CHENNAI INSTITUTE OF TECHNOLOGY
(Autonomous)
(Affiliated to Anna University, Approved by AICTE, Accredited by NAAC & NBA)
Sarathy Nagar, Kundrathur, Chennai – 600069, India.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CCS335-CLOUD COMPUTING

LECTURE NOTES UNIT 2

UNIT II VIRTUALIZATION BASICS

UNIT II VIRTUALIZATION BASICS

Virtual Machine Basics – Taxonomy of Virtual Machines – Hypervisor – Key Concepts – Virtualization structure – Implementation levels of virtualization – Virtualization Types: Full Virtualization – ParaVirtualization – Hardware Virtualization – Virtualization of CPU, Memory and I/O devices

1. VIRTUALIZATION STRUCTURE:

- **Definition:** Virtualization is a computer architecture technology where multiple virtual machines are multiplexed in the same hardware machine.

- **Classes of VM architecture:-**
  1. Hypervisor architecture
  2. Para virtualization
  3. Host based virtualization
- **Hypervisor and XEN architecture:**

  Directly between physical hardware and its OS and provides hyper calls.

  Architecture: Two types

  □ microkernel architecture: called Hyper-Functions for basic.

  □ monolithic hypervisor architecture: VMware.

- **XEN architecture:**

  □ Processing:

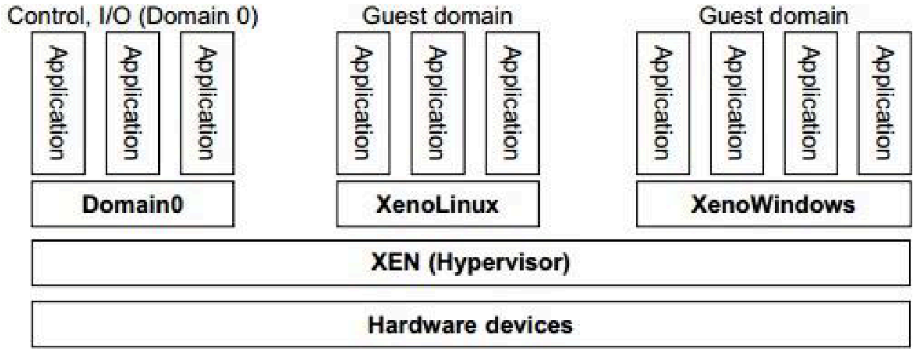
  ▪ It is an open source hypervisor, it is a microkernel hypervisor that provides a virtual environment located between Hardware and OS.

  ▪ The Guest OS, control sharing is called Domain 0, other are Domain U

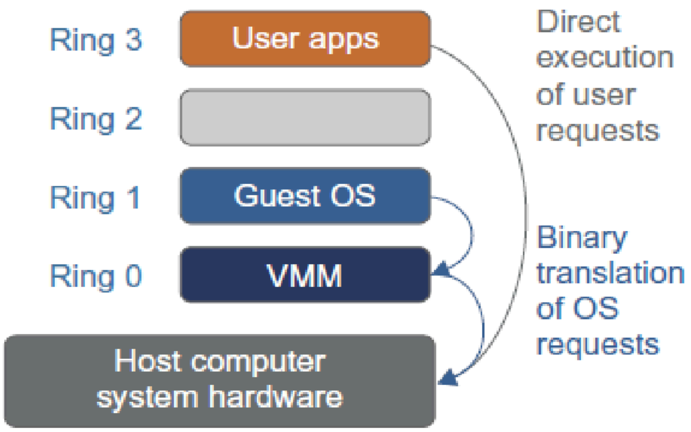
  ▪ When Xen boots without any file system, Domain 0 accesses hardware directly and manages devices.
- **Binary Translation with full virtualization:**
  - Non critical instructions run on the hardware directly, while critical instructions are discovered and replaced with traps into the VMM.
  - Full virtualization does not need to modify the host OS.

- In a host based system both host OS and guest OS are used.

- **Para virtualization:**
  - para virtualization needs to modify the guest OS, to replace non-virtualizable instructions with hyper calls for the hypervisor (or) VMM to carry out the virtualization.
- **para virtualization architecture:**
  - when X86 processor is virtualized, a virtualization layer is inserted between the hardware and OS.
  - para virtualization replaces non virtualizable instructions with hyper calls that communicate directly with hypervisor (or) VMM.
- **Host-Based virtualization**
  - ➔ First, the user can install this VM architecture without modifying the host OS virtualization software can rely on the host OS to provide drivers and other low level services.
  - ➔ Second, compared hypervisor/VMM architecture the performance of the host based architecture may also be slow.
- **Kernel based VM:**
  - KVM is a hardware assisted para virtualization tool, part of Linux version 2.6.20 kernel.
  - para virtualization with compiler support:
    - ➔ The guest OS kernel is modified to replace the privileged and sensitive instructions with hyper calls to the VMM.

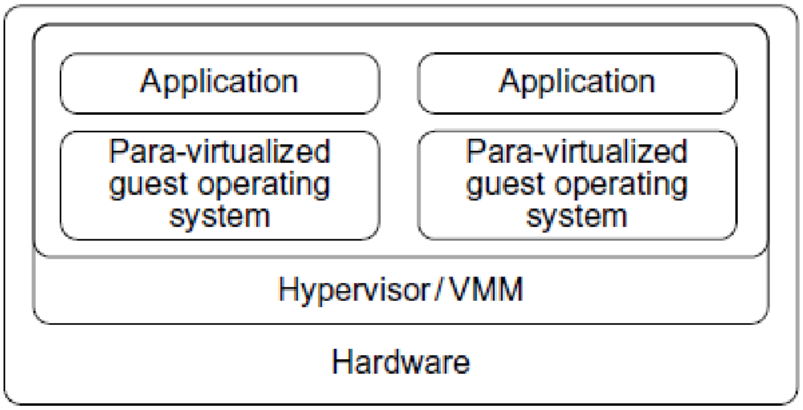


Host Based Virtualization


Ring 3
User apps
Ring 2
Ring 1
Guest OS
Ring 0
VMM
Host computer system hardware
Direct execution of user requests
Binary translation of OS requests

Indirect execution of complex instructions via binary translation of guest OS requests using the VMM plus direct execution of simple instructions on the same host.

