BITS Pilani

Cloud Computing – CS3

Agenda

- Virtualization Recap
- Virtualization Approaches
  - Motivations
  - Full Virtualization
  - Para Virtualization
  - Hardware Assisted Virtualization
  - Compare & Contrast architectures
- X86 Hardware Virtualization
  - Motivation & Challenges
  - X86 Hardware Virtualization
  - NFV - VNF

innovate achieve lead

BITS Pilani

Pilani|Dubai|Goa|Hyderabad

Recap

What is Virtualization?

Virtualization Defined

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

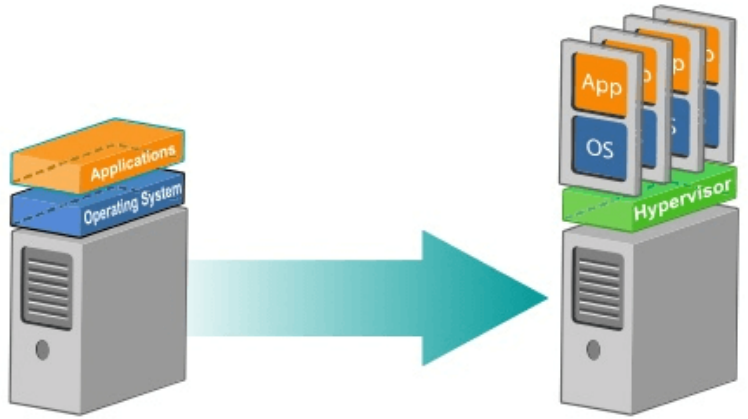
Virtualization allows multiple operating system instances to run concurrently on a single computer

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

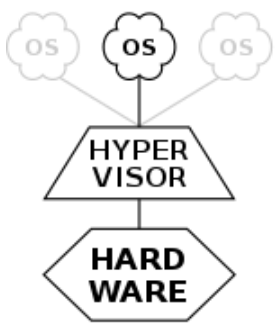


innovate achieve lead

BITS Pilani

Pilani|Dubai|Goa|Hyderabad

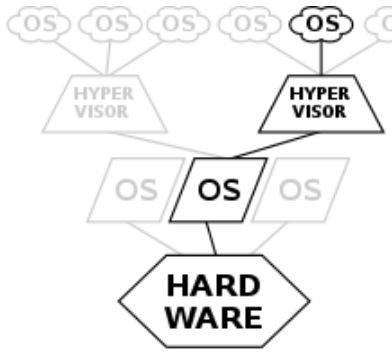
Understanding Hypervisor



TYPE 1

native

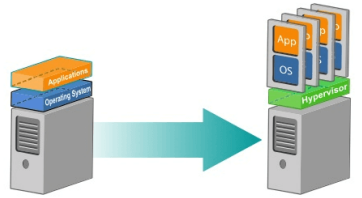
(bare metal)



TYPE 2

hosted

What is Hypervisor



Hypervisor Demystified

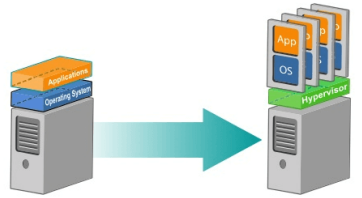
A hypervisor is a form of virtualization software used in Cloud hosting to divide and allocate the resources on various pieces of hardware.

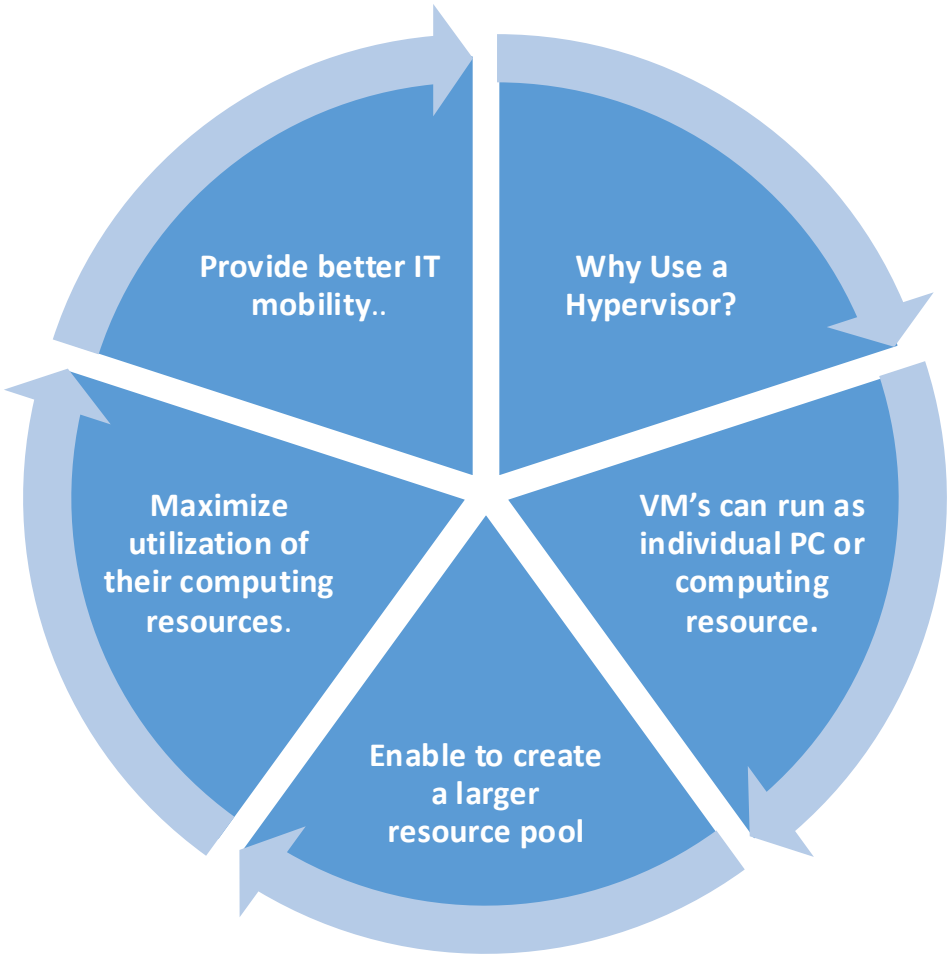
A hypervisor is a crucial piece of software that makes virtualization possible. It creates a virtualization layer that separates the actual hardware components - processors, RAM, and other physical resources - from the virtual machines and the operating systems they run.

Hypervisors emulate available resources so that guest machines can use them. No matter what operating system boots up on a virtual machine, it will think that actual physical hardware is at its disposal..

From a VM's standpoint, there is no difference between the physical and virtualized environment. Guest machines do not know that the hypervisor created them in a virtual environment or that they share available computing power.

Why use Hypervisor

