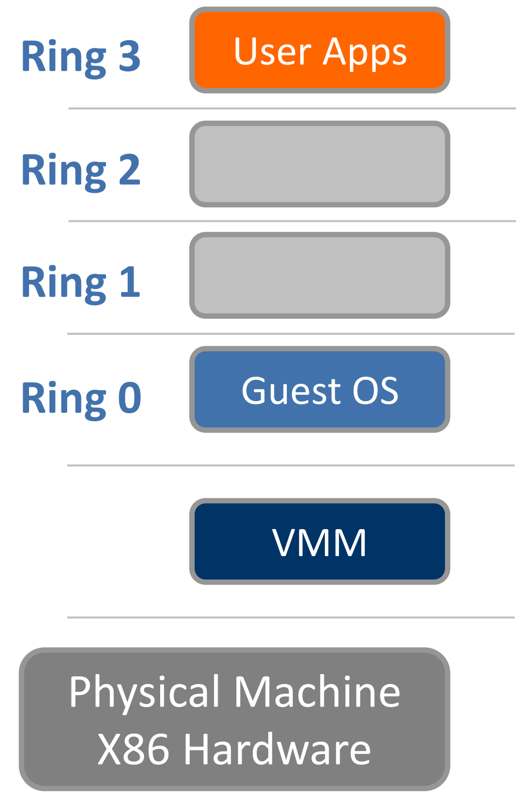
Notes

- PV originally introduced by Xen Project, later adopted by other virtualization platforms
- PV-enabled kernels exist for Linux, NetBSD, FreeBSD and OpenSolaris. Linux kernels have been PV-enabled from 2.6.24 using the Linux pvops framework.
- In practice this means that PV will work with most Linux distributions



Hardware Assisted Virtualization

- Achieved by using hypervisor-aware CPU to handle privileged instructions
  - Reduces virtualization overhead caused due to full and paravirtualization
  - CPU and Memory virtualization support is provided in hardware
- Enabled by AMD-V and Intel VT technologies in the x86 processor architecture



- *Well-known implementations of hardware-assisted x86 virtualization include VMware Workstation (for 64-bit guests only), Xen 3.x (including derivatives like Virtual Iron), Linux KVM and Microsoft Hyper-V.
- * http://en.wikipedia.org/wiki/Hardware-assisted_virtualization

Hardware Assisted Virtualization (why?)*

- In the traditional x86 architecture, operating system kernels expect direct CPU access running in Ring 0, which is the most privileged level.

- With software virtualization, guest operating systems cannot run in Ring 0 because the VMM sits there.

- The guest operating systems must therefore run in Ring 1, but there's a catch: Some x86 instructions work only in Ring 0, so the operating systems must be recompiled to avoid them.
- *http://www.webopedia.com/DidYouKnow/Computer_Science/hardware_assisted_virtualization.asp

EMC² where information lives

Hardware Assisted Virtualization (Pros)

- This process is called paravirtualization, and it is impractical — especially if the source code for the OS is not available.
- To get around this, VMMs traps these instructions and emulates them, which unfortunately results in an enormous performance hit: Virtual machines can be significantly slower than real physical ones.

Hardware Assisted Virtualization (Pros)

Thus, Intel and AMD have introduced their new virtualization technologies, a handful of new instructions and — crucially — a new privilege level.

The hypervisor can now run at "Ring -1"; so the guest operating systems can run in Ring 0.

There's no need for paravirtualization, the VMM does less work, and the performance hit is reduced.

Virtualization products

Check out this page:
http://en.wikipedia.org/wiki/VMware_vSphere

Module 3: Virtualized Data Center – Compute

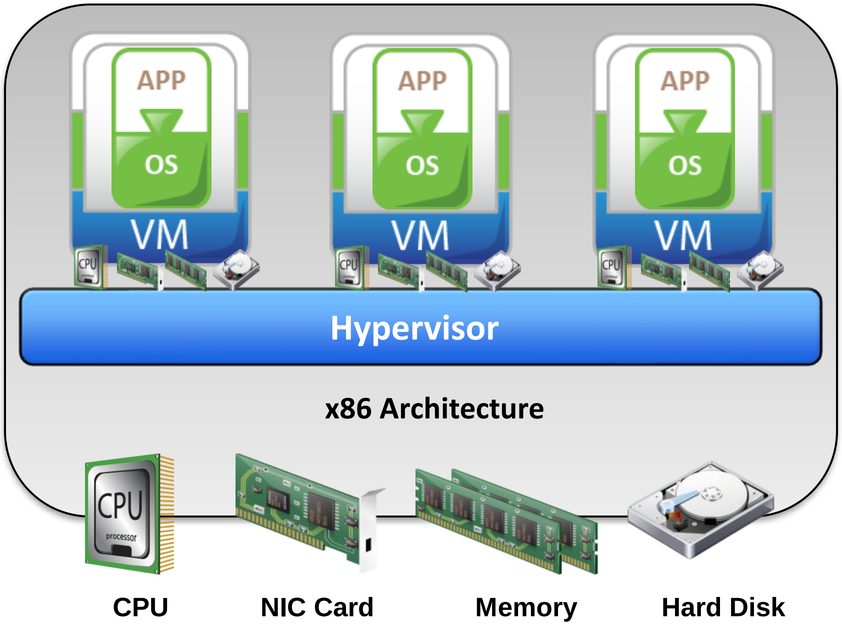
Lesson 3: Virtual Machine

Topics covered in this lesson:

- Virtual machine (VM) files
- File systems that manage Virtual machine files
- Virtual machine hardware
- Virtual machine console

Virtual Machine

- From a user's perspective, a logical compute system
  - Runs an operating system (OS) and application like a physical machine
  - Contains virtual components such as CPU, RAM, disk, and NIC
- From a hypervisor's perspective
  - Virtual machine (VM) is a discrete set of files such as configuration file, virtual disk files, virtual BIOS file, VM swap file, and log file



