Server sprawl unnecessary eats the capitals

Server Virtualization

Before that, First we need to focus on term server sprawl

Assistant. Prof. Ashwini Mathur

Module - 2

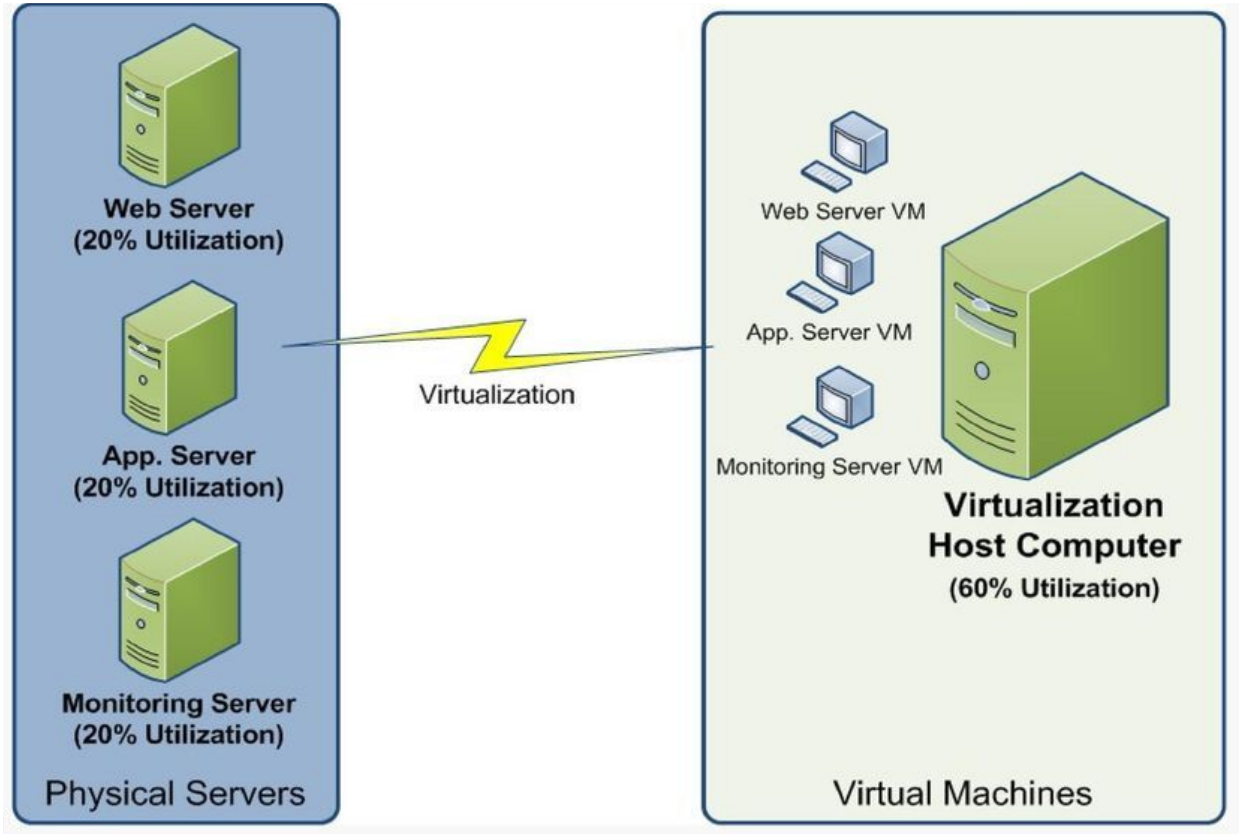
Server Sprawl

Server sprawl can be defined as a situation in which IT resources including hardware and software applications housed in a data center remains under-utilized, leading to poor productivity and proficiency.