Para Virtualization Architecture


Application
Application
Para-virtualized guest operating system
Para-virtualized guest operating system
Hypervisor / VMM
Hardware

Para-virtualized VM architecture, which involves modifying the guest OS kernel to replace nonvirtualizable instructions with hypercalls for the hypervisor or the VMM to carry out the virtualization process

2. Implementation Levels of Virtualization:

Definition: Virtualization is a computer architecture technology where multiple virtual machines are multiplexed in the same hardware machine.

Hardware Resources: Cpu, Memory, I/O

Software Resources: Operating system, Library.

Levels Of Virtualization Implementation:

VMM:(Virtual machine monitor)-Different user applications managed at same H/W systems.

Levels:

- ➔ Instruction set Architecture Level: Code Interpretation (source to destination)
- ➔ Hardware Abstraction level: It generates virtual environment for VM
- ➔ Operating System Level: Allocates hardware resources
- ➔ Library Support Level: communication between application and system
- ➔ User Application Level: virtualize an application in a virtual machine.

VMM design requirements and providers:

-Three requirements

1. VMM provide environment for program
2. program run this environment, only minor decreases in speed
3. VMM should complete control of system resources.

Virtualization support at the OS level:

- ➔ The hardware level virtualization issues:

  1)storing VM images

  2)Full virtualization at hardware level leads to slow performance, low density.
- ➔ To solve this problem OS level virtualization is needed.

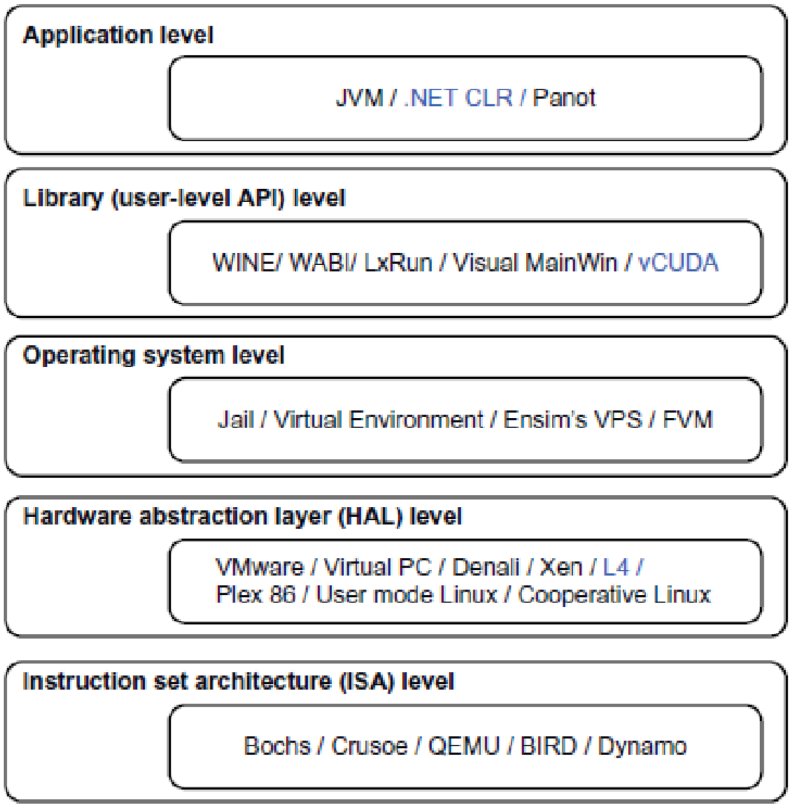
Advantages of OS level:

- minimal startup
- low resource requirement
- high scalability.

Middleware support for Virtualization:

- ➔ It provides an library level virtualization system

- windows Application Binary Interface(WABI):convert system calls to solaris system calls
- Visual Mainwin:To develop windows applications using visual studio.
- VCUDA.


Application level
JVM / .NET CLR / Panot
Library (user-level API) level
WINE/ WABI/ LxRun / Visual MainWin / vCUDA
Operating system level
Jail / Virtual Environment / Ensim's VPS / FVM
Hardware abstraction layer (HAL) level
VMware / Virtual PC / Denali / Xen / L4 /
Plex 86 / User mode Linux / Cooperative Linux
Instruction set architecture (ISA) level
Bochs / Crusoe / QEMU / BIRD / Dynamo

Virtualization ranging from hardware to applications in five abstraction levels.

1. **Instruction Set architecture Level**

   In ISA, virtualization works through an ISA emulation. This is helpful to run heaps of legacy code which was originally written for different hardware configurations.

   A binary code that might need additional layers to run can now run on an x86 machine or with some tweaking, even on x64 machines. ISA helps make this a hardware-agnostic virtual machine.

2. **Hardware abstraction Layer Level**

   This level helps perform virtualization at the hardware level. It uses a bare hypervisor for its functioning. This level helps form the virtual machine and manages the hardware through virtualization.It enables virtualization of each hardware component such as I/O devices, processors, memory, etc.This way multiple users can use the same hardware with numerous instances of virtualization at the same time.

3. Operating System Level

At the operating system level, the virtualization model creates an abstract layer between the applications and the OS.It is like an isolated container on the physical server and operating system that utilizes hardware and software. Each of these containers functions like servers.When the number of users is high, and no one is willing to share hardware, this level of virtualization comes in handy.

