BITS Pilani

# Cloud Computing

## CC ZG527

Aditya Goel

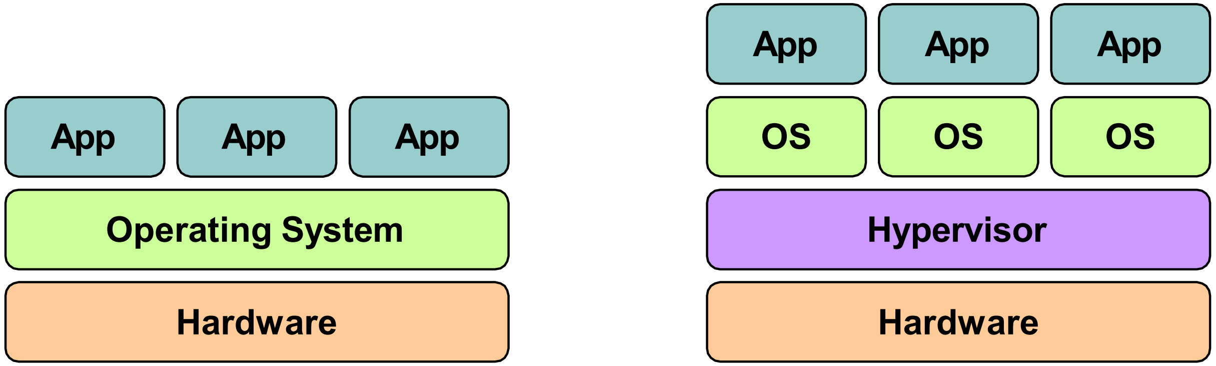
# Agenda

Virtualization Techniques and Types

- ❑ Introduction to Virtualization
- ❑ Use & demerits of Virtualization
- ❑ Types of Virtualization
  - Examples
- ❑ Types of Hypervisors

# Technology made cloud possible

## Key Technology is Virtualization



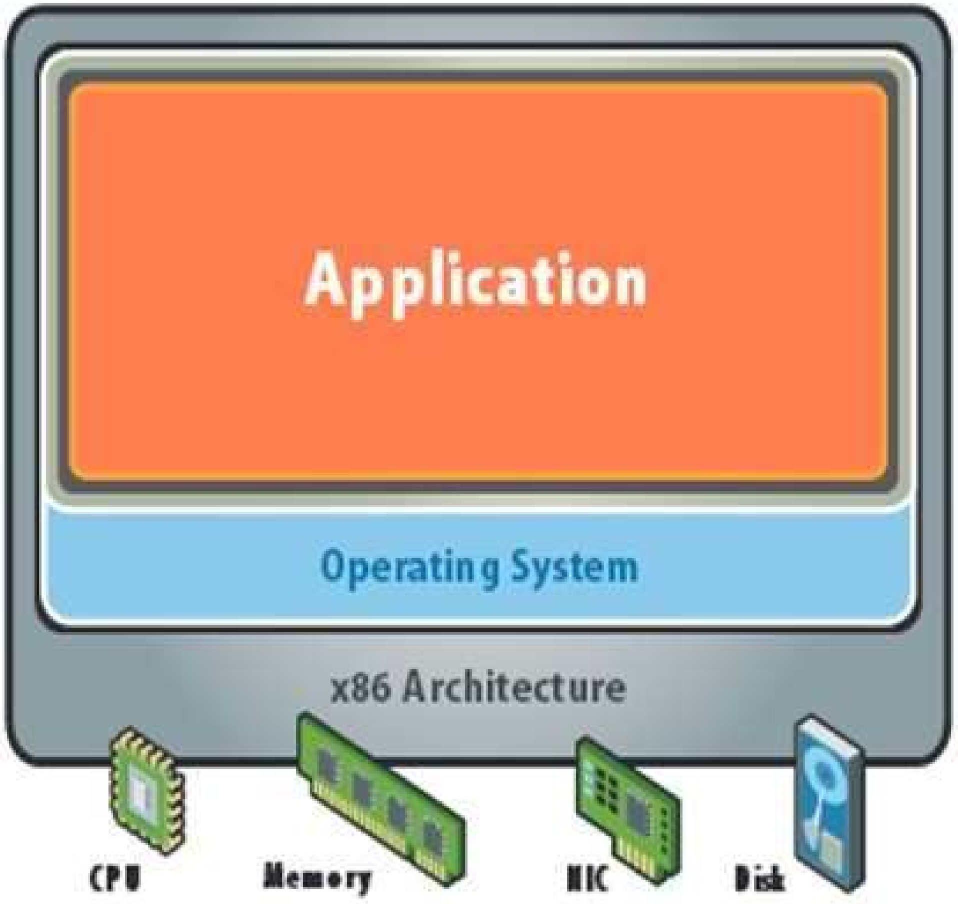

Virtualization plays an important role:

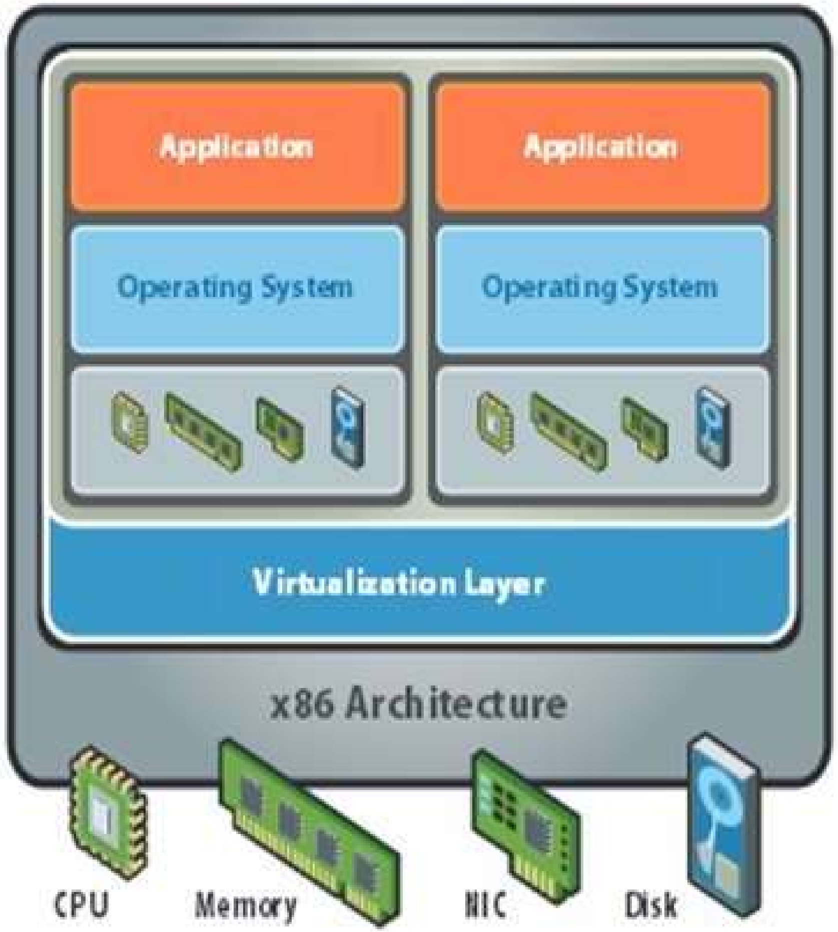
- as an enabling technology for datacentre implementation
- abstract compute, network, and storage service platforms from the underlying physical hardware