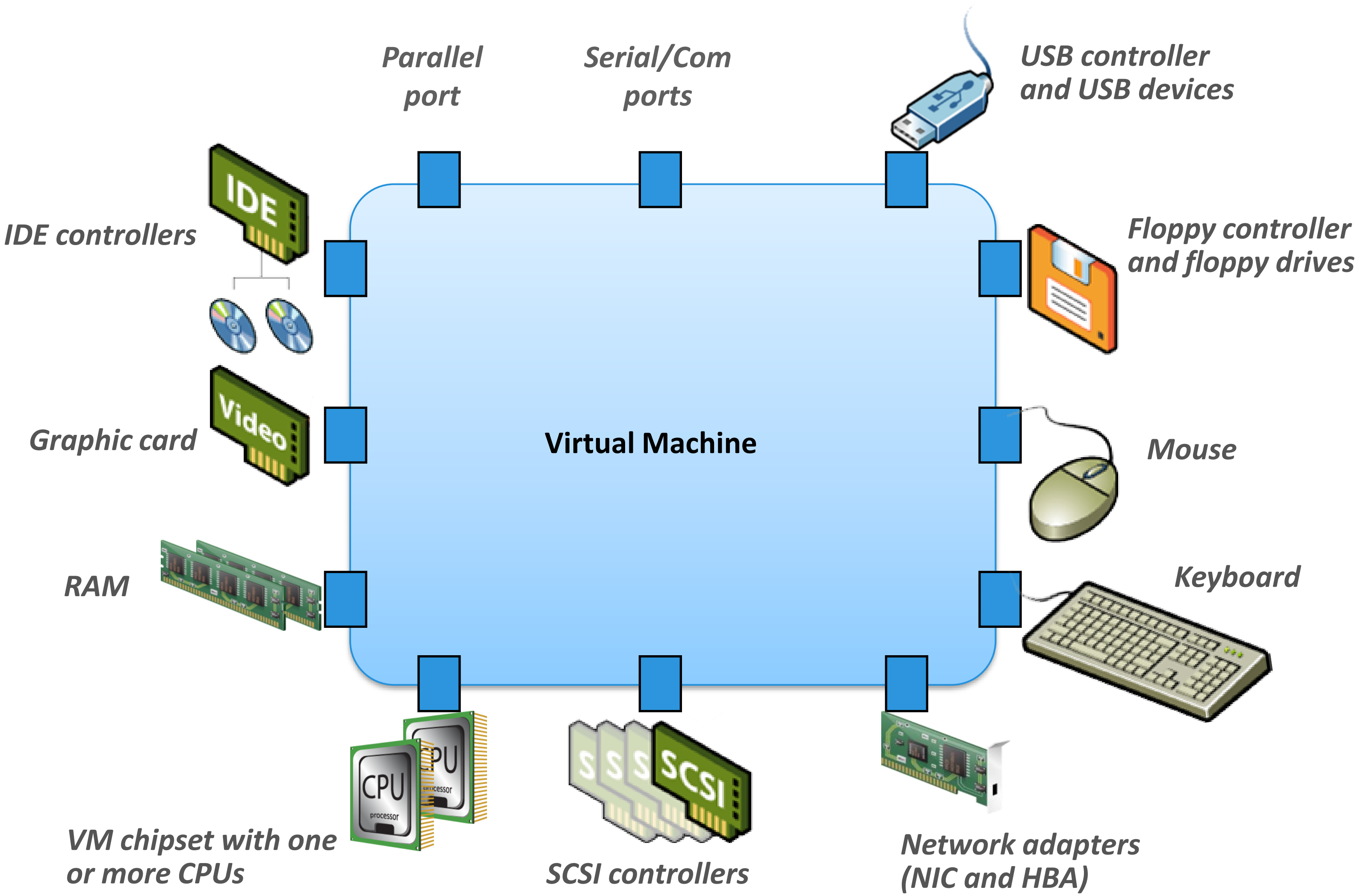
Virtual Machine Files

File name	Description
Virtual BIOS File	• Stores the state of the virtual machine's (VM's) BIOS
Virtual Swap File	• Is a VM's paging file which backs up the VM RAM contents • The file exists only when VM is running
Virtual Disk File	• Stores the contents of the VM's disk drive • Appears like a physical disk drive to VM • VM can have multiple disk drives
Log File	• Keeps a log of VM activity • Is useful for troubleshooting
Virtual Configuration File	• Stores the configuration information chosen during VM creation • Includes information such as number of CPUs, memory, number and type of network adaptors, and disk types

File System to Manage VM Files

- The file systems supported by hypervisor are Virtual Machine File System (VMFS) and Network File System (NFS)
- VMFS
  - Is a cluster file system that allows multiple physical machines to perform read/write on the same storage device concurrently
  - Is deployed on FC and iSCSI storage apart from local storage
- NFS
  - Enables storing VM files on a remote file server (NAS device) using IP
  - NFS client is built into hypervisor

Virtual Machine Hardware



Virtual Machine Hardware

Notes:

- All virtual machines have standardized hardware. Standardized hardware makes virtual machine portable across physical machines.

- In the current implementation of hypervisors, not all devices are available to add and configure; for example video devices cannot be added, but the available video device can be configured.

VM Hardware Components

Virtual Hardware	Description
vCPU	• Virtual machine (VM) can be configured with one or more virtual CPUs • Number of CPUs allocated to a VM can be changed
vRAM	• Amount of memory presented to the guest operating system (OS) • Memory size can be changed based on requirement
Virtual Disk	• Stores VM's OS and application data • A VM should have at least one virtual disk
vNIC	• Enables a VM to connect to other physical and virtual machines
Virtual DVD/CD-ROM Drive	• It maps a VM’s DVD/CD-ROM drive to either a physical drive or an .iso file
Virtual Floppy Drive	• It maps a VM’s floppy drive to either a physical drive or an .flp file
Virtual SCSI Controller	• VM uses virtual SCSI controller to access virtual disk
Virtual USB Controller	• Maps VM’s USB controller to the physical USB controller

Virtual Machine Console

- Provides mouse, keyboard, and screen functionality
- Sends power changes (on/off) to the virtual machine (VM)
- Allows access to BIOS of the VM
- Typically used for virtual hardware configuration and troubleshooting issues

Module 3: Virtualized Data Center – Compute

Lesson 4: Resource Management

Topics covered in this lesson:

- Resource management and resource pool
- Share, limit, and reservation
- CPU and memory resource optimization techniques

Resource Management

Resource management

A process of allocating resources from physical machine or clustered physical machines to virtual machines (VMs) to optimize the utilization of resources.

- Goals of resource management
  - Controls utilization of resources
  - Prevents VMs from monopolizing resources
  - Allocates resources based on relative priority of VMs
- Resources must be pooled to manage them centrally
- *Note: Resource management includes management of CPU, memory, network, and storage. This module focuses on the management of CPU and memory resources. Network and storage resource management are covered in Module 4 and 5 of this course*

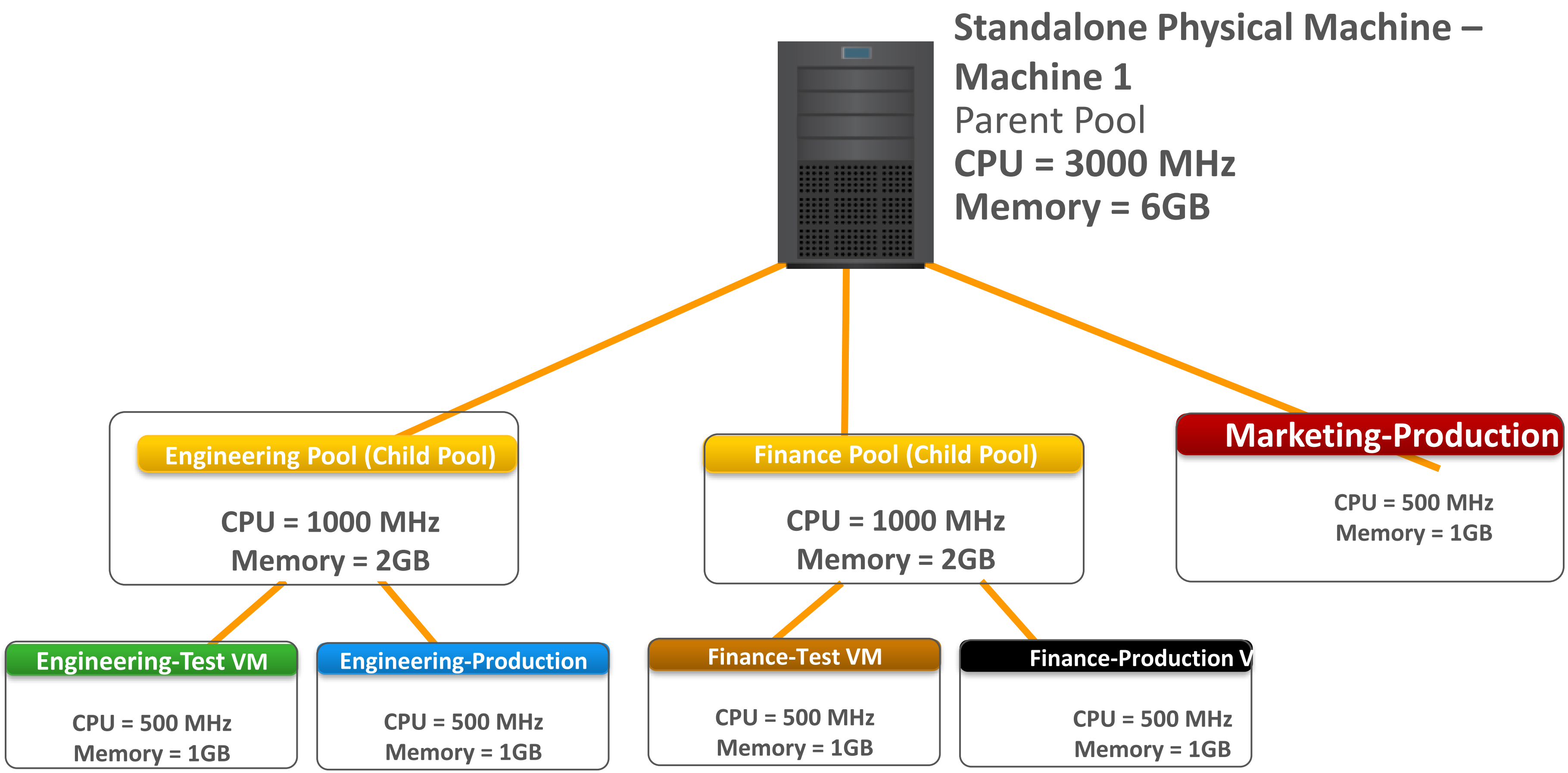
Resource Pool

Resource pool

It is a logical abstraction of aggregated physical resources that are managed centrally.