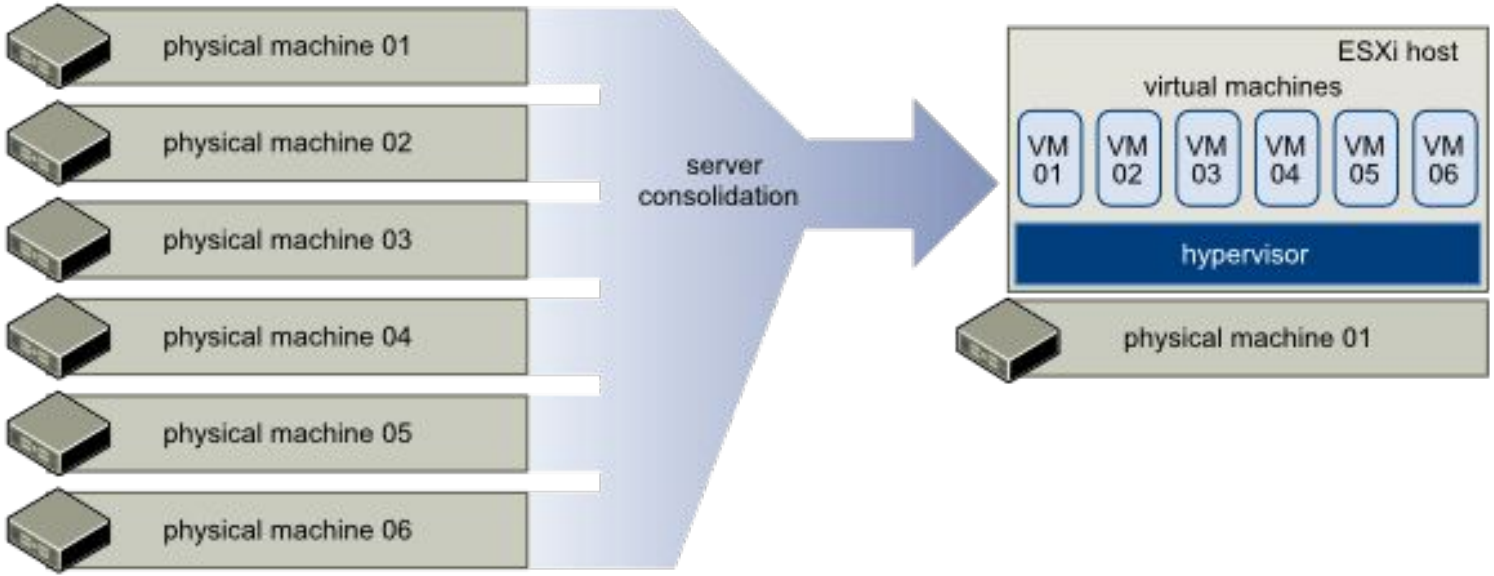
Example of Server sprawl

The situation of Server Sprawl usually occurs where an organization acquires large number of servers than it should depending on their present and future growth. Server system sprawl condition occurs due to the uncoordinated planning.

Say, an organization has 5 active servers, even though it only need two to properly run its IT operations. The remaining three servers could either be under-utilized or not utilized at all. While having some additional resource pays off with greater flexibility, too much extra resource can get really expensive. Even if the organization says it is utilizing all its servers, the servers are certainly running at extremely low capacity -10% or less. Notwithstanding the servers are using only 10% space, they are consuming almost the same power a utilized server would use. Thus, it is essential to deal with the problem of **Server Sprawl**.


Web Server
(20% Utilization)
App. Server
(20% Utilization)
Monitoring Server
(20% Utilization)
Physical Servers
Virtualization
Web Server VM
App. Server VM
Monitoring Server VM
Virtualization
Host Computer
(60% Utilization)
Virtual Machines