4. Library (user-Level API) Level

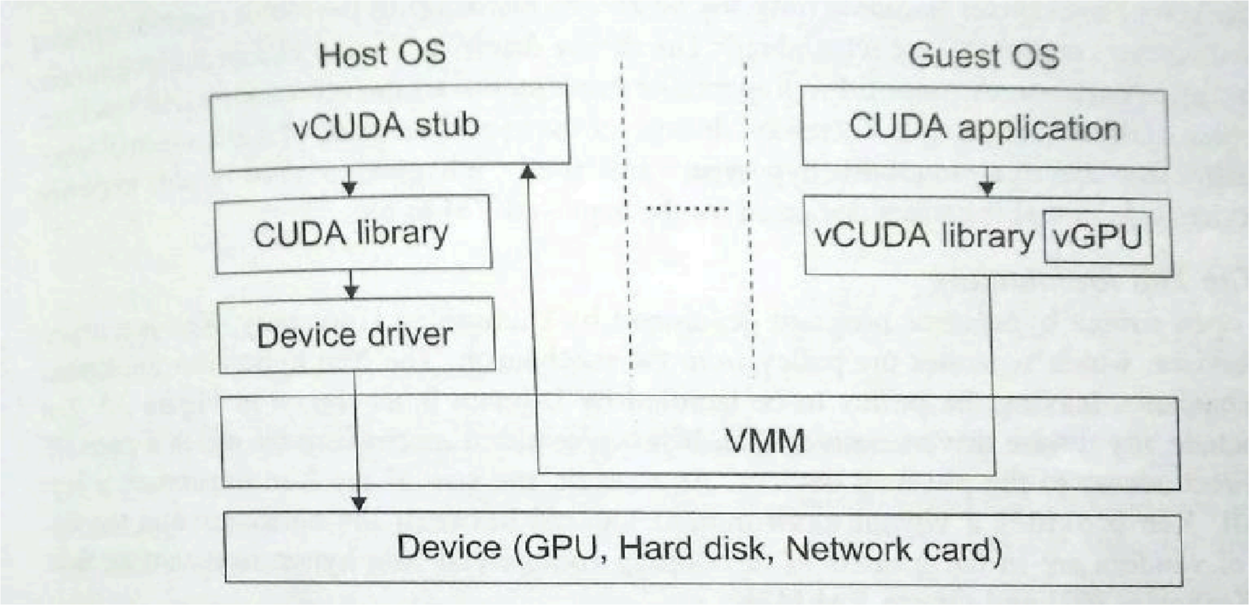
OS system calls are lengthy and cumbersome. Which is why applications opt for APIs from user-level libraries.Most of the APIs provided by systems are rather well documented. Hence, library level virtualization is preferred in such scenarios.Library interfacing virtualization is made possible by API hooks. These API hooks control the communication link from the system to the applications.Some tools available today, such as vCUDA and WINE, have successfully demonstrated this technique.

5. Application Level

Application-level virtualization comes handy when you wish to virtualize only an application. It does not virtualize an entire platform or environment.On an operating system, applications work as one process. Hence it is also known as process-level virtualization.It is generally useful when running virtual machines with high-level languages. Here, the application sits on top of the virtualization layer, which is above the application program.The application program is, in turn, residing in the operating system.Programs written in high-level languages and compiled for an application-level virtual machine can run fluently here.

Example:VCUDA

It is a programming model and library for general purpose GPUs.It employs client server model to implement CUDA virtualization.



3.Virtualization of CPU,Memory and I/O devices:

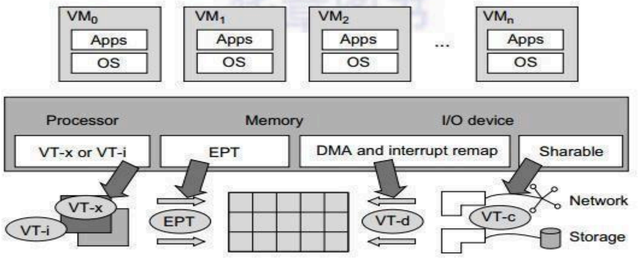
Definition: Virtualization is a computer architecture technology where multiple virtual machines are multiplexed in the same hardware machine.

Hardware support for virtualization:

Two instructions

1. **privileged instruction**
2. **non privileged instruction.**

- ➔ kernel based virtual machine: Linux kernel virtualization
- ➔ Architecture ->hardware support for virtualization Intel X86 processor.



CPU virtualized:

➔ **critical instruction**: Three instructions

➔ **privileged instruction**: execute in a privileged mode and will be trapped if executed outside this mode.

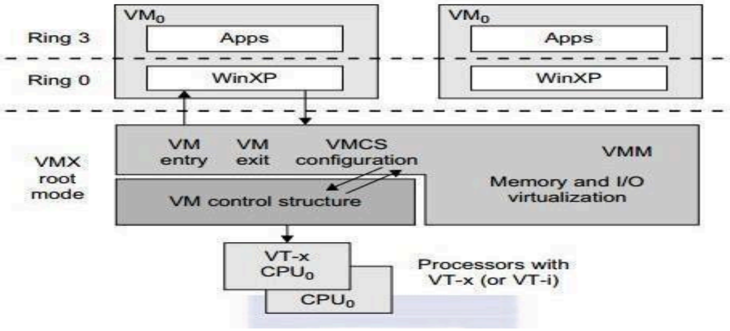
➔ **control sensitive instructions**: attempt to change the configuration of resources used.

➔ **behavior sensitive instructions**: including the load and store operations over the virtual memory

Hardware Assisted CPU virtualization:

This technique attempts to simplify virtualization because full or paravirtualization is complicated. Intel and AMD add an additional mode called privilege mode level (some people call it Ring-1) to x86 processors. Therefore, operating systems can still run at Ring 0 and the hypervisor can run at Ring -1. All the privileged and sensitive instructions are trapped in the hypervisor automatically. This technique removes the difficulty of implementing binary translation of full virtualization. It also lets the operating system run in VMs without modification.

There are two modes to run under virtualization: root operation and non-root operation. Usually only the virtualization controlling software, called Virtual Machine Monitor (VMM), runs under non-root operation. Software running on top of virtual machines is also called ‘guest software’.



