BITS Pilani presentation

D. Powar
Lecturer,
BITS-Pilani, Hyderabad Campus

BITS Pilani
Hyderabad Campus

innovate achieve lead

BITS Pilani
Hyderabad Campus

CS ZG527
Cloud Computing
Lecture -03

Objectives

- Introduction to Virtualization
- Uses of Virtualization
- Types of Virtualization
  - Examples
- x86 Hardware Virtualization
- Demerits of Virtualization
- Who manages the resources for the SaaS, PaaS and IaaS models

Virtualization (abstraction layer)

Servers

Desktop PC's

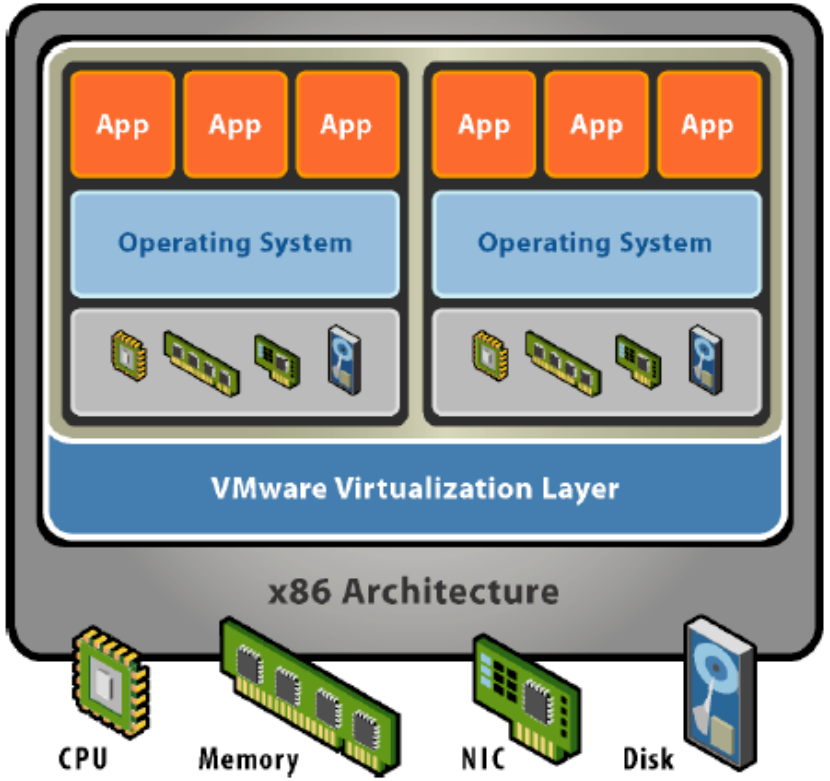
Storage

Application

Networks

Virtualization Technology

- **Virtualization** is the creation of a virtual (rather than actual) version of something, such as a hardware platform, operating system (OS), storage device, or network resources. (Wiki)



Source: white paper on "Understanding Full Virtualization, Paravirtualization, and Hardware Assist Virtualization"

Virtualization Technology (contd..)

- Virtualization is not new, but it has recently become more popular
- In the late 1960s, mainframe computers were virtualized and rented out to smaller companies that could not afford an actual mainframe as a cost effective solution
- In the 1980s and 1990s, with the advent of the personal computer (PC) and the drop in hardware costs, virtualization lost steam
- Recently however, virtualization has grown in popularity again, desktop and server virtualization are common to see in organizations

Virtualization Technology (contd..)

- Virtualization has its beginnings in multiprocessing; one computer, one operating system, and many applications made way for one computer, many operating systems, and many applications
- Virtualization refers to the partitioning the resources of a physical system (such as computing, storage, network and memory) into multiple virtual resources.
- Key enabling technology of cloud computing that allow pooling of resources.
- In cloud computing, resources are pooled to serve multiple users using multi-tenancy.

Virtualization Technology (contd..)

- In computing, virtualization is a broad term that refers to the abstraction of computer resources

- It is "a technique for hiding the physical characteristics of computing resources from the way in which other systems, applications, or end users interact with those resources. This includes making a single physical resource (such as a server, an operating system, an application, or storage device) appear to function as multiple logical resources; or it can include making multiple physical resources (such as storage devices or servers) appear as a single logical resource."

Virtualization Categories

There are broadly two main categories of software virtualization

- Process Virtualization
- System Virtualization

Process Virtualization

- The virtualization software runs above the OS and hardware combination
- Sun's JVM, MS .Net, Rosetta, Pin, etc.

JVM:

- Interprets, then compiles "byte code" files
- "Write once, run anywhere"
- extensive libraries – extend OS API as Java standard

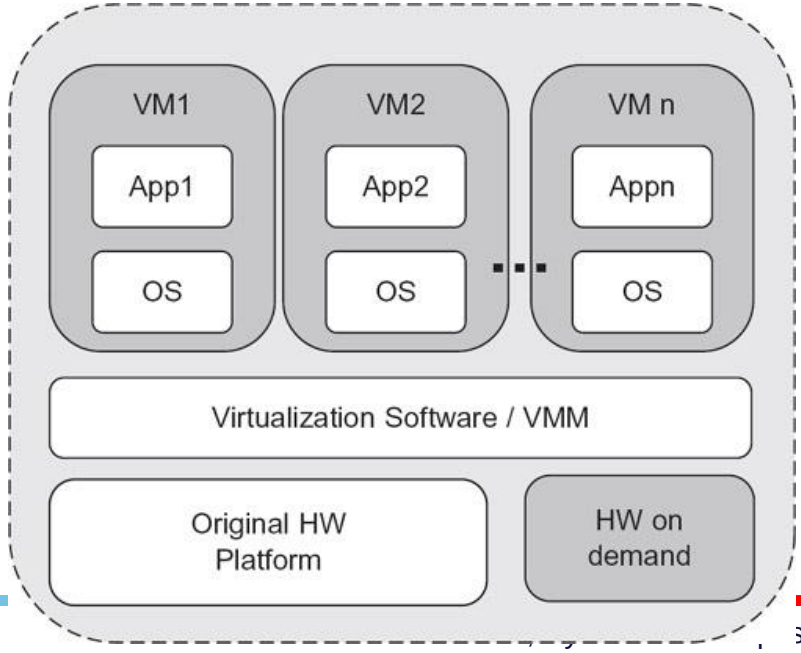
Rosetta:

- Translates PowerPC binaries "on-the-fly" to x86

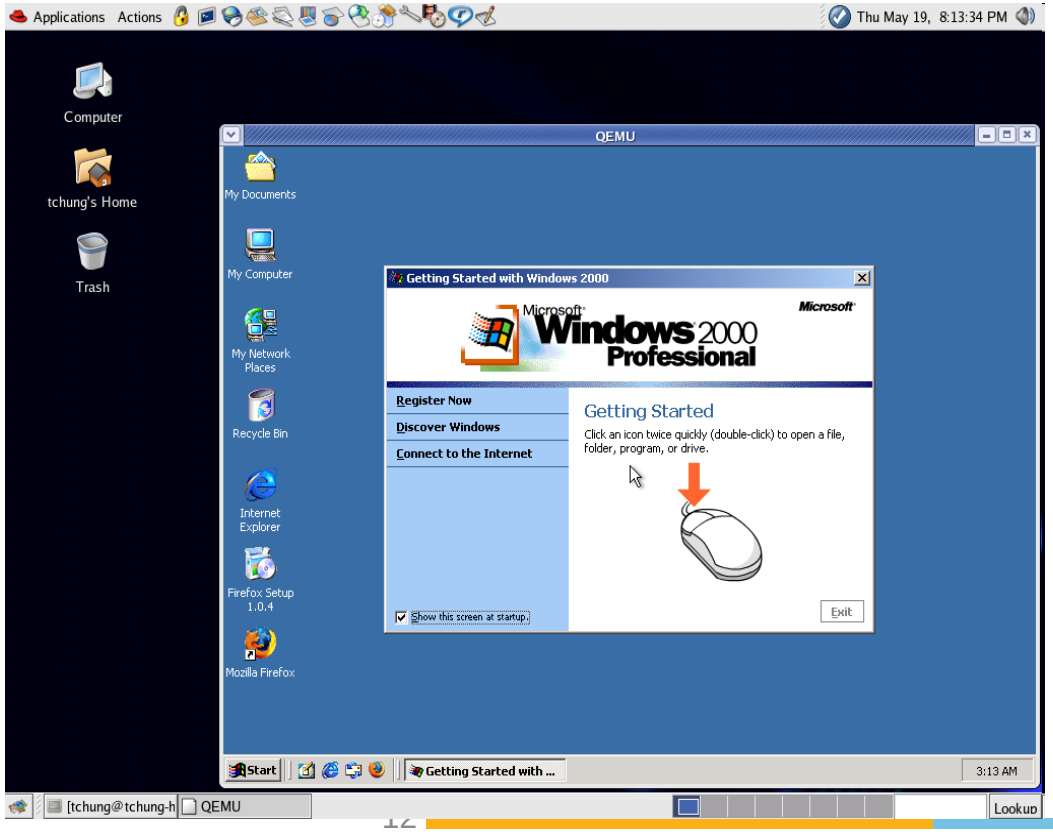
System Virtualization

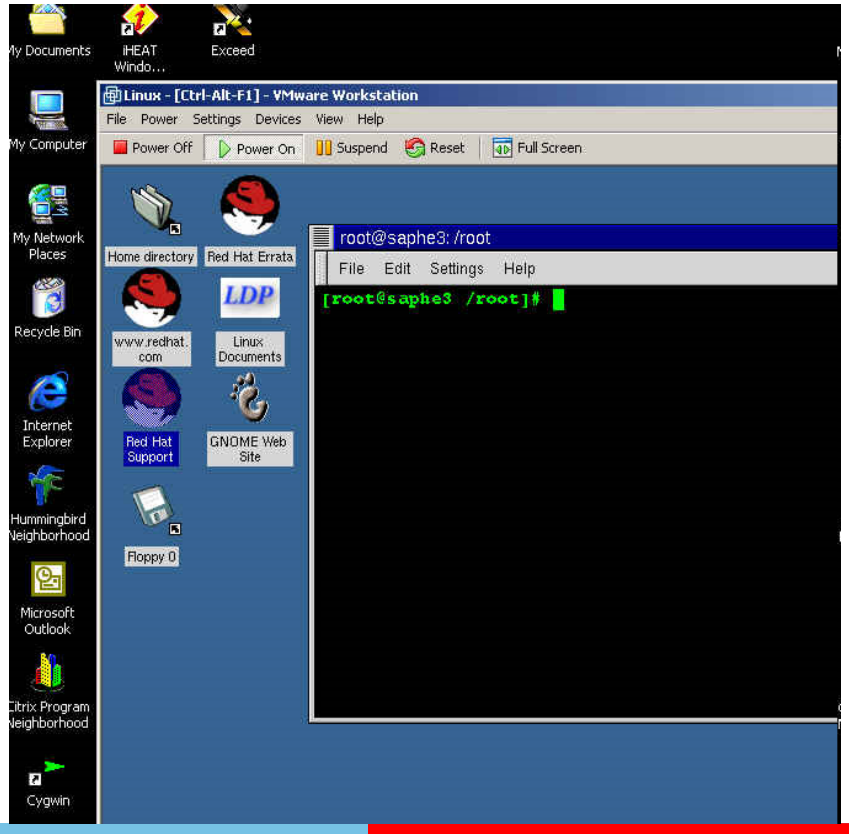
- Run an operating system under the control of a layer of software
- The Virtualization software runs in between the host hardware and the guest software.
- Goal is to provide virtualized hardware resources
- Enables elasticity in hardware without effecting the guest software.
- Ex: VMware, Xen, KVM

Note: In a legacy OS, kernel is the entity which controls hardware, and it has the highest privileges. To virtualize a VM, OS should be tricked to think that it is controlling hardware, while actually another entity named Virtual Machine Monitor (VMM) or hypervisor is controlling it



- Virtualization is way to run **multiple operating systems** and **user applications** on the same hardware
  - E.g., run both Windows and Linux on the same laptop
- How is it different from **dual-boot**?
  - Both OSes run simultaneously
- The OSes are completely **isolated** from each other