Solution of the Server Sprawl: Server Consolidation



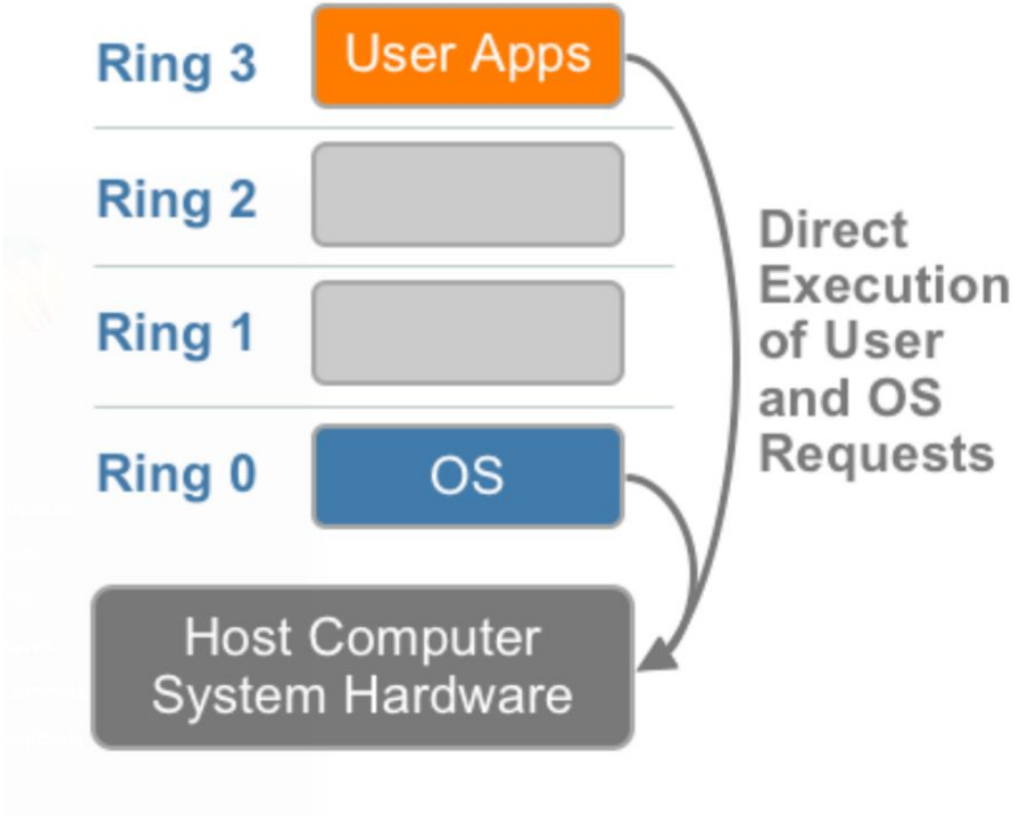
Let's go to little bit in Details inside the server (compute resources) virtualization.

1. Virtualize the CPU in x86 Architecture
2. Memory Virtualization
3. Device and IO Virtualization

Virtualize the CPU in x86 Architecture

Full Virtualization , Paravirtualization and Hardware Assisted Virtualization

Scenario: Without Virtualization – x86 or x64 privilege level architecture without virtualization



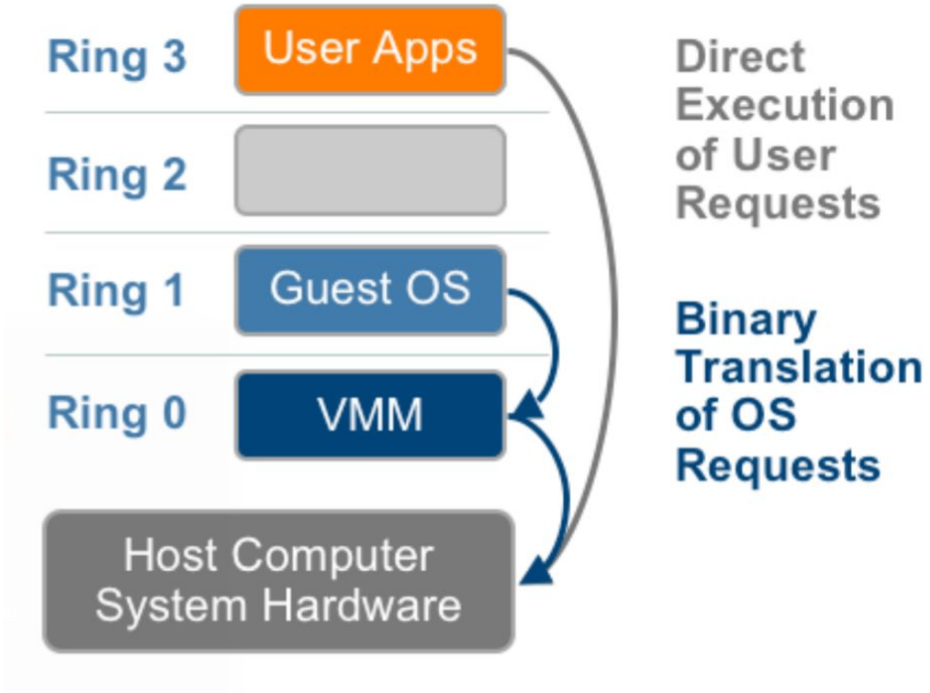
– x86 Hardware privilege level architecture without virtualization