# Importance of Virtualization in Cloud Computing

- 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

# What is Virtualization?



# What does Virtualization do?

- Virtualization allows multiple operating system instances to run concurrently on a single computer
- Each "guest" OS is managed by a Virtual Machine Monitor (VMM), also known as a hypervisor.
- Because the virtualization layer sits between the guest and the hardware,
  - it can control the guests' use of CPU, memory, and storage
  - allows a guest OS to migrate from one machine to another

# Changes after Virtualization

**Before Virtualization**

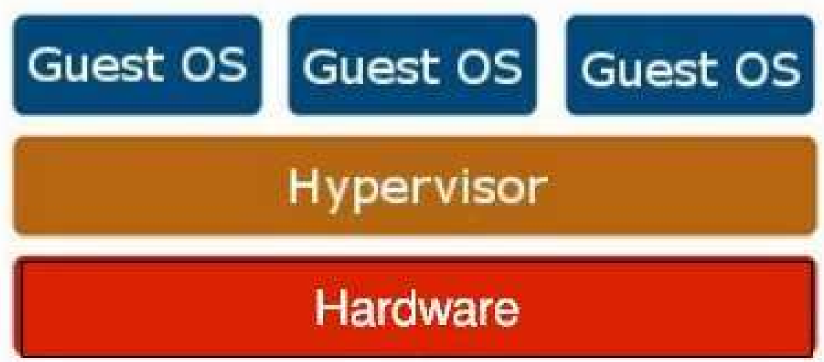
- Single OS image per machine
- Software and hardware tightly coupled
- Running multiple applications on same machine often creates conflict
- Underutilized resources
- Inflexible and costly infrastructure

**After Virtualization**

- 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 Architecture

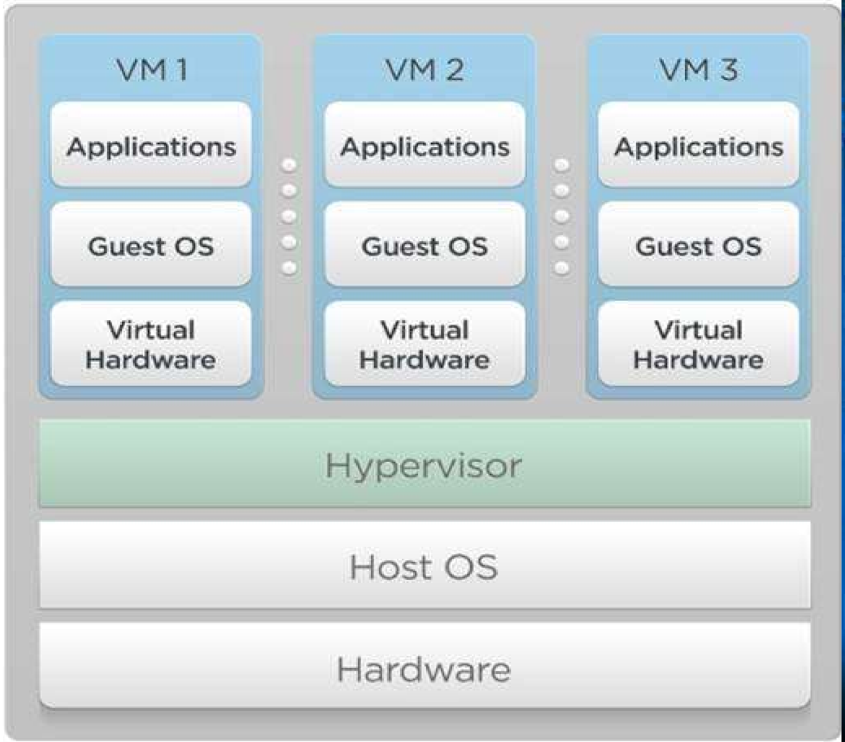
- OS assumes complete control of the underlying hardware.
- Virtualization architecture provides this illusion through a hypervisor/VMM.
- Hypervisor/VMM is a software layer which:
  - Allows multiple Guest OS (Virtual Machines) to run simultaneously on a single physical host
  - Provides hardware abstraction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