- Created from a physical machine or cluster
- Administrators may create child resource pool or virtual machine (VM) from the parent resource pool
- Reservation, limit, and share are used to control the resources consumed by resource pools or VMs

Resource Pool Example



Share, Limit, and Reservation

- Parameters that control the resources consumed by a child resource pool or a virtual machine (VM) are as follows:
  - Share
    - Amount of CPU or memory resources a VM or a child resource pool can have with respect to its parent's total resources
  - Limit
    - Maximum amount of CPU and memory a VM or a child resource pool can consume
  - Reservation
    - Amount of CPU and memory reserved for a VM or a child resource pool

Optimizing CPU Resources

- Modern CPUs are equipped with multiple cores and hyper-threading
  - Multi-core processors have multiple processing units (cores) in a single CPU
    - A multi-core CPU is an integrated circuit to which two or more processing units (cores) have been attached for enhanced performance and more efficient, simultaneous processing of multiple processes
  - Hyper-threading makes a physical CPU appear as two or more logical CPUs
    - Hyper-threading makes a physical CPU appear as two or more logical CPUs.

EMC² where information lives

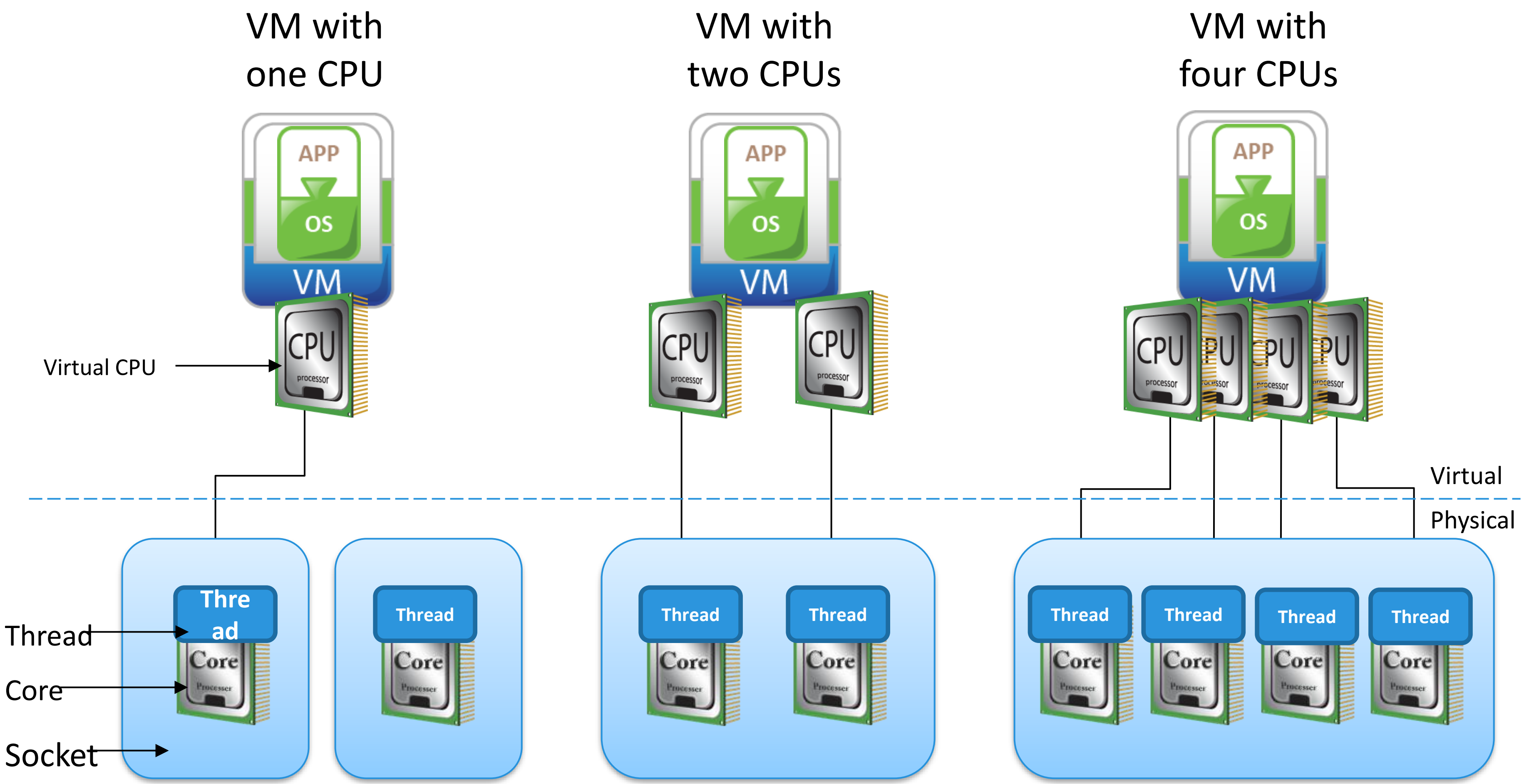
Optimizing CPU Resources

- Allocating a CPU resource efficiently and fairly is critical
- Hypervisor schedules virtual CPUs on the physical CPUs
- Hypervisors support multi-core, hyper-threading, and CPU load-balancing features to optimize CPU resources

- The role of the hypervisor scheduler is to assign a physical CPU resource to the virtual CPU in a way that meets system objectives, such as responsiveness, throughput, and utilization.

- A conventional Operating System schedules a process or thread on a CPU, while a hypervisor schedules virtual CPUs of virtual machines on the physical machines.