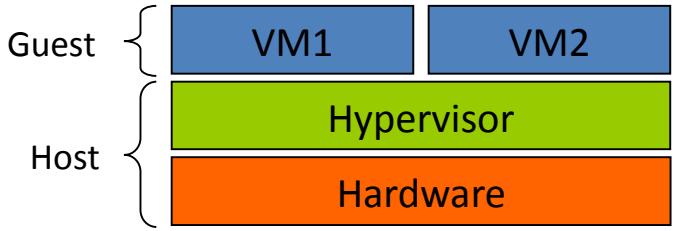


Two types of hypervisors

Definitions

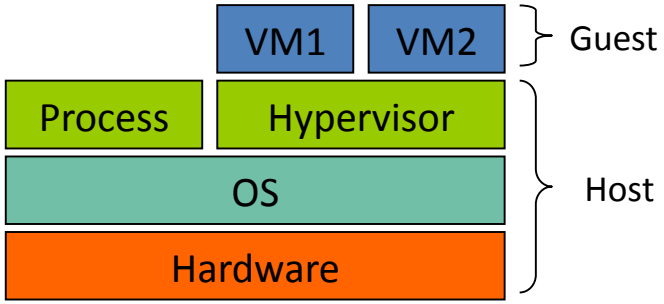
- **Hypervisor** (or **VMM** – Virtual Machine Monitor) is a software layer that allows several **virtual machines** to **run** on a **physical machine <**software responsible for system virtualization**>**
- The physical OS and hardware are called the **Host**
- The virtual machine OS and applications are called the **Guest**

Type 1 (bare-metal)
OR Native Hypervisor



VMware ESX, Hyper V

Type 2 (hosted) Hypervisor



VMware Workstation, Microsoft Virtual PC, Sun VirtualBox, QEMU

Techniques for Hypervisors

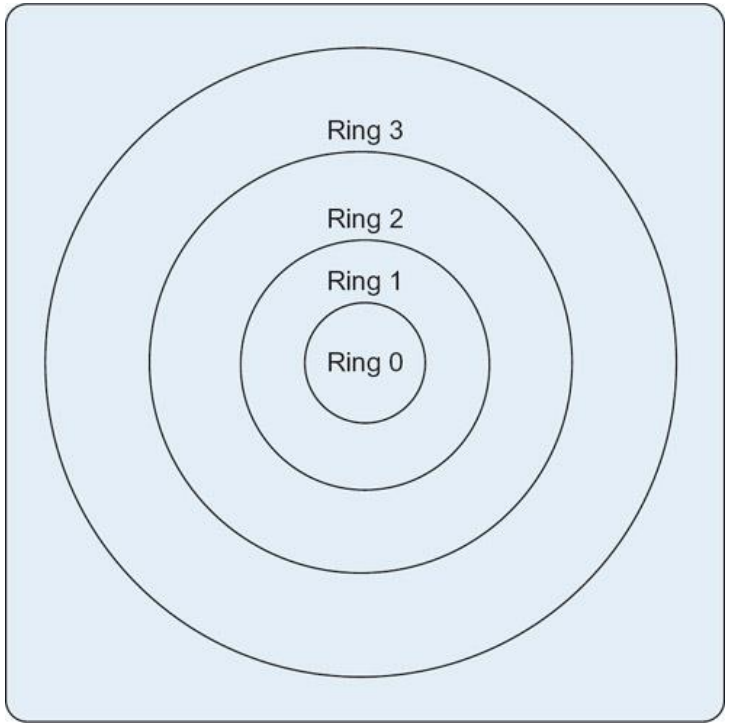
- There are different techniques used for hypervisor-based virtualization

Trap and emulate virtualization: is a basic technique used from the days of the earliest hypervisors.

- Both the hypervisors operate in a similar manner
- The guests continue execution until they try to access a shared physical resource of the hardware (such as an I/O device), or an interrupt is received.
- When this happens, the hypervisor regains control and mediates access to the hardware, or handles the interrupt

The **privilege level** or **protection ring**

- Instructions that modify the physical hardware configuration are permitted at the highest level
- Programs executing in Ring 0 have the highest privileges, and are allowed to execute any instructions or access any physical resources.
- Guests are typically made to execute in ring 3
- Virtualizing the x86 architecture requires placing a virtualization layer under the operating system (which expects to be in the most privileged Ring 0) to create and manage the virtual machines that deliver shared resources.



Protection rings in the Intel x86 architecture

Limitations of Trap and Emulate Virtualization

- Amount of performance overhead
- Not all architectures are suitable for implementing (Trap and emulate may fail)

Software Extensions to Trap and Emulate Virtualization

- Binary translation
- Paravirtualization

Binary translation:

- Hypervisor includes a **binary translator** which replaces the sensitive instructions by equivalent non-sensitive instructions at run-time.
- The technique is similar to just-in-time translation in JVMs

Paravirtualization:

- The guest OS is modified so that instead of working with hardware directly, the OS uses hypervisor calls to VMM.
- OS is aware that it is going to be run in a virtualized environment.
- OS is modified to make kernel runs in ring 1, and user space applications runs in ring 3.

Disadv: Not hypervisor-independent; the modifications have to be carried out for every hypervisor under which the guest could run. So, paravirtualization requires rewriting of the guest OS

x86 Hardware Virtualization

- Virtualizing the x86 architecture requires placing a virtualization layer under the operating system (which expects to be in the most privileged Ring 0) to create and manage the virtual machines that deliver shared resources.
- Some sensitive instructions can't effectively be virtualized as they have different semantics when they are not executed in Ring 0.
- The difficulty in trapping and translating these sensitive and privileged instruction at runtime was the challenge that originally made x86 architecture virtualization impossible.
- VMware resolved the challenge in 1998, developing binary translation techniques that allow the VMM to run in Ring 0 for isolation and performance

3 techniques exists for handling sensitive and privileged instructions to virtualize the CPU on the x86 architecture

