BITS Pilani

Cloud Computing - Virtualization

Agenda

- Cloud Recap
  - What is NIST 3-4-5 Rule
  - Advantages of Cloud
  - Disadvantages
- Introduction to Virtualization
  - What is Virtualization
  - Use & demerits of Virtualization
  - Introducing the Hypervisor
  - Purpose, Design Goals & Types of Hypervisor
- Virtualization
  - Types of Virtualization
  - X86 Hardware Virtualization
  - NFV - VNF

innovate achieve lead

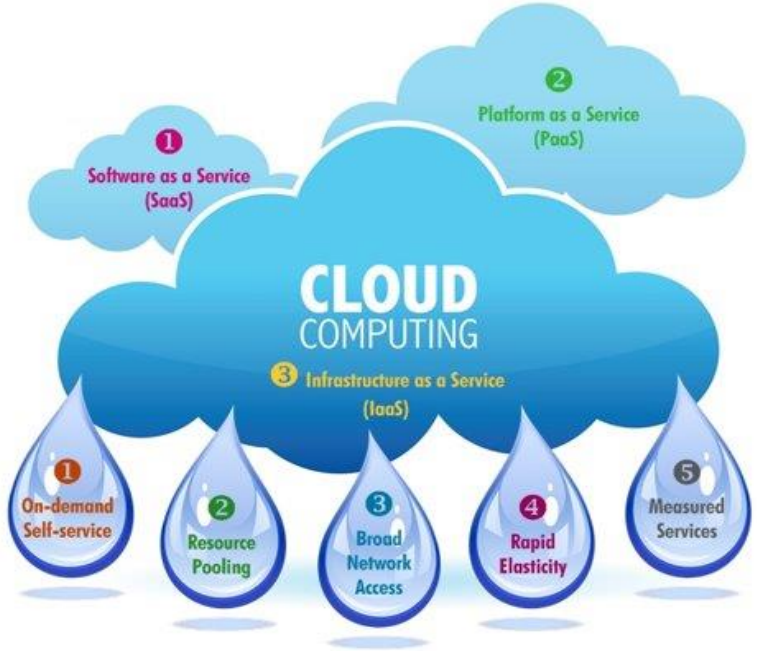
BITS Pilani

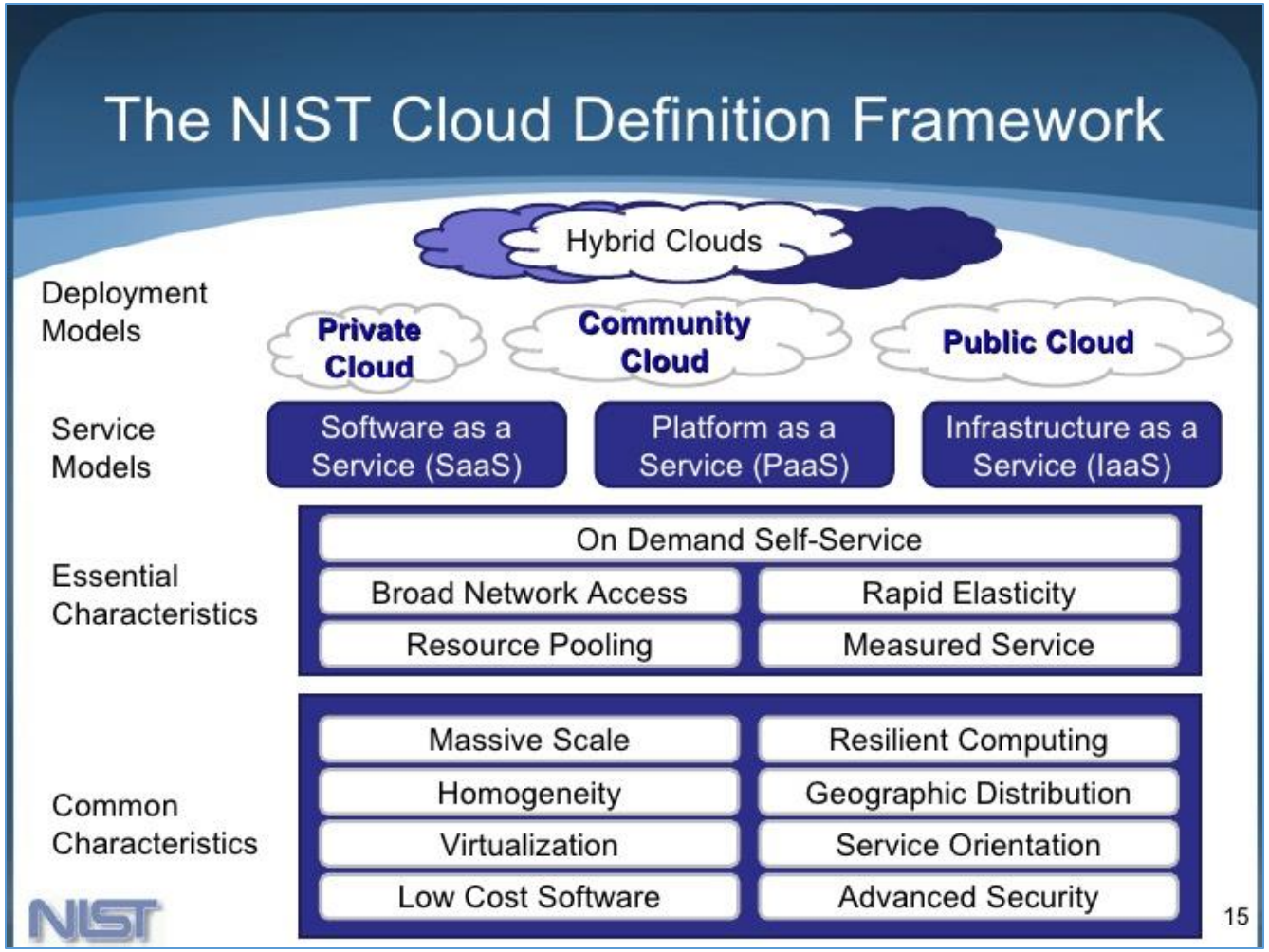
Pilani|Dubai|Goa|Hyderabad

Recap

NIST Definitions

- 3 cloud service models or service types for any cloud platform
- 4 Deployment models
- 5 Essential characteristics of cloud computing infrastructure



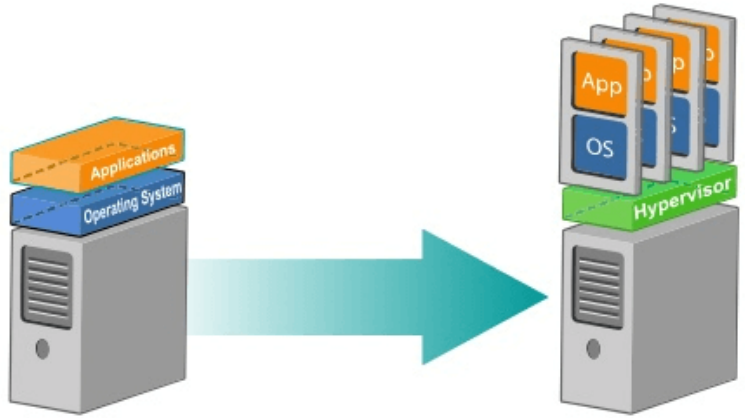


innovate achieve lead

BITS Pilani

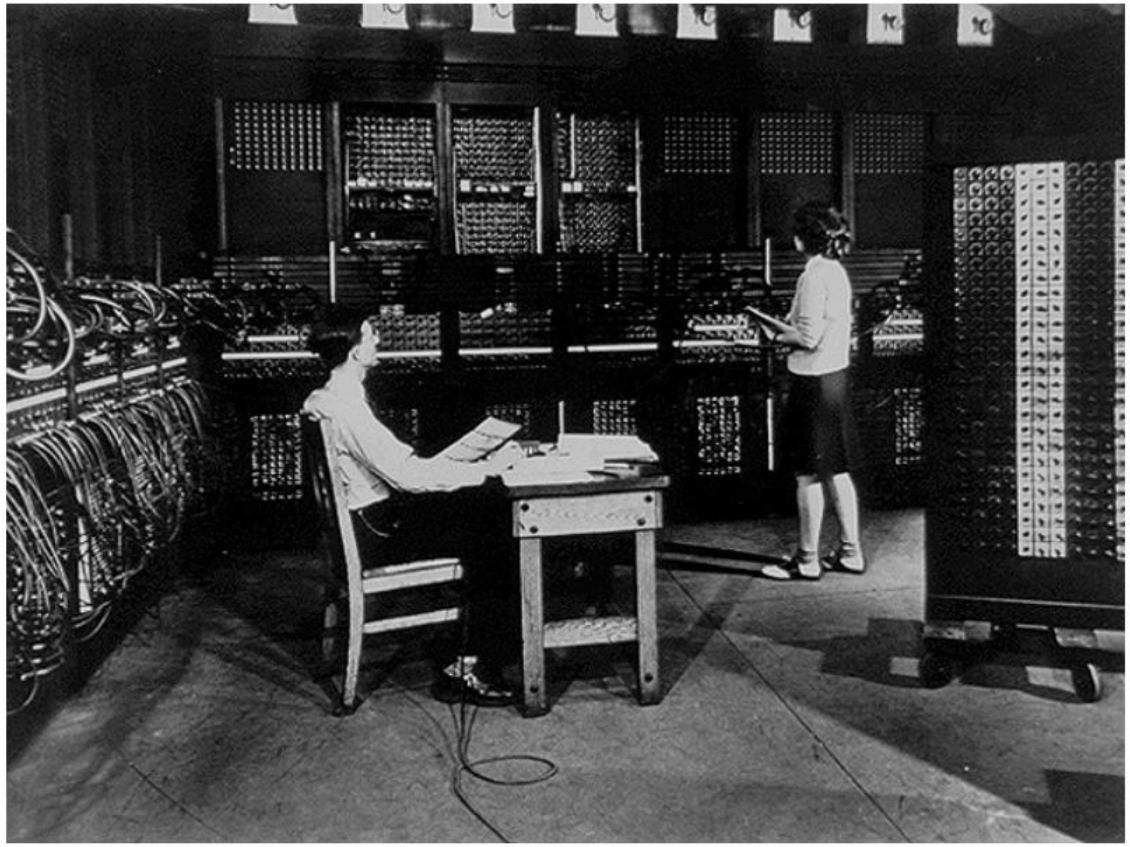
Pilani|Dubai|Goa|Hyderabad

Introduction to Virtualization



Virtualization History

History



ENIAC, the first electronic computing machine

***E**lectronic **N**umerical **I**ntegrator **A**nd **C**omputer*

IBM 7094 console, two magnetic tape drives and a punch card reader. This computer took the whole room. © NASA Ames Resarch Center

Motivations & Origins

Motivation

1 machine → 1 OS → several applications

Applications can affect each other

Big disadvantage: machine utilization is very low, most of the times it is below than 25%

Origins

- Server virtualization has existed for several decades
- IBM pioneered more than 30 years ago with the capability to "multitask"
- The inception was in specialized, proprietary, high-end server and mainframe systems. By 1980/90 servers virtualization adoption reduced
- Inexpensive x86 hardware platforms
- Windows/Linux adopted as server

Video – Virtualization

Learning Objectives

- Introduce Oracle Virtual Box, a hosted hypervisor.
- Demonstrate what a host system is what a guest VM is and what is the role of the hypervisor.
- Students will use the same as home work and install virtual box and a choice of their own OS after class.

What is Virtualization?

Virtualization Defined

Virtualization is a computer architecture technology by which multiple virtual machines (VMs) are multiplexed in the same hardware machine.

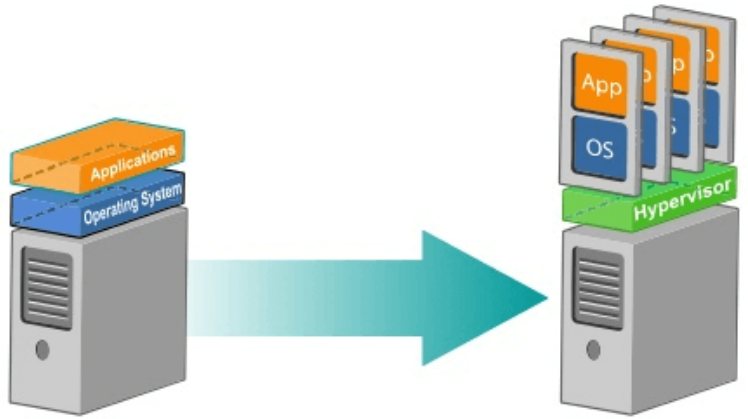
Virtualization allows multiple operating system instances to run concurrently on a single computer

Instead of purchasing and maintaining an entire computer for one application, each application can be given its own operating system, and all those operating systems can reside on a single piece of hardware.

Virtualization allows an operator to control a guest operating system's use of CPU, memory, storage, and other resources, so each guest receives only the resources that it needs.

Key Terms:

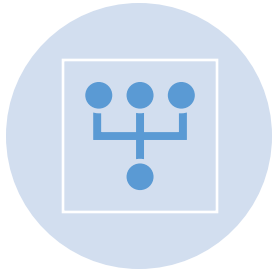
- VM ➔ Virtual Machine
- VMM ➔ Virtual Machine Monitor
- Hypervisor ➔ VMM
- Multiplexed ➔ Many or several
- Host ➔ System where the VMM resides
- Guest ➔ Virtual Machines created



What is Virtualization?

Virtualization Objectives

ABSTRACTION – TO SIMPLIFY THE USE OF THE UNDERLYING RESOURCE (E.G., BY REMOVING DETAILS OF THE RESOURCE'S STRUCTURE)



REPLICATION – TO CREATE MULTIPLE INSTANCES OF THE RESOURCE (E.G., TO SIMPLIFY MANAGEMENT OR ALLOCATION)

ISOLATION – TO SEPARATE THE USES WHICH CLIENTS MAKE OF THE UNDERLYING RESOURCES (E.G., TO IMPROVE SECURITY)

Key Terms:

- VM ➔ Virtual Machine
- VMM ➔ Virtual Machine Monitor
- Hypervisor ➔ VMM
- Multiplexed ➔ Many or several
- Host ➔ System where the VMM resides
- Guest ➔ Virtual Machines created

What is Virtualization?

Need of Virtualization

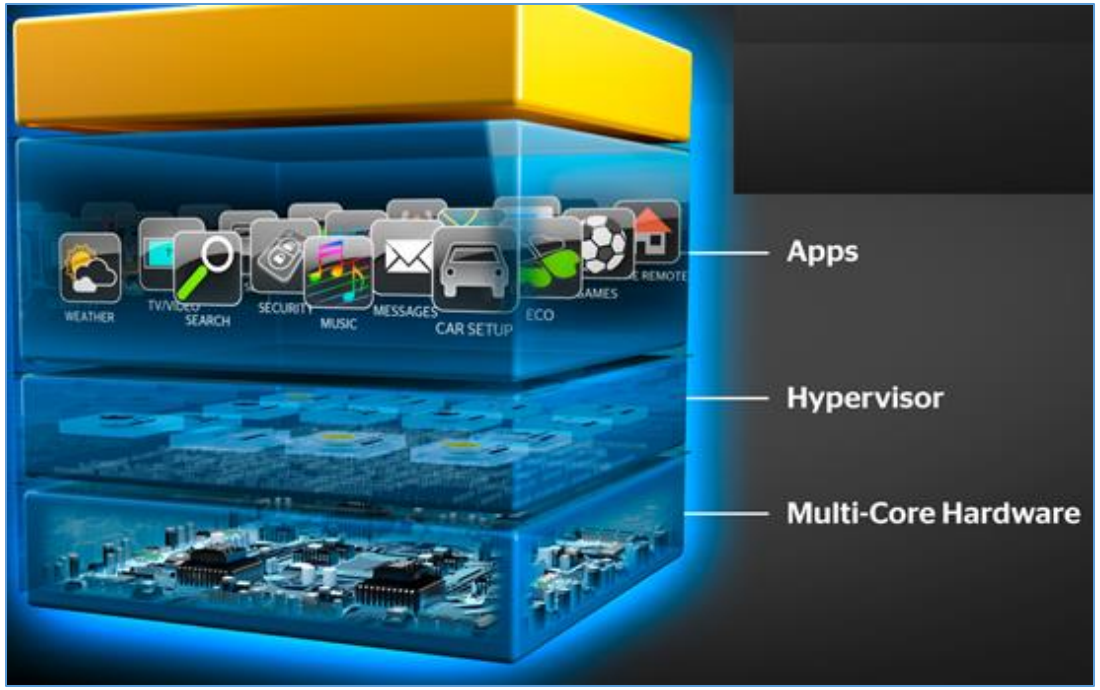
- Cloud can exist without Virtualization, although it will be difficult and inefficient.
- Cloud makes notion of "Pay for what you use", "infinite availability- use as much you want".
- These notions are practical only if we have
  - lot of flexibility
  - efficiency in the back-end.
- This efficiency is readily available in Virtualized Environments and Machines

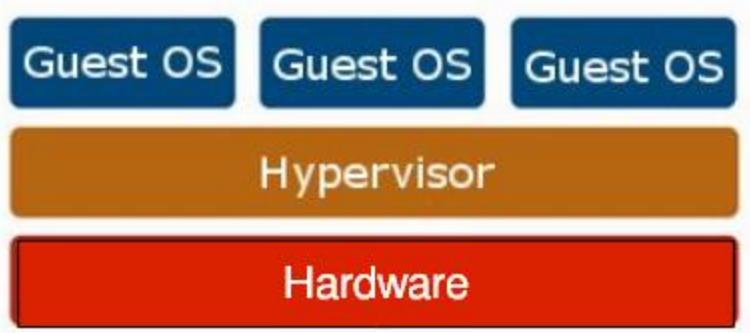
Key Terms:

- VM ➔ Virtual Machine
- VMM ➔ Virtual Machine Monitor
- Hypervisor ➔ VMM
- Multiplexed ➔ Many or several
- Host ➔ System where the VMM resides
- Guest ➔ Virtual Machines created

Virtualization Architecture

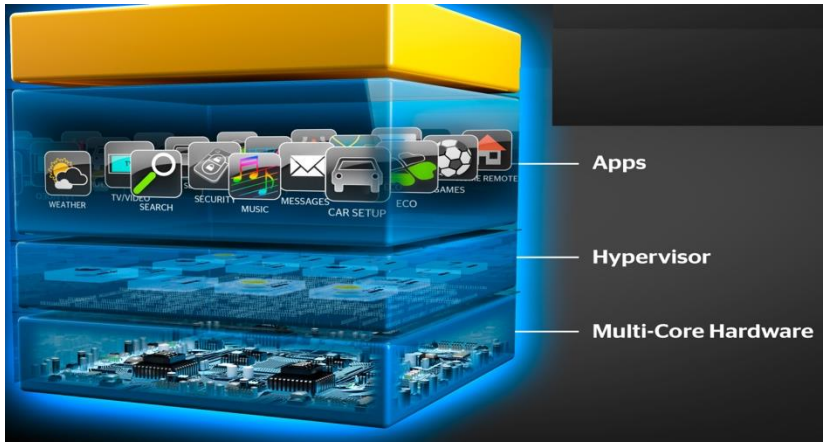
- OS assumes complete control of the underlying hardware.
- Virtualization architecture provides this illusion through a hypervisor/VMM.
- Hypervisor/VMM is a software layer which:
- Allows multiple Guest OS (Virtual Machines) to run simultaneously on a single physical host
- Provides hardware abstraction to the running Guest OSs and efficiently multiplexes underlying hardware resources



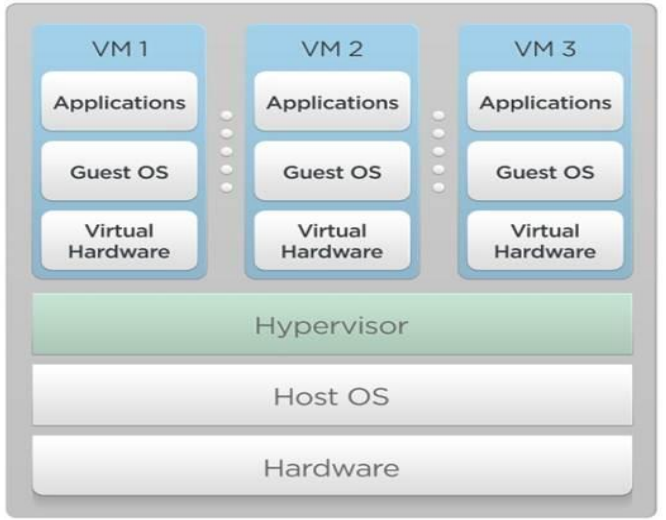


Hypervisor

- A **hypervisor** or **virtual machine monitor (VMM)** is computer software, firmware, or hardware. VMM creates and runs virtual machines.
- A computer on which a hypervisor runs one or more virtual machines is called a *host machine*,
- Each virtual machine is called a *guest machine*
- The hypervisor presents the guest systems with a virtual operating platform and manages the execution of the guest operating systems.
- Multiple instances of a variety of operating systems may share the virtualized hardware resources:



Bare Metal Hypervisor



Hosted Hypervisor

Hypervisor Goals

The main function of the software layer for virtualization is to **virtualize the physical hardware** of the host machine to virtual resources to be used by guest VM's

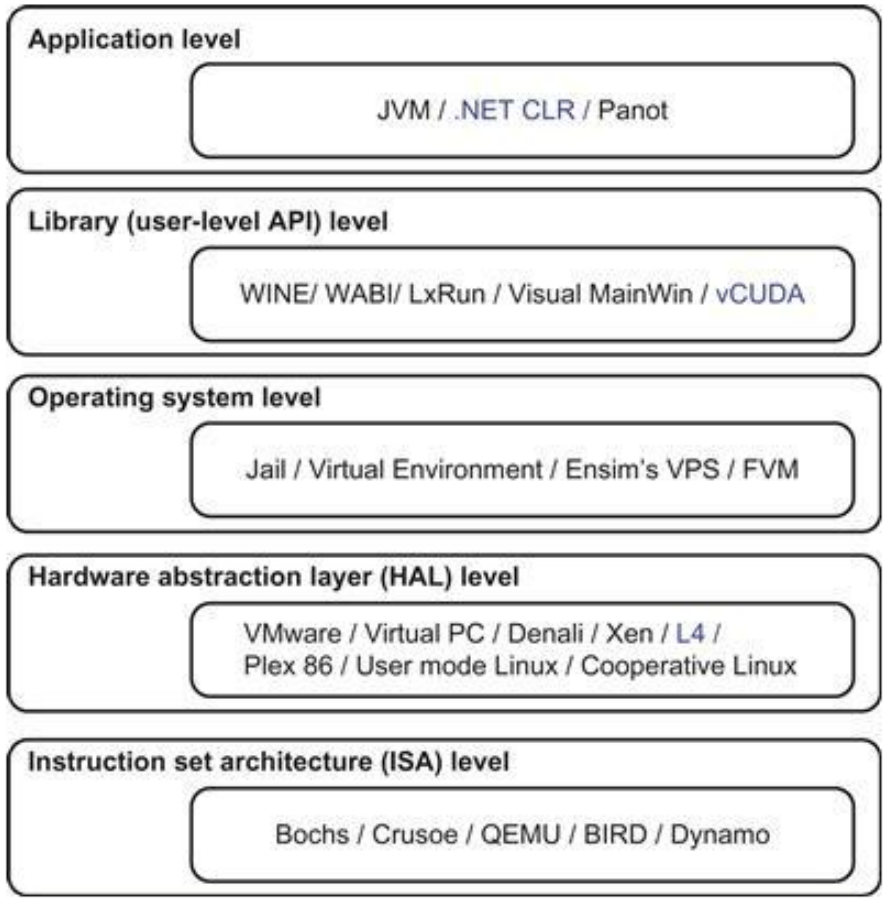
The virtualization software creates the abstraction of VM's by interposing a virtual layer at various levels of the computer system.

At the ISA level, virtualization is performed by emulating a given ISA on the ISA of the host machine.

For example, MIPS binary code can run on an x86-based host machine with the help of ISA emulation.

With this approach, it is possible to run a large amount of legacy binary code written for various processors on any given new hardware host machine.

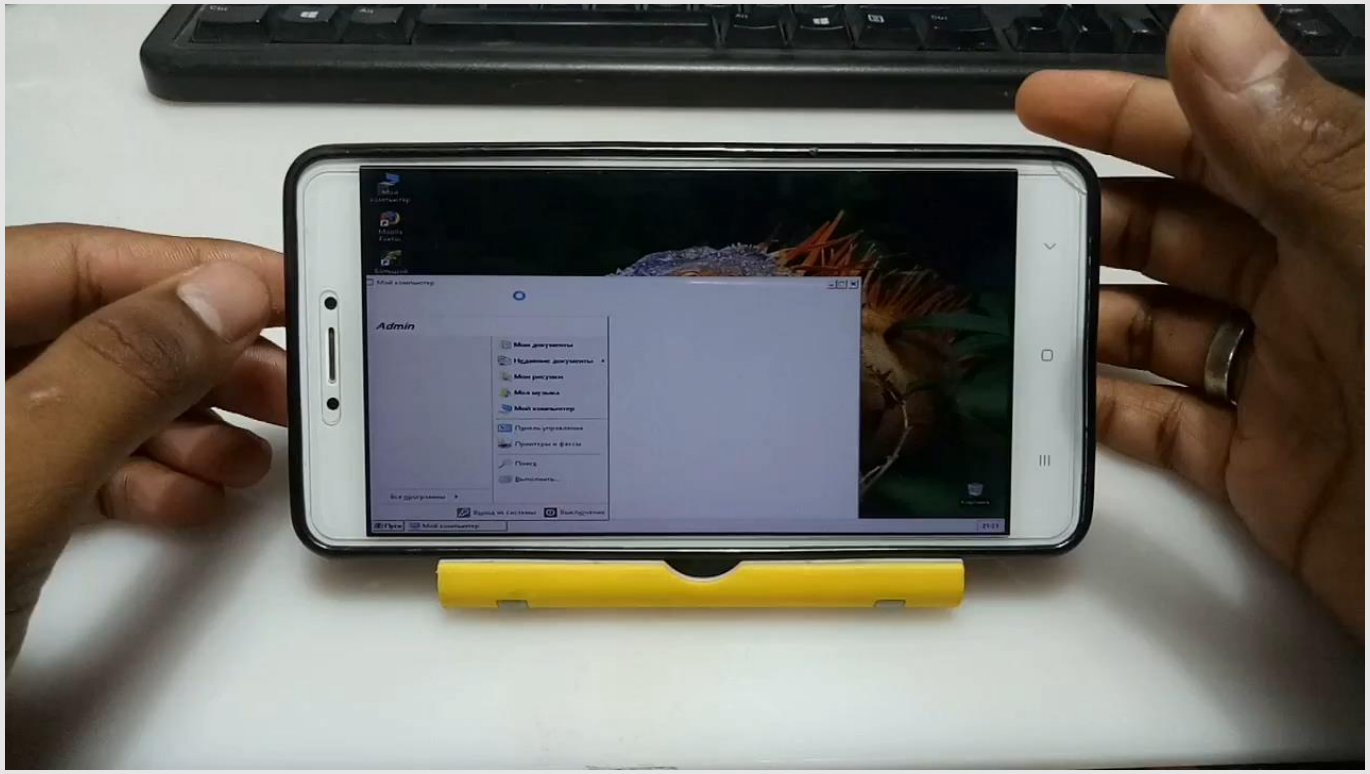
Instruction set emulation leads to virtual ISAs created on any hardware machine.



Hypervisor - Samples

- **BOCHS** :
  - Bochs is a portable IA-32 and x86-64 IBM PC compatible emulator and debugger mostly written in C++ and distributed as free software under the GNU Lesser General Public License.
  - It supports emulation of the processor, memory, disks, display, Ethernet, BIOS and common hardware peripherals of PCs.
- **BSD Jail** :
  - The jail mechanism is an implementation of FreeBSD's OS-level virtualisation that allows system administrators to partition a FreeBSD-derived computer system into several independent mini-systems called jails, all sharing the same kernel, with very little overhead.

Video – BOCHS

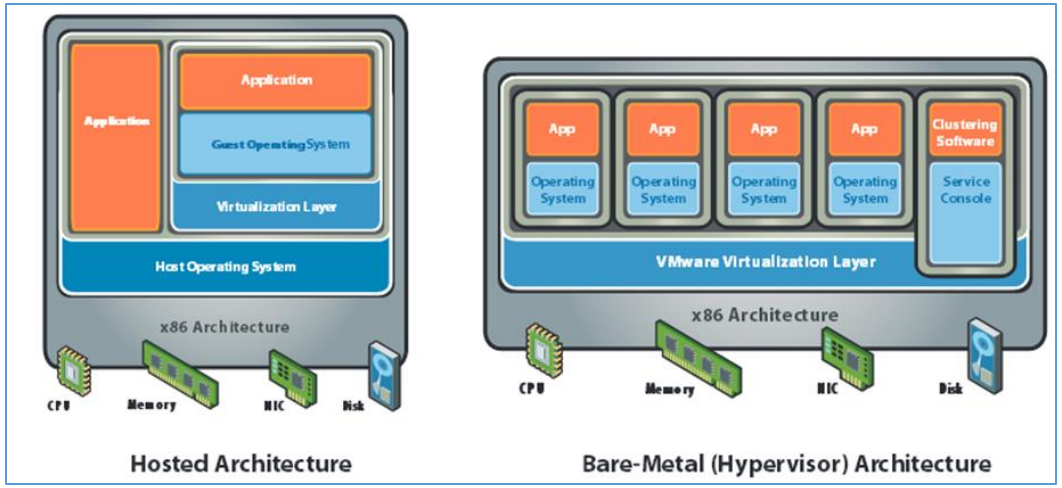


Learning Objectives

- Hypervisors can be used at any abstraction level.
- Oracle Virtual Box was an example of an hardware abstraction.
- We will see Bochs (pronounced Box), which is an ISA abstraction.
- It enables emulation to disparate Instruction sets

Hypervisor Types

- **Hosted**: A **hosted** architecture installs and runs the virtualization layer as an application on top of an operating system and supports the broadest range of hardware configurations. (VMware Player, ACE)
- **Bare Metal** : The architecture installs the virtualization layer directly on a clean x86-based system. Since it has direct access to the hardware resources rather than going through an operating system, a hypervisor is more efficient than a hosted architecture and delivers greater scalability, robustness and performance. (ESX Server)
- **Hybrid**: The architecture installs the VM layer directly on the hardware like a bare metal, but also leverages the features of the host OS. Xen and Microsoft's Hyper-V are examples of hybrid hypervisors



Hosted Architecture | Bare-Metal (Hypervisor) Architecture

Design Goals

- **Reliability**
  - Minimal code base
  - Strictly layered design
  - Not extensible
- **Isolation**
  - Security isolation
  - Fault isolation
  - Resource isolation
- **Scalability**
  - Scale to large number of cores
  - Large memory systems

Hypervisor Architecture

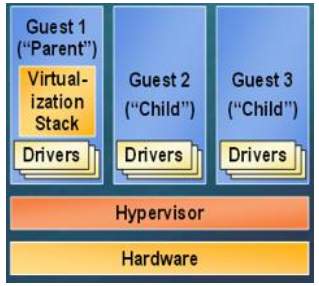
Monolithic hypervisor

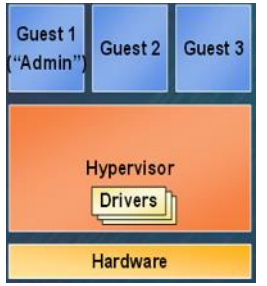
- Simpler than a modern kernel, but still complex
- Contains its own drivers model

Microkernel hypervisor

- Simple partitioning functionality
- Increase reliability and minimize lowest level of the TCB
- No third-party code
- Drivers run within guests

BASIS FOR COMPARISON	MICROKERNEL	MONOLITHIC KERNEL
Basic	In microkernel user services and kernel, services are kept in separate address space.	In monolithic kernel, both user services and kernel services are kept in the same address space.
Size	Microkernel are smaller in size.	Monolithic kernel is larger than microkernel.
Execution	Slow execution.	Fast execution.
Extendible	The microkernel is easily extendible.	The monolithic kernel is hard to extend.
Security	If a service crashes, it does effect on working of microkernel.	If a service crashes, the whole system crashes in monolithic kernel.
Code	To write a microkernel, more code is required.	To write a monolithic kernel, less code is required.
Example	QNX, Symbian, L4Linux, Singularity, K42, Mac OS X, Integrity, PikeOS, HURD, Minix, and Coyotos.	Linux, BSDs (FreeBSD, OpenBSD, NetBSD), Microsoft Windows (95,98,Me), Solaris, OS-9, AIX, HP-UX, DOS, OpenVMS, XTS-400 etc.



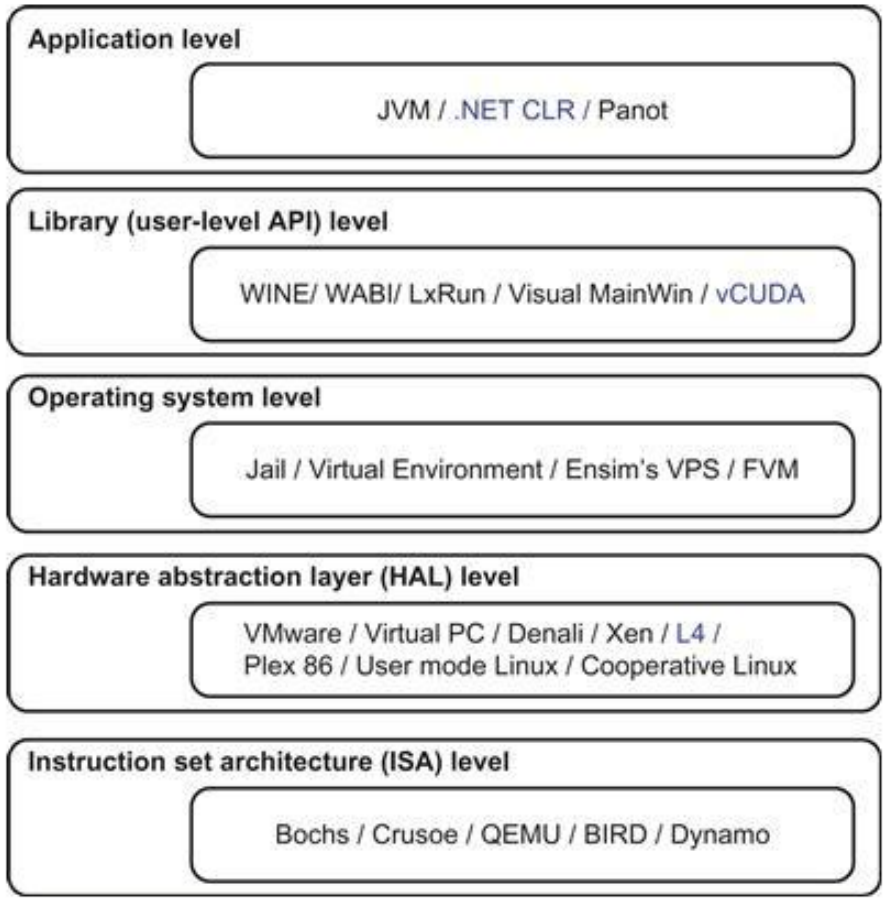


Comparison

Level of Implementation	Higher Performance	Application Flexibility	Implementation Complexity	Application Isolation
ISA	X	XXXXX	XXX	XXX
Hardware-level virtualization	XXXXX	XXX	XXXXX	XXXX
OS-level virtualization	XXXXX	XX	XXX	XX
Runtime library support	XXX	XX	XX	XX
User application level	XX	XX	XXXXX	XXXXX

The number of X's in the table cells **reflects the advantage points** of each implementation level. **Five X's implies the best case** and **one X implies the worst case**.

Overall, **hardware and OS support** will yield the **highest performance**. However, the hardware and application levels are also the most expensive to implement. User isolation is the most difficult to achieve. ISA implementation offers the best application flexibility.



Comparison

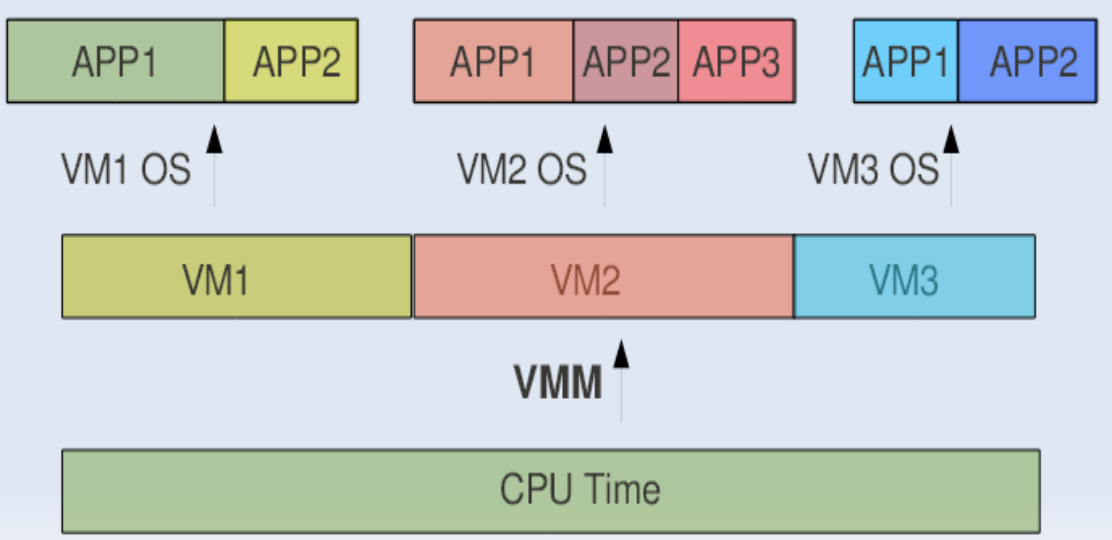
Aspect	Type 1 Hypervisor (Bare Metal)	Type 2 Hypervisor (Hosted)
Deployment	Installed directly on the physical hardware	Installed on top of a host operating system
Examples	VMware vSphere/ESXi, Microsoft Hyper-V, Xen	Oracle VirtualBox, VMware Workstation, Parallels Desktop
Performance	Generally higher performance due to direct access to hardware resources	Lower performance due to reliance on host OS for resource allocation
Resource Overhead	Minimal overhead as it operates directly on hardware	Higher overhead due to running within a host OS
Isolation	Better isolation between virtual machines (VMs)	Weaker isolation, as issues in host OS can affect VMs
Security	Generally more secure due to reduced attack surface	Less secure due to reliance on host OS security
Scalability	Typically more scalable for large deployments	Limited scalability compared to Type 1 hypervisors
Management	May require additional management tools	Often easier to manage with GUI interfaces
Disadvantages	• Requires dedicated hardware • Potentially higher upfront costs • More complex setup and management	• Performance overhead • Limited scalability • Reduced security compared to Type 1 hypervisors

Resource Sharing in VM - CPU

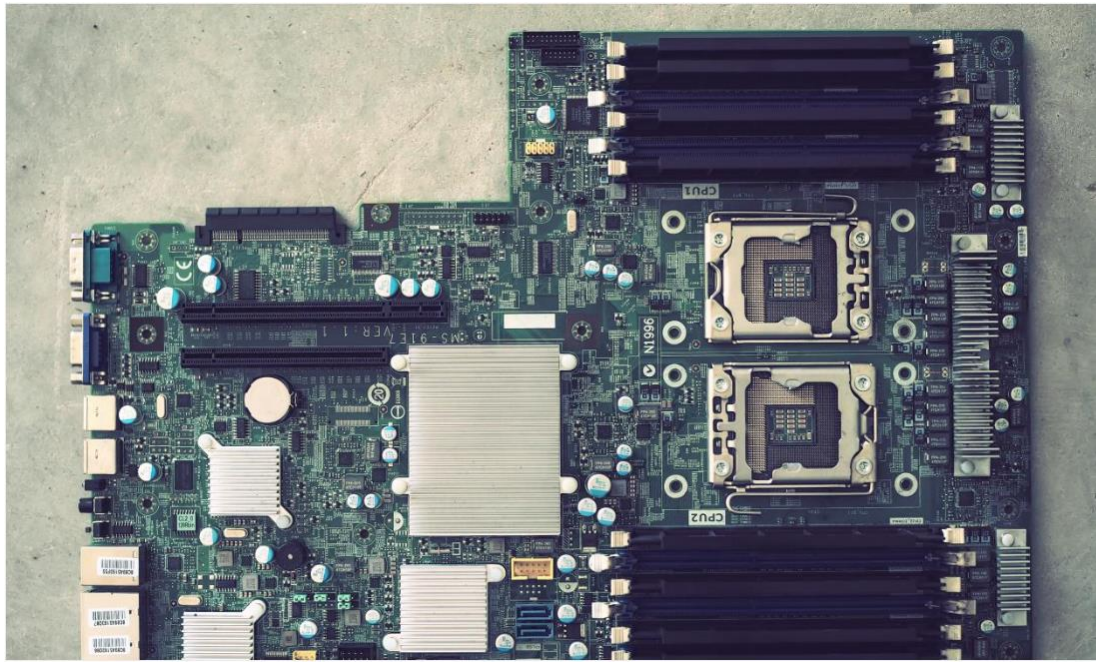
VMM or Hypervisor provides a virtual view of CPU to VMs.

In multi processing, CPU is allotted to the different processes in form of time slices by the OS.

Similarly VMM or Hypervisor allots CPU to different VMs.



Resource Sharing in VM - CPU



A CPU Socket is a physical connector on the motherboard to which a single physical CPU is connected.

A CPU (central processing unit, microprocessor chip, or processor) is a computer component. It is the electronic circuitry with transistors that is connected to a socket.

A CPU core is the part of a processor(CPU) containing the L1 cache. The CPU core performs computational tasks independently without interacting with other cores and external components of a "big" processor that are shared among cores. Basically, a core can be considered as a small processor built into the main processor that is connected to a socket. Applications should support parallel computations to use multicore processors rationally.

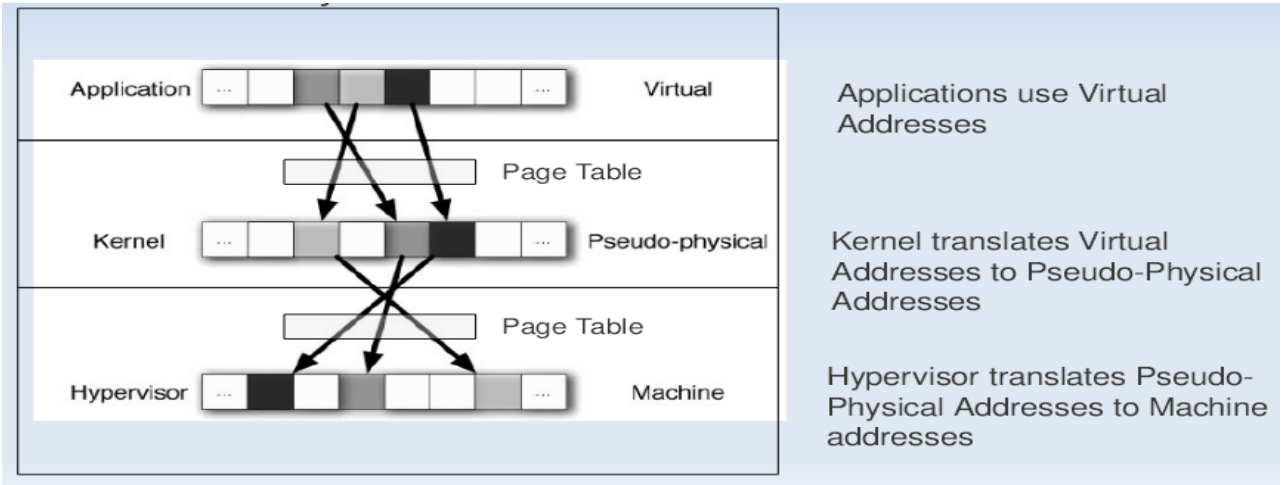
Hyper-threading is a technology developed by Intel engineers to bring parallel computation to processors that have one processor core. The debut of hyper-threading was in 2002 when the Pentium 4 HT processor was released and positioned for desktop computers. An operating system detects a single-core processor with hyper-threading as a processor with two logical cores (not physical cores). Similarly, a four-core processor with hyper-threading appears to an OS as a processor with 8 cores.

A vCPU is a virtual processor that is configured as a virtual device in the virtual hardware settings of a VM. A virtual processor can be configured to use multiple CPU cores. A vCPU is connected to a virtual socket.

Resource Sharing in VM - Memory

In Multiprogramming there is a single level of indirection maintained by Kernel.

In case of Virtual Machines there is one more level of indirection maintained by VMM



Memory sharing relies on the observation that several virtual machines might be running instances of the same guest operating system.

These virtual machines might have the same applications or components loaded, or contain common data.

In such cases, a host uses a proprietary Transparent Page Sharing (TPS) technique to eliminate redundant copies of memory pages.

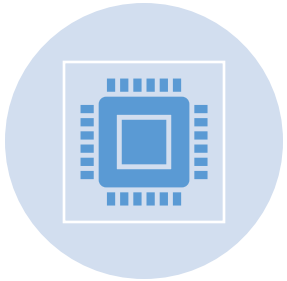
With memory sharing, a workload running on a virtual machine often consumes less memory than it might when running on physical machines.

As a result, higher levels of overcommitment can be supported efficiently.

The amount of memory saved by memory sharing depends on whether the workload consists of nearly identical machines which might free up more memory.

A more diverse workload might result in a lower percentage of memory savings.

Resource Sharing in VM - IO



Device needs to use Physical Memory location.

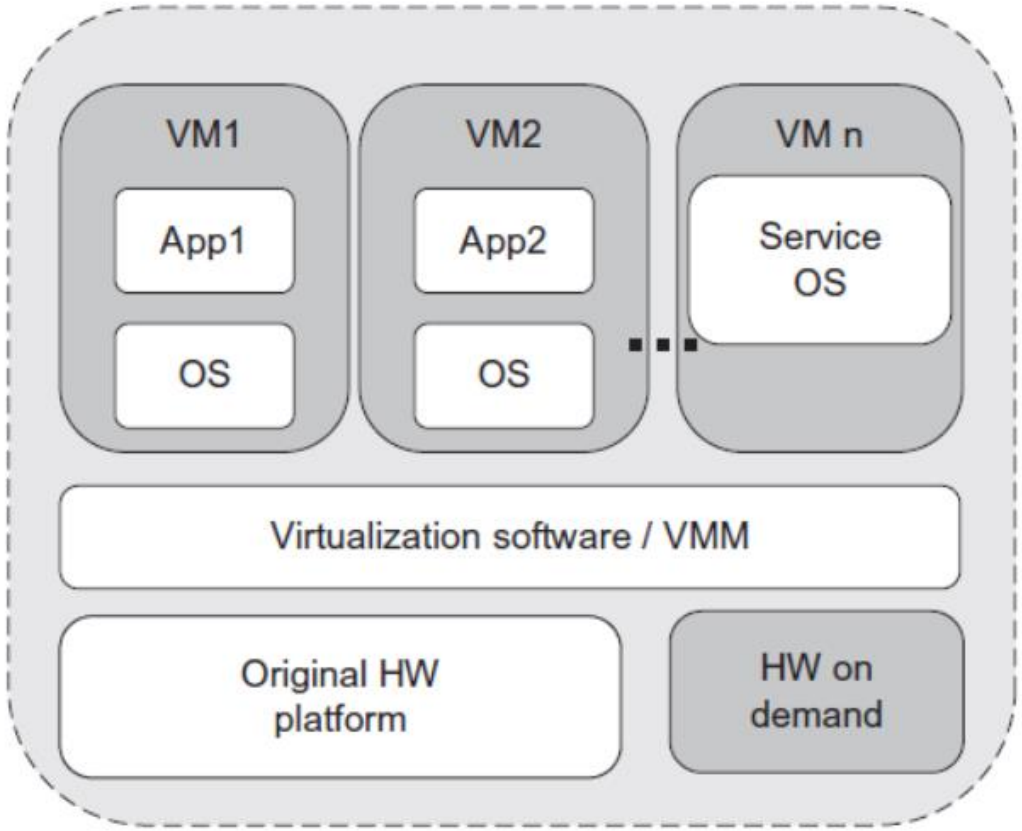
In a virtualized environment, the kernel is running in a hypervisor-provided virtual address space

Allowing the guest kernel to convey an arbitrary location to device for writing is a serious security hole

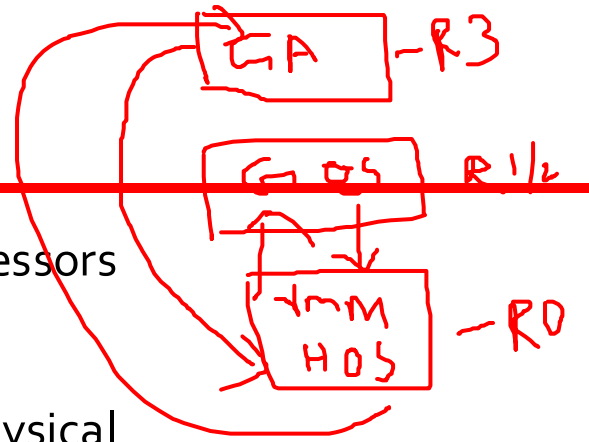
Each device defines its own protocol for talking to drivers

Hypervisor Techniques

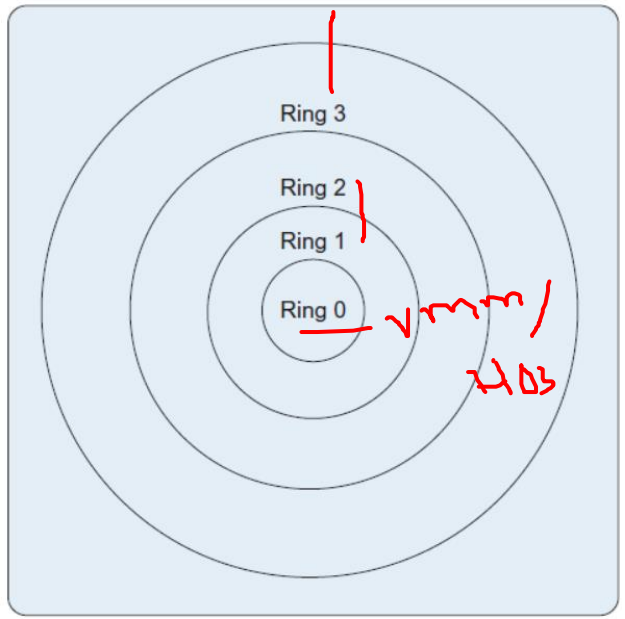
- At a very high level, all three types of hypervisors described earlier operate in a similar manner.
- In each case, the guests continue execution until they try to access a shared physical resource of the hardware (such as an I/O device), or an interrupt is received.
- When this happens, the hypervisor regains control and mediates access to the hardware, or handles the interrupt.



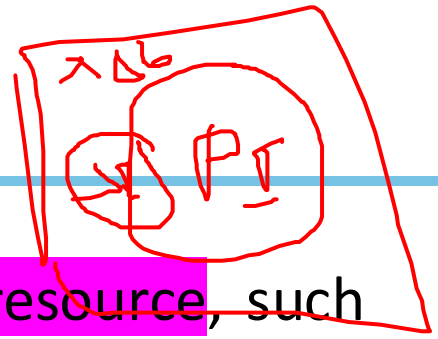
Hypervisor Techniques

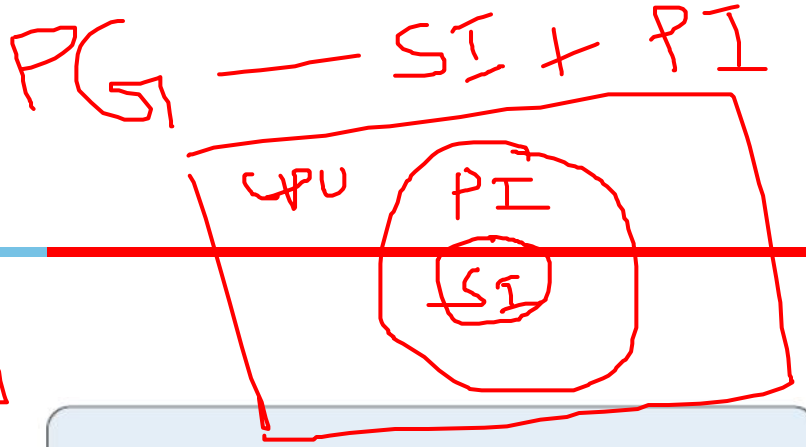


- To accomplish this functionality, hypervisors rely on a feature of modern processors known as the privilege level or protection ring.
- The basic idea behind privilege levels is that all instructions that modify the physical hardware configuration are permitted at the highest level,
- At lower levels, only restricted sets of instructions can be executed.
- There are four rings, numbered from 0 to 3.
- Programs executing in Ring 0 have the highest privileges, and are allowed to execute any instructions or access any physical resources such as memory pages or I/O devices.
- Guests are typically made to execute in ring 3. This is accomplished by setting the Current Privilege Level (CPL) register of the processor to 3 before starting execution of the guest.

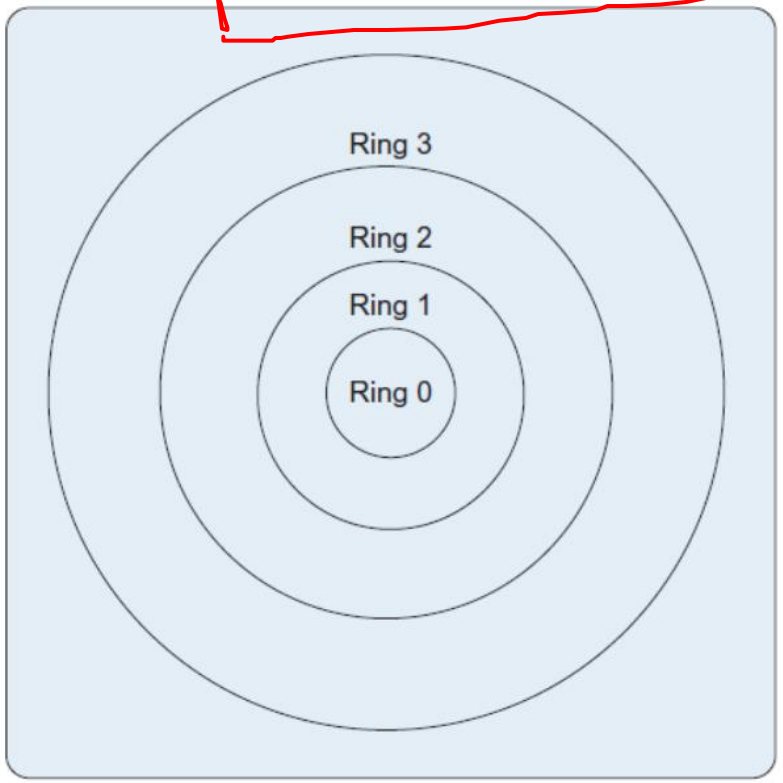


Hypervisor Techniques





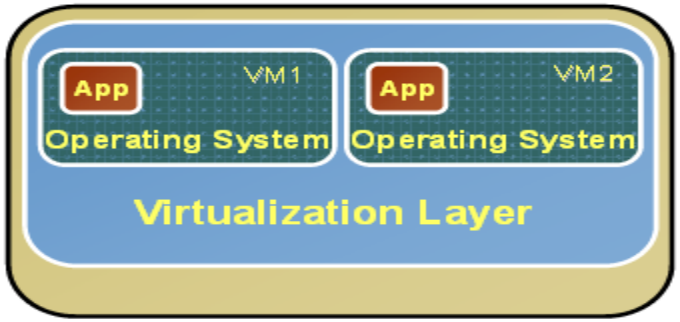
- If the guest tries to access a protected resource, such as an I/O device, an interrupt takes place, and the hypervisor regains control.
- The hypervisor then emulates the I/O operation for the guest.
- The exact details depend upon the particular hypervisor (e.g., Xen or Hyper-V).
- Note that in order to emulate the I/O operation, it is necessary for the hypervisor to have **maintained the state of the guest and its virtual resources**



Benefits of Virtualization

- Single OS image per machine
- Software and hardware tightly coupled
- Running multiple applications on same machine often creates conflict
- Underutilized resources
- Inflexible and costly infrastructure

- Hardware-independence of operating system and applications
- Virtual machines can be provisioned to any system
- Can manage OS and application as a single unit by encapsulating them into virtual machines



Virtualization Summary

•Virtualization allows multiple operating system instances to run concurrently on a single computer. It is a means of separating hardware from a single operating system.

•Each "guest" OS is managed by a Virtual Machine Monitor (VMM), also known as a hypervisor.

•Because the virtualization system sits between the guest and the hardware, it can control the guests' use of CPU, memory, and storage, even allowing a guest OS to migrate from one machine to another.

•Instead of purchasing and maintaining an entire computer for one application, each application can be given its own operating system, and all those operating systems can reside on a single piece of hardware.

•Virtualization allows an operator to control a guest operating system's use of CPU, memory, storage, and other resources, so each guest receives only the resources that it needs.

Key Terms to Remember

Key Terms:

VM : Virtual Machine

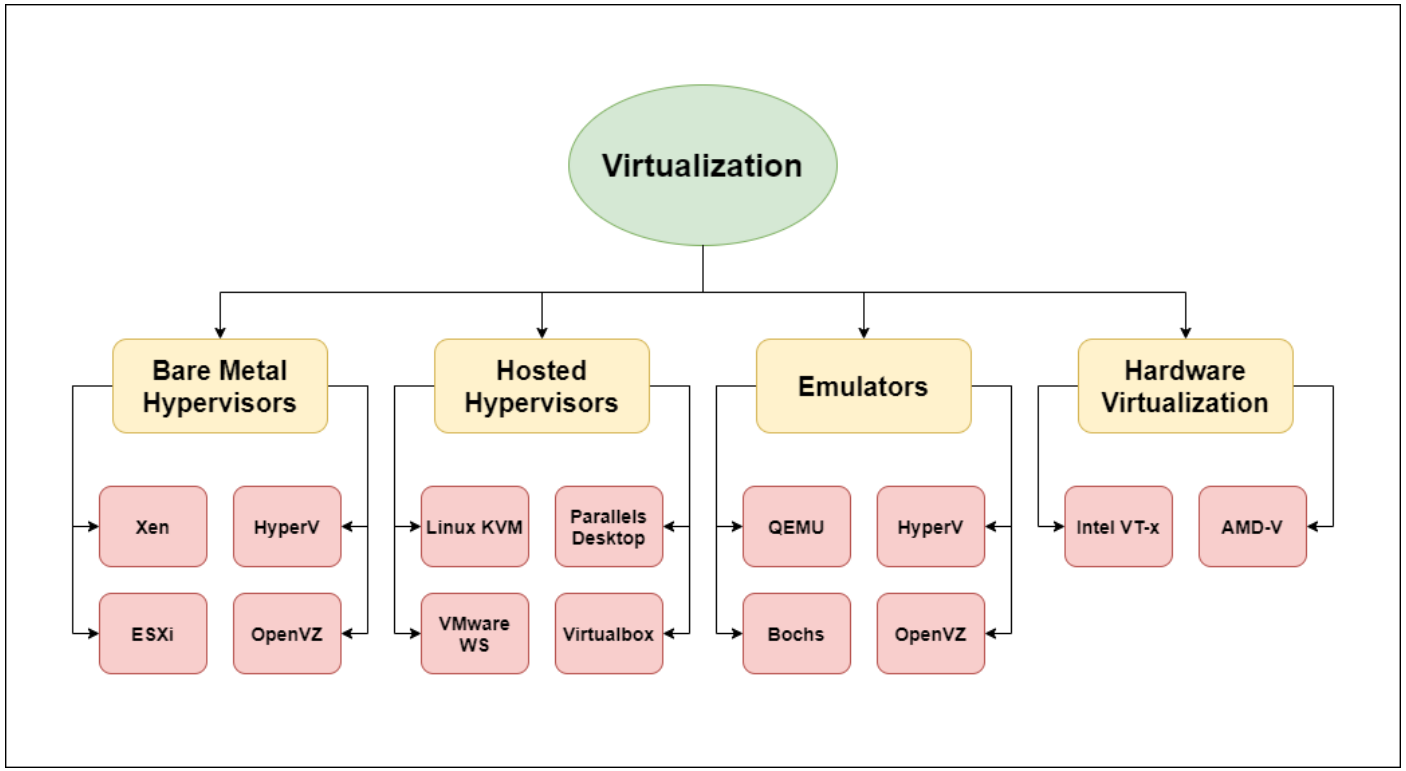
VMM: Virtual Machine Monitor

Hypervisor : VMM

Multiplexed: Many or several

Host: System where the VMM resides

Guest : Virtual Machines created



Q & A……..

BITS Pilani

Pilani|Dubai|Goa|Hyderabad

Credits

•Hwang, Kai; Dongarra, Jack; Fox, Geoffrey C.. Distributed and Cloud Computing: From Parallel Processing to the Internet of Things (Kindle Locations 3532-3533). Elsevier Science. Kindle Edition.