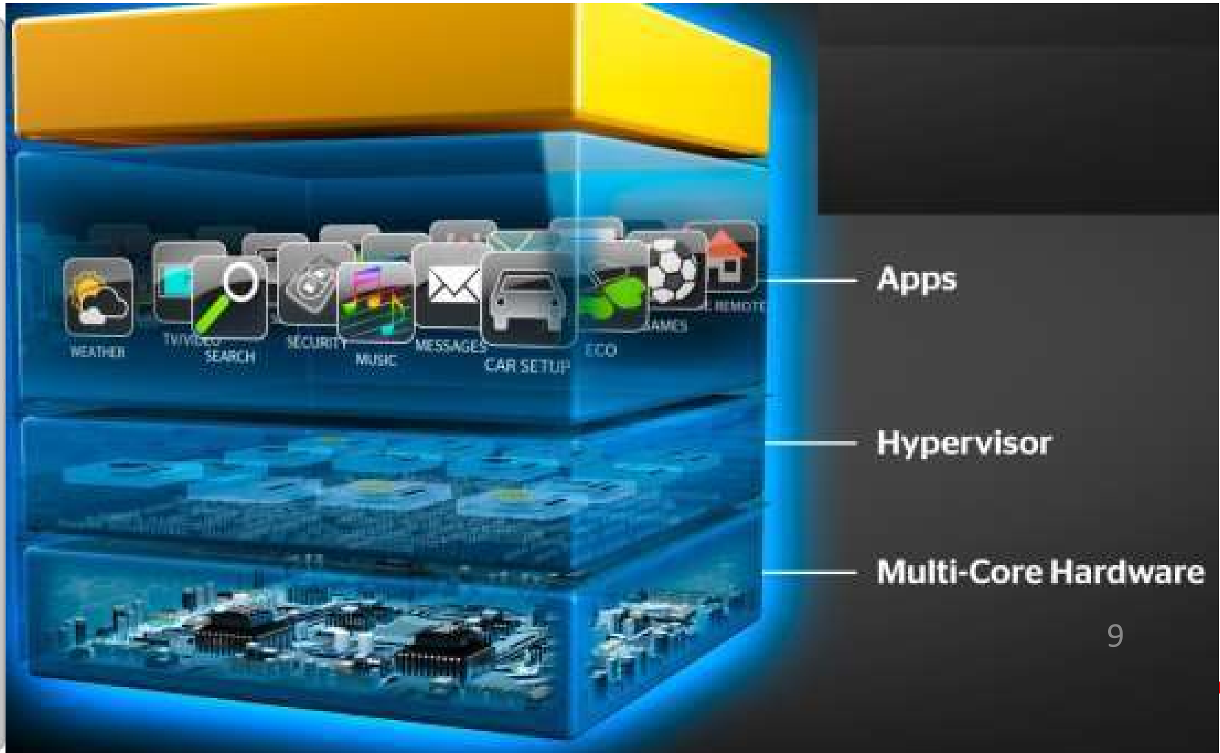


# Hypervisor

A layer of software that generally provides virtual partitioning capabilities which runs directly on hardware.

Sometimes referred to as a "bare metal" approach.



# Hypervisor Design Goals

- Isolation
  - Security isolation
  - Fault isolation
  - Resource isolation
- Reliability
  - Minimal code base
  - Strictly layered design
- Scalability
  - Scale to large number of cores
  - Large memory systems

# How Hypervisor goals are achieved?

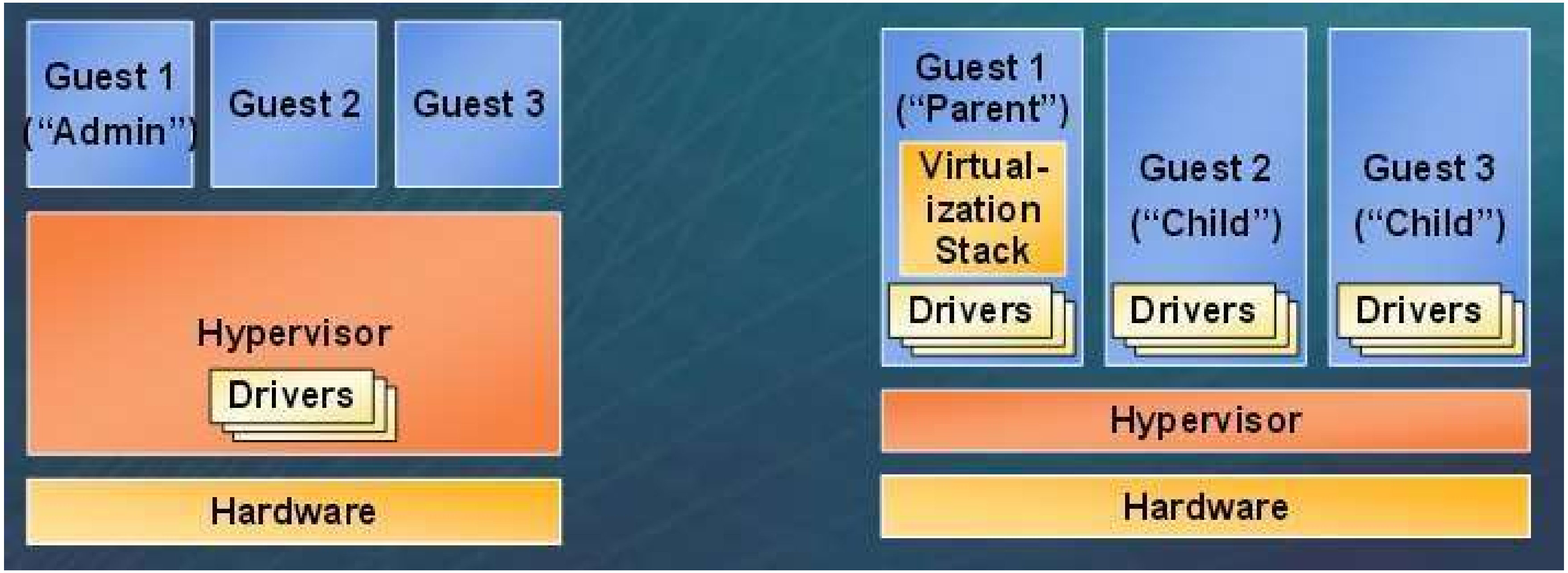
- Partitioning Kernel
  - "Partition" is isolation boundary
  - Few virtualization functions; relies on virtualization stack
- Very thin layer of software
  - Microkernel
  - Highly reliable
  - Basis for smaller Trusted Computing Base (TCB)
- No device drivers
  - Drivers run in a partition
- Well-defined interface
  - Allow others to create support for their OSes as guests