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.

Developing binary translation techniques that allow the **VMM (Virtual Machine Monitor - Hypervisor) to run in Ring 0** for isolation and performance, while moving the **operating system to a user level ring** with greater privilege than applications in Ring 3 but less privilege than the virtual machine monitor in Ring 0.

Full Virtualization

Technique 1 - **Full Virtualization** using Binary Translation



Hypervisor ---------->

Translates kernel code to replace non virtualizable instructions with new sequences of instructions that have the intended effect on the virtual hardware. Meanwhile, user level code is directly executed on the processor for high performance virtualization.

Virtual machine monitor **(Hypervisor)** provides each Virtual Machine with all the services of the physical system, including a virtual BIOS, virtual devices and virtualized memory management.

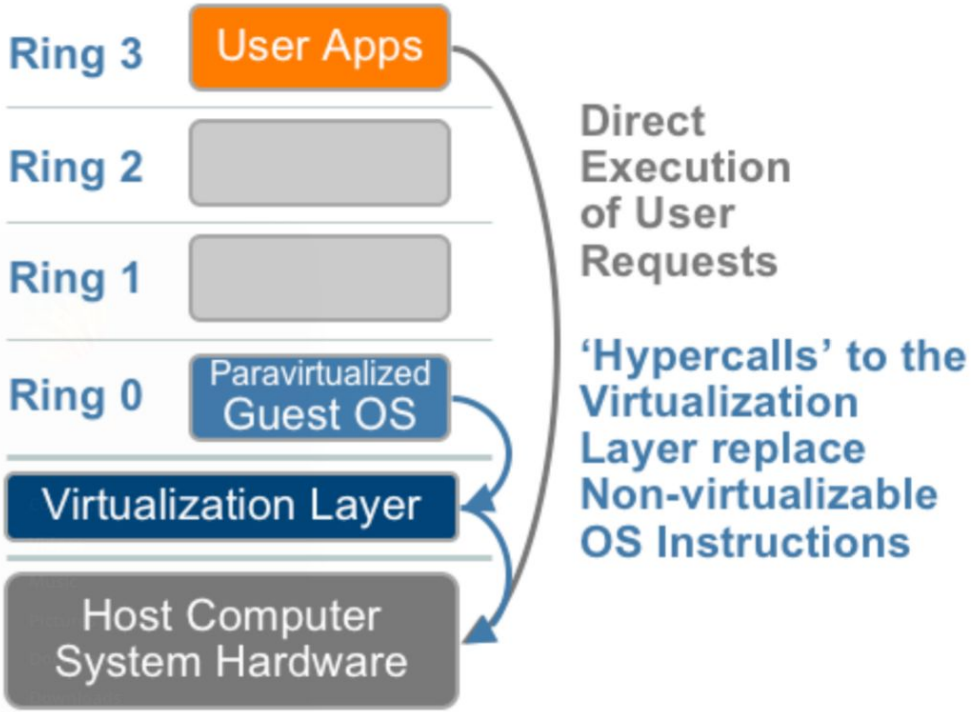
Example:

Full virtualization offers the best isolation and security for virtual machines, and simplifies migration and portability as the same guest OS instance can run virtualized or on native hardware.