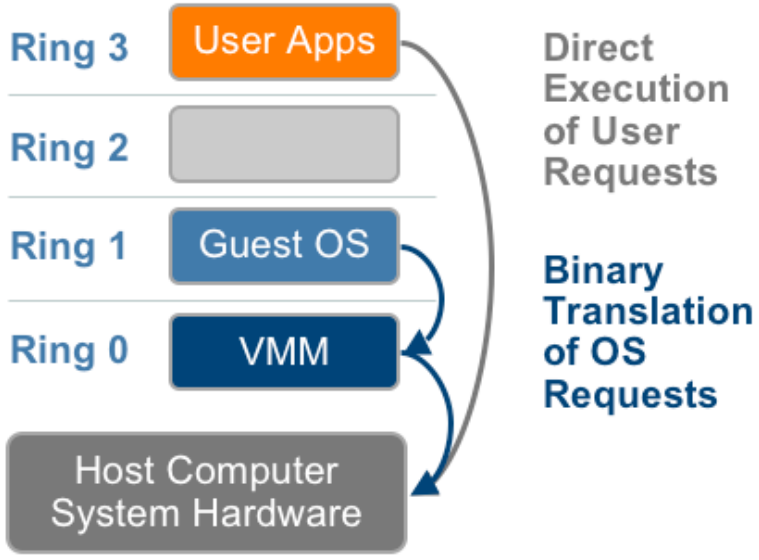
- Full virtualization with binary translation
- OS assisted virtualization or paravirtualization
- Hardware assisted virtualization (first generation)

Source: white paper on “Understanding Full Virtualization, Paravirtualization, and Hardware Assisted virtualization”

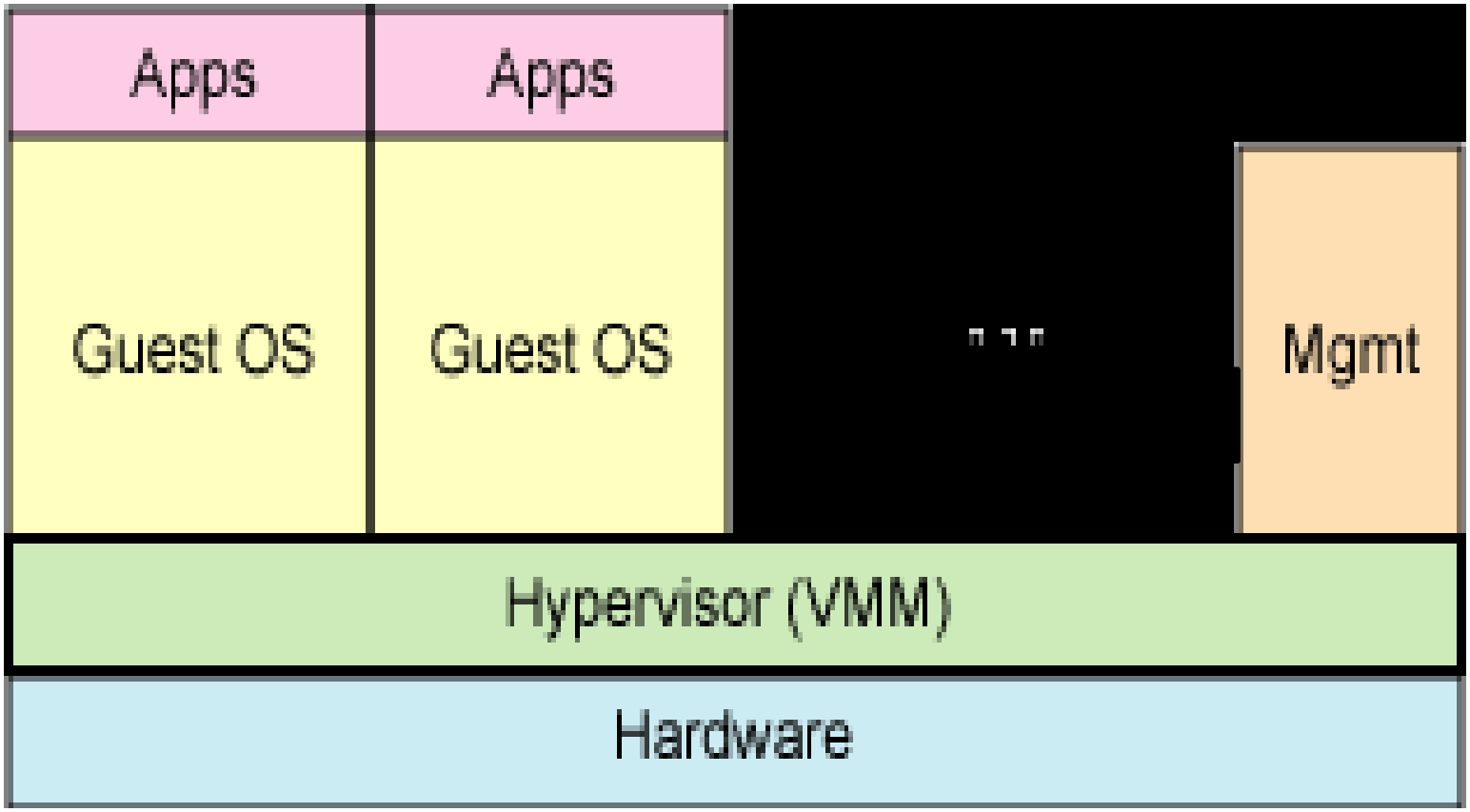
Full virtualization

- Combination of binary translation and direct execution techniques
- Translates kernel code to replace non-virtualizable instructions with new sequences of instructions that have the intended effect on the virtual hardware.
- This combination of binary translation and direct execution provides Full Virtualization as the guest OS is completely decoupled from the underlying hardware by the virtualization layer

- The guest OS is not aware it is being virtualized and requires no modification
- Requires no hardware assist or operating system assist to virtualize sensitive and privileged instructions
- VMM provides each VM with all the services of the physical system, including a virtual BIOS, virtual devices and virtualized memory management