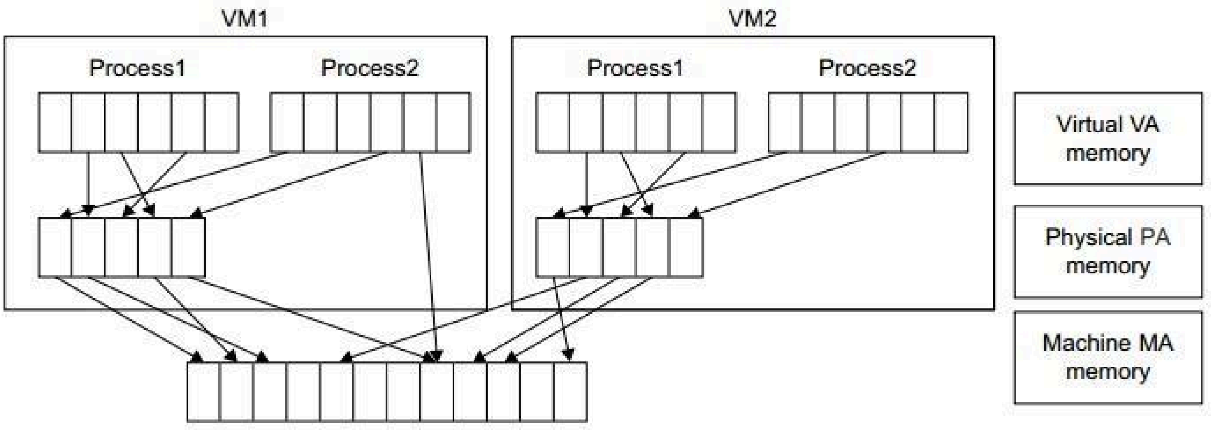
Memory virtualization:

- In a traditional execution environment the OS maintains mappings of virtual memory to machine memory using page tables, which is one stage mapping from virtual memory to machine memory.
- All modern x86 CPUs include a Memory management Unit and a translation Look-aside Buffer to optimize virtual memory performance.
- Three types of memory
  - ➔ virtual memory
  - ➔ machine memory
  - ➔ physical memory

Two stage:

1)virtual memory to physical memory

2)physical memory to machine memory.



I/O virtualization:

➔ I/O virtualization manages the route of I/O requests between virtual and physical hardware.

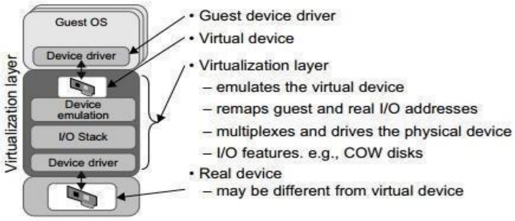
➔ Three ways to implement I/O virtualization

❖ **full device emulation**->I/O access requests of the guest OS are trapped in the VMM which interacts with the I/O devices.

❖ **para-virtualization**: I/O virtualization used in Xen, frontend and backend driver.

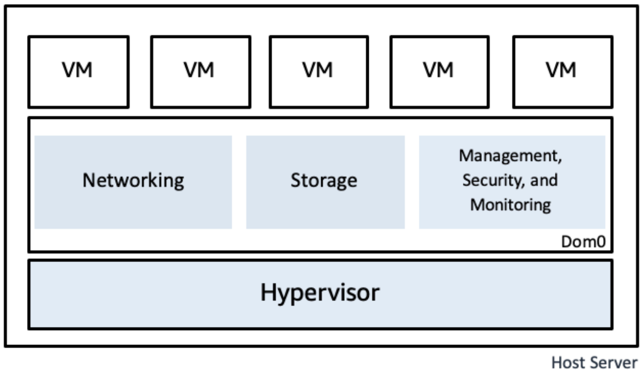
❖ **direct I/O:** implements focus on networking for mainframes.

All the functions of a device or bus infrastructure, such as device enumeration, identification, interrupts, and DMA, are replicated in software. This software is located in the VMM and acts as a virtual device. The I/O access requests of the guest OS are trapped in the VMM which interacts with the I/O devices.


Guest OS
Device driver
Virtualization layer
Device emulation
I/O Stack
Device driver
• Guest device driver
• Virtual device
• Virtualization layer
– emulates the virtual device
– remaps guest and real I/O addresses
– multiplexes and drives the physical device
– I/O features. e.g., COW disks
• Real device
– may be different from virtual device

4.Hypervisor and XEN Architecture:

A hypervisor is a software that you can use to run multiple virtual machines on a single physical machine.The hypervisor supports hardware level virtualization on bare metal devices like CPU, memory, disk and network interfaces.The hypervisor software sits directly between the physical hardware and its OS. This virtualization layer is referred to as either the VMM or the hypervisor.



The hypervisor provides hypercalls for the guest OSes and applications. Depending on the functionality, a hypervisor can assume microkernel architecture like the Microsoft Hyper-V.

It can assume monolithic hypervisor architecture like the VMware ESX for server virtualization.

A microkernel hypervisor includes only the basic and unchanging functions (such as physical memory management and processor scheduling).

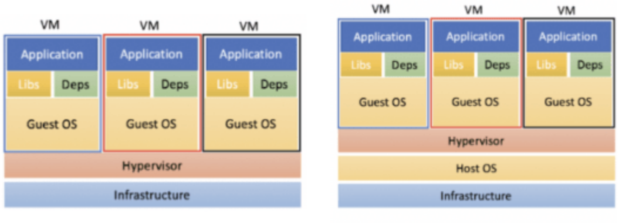
The device drivers and other changeable components are outside the hypervisor.

TYPE-1 Hypervisor:

The hypervisor runs directly on the underlying host system. It is also known as a “Native Hypervisor” or “Bare metal hypervisor”. It does not require any base server operating system. It has direct access to hardware resources. Examples of Type 1 hypervisors include VMware ESXi, Citrix XenServer, and Microsoft Hyper-V hypervisor.

Pros & Cons of Type-1 Hypervisor:

Pros: Such kinds of hypervisors are very efficient because they have direct access to the physical hardware resources(like Cpu, Memory, Network, and Physical storage). This causes the empowerment of the security because there is nothing of any kind of the third party resource so that attacker couldn’t compromise with anything.

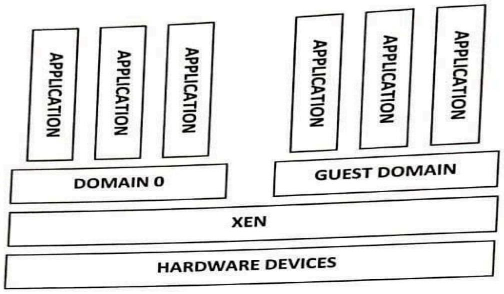


Type-1

Type-2

Xen architecture:

Xen is an open source hypervisor program developed by Cambridge University. Xen is a microkernel hypervisor, which separates the policy from the mechanism.The Xen hypervisor implements all the mechanisms, leaving the policy to be handled by Domain .Figure shows architecture of Xen hypervisor.Xen does not include any device drivers natively. It just provides a mechanism by which a guest OS can have direct access to the physical devices.



The core components of a Xen system are the hypervisor, kernel, and applications.

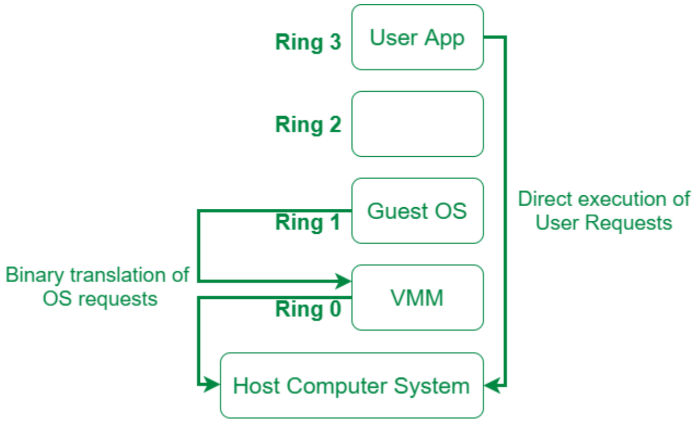
The organization of the three components is important. Like other virtualization systems, many guest OSes can run on top of the hypervisor. However, not all guest OSes are created equal, and one in particular controls the others.The guest OS, which has control ability, is called Domain 0, and the others are called Domain U.Domain 0 is a privileged guest OS of Xen. It is first loaded when Xen boots without any file system drivers being available. Domain 0 is designed to access hardware directly and manage devices. Therefore, one of the responsibilities of Domain 0 is to allocate and map hardware resources for the guest domains (the Domain U domains).

For example, Xen is based on Linux and its security level is C2. Its management VM is named Domain 0 which has the privilege to manage other VMs implemented on the same host. If Domain 0 is compromised, the hacker can control the entire system. So, in the VM system, security policies are needed to improve the security of Domain 0. Domain 0, behaving as a VMM, allows users to create, copy, save, read, modify, share, migrate and roll back VMs as easily as manipulating a file, which flexibly provides tremendous benefits for users.

5.Virtualization Types:

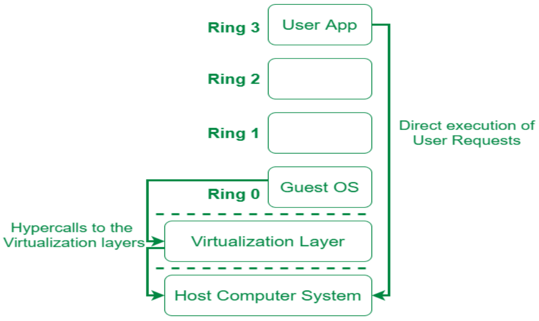
Hardware virtualization provides an abstract execution environment by Hardware assisted virtualization, Full virtualization, Para virtualization and Partial virtualization techniques.

Full Virtualization: Full Virtualization was introduced by IBM in the year 1966. It is the first software solution for server virtualization and uses binary translation and direct approach techniques. In full virtualization, guest OS is completely isolated by the virtual machine from the virtualization layer and hardware. Microsoft and Parallels systems are examples of full virtualization.


Ring 3
User App
Ring 2
Ring 1
Guest OS
Direct execution of User Requests
Binary translation of OS requests
VMM
Ring 0
Host Computer System

Full Virtualization

Paravirtualization: Paravirtualization is the category of CPU virtualization which uses hypercalls for operations to handle instructions at compile time. In paravirtualization, guest OS is not completely isolated but it is partially isolated by the virtual machine from the virtualization layer and hardware. VMware and Xen are some examples of paravirtualization.



Paravirtualization

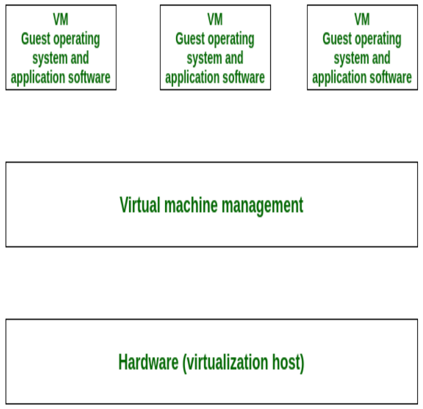
The difference between Full Virtualization and Para virtualization are:

S.No.	Full Virtualization	Paravirtualization
1.	In Full virtualization, virtual machines permit the execution of the instructions with the running of unmodified OS in an entirely isolated way.	In paravirtualization, a virtual machine does not implement full isolation of the OS but rather provides a different API which is utilized when the OS is subjected to alteration.

S.No.	Full Virtualization	Paravirtualization
2.	Full Virtualization is less secure.	Paravirtualization is more secure than Full Virtualization.
3.	Full Virtualization uses binary translation and a direct approach as a technique for operations.	Paravirtualization uses hypercalls at compile time for operations.
4.	Full Virtualization is slow than paravirtualization in operation.	Paravirtualization is faster in operation as compared to full virtualization.
5.	Full Virtualization is more portable and compatible.	Paravirtualization is less portable and compatible.
6.	Examples of full virtualization are Microsoft and Parallels systems.	Examples of paravirtualization are Microsoft Hyper-V, Citrix Xen, etc.
7.	It supports all guest operating systems without modification.	The guest operating system has to be modified and only a few operating systems support it.
8.	The guest operating system will issue hardware calls.	Using the drivers, the guest operating system will directly communicate with the hypervisor.
9.	It is less streamlined compared to para-virtualization.	It is more streamlined.
10.	It provides the best isolation.	It provides less isolation compared to full virtualization.

Hardware Based Virtualization:

A platform virtualization approach that allows efficient full virtualization with the help of hardware capabilities, primarily from the host processor is referred to as Hardware based virtualization in computing. To simulate a complete hardware environment, or virtual machine, full virtualization is used in which an unchanged guest operating system (using the common instruction set as the host machine) executes in sophisticated isolation.


VM
Guest operating system and application software
VM
Guest operating system and application software
VM
Guest operating system and application software
Virtual machine management
Hardware (virtualization host)

An abstract execution environment in terms of computer hardware in which guest OS can be run, referred to as Hardware-level virtualization. In this, an operating system represents the guest, the physical computer hardware represents a host, its emulation represents a virtual machine, and the hypervisor represents the Virtual Machine Manager. When the virtual machines are allowed to interact with hardware without any intermediary action requirement from the host operating system generally makes hardware-based virtualization more efficient. A fundamental component of hardware virtualization is the hypervisor, or virtual machine manager (VMM).

Features of hardware-based virtualization are:

Isolation: Hardware-based virtualization provides strong isolation between virtual machines, which means that any problems in one virtual machine will not affect other virtual machines running on the same physical host.

Security: Hardware-based virtualization provides a high level of security as each virtual machine is isolated from the host operating system and other virtual machines, making it difficult for malicious code to spread from one virtual machine to another.

Performance: Hardware-based virtualization provides good performance as the hypervisor has direct access to the physical hardware, which means that virtual machines can achieve close to native performance.

Resource allocation: Hardware-based virtualization allows for flexible allocation of hardware resources such as CPU, memory, and I/O bandwidth to virtual machines.

Snapshot and migration: Hardware-based virtualization allows for the creation of snapshots, which can be used for backup and recovery purposes. It also allows for live migration of virtual machines between physical hosts, which can be used for load balancing and other purposes.

Support for multiple operating systems: Hardware-based virtualization supports multiple operating systems, which allows for the consolidation of workloads onto fewer physical machines, reducing hardware and maintenance costs.

Compatibility: Hardware-based virtualization is compatible with most modern operating systems, making it easy to integrate into existing IT infrastructure.

Advantages of hardware-based virtualization –

It reduces the maintenance overhead of paravirtualization as it reduces (ideally, eliminates) the modification in the guest operating system. It is also significantly convenient to attain enhanced performance. A practical benefit of hardware-based virtualization has been mentioned by VMware engineers and Virtual Iron.

Disadvantages of hardware-based virtualization –

Hardware-based virtualization requires explicit support in the host CPU, which may not available on all x86/x86_64 processors. A "pure" hardware-based virtualization approach, including the entire unmodified guest operating system, involves many VM traps, and thus a rapid increase in CPU overhead occurs which limits the scalability and efficiency of server consolidation. This performance hit can be mitigated by the use of para-virtualized drivers; the combination has been called "hybrid virtualization".

6.Key Concepts of Virtualization:

Hypervisor vs Increased security

The ability to control the execution of a guest in a completely transparent manner opens new possibilities for delivering a secure, controlled execution environment.

The virtual machine represents an emulated environment in which the guest is executed. This level of indirection allows the virtual machine manager to control and filter the activity of the guest, thus preventing some harmful operations from being performed.

Managed execution Virtualization of the execution environment not only allows increased security, but a wider range of features also can be implemented.

In particular, sharing, aggregation, emulation, and isolation are the most relevant features.
Sharing

Virtualization allows the creation of a separate computing environment within the same host.

In this way it is possible to fully exploit the capabilities of a powerful guest, which would otherwise be underutilized.

Aggregation

Not only is it possible to share physical resources. among several guests but Virtualization also allows aggregation, which is the opposite process.

A group of separate hosts can be tied together and represented to guests as a single virtual host.

Emulation

Guest programs are executed within an environment that is controlled by the virtualization layer, which ultimately is a program.This allows for controlling and tuning the environment that is exposed to guests.

Isolation

Virtualization allows providing guests whether they are operating systems, applications, or other entities with a completely separate environment, in which they are executed. • The guest program performs its activity by interacting with an abstraction layer, which provides access to the underlying resources.

Benefits of Isolation

First it allows multiple guests to run on the same host without interfering with each other. Second, it provides a separation between the host and the guest.

Another important capability Enabled by virtualization is performance tuning. This feature is a reality at present, given the considerable advances in hardware and software supporting virtualization.

It becomes easier to control the performance of the guest by finely tuning the properties of the resources exposed through the virtual environment.

This capability provides a means to effectively implement a quality of service (QoS) infrastructure that more easily fulfills the service level agreement (SLA) established for the guest

VM Portability

The concept of portability applies in different ways according to the specific type of virtualization considered.

In the case of a hardware virtualization solution, the guest is packaged into a virtual image that, in most cases, can be safely moved and executed on top of different virtual machines

7. Taxonomy of virtual machines:

- ➔ Virtualization is mainly used to emulate execution environments, storage and networks.
- ➔ Execution virtualization techniques into two major categories by considering the type of host they require.
- ➔ Process level techniques are implemented on top of an existing operating system, which has full control of the hardware. System level techniques are implemented directly on hardware and do not require or require a minimum of support from existing operating system.Within these two categories we can list various techniques that offer the guest a different type of virtual computing environment:

Bare hardware

1.Operating system resources

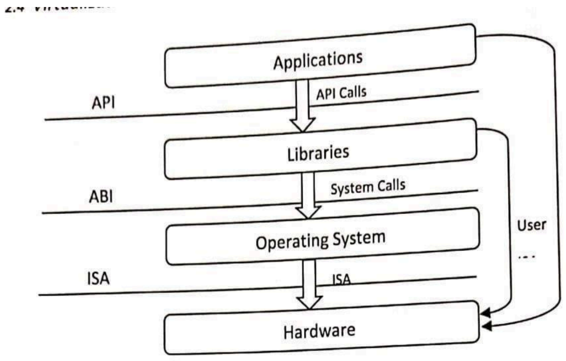
2.Low level programming language

3.Application libraries

Execution virtualization includes all techniques that aim to emulate an execution environment that is separate from the one hosting the virtualization layer.

All these techniques concentrate their interest on providing support for the execution of programs, whether these are the operating system, a binary specification of a program compiled against an abstract machine model or an application.

Therefore, execution virtualization can be implemented directly on top of the hardware by the operating system, an application and libraries (dynamically or statically) linked to an application image.



At the bottom layer, the model for the hardware is expressed in terms of the Instruction Set Architecture (ISA), which defines the instruction set for the processor, registers, memory and interrupt management.

ISA is the interface between hardware and software.

ISA is important to the operating system (OS) developer (System ISA) and developers of applications that directly manage the underlying hardware (User ISA).

The application binary interface (ABI) separates the operating system layer from the applications and libraries, which are managed by the OS. ABI covers details such as low level data types, alignment, call conventions and defines a format for executable programs.

System calls are defined at this level. This interface allows portability of applications and libraries across operating systems that implement the same ABI.

The highest level of abstraction is represented by the application programming interface (API), which interfaces applications to libraries and the underlying operating system.

For this purpose, the instruction set exposed by the hardware has been divided into different security classes that define who can operate with them. The first distinction can be made between privileged and non privileged instructions.

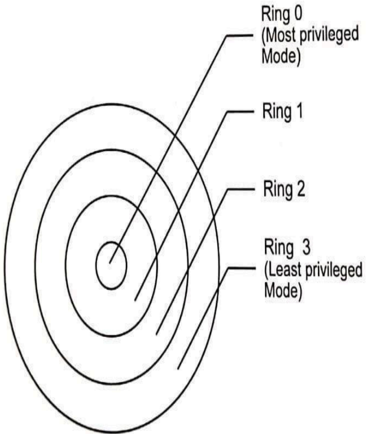
- Non privileged instructions are those instructions that can be used without interfering with other tasks because they do not access shared resources.

This category contains all the floating, fixed-point, and arithmetic instructions.

- Privileged instructions are those that are executed under specific restrictions and are mostly used for sensitive operations, which expose (behavior-sensitive) or modify (control-sensitive) the privileged state.

Some types of architecture feature more than one class of privileged instructions and implement a finer control of how these instructions can be accessed.

For instance, a possible implementation features a hierarchy of privileges illustrate in the figure in the form of ring-based security: Ring 0, Ring 1, Ring 2, and Ring 3;
Ring 0 is in the most privileged level and Ring 3 in the least privileged level.
Ring 0 is used by the kernel of the OS, rings 1 and 2 are used by the OS level services, and Ring 3 is used by the user.
Recent systems support only two levels, with Ring 0 for supervisor mode and Ring 3 for user mode.



The supervisor mode denotes an execution mode in which all the instructions (privileged and non privileged) can be executed without any restriction.This mode, also called master mode or kernel mode, is generally used by the operating system (or the hypervisor) to perform sensitive operations on hardware level resources.

In user mode, there are restrictions to control the machine level resources. The distinction between user and supervisor mode allows us to understand the role of the hypervisor and why it is called that.

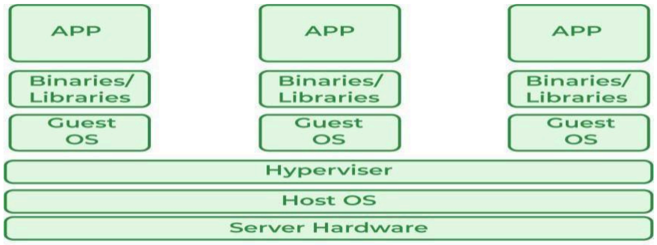
- Conceptually, the hypervisor runs above the supervisor mode and from here the prefix "hyper" is used.
- In reality, hypervisors are run in supervisor mode and the division between privileged and non privileged instructions has posed challenges in designing virtual machine managers.

S.NO	Cloud Computing	Virtualization
1.	Cloud computing is used to provide pools and automated resources that can be accessed on-demand.	It is used to make various simulated environments through a physical hardware system.
2.	Cloud computing setup is tedious, complicated.	While virtualization setup is simple as compared to cloud computing.
3.	Cloud computing is high scalable.	While virtualization is low scalable compared to cloud computing.

4.	Cloud computing is Very flexible.	While virtualization is less flexible than cloud computing.
5.	In the condition of disaster recovery, cloud computing relies on multiple machines.	While it relies on single peripheral device.
6.	In cloud computing, the workload is stateless.	In virtualization, the workload is stateful.
7.	The total cost of cloud computing is higher than virtualization.	The total cost of virtualization is lower than Cloud Computing.
8.	Cloud computing requires many dedicated hardware.	While single dedicated hardware can do a great job in it.
9.	Cloud computing provides unlimited storage space.	While storage space depends on physical server capacity in virtualization.
10.	Cloud computing is of two types : Public cloud and Private cloud.	Virtualization is of two types : Hardware virtualization and Application virtualization.
11.	In Cloud Computing, Configuration is image based.	In Virtualization, Configuration is template based.
12.	In cloud computing, we utilize the entire server capacity and the entire servers are consolidated.	In Virtualization, the entire servers are on-demand.

8. Virtualization in Cloud Computing and Types:

Virtualization is a technique to separate a service from the underlying physical delivery of that service. It is the process of creating a virtual version of something like computer hardware. It was initially developed during the mainframe era. It involves using specialized software to create a virtual or software-created version of a computing resource rather than the actual version of the same resource. With the help of Virtualization, multiple operating systems and applications can run on the same machine and its same hardware at the same time, increasing the utilization and flexibility of hardware.In other words, one of the main cost-effective, hardware-reducing, and energy-saving techniques used by cloud providers is Virtualization. Virtualization allows sharing of a single physical instance of a resource or an application among multiple customers and organizations at one time. It does this by assigning a logical name to physical storage and providing a pointer to that physical resource on demand. The term virtualization is often synonymous with hardware virtualization, which plays a fundamental role in efficiently delivering Infrastructure-as- a-Service (IaaS) solutions for cloud computing. Moreover, virtualization technologies provide a virtual environment for not only executing applications but also for storage, memory, and networking.



Virtualization

- Host Machine: The machine on which the virtual machine is going to be built is known as Host Machine.
- Guest Machine: The virtual machine is referred to as a

Guest Machine. Work of Virtualization in Cloud Computing

Benefits of Virtualization

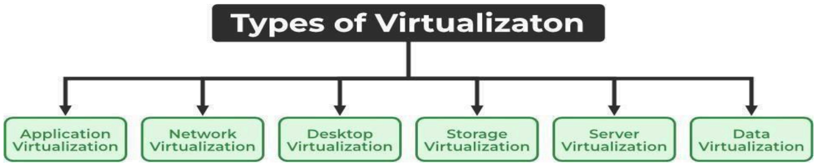
- More flexible and efficient allocation of resources.
- Enhance development productivity.
- It lowers the cost of IT infrastructure.
- Remote access and rapid scalability.
- High availability and disaster recovery.
- Pay per use of the IT infrastructure on demand.

Drawback of Virtualization:

- High Initial Investment
- Learning New Infrastructure
- Risk of Data

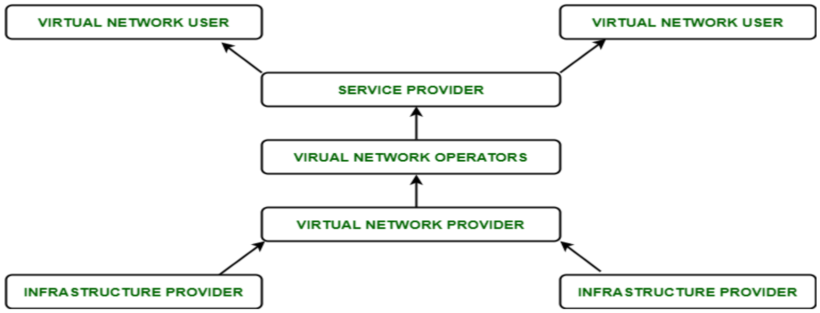
Types of Virtualization

1. Application Virtualization
2. Network Virtualization
3. Desktop Virtualization
4. Storage Virtualization
5. Server Virtualization
6. Data virtualization



Application Virtualization: Application virtualization helps a user to have remote access to an application from a server. The server stores all personal information and other characteristics of the application but can still run on a local workstation through the internet. An example of this would be a user who needs to run two different versions of the same software. Technologies that use application virtualization are hosted applications and packaged applications.

Network Virtualization: The ability to run multiple virtual networks with each having a separate control and data plan. It co-exists together on top of one physical network. It can be managed by individual parties that are potentially confidential to each other. Network virtualization provides a facility to create and provision virtual networks, logical switches, routers, firewalls, load balancers, Virtual Private Networks (VPN), and workload security within days or even weeks.

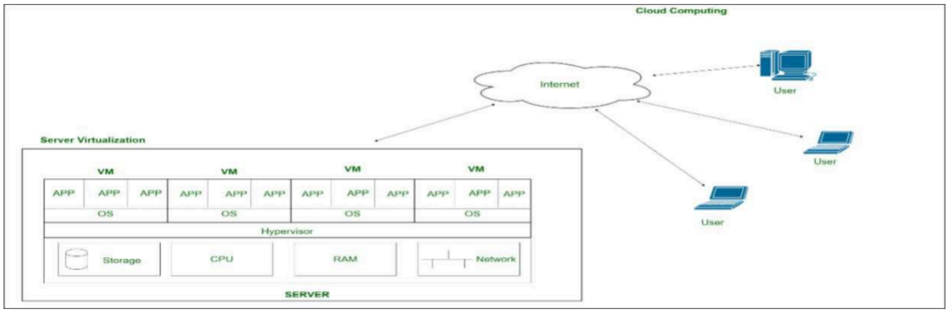


Desktop Virtualization: Desktop virtualization allows the users' OS to be remotely stored on a server in the data center. It allows the user to access their desktop virtually, from any location by a different machine. Users who want specific operating systems other than Windows Server will need to have a virtual desktop. The main benefits of desktop virtualization are user mobility, portability, and easy management of software installation, updates, and patches.

Storage virtualization is an array of servers that are managed by a virtual storage system. The servers aren't aware of exactly where their data is stored and instead function more like worker bees in a hive. It makes managing storage from multiple sources be managed and utilized as a single repository. storage virtualization software maintains smooth

Operations.

Server Virtualization: This is a kind of virtualization in which the masking of server resources takes place. Here, the central server (physical server) is divided into multiple different virtual servers by changing the identity number, and processors. So, each system can operate its operating systems in an isolated manner. Where each sub-server knows the identity of the central server. It causes an increase in performance and reduces the operating cost by the deployment of main server resources into a sub-server resource. It’s beneficial in virtual migration, reducing energy consumption, reducing infrastructural costs, etc.



Data Virtualization: This is the kind of virtualization in which the data is collected from various sources and managed at a single place without knowing more about the technical information like how data is collected, stored & formatted then arranged that data logically so that its virtual view can be accessed by its interested people and stakeholders, and users through the various cloud services remotely. Many big giant companies are providing their services like Oracle, IBM, At scale, Cdata, etc.

Uses of Virtualization

- Data-integration
- Business-integration
- Service-oriented architecture data-services
- Searching organizational data