VMware's virtualization products and Microsoft Virtual Server are examples of full virtualization.

Para-Virtualization

Technique 2 – OS Assisted Virtualization or Paravirtualization

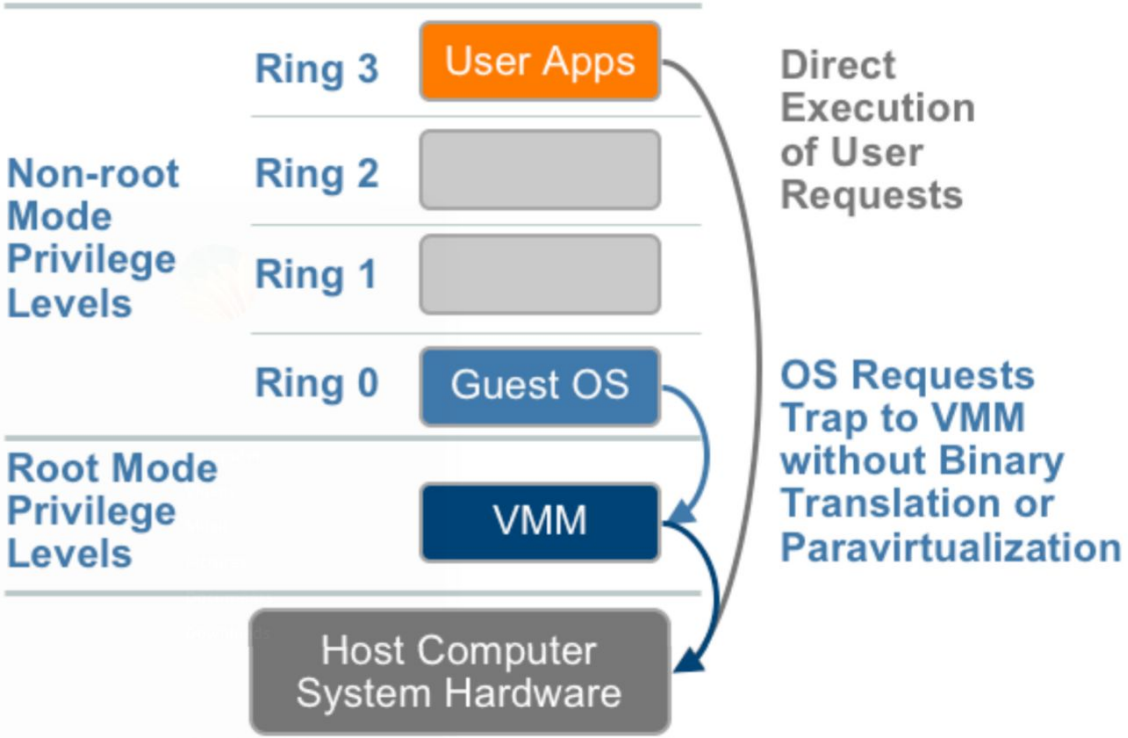


Modifying the OS kernel to replace non virtualizable instructions with hypercalls that communicate directly with the virtualization layer hypervisor. The hypervisor also **provides hypercall interfaces** for other critical kernel operations such as memory management, interrupt handling and time keeping.

Paravirtualization is different from full virtualization, where the **unmodified OS does not know it is virtualized** and sensitive OS calls are trapped using binary translation. The value proposition of paravirtualization is in lower virtualization overhead, but the performance advantage of paravirtualization over full virtualization can vary greatly depending on the workload.