Full virtualization



- Examples include Virtual Iron, VMware Workstation, Microsoft Virtual Server

Para virtualization

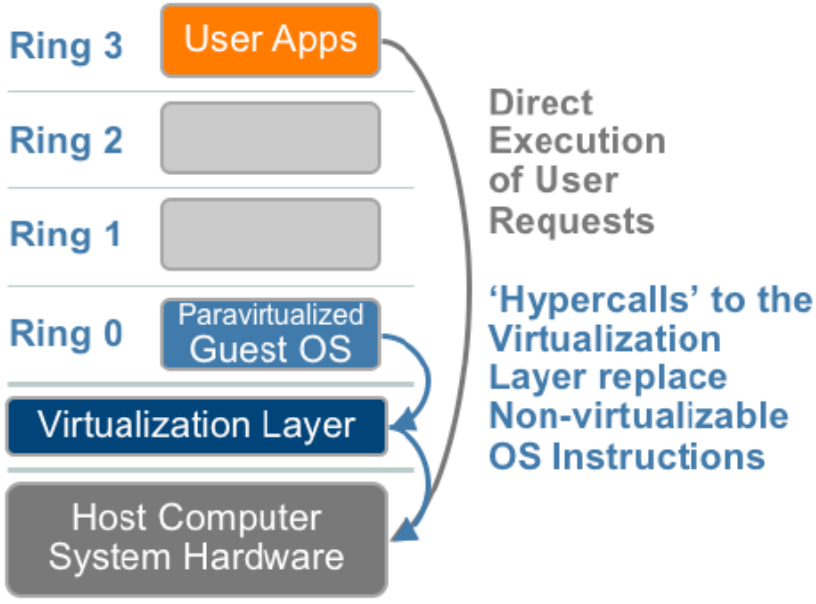
Para virtualization:

- Para virtualization is different from full virtualization, where the unmodified OS does not know it is virtualized and sensitive OS calls are trapped using binary translation
- Involves modifying the OS kernel to replace non-virtualizable instructions with hypercalls that communicate directly with the virtualization layer
- The hypervisor also provides hypercall interfaces for other critical kernel operations such as memory management, interrupt handling etc.
- What is the % modification (Guest OS)

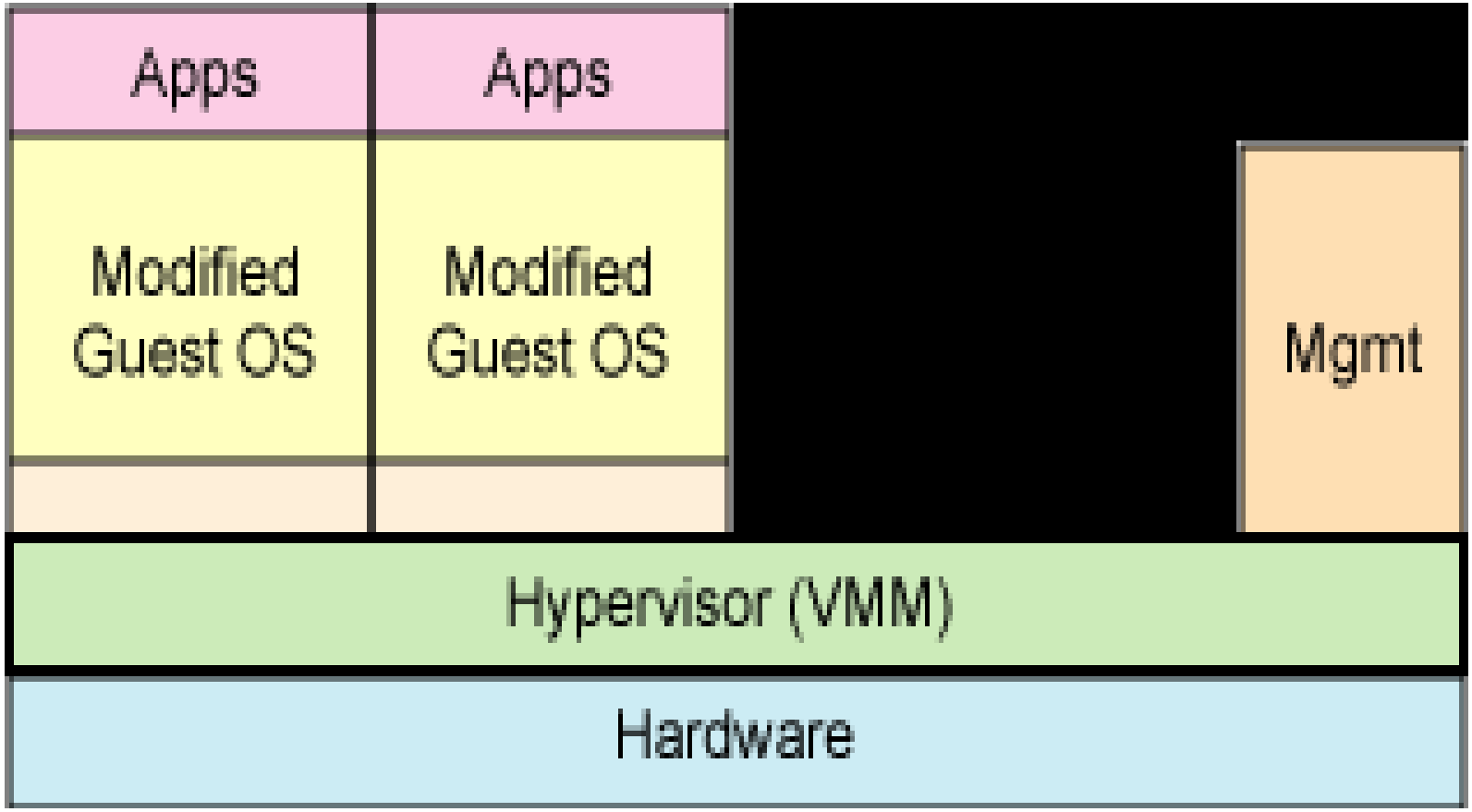
Para virtualization (contd..)