Multi-core Processors



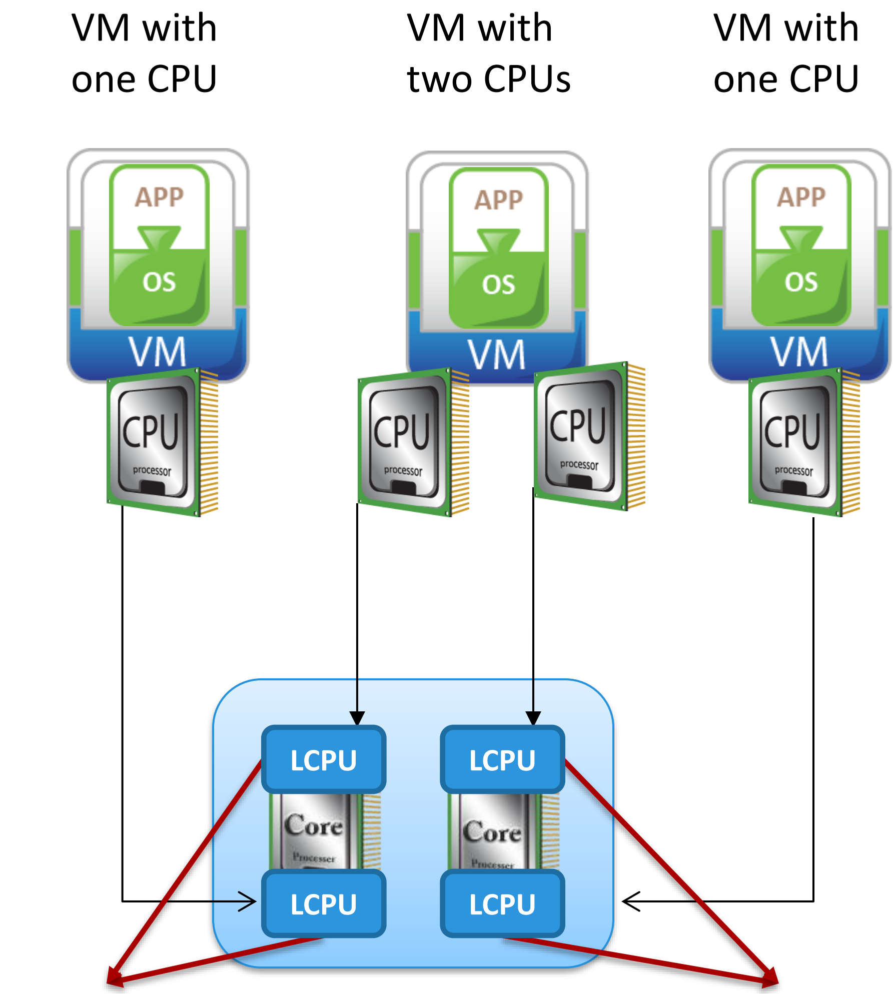
Single – core
Dual – socket system

Dual – core
Single – socket system

Quad – core
Single – socket system

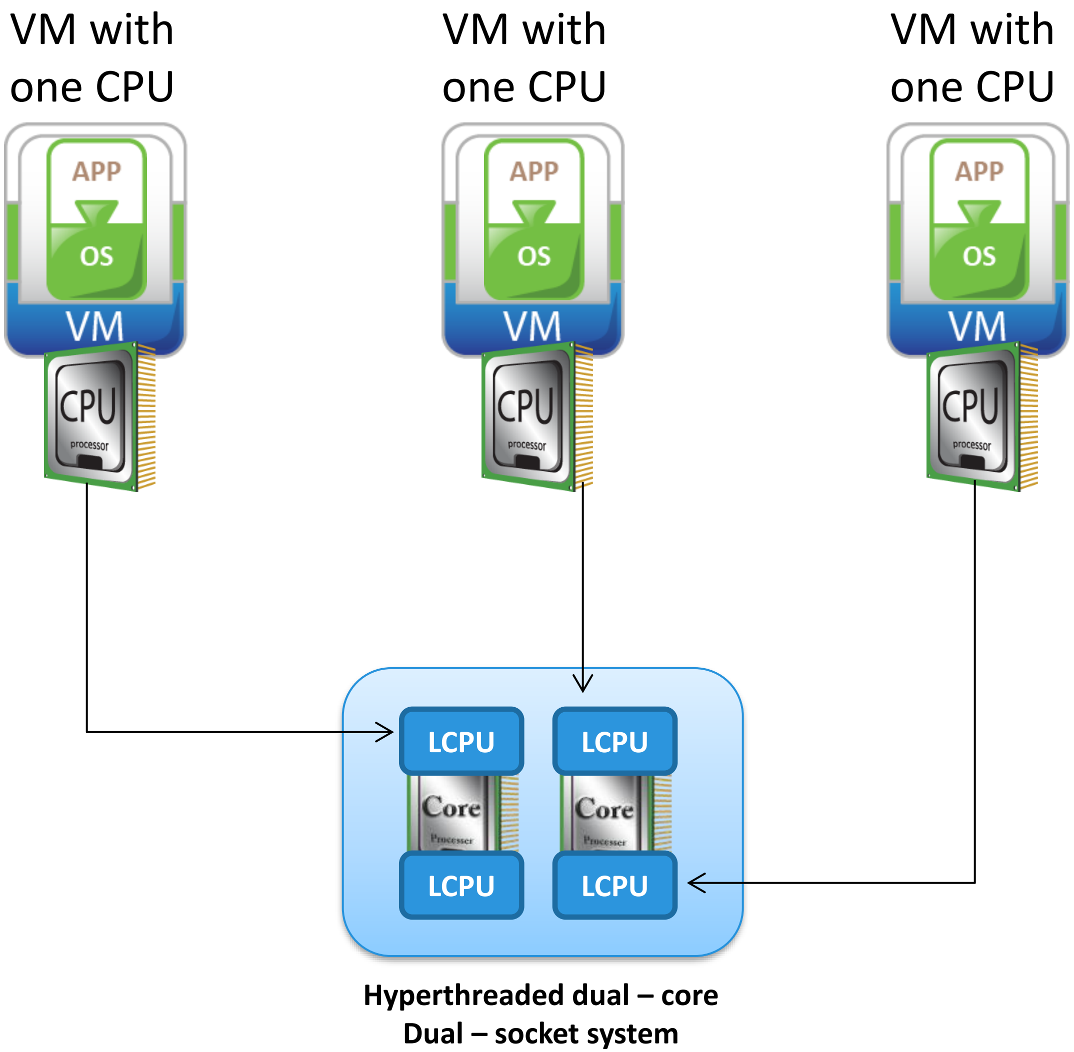
Hyper-threading

- Makes a physical CPU appear as two Logical CPUs (LCPUs)
  - Enables operating system (OS) to schedule two or more threads simultaneously
- Two LCPUs share the same physical resources
  - While the current thread is stalled, CPU can execute another thread
- Hypervisor running on a hyper-threading-enabled CPU provides improved performance and utilization



Dual – core
Single – socket system with hyperthreading

CPU Load Balancing



, load balancing is achieved by migrating a thread from one logical CPU (over utilized) to another (under utilized) to keep the load balanced

Question

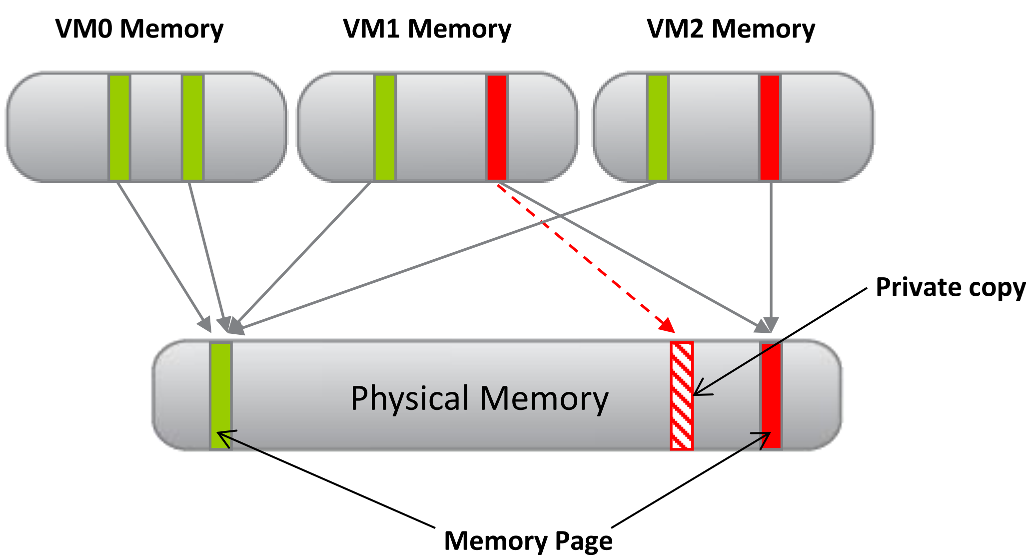
- Can VMs can be configured with more memory than physically available?

Optimizing Memory Resource

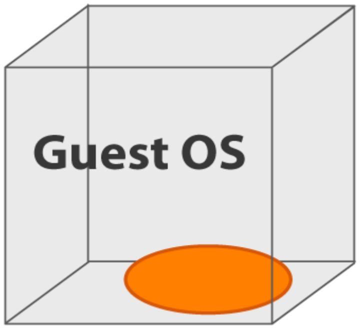
- Hypervisor manages a machine's physical memory
  - Part of this memory is used by the hypervisor
  - Rest is available for virtual machines (VMs)
- VMs can be configured with more memory than physically available, called 'memory overcommitment'
  - Memory optimization is done to allow overcommitment
- Memory management techniques are Transparent page sharing, memory ballooning, and memory swapping

Transparent Page Sharing

- Hypervisor detects identical memory pages of virtual machines (VMs) and maps them to same physical page
  - Read-only when shared
- For writes, hypervisor treats the shared pages as copy-on-write
- Attempts to write on shared page
  - Generates minor page fault
  - Creates private copy after write and remaps the memory



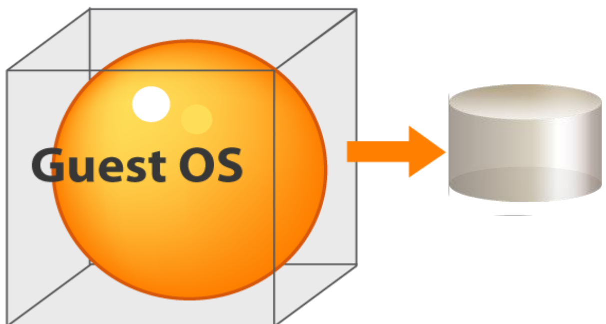
Memory Ballooning



Virtual Machine (VM)

No memory shortage, balloon remains uninflated

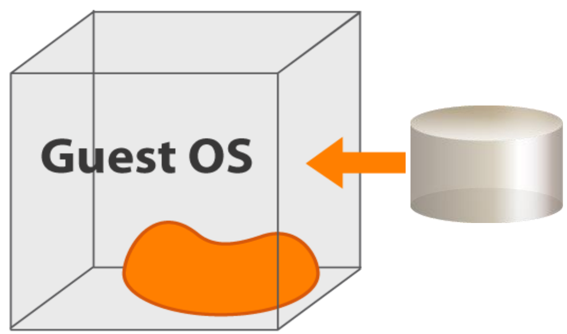
When a virtual machine (VM) must yield memory, the best thing is to let the guest operating system of the VM select the memory pages to give up.