# Hypervisor
## Monolithic versus Microkernelized

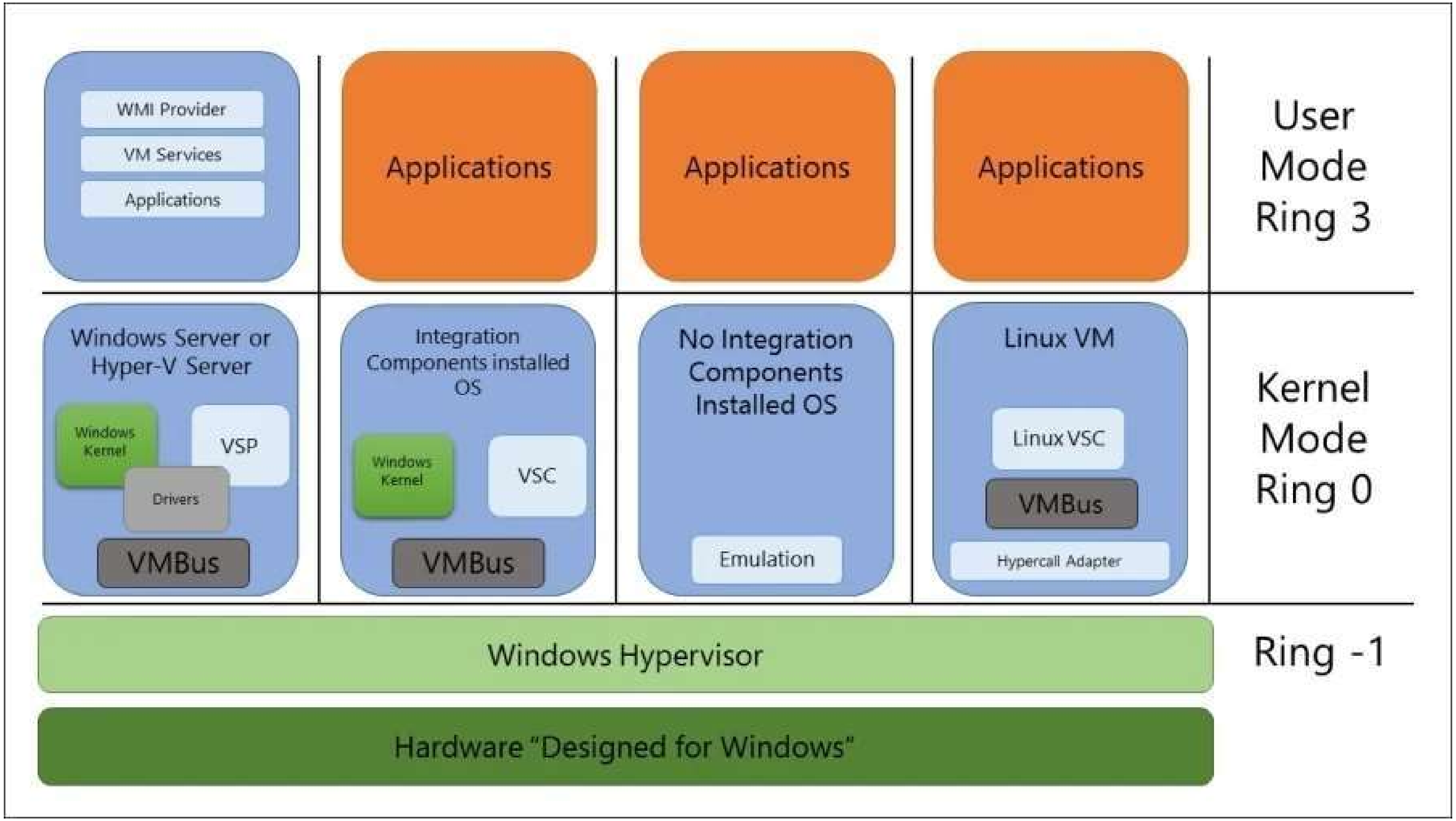
- Monolithic hypervisor
  - Simpler than a modern kernel, but still complex
  - Contains its own drivers

model

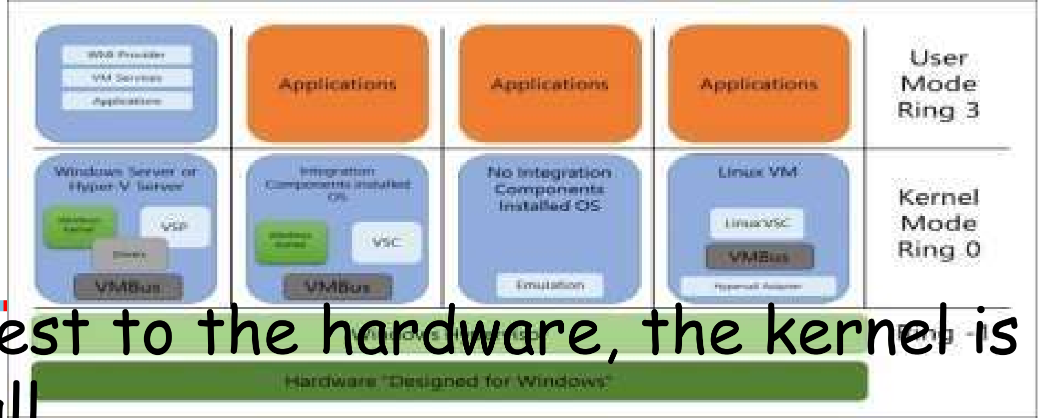
- Microkernelized hypervisor
  - Simple partitioning functionality
  - Increase reliability and minimize lowest level of the TCB
  - No third-party code
  - Drivers run within guests



# Hyper-V as a Microkernel Type 1 Hypervisor



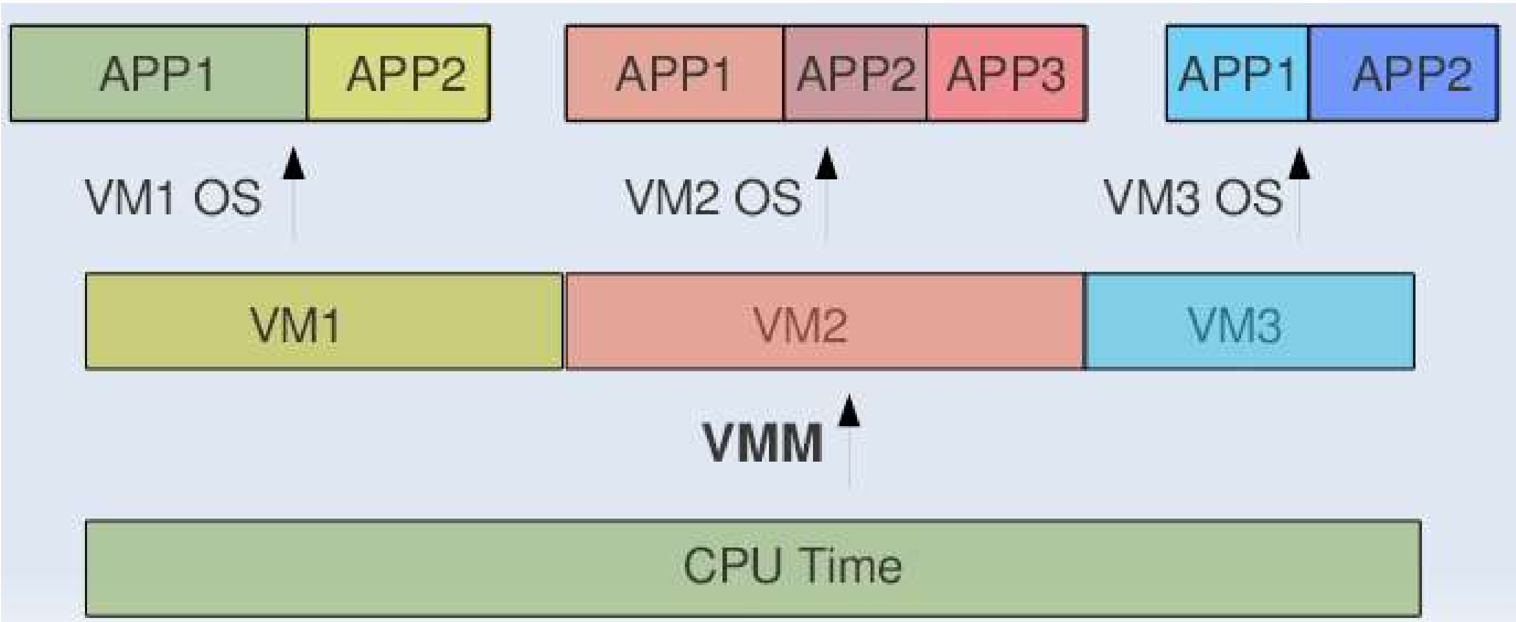
# Hyper-V as a Microkernel Type 1 Hypervisor