Hardware Assisted Virtualization

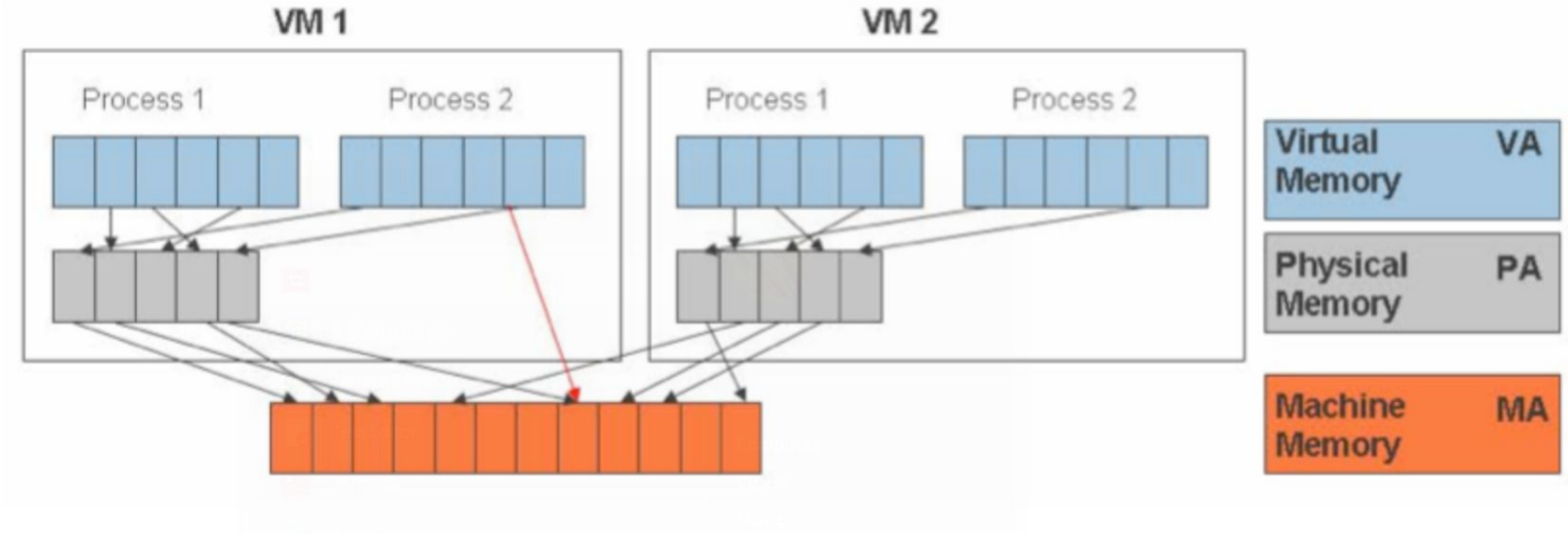
Technique 3 – Hardware Assisted Virtualization



Hardware vendors are rapidly embracing virtualization and developing new features to simplify virtualization techniques. First generation enhancements include 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.