Virtual Machine (VM)

1. Memory shortage, balloon inflates
2. Driver demands memory from guest operating system (OS)
3. Guest OS forces page out
4. Hypervisor reclaims memory

1. Memory shortage resolved, deflates balloon
2. Driver relinquishes memory
3. Guest OS can use pages
4. Hypervisor grants memory



Virtual Machine (VM)

Memory Swapping

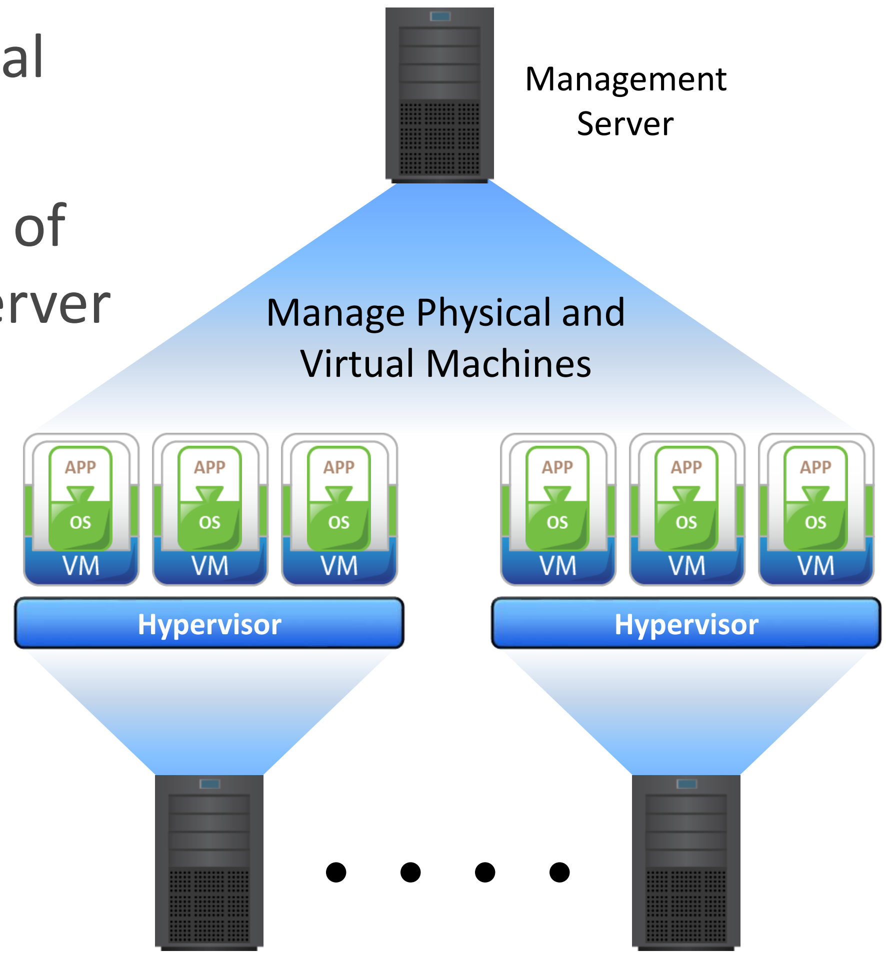
- Each powered-on virtual machine (VM) needs its own swap file
  - Created when the VM is powered-on
  - Deleted when the VM is powered-off
- Swap file size is equal to the difference between the memory limit and the VM memory reservation
- Hypervisor swaps out the VM's memory content if memory is scarce
- Swapping is the last option because it causes notable performance impact
  - If a physical machine cannot get enough memory through page sharing and memory ballooning, the hypervisor forcibly reclaims memory from virtual machines by memory swapping.
  - The hypervisor copies the VM page contents to their corresponding swap files before assigning the pages to the virtual machines that need memory.

Virtual Machine Affinity

- VM to VM affinity:
  - Selected VMs should run on **same hypervisor**
    - To improve performance ,if VMs are communicating with each other heavily
  - Anti-affinity ensures that selected VMs are not together on a hypervisor (ex: for availability reasons)
- VM to physical server (hypervisor) affinity:
  - Specify whether selected VM can be placed only on a particular hypervisor (ex: for licensing issues)
  - Anti-affinity is allowing VM to move on different hypervisors in a cluster (ex: for high availability or performance requirements)

Resource Management Tool

- Provides ability to manage physical machines running hypervisor
- Enables centralized management of resources from a management server
- Enables pooling of resources and allocates capacity to VMs
  - Communicates with hypervisors to perform management
- Provides operational automation



Module 3: Virtualized Data Center – Compute

Lesson 5: Physical to Virtual Machines Conversion

Topics covered in this lesson:

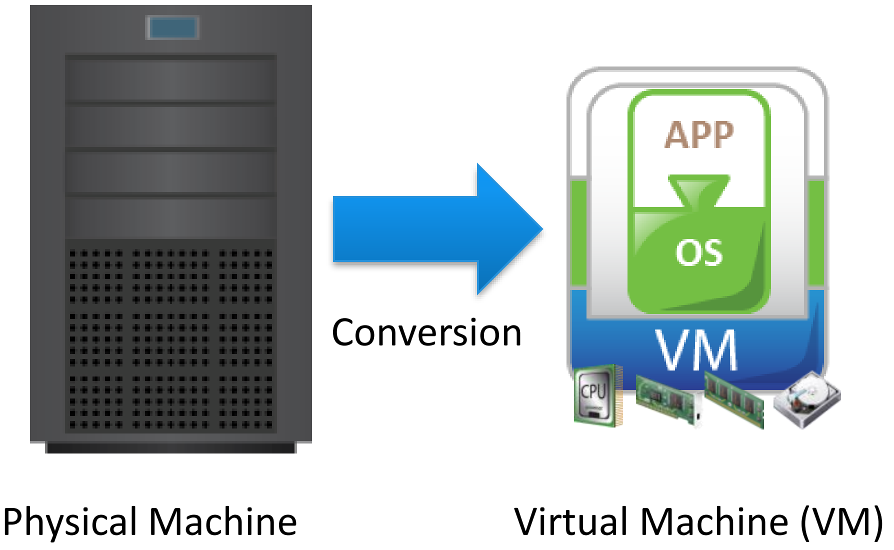
- Converter components
- Conversion options
- Conversion process
- Conversion considerations

Physical to Virtual Machine (P2V) Conversion

P2V Conversion

It is a process through which physical machines are converted into virtual machines (VMs).

- Clones data from physical machine's disk to VM disk
- Performs system reconfiguration of the destination VM such as:
  - Change IP address and computer name
  - Install required device drivers to enable the VM to boot



Benefits of P2V Converter

- Reduces time needed to setup new virtual machine (VM)
- Enables migration of legacy machine to a new hardware without reinstalling operating system (OS) or application
- Performs migration across heterogeneous hardware

Components of P2V Converter

- There are three key components:
  - **Converter server**
    - An application that is loaded on a separate physical machine
    - Is responsible for controlling conversion process
    - Is used for hot conversion only (when source is running its OS)
    - Pushes and installs converter agent on the source machine
  - **Converter agent**
    - Is responsible for performing the conversion
    - Is used in hot mode only
    - Is installed on physical machine to convert it to virtual machine (VM)
  - **Converter Boot CD**
    - Bootable CD contains its operating system (OS) and converter application
    - Converter application is used to perform cold conversion