1. Application tries to send a request to the hardware, the kernel is responsible for interpreting this call.
2. As this OS is running on an **Enlightened Child Partition** (Means that IC is installed), the Kernel will send this call to the **Virtual Service Client (VSC)** that operates as a synthetic device driver.
3. The VSC is responsible for communicating with the **Virtual Service Provider (VSP)** on the parent partition, through VMBus
4. The VMBus will then be able to communicate with the hardware for the VM. The VMBus, a channel-based communication, is actually responsible for communicating with the parent partition and hardware.
5. VMBus to access the hardware, it will communicate directly with a component on the Hypervisor called **hypercalls**.
6. The hypercalls are then redirected to the hardware. Only the parent partition can actually access the physical processor and memory.

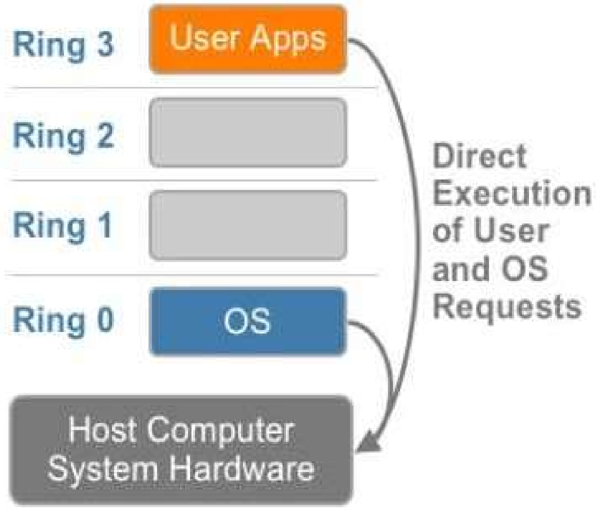
# CPU Virtualization

- 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.



# CPU Virtualization

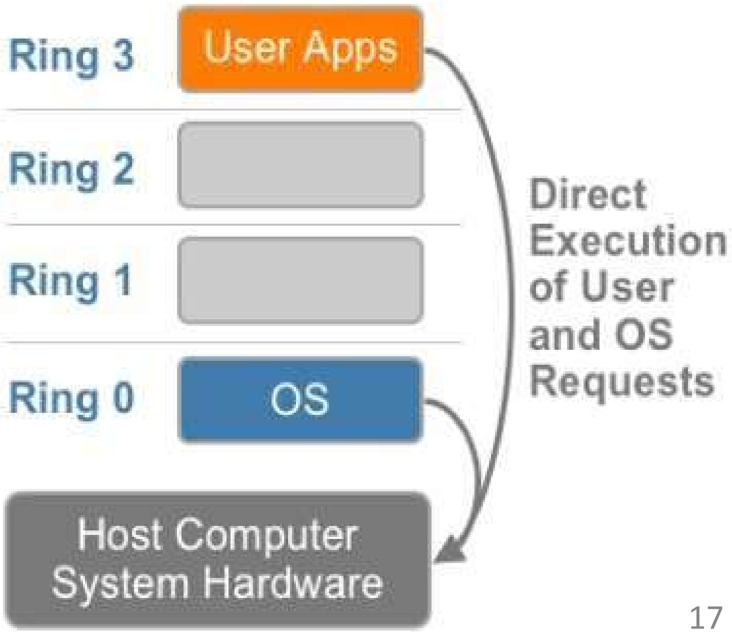
- x86 architecture offers four levels of privilege known as Ring 0, 1, 2 and 3 to operating systems and applications to manage access to the computer hardware
- Virtualizing the x86 architecture requires placing a virtualization layer under the operating system (which expects to be in the most privileged Ring 0)
- Some sensitive instructions can't effectively be virtualized as they have different semantics when they are not executed in Ring 0



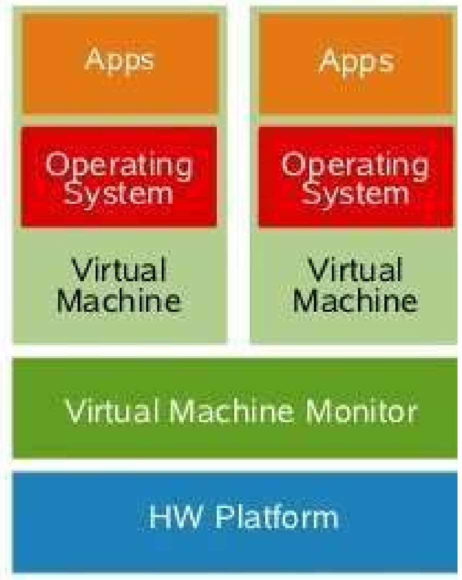
# Approaches to CPU Virtualization

Three techniques now exist for handling sensitive and privileged instructions to virtualize the CPU on the x86 architecture:

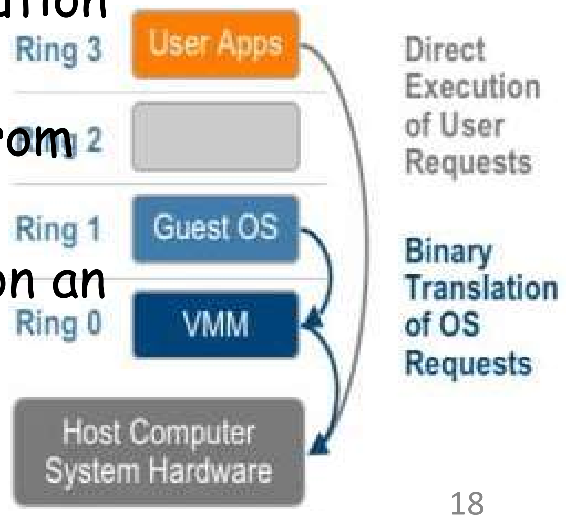
- Full virtualization using binary translation
- OS assisted virtualization or para-virtualization
- Hardware assisted virtualization



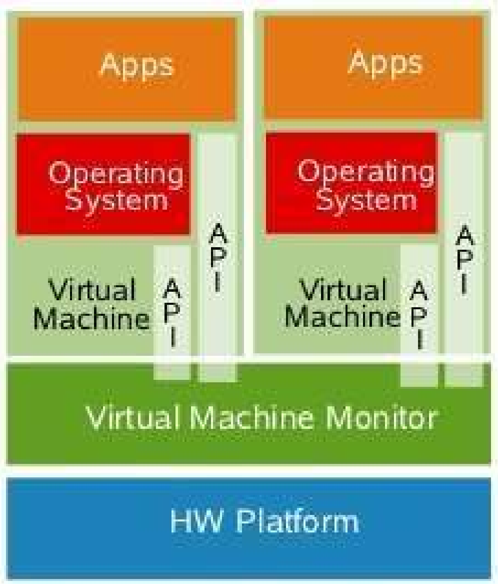
# Full Virtualization using Binary Translation




- Vmware(Full Virtualization) can virtualize any x86 operating system using a combination of
  - binary translation
  - direct execution techniques
- User level code is directly executed on the processor for high performance virtualization
- Kernel code is translated to replace non-virtualizable instructions with new sequences of instructions
- This combination of binary translation and direct execution provides Full Virtualization
- Guest OS is fully abstracted (completely decoupled) from the underlying hardware by the virtualization layer
- Flexibility: one could run a RISC-based OS as a guest on an Intel-based host.



# ParaVirtualization



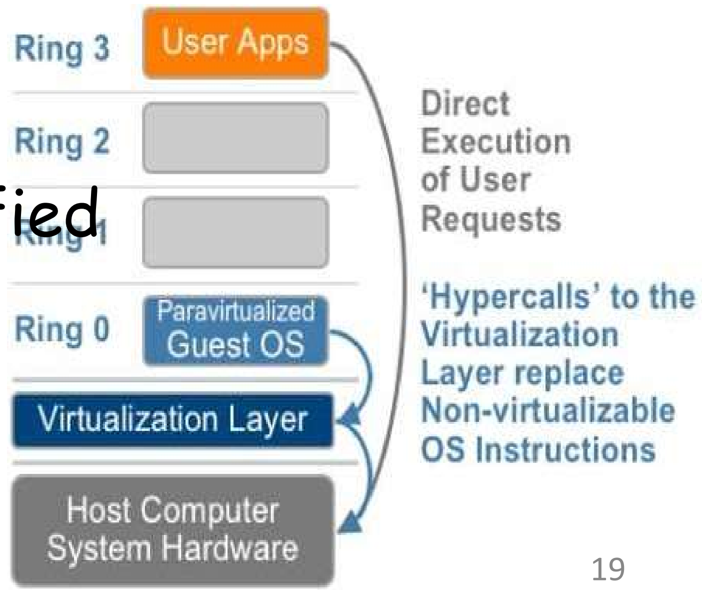

- Paravirtualization involves modifying the OS kernel to replace non-virtualizable instructions with hypercalls
- Hypercalls communicate directly with the virtualization layer
- The hypervisor also provides hypercall interfaces for other critical kernel operations such as: .

.memory management

.interrupt handling

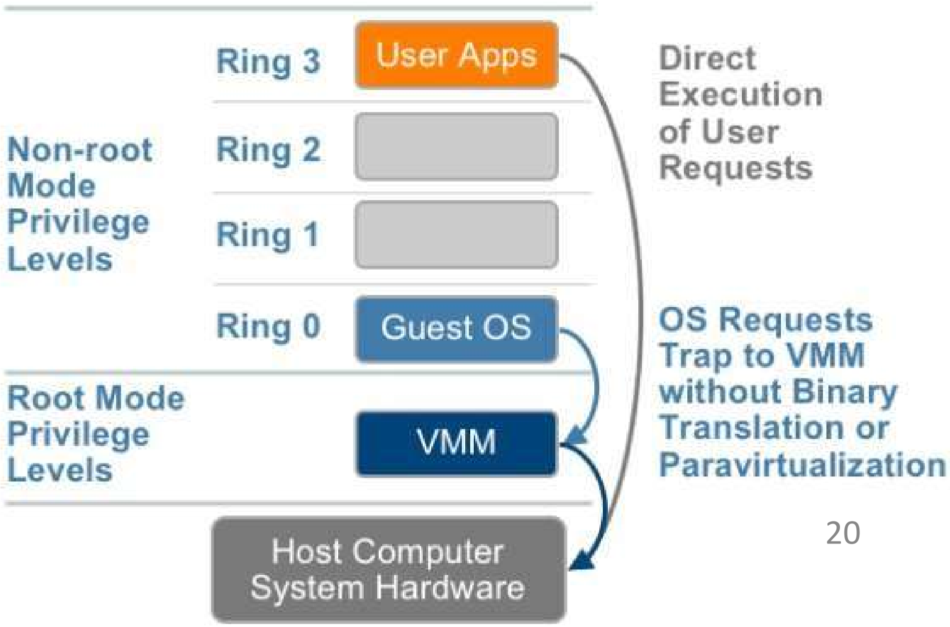
.time keeping

- Paravirtualization cannot support unmodified operating systems
- Compatibility and portability is poor
- Ex: open-source Xen