- As para virtualization cannot support unmodified operating systems (e.g. Windows 2000/XP), its compatibility and portability is poor
- The performance advantage of para-virtualization over full virtualization can vary greatly depending on the workload

Example: The open source Xen project - virtualizes the processor and memory using a modified Linux kernel and virtualizes the I/O using custom guest OS device drivers

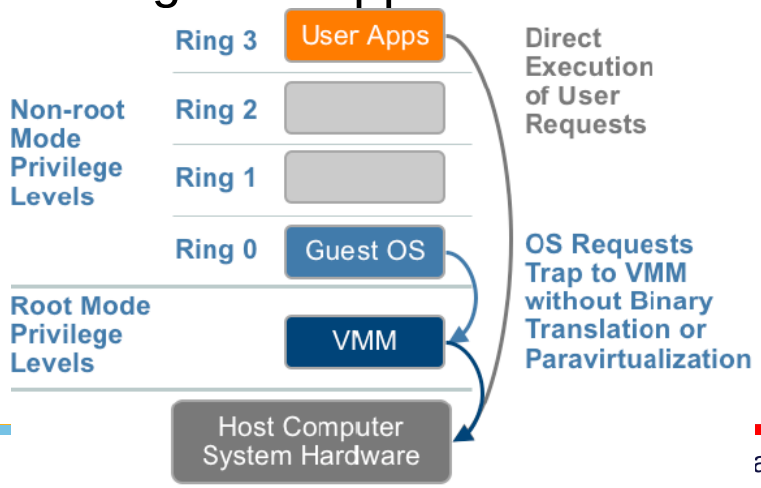


Para virtualization



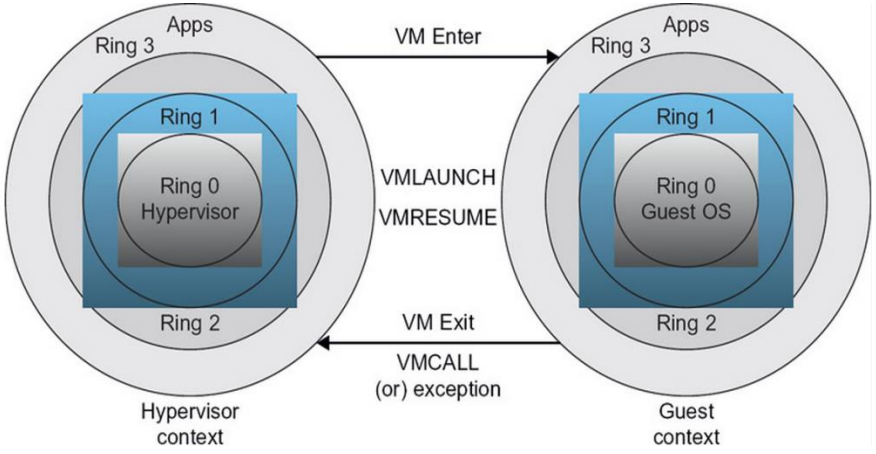
Hardware assisted virtualization

- Intel's Virtualization Technology (VT-x) (e.g. Intel Xeon) and AMD's AMD-V both target privileged instructions with a new CPU execution mode feature that allows the VMM to run in a new root mode below ring 0, also referred to as Ring 0P (for privileged root mode) while the Guest OS runs in Ring 0D (for de-privileged non-root mode).
- Privileged and sensitive calls are set to automatically trap to the hypervisor and handled by hardware, removing the need for either binary translation or para-virtualization
- VMware only takes advantage of these first generation hardware features in limited cases such as for 64-bit guest support on Intel processors.
- Also called as native virtualization



Hardware assist virtualization (contd..)

- Introduction of virtualization technologies in CPUs (Intel VT-x, AMD-sv) , virtualizing an unmodified OS has become possible
- Two security modes in CPU architecture
  - Root mode
  - Non-root mode



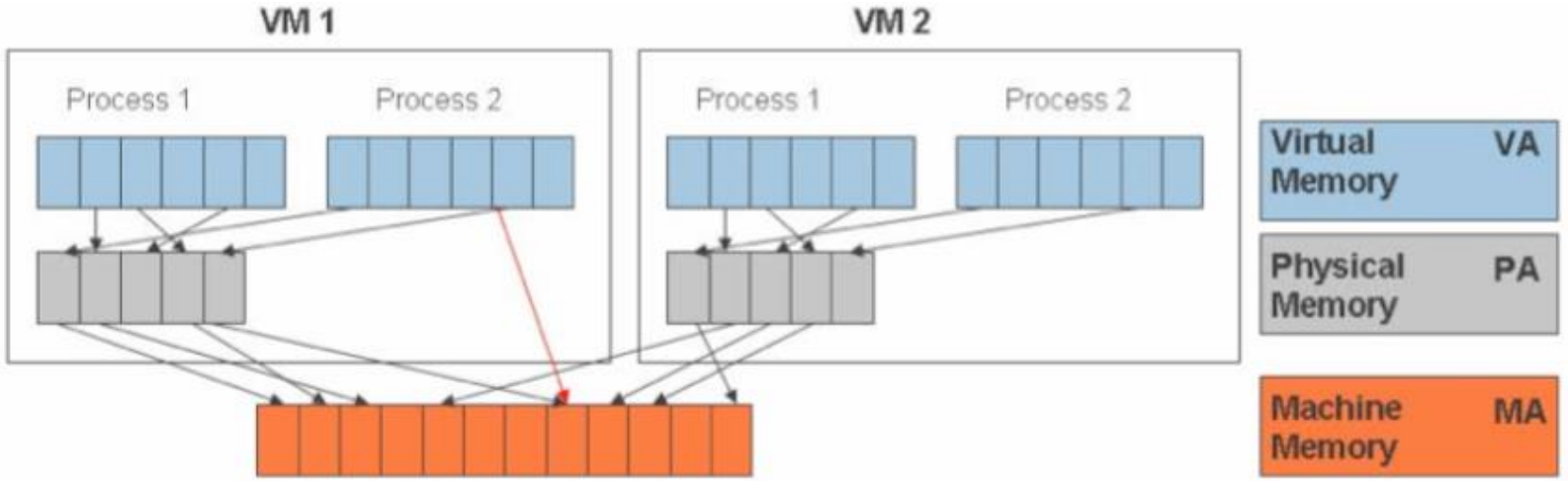
- The root mode allows VMM to control the guest VM kernel.
- Executing instructions which need higher privileges like accessing hardware causes CPU to generate an exception interrupt, which triggers VMM to take control. VMM then decides how to handle the situation and makes guest OS believe it is controlling hardware.
- The transition from the non-root mode to the root mode is called VM EXIT, and the reverse is called VM Entry

Virtualization (contd..)