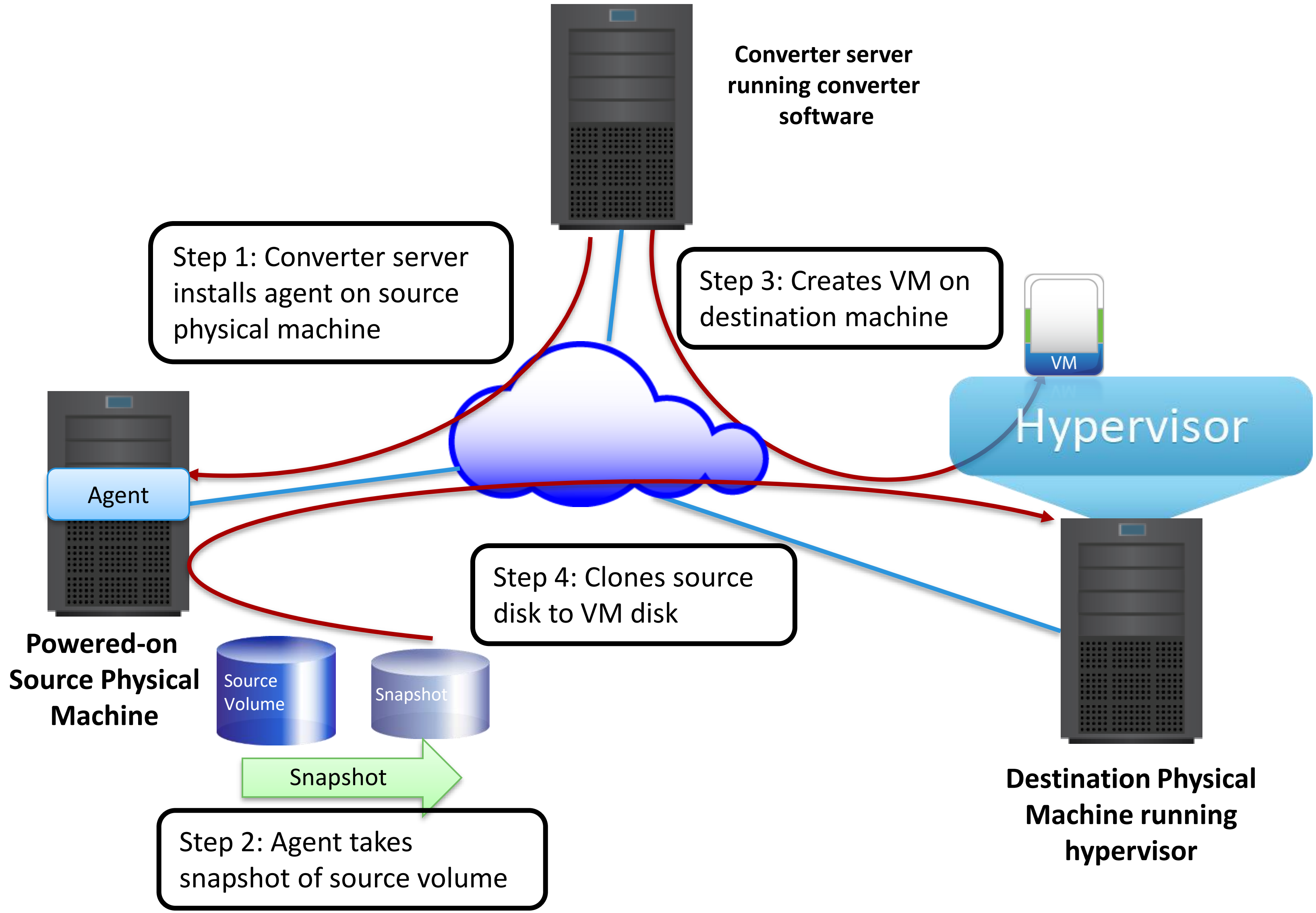
Conversion Options

- Hot conversion
  - Occurs while physical machine is running
  - Performs synchronization
    - Copies blocks that were changed during the initial cloning period
  - Performs power off at source and power on at target virtual machine (VM)
  - Changes IP address and machine name of the selected machine, if both machines must co-exist on the same network
  - Because processes continue to run on the source machine during conversion, the resulting virtual machine is not an exact copy of the source physical machine.
    - After conversion is complete, the destination virtual machine is synchronized with the source machine

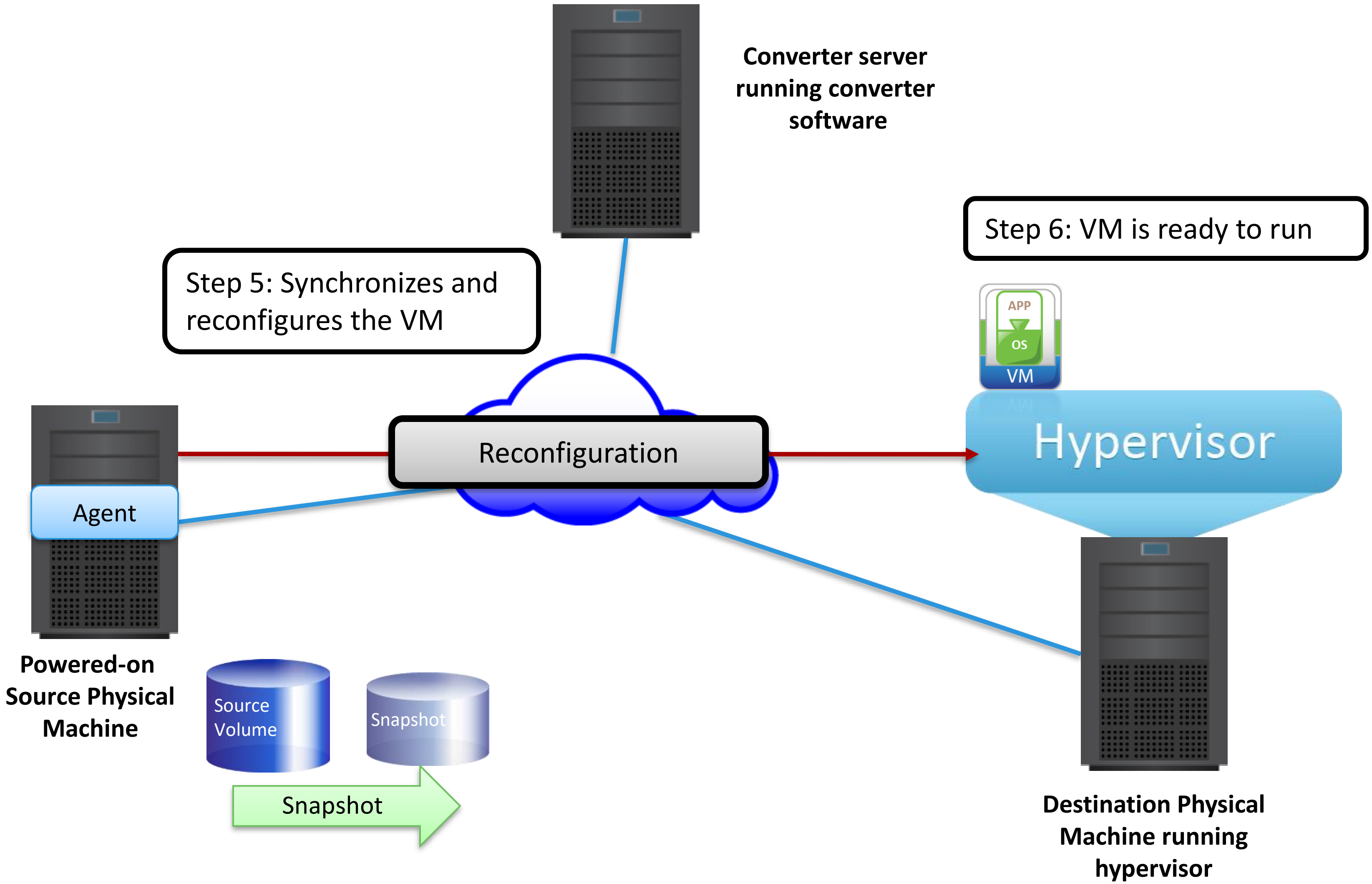
Conversion Options

- Cold conversion (called offline conversion)
  - Occurs while physical machine is not running OS and application
  - Boots the physical machine using converter boot CD
  - Creates consistent copy of the physical machine
    - because no changes occur on the source machine during the conversion