Memory Virtualization

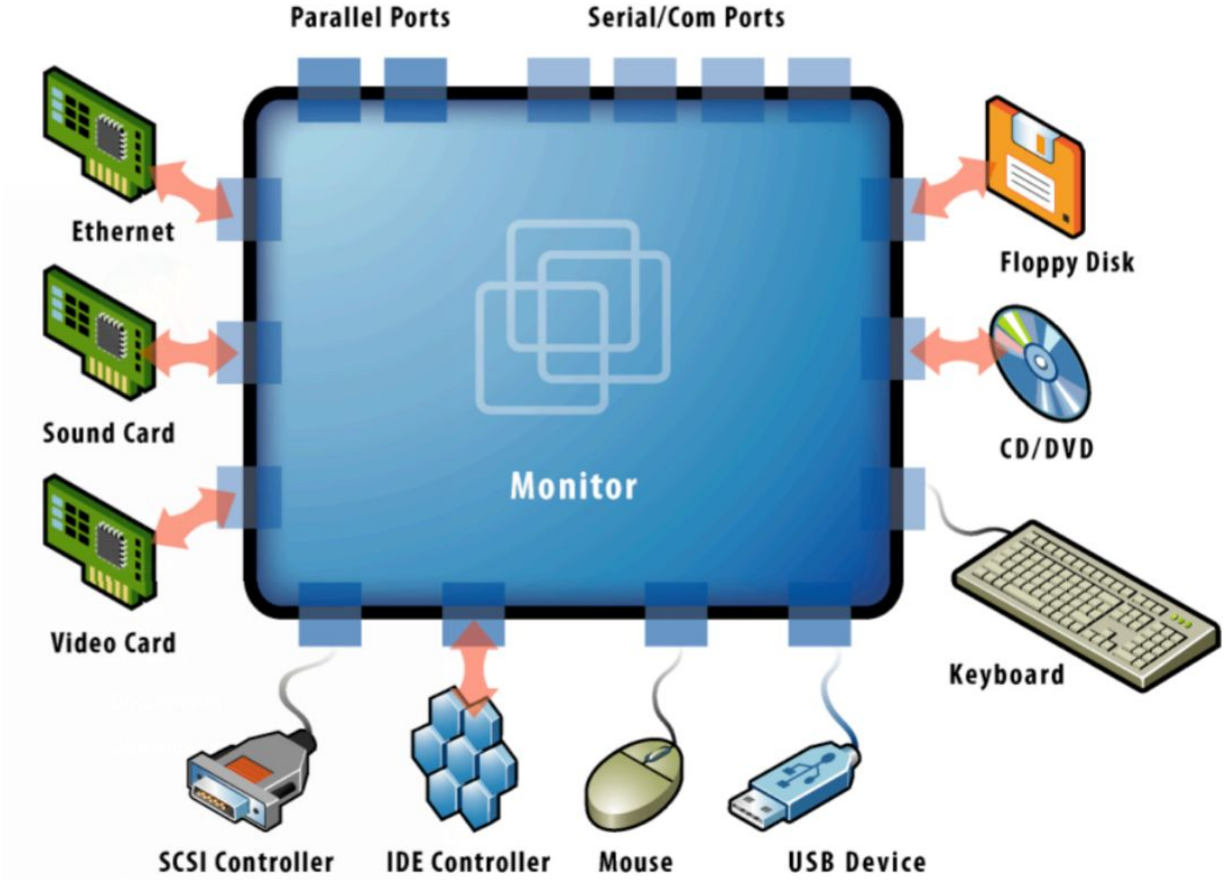

VM 1
VM 2
Process 1
Process 2
Process 1
Process 2
Virtual Memory VA
Physical Memory PA
Machine Memory MA

To run multiple virtual machines on a single system, another level of memory virtualization is required. In other words, one has to virtualize the **MMU to support the guest OS.**

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 (hypervisor) is responsible for mapping guest physical memory to the actual machine memory, and it uses shadow page tables to accelerate the mappings. As depicted by the red line in above figure.

Device and IO Virtualization


Parallel Ports
Serial/Com Ports
Ethernet
Sound Card
Video Card
Monitor
Floppy Disk
CD/DVD
Keyboard
SCSI Controller
IDE Controller
Mouse
USB Device

The final component required beyond CPU and memory virtualization is device and I/O virtualization. This involves **managing the routing of I/O requests between virtual devices and the shared physical hardware.**

These virtual devices effectively **emulate** well-known hardware and translate the virtual machine requests to the system hardware.

Summary comparison of x86 processor virtualization techniques

	Full Virtualization with Binary Translation	Hardware Assisted Virtualization	OS Assisted Virtualization / Paravirtualization
Technique	Binary Translation and Direct Execution	Exit to Root Mode on Privileged Instructions	Hypercalls
Guest Modification / Compatibility	Unmodified Guest OS Excellent compatibility	Unmodified Guest OS Excellent compatibility	Guest OS codified to issue Hypercalls so it can't run on Native Hardware or other Hypervisors Poor compatibility; Not available on Windows OSes
Performance	Good	Fair Current performance lags Binary Translation virtualization on various workloads but will improve over time	Better in certain cases
Used By	VMware, Microsoft, Parallels	VMware, Microsoft, Parallels, Xen	VMware, Xen
Guest OS Hypervisor Independent?	Yes	Yes	XenLinux runs only on Xen Hypervisor VMI-Linux is Hypervisor agnostic