- Platform virtualization
- Storage virtualization
- Network virtualization
- Memory virtualization
- Application virtualization
- Desktop virtualization
- Data Center Virtualization

Memory Virtualization

- This involves sharing the physical system memory and dynamically allocating it to virtual machines
- Very similar to the virtual memory support provided by modern operating systems
- Virtualization of the MMU is required
- The guest OS continues to control the mapping of virtual addresses to the guest memory physical addresses, but the guest OS cannot have direct access to the actual machine memory.
- The VMM is responsible for mapping guest physical memory to the actual machine memory
- VMM uses TLB hardware to map the virtual memory directly to the machine memory to avoid the two levels of translation on every access (red arrow)

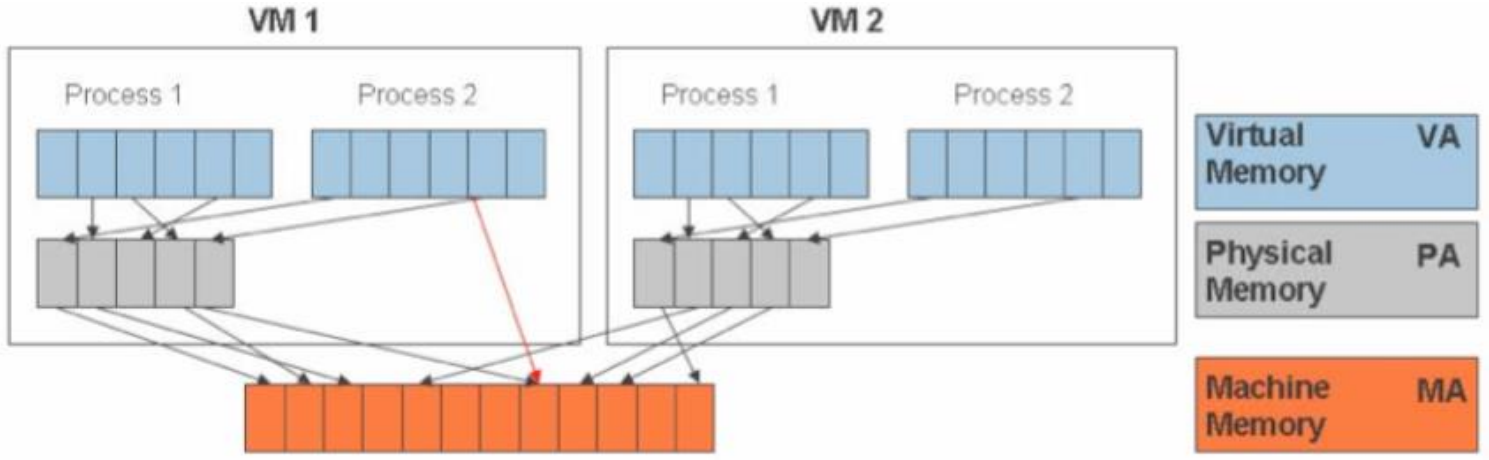


Memory Virtualization

Guest virtual addr

Guest Physical addr

Physical addr

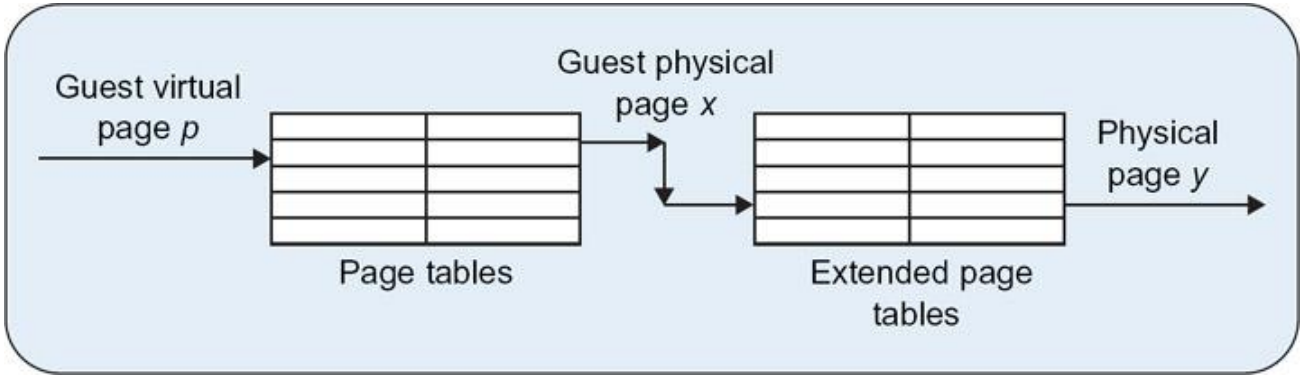


Auxiliary mapping — Guest Physical addr → Physical addr

Shadow page table — Guest virtual addr → Physical addr

Memory Virtualization (contd..)

- Trapping the guest when it tries to modify the page tables is a source of performance overhead
- Maintenance of shadow page tables is complex and prone to error
- Purging the TLB each time a new guest executes, since translations valid for one guest would not be valid for another guest
- To overcome these performance overheads, there are two hardware assists for memory virtualization in Intel x86 processors.
  - **Extended Page Tables (EPT)**
  - **Virtual Processor ID (VPID)**



Storage Virtualization

- Storage Virtualization refers to the process of completely abstracting logical storage from physical storage, and is commonly used in SANs.
- The physical storage resources are aggregated into storage pools, from which the logical storage is created. Multiple independent storage devices, which may be scattered over a network, appear to the user as a single, location-independent, monolithic storage device, which can be managed centrally

- RAID and volume managers combine many disks into one large logical disk.

Uses of virtualization

Server consolidation

- Run a **web server** and a **mail server** on the **same physical server**

Easier development

- Develop critical **operating system components** (file system, disk driver) without affecting **computer stability**

Testing

- Testing a network product (e.g., a firewall) may require **tens of computers**
- Try testing thoroughly a product at each pre-release milestone… and have a straight face when your boss shows you the **electricity bill**

Cloud computing

- The modern buzz-word
- Amazon sells computing power
- You pay for e.g., 2 CPU cores for 3 hours plus 10GB of network traffic

Virtualization demerits?