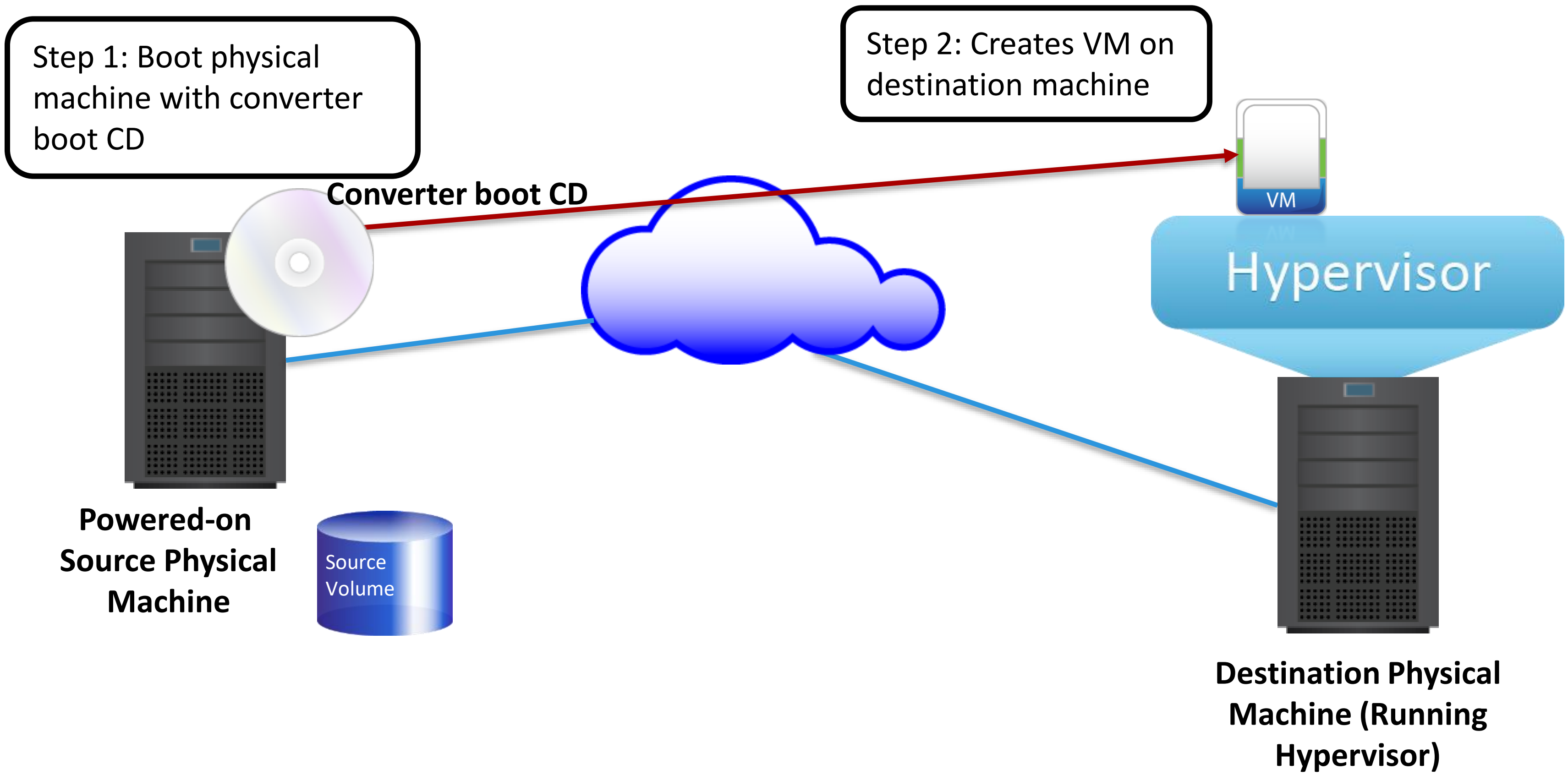
Hot Conversion Process



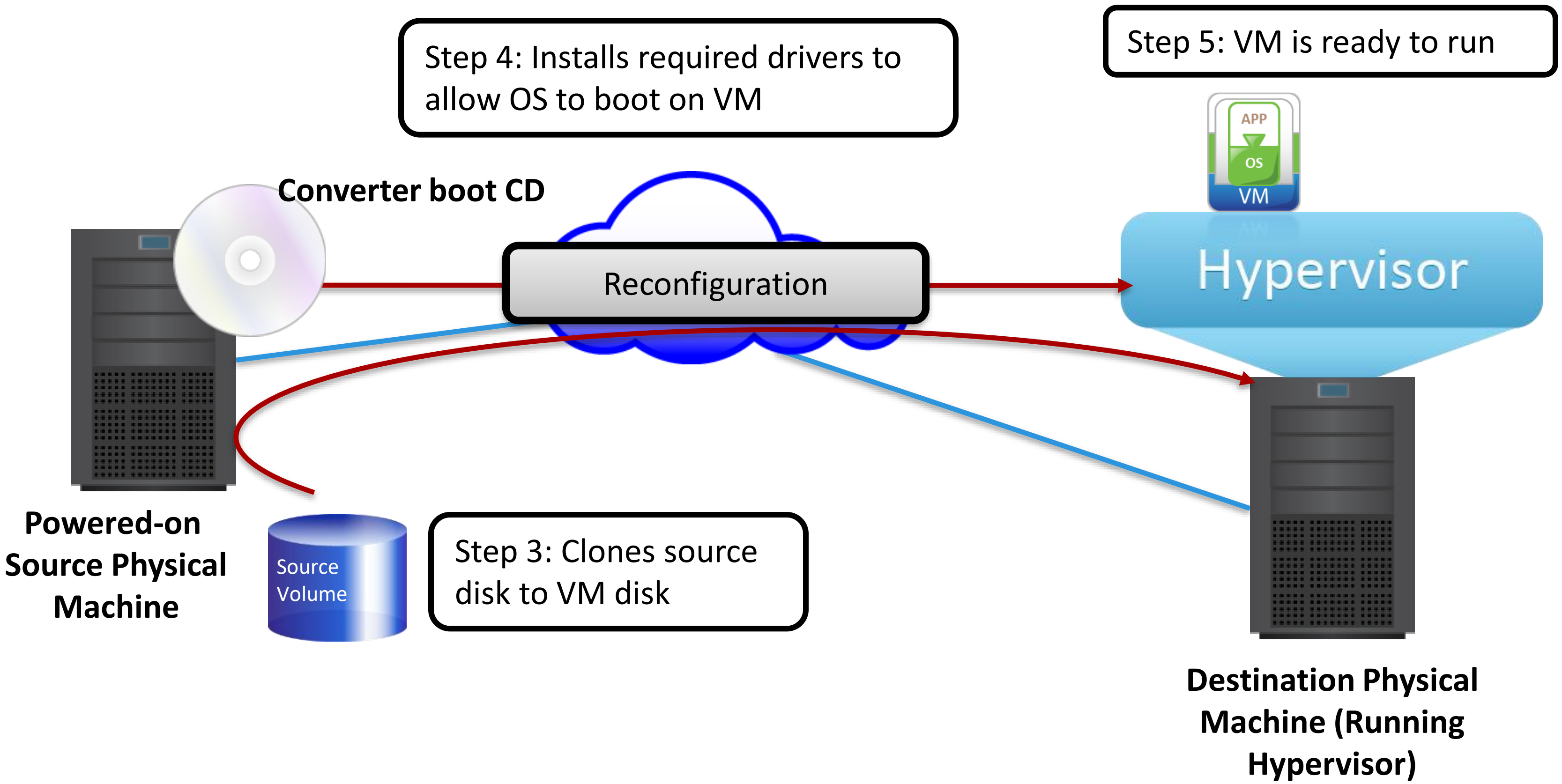
Hot Conversion Process (contd.)



Cold Conversion Process



Cold Conversion Process (contd.)