# Hardware Assisted Virtualization

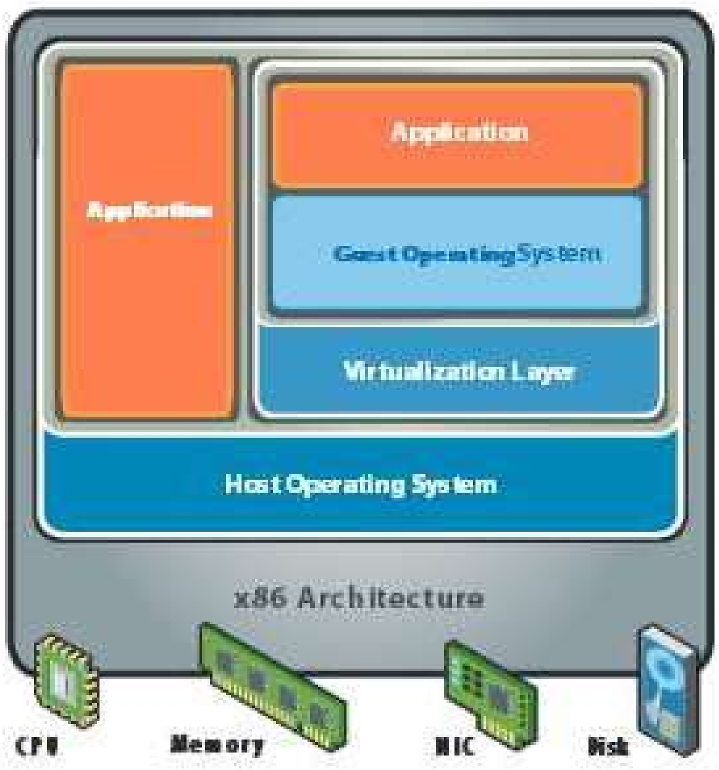
- Intel Virtualization Technology (VT-x) and AMD's AMD-V which both target privileged instructions with a new CPU execution mode feature that allows the VMM to run in a new root mode below ring 0
- Privileged and sensitive calls are set to automatically trap to the hypervisor, removing the need for either binary translation or paravirtualization.
- Underlying hardware provides special CPU instructions to aid virtualization



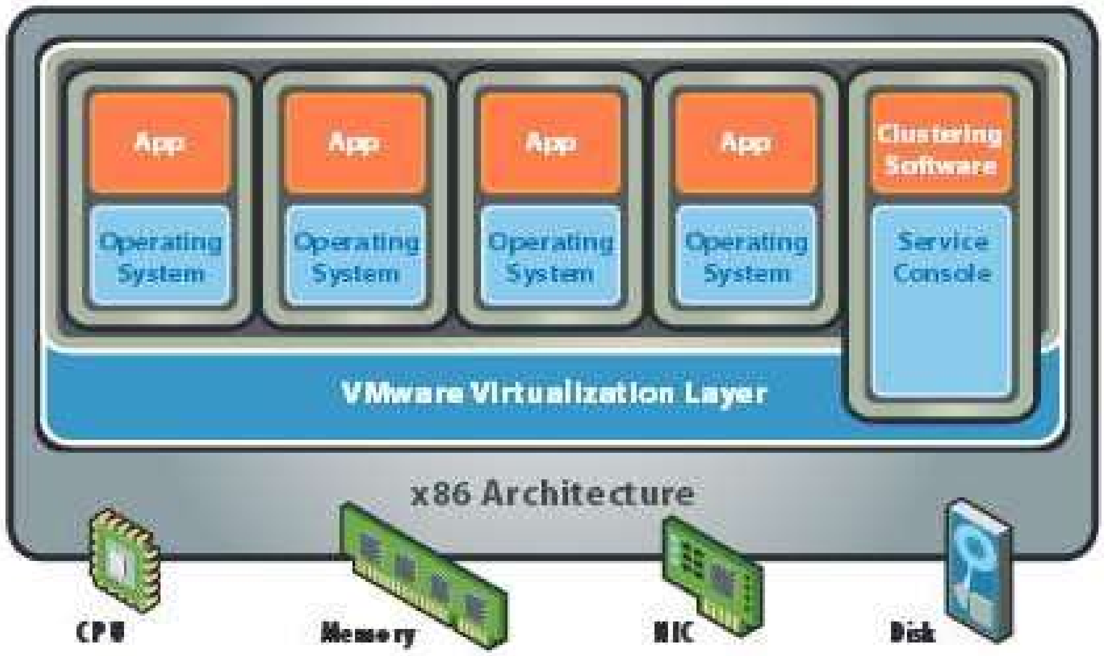
# Types of Hypervisors

- For Industry-standard x86 systems, the two approaches typically used with software-based partitioning are
  - Hosted
  - Bare-metal architectures
- A hosted approach provides partitioning services on top of a standard operating system and supports the broadest range of hardware configurations.
- In contrast, a hypervisor architecture is the layer of software installed on a clean x86-based system (hence it is often referred to as a "bare metal" approach).
  - Since it has direct access to the hardware resources, a hypervisor is more efficient than hosted architectures, enabling greater scalability, robustness and performance