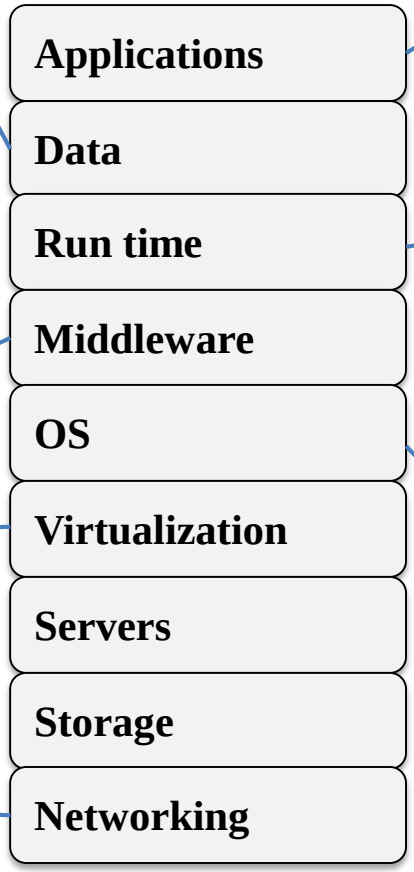
- The wide spread use of virtualization in data centers has created many concerns on the security front
- Collocating many virtual machines on a single physical computer changes the security paradigm
- Attacker my get control of the VMM (Hypervisor)

The Cloud stack

Data Management:
New generation cloud specific databases and management systems
E.g., Hbase, Cassandra, Hive, Pig etc.

Management platforms that enable:
Resource Management, Monitoring, Provisioning, Identity Management and Security

Virtualization layer:
Amazon EC2 is based on the Xen virtualization platform

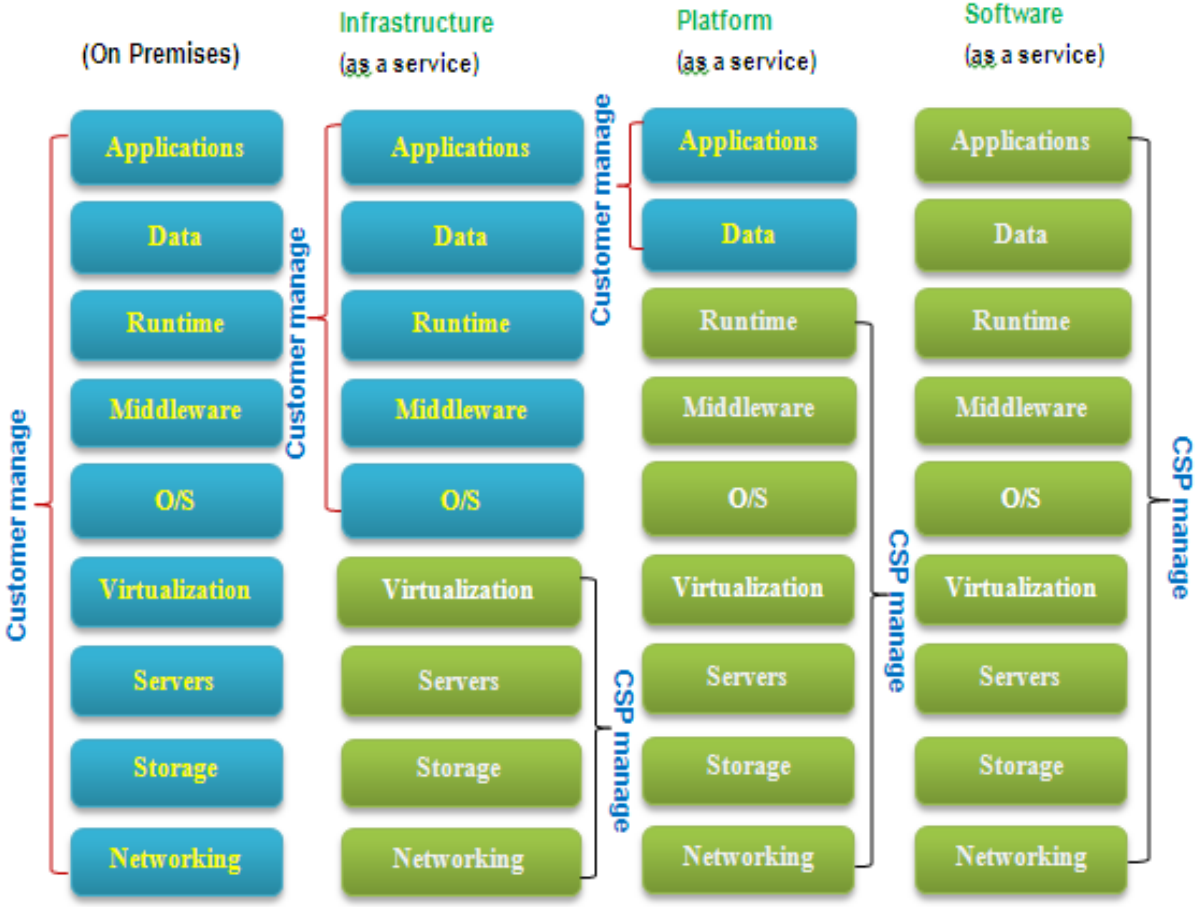


Cloud applications can range from Web applications to scientific computational jobs

Runtime platforms to support cloud programming models
E.g., MPI, MapReduce, Pregel etc.

OS Packaged with libraries and software for quick deployment and provisioning
E.g., Amazon Machine Images (AMI) contain OS as well as required software packages as a "snapshot" for instant deployment

Who manages???


(On Premises)
Infrastructure
(as a service)
Platform
(as a service)
Software
(as a service)
Applications
Data
Runtime
Middleware
O/S
Virtualization
Servers
Storage
Networking
Customer manage
Customer manage
Customer manage
CSP manage
CSP manage
CSP manage

Summary:

- Introduction to Virtualization
- Uses of Virtualization
- Types of Virtualization
  - Examples
- x86 Hardware Virtualization
- Demerits of Virtualization
- Who manages the resources for the SaaS, PaaS and IaaS models