P2V Conversion: Considerations

- Some hardware-dependent drivers and mapped drive letters might not be preserved
- Source machine configuration remains unchanged such as:
  - Operating system (OS) configuration, such as computer name, security ID, user accounts, profiles, and preferences
  - Applications and data files
  - Volume serial number for each disk partition
- Source and target machines will have the same identities
  - Running them on the same network might result in conflicts
- Applications that depend on characteristics of the hardware may not work

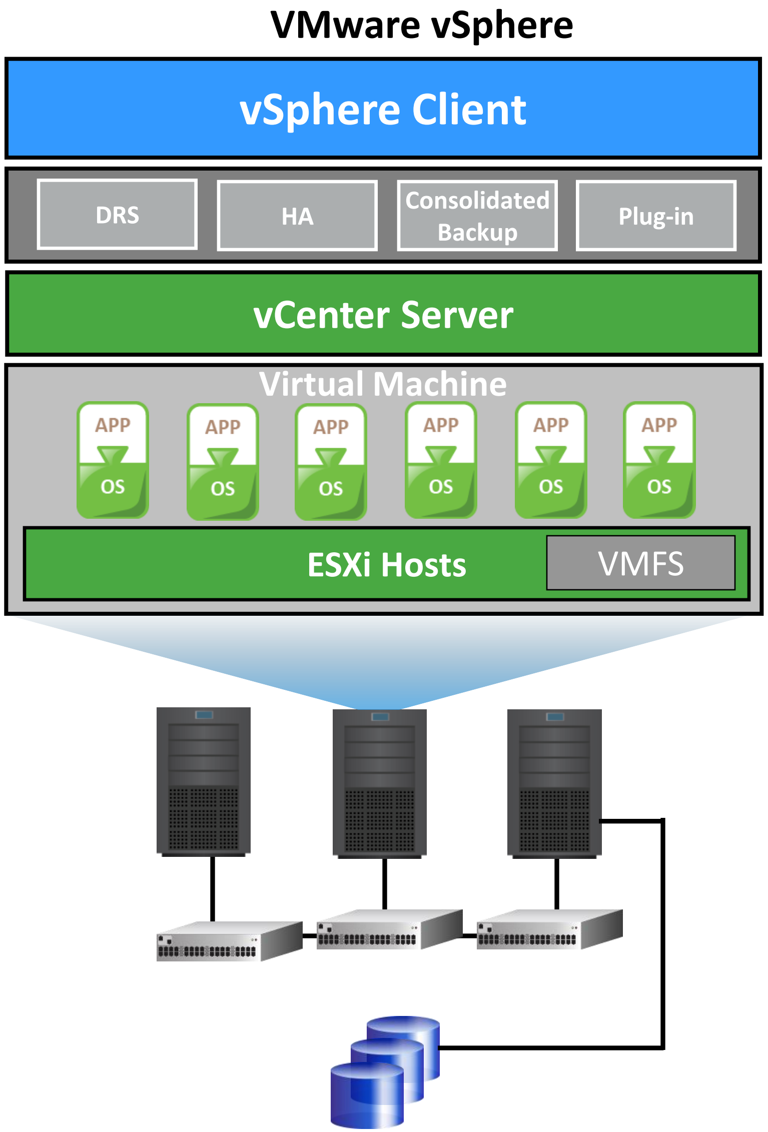
Module 3: Virtualized Data Center – Compute

Concept in Practice

- VMware vSphere
- VMware vCenter Converter

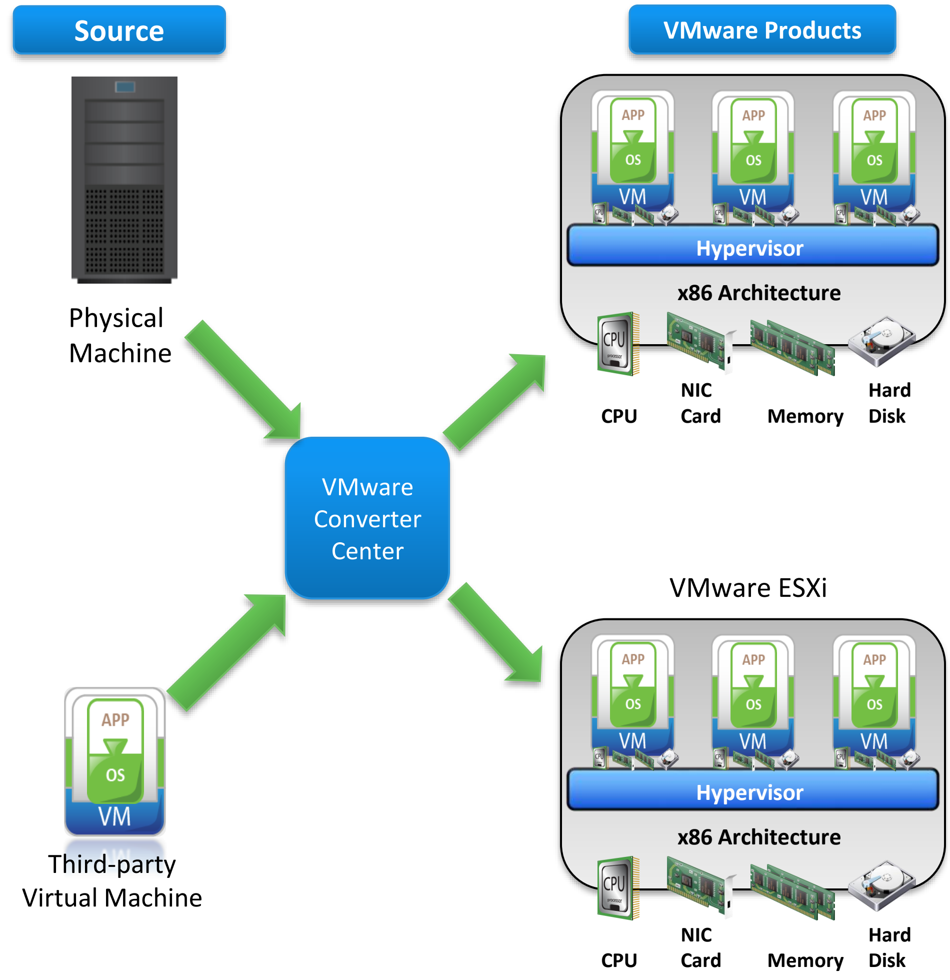
Concept in Practice: VMware vSphere

- An infrastructure virtualization suite that provides:
  - Virtualization
  - Resources management and optimization
  - High availability
  - Operational automation
- vSphere consists of the following key components:
  - VMware ESXi
  - VMware vCenter Server
  - VMware vCenter Client
  - VMware vStorage VMFS



Concept in Practice: VMware vCenter Converter

- Converts physical machines to virtual machines (VMs)
- Supports conversion of VM created third party software to VMware VM



Concept in Practice: VMware vCenter Converter

- VMware vCenter Converter is a highly robust and scalable enterprise-class migration tool that automates the process of creating virtual machines from physical machines and other virtual machine formats (for example virtual machines created using Microsoft Hyper-V).

-VMware vCenter Converter lets users quickly, easily, and affordably convert Microsoft Windows and Linux-based physical machines to VMware virtual machines.

- It also converts virtual machines between VMware platforms.

Module 3: Summary

Key points covered in this module:

- Drivers of compute virtualization
- Challenges of virtualizing x86 hardware
- Techniques to virtualize x86 hardware
- Components of a virtual machine (VM)
- Resource optimization techniques
- Conversion of physical machine to VM

Check Your Knowledge

1. Explain two types of hypervisor.
2. What are the challenges of x86 hardware virtualization?
3. Explain the function of each virtual machine (VM) file.
4. Explain the three memory optimization techniques.
5. What are the two options to convert physical to VM?

Check Your Knowledge

1. Explain two types of hypervisor.

   Ans. Bare-Metal and Hosted hypervisor

2. What are the challenges of x86 hardware virtualization?

- Ans. x86 based operating systems (OSs) need to have direct access to the hardware and must execute its privileged instructions in Ring 0.

- Virtualizing x86 architecture requires placing a virtualization layer below the operating system.
- The difficulty in capturing and translating these privileged instructions at runtime was the challenge that originally made x86 architecture virtualization look impossible.
- The three techniques used for handling privileged instructions to virtualize the CPU on x86 architecture are: Full virtualization, Paravirtualization, and Hardware assisted virtualization.

EMC² where information lives

Check Your Knowledge

Explain the function of each virtual machine (VM) file.

Ans. logical compute system like a physical machine that runs an operating system and applications

These virtual machines have the following virtual hardware components: vCPU, vRAM, virtual disk, vNIC, virtual DVD/CD-ROM, virtual floppy drive, virtual SCSI controller, and virtual USB controller.

Check Your Knowledge

Explain the three memory optimization techniques

- Transparent Page Sharing
- Memory Ballooning
- Memory Swapping

Check Your Knowledge

- What are the two options to convert physical to VM?
- - Hot and cold conversion

MODULE 3 QUIZ