# x86 Hardware Virtualization



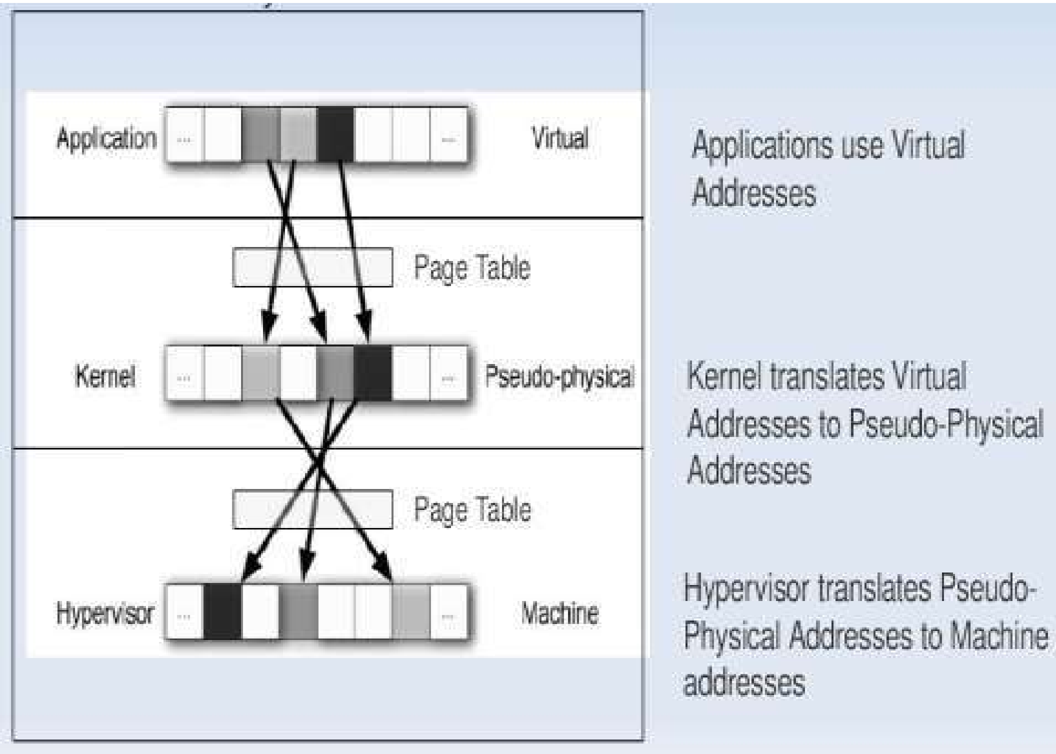

Hosted Architecture

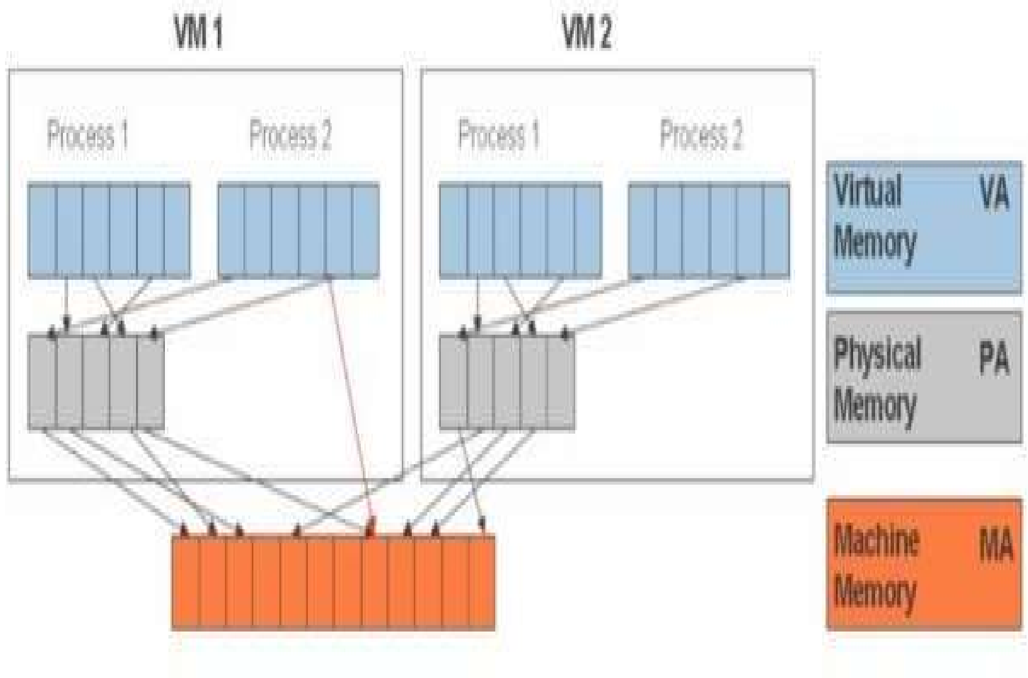



Bare-Metal (Hypervisor) Architecture

# Memory Virtualization

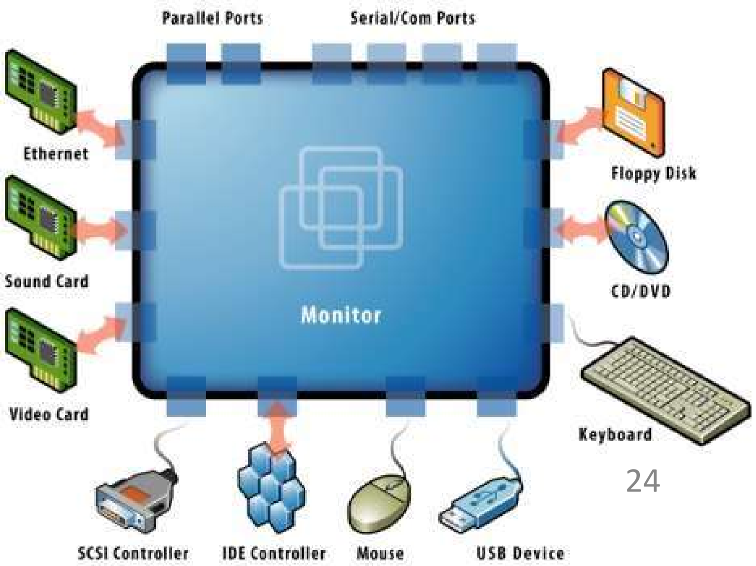
- In Multiprogramming there is a single level of indirection maintained by Kernel.
- In case of Virtual Machines there is one more level of indirection maintained by VMM



# Device and IO Virtualization

- This involves managing the routing of I/O requests between virtual devices and the shared physical hardware
- Virtual NICs and switches create virtual networks between virtual machines without the network traffic consuming bandwidth on the physical network
- The hypervisor virtualizes the physical hardware and presents each virtual machine with a standardized set of virtual devices
- These virtual devices effectively emulate well-known hardware and translate the virtual machine requests to the system hardware



# Advantages of Virtualization

- Instant provisioning - fast scalability
- Live Migration is possible
- Load balancing and consolidation in a Data Center is possible.
- Low downtime for maintenance
- Virtual hardware supports legacy operating systems efficiently
- Security and fault isolation

# Issues to be aware of

- Software licensing
- IT training
- Hardware investment

# Issues(technical) to be aware of

- Performance can be a concern, especially for in-band deployments, where the virtualization controller or appliance can become a bandwidth bottleneck.
- Interoperability among vendor products is still evolving.
- Failure of the virtualization device, leading to loss of the mapping table.

# Summary

- Virtualization is Key to Exploiting Trends
- Allows most efficient use of the compute resources
  - Few apps take advantage of 16+ CPUs and huge memory as well as virtualization
  - Virtualization layer worries about NUMA, not apps
- Maximize performance per watt across all servers
  - Run VMs on minimal # of servers, shutting off the others
  - Automated, live migration critical:
    - Provide performance guarantees for dynamic workloads
    - Balance load to minimize number of active servers
- Stateless, Run-anywhere Capabilities
  - Shared network and storage allows flexible mappings
  - Enables additional availability guarantees

End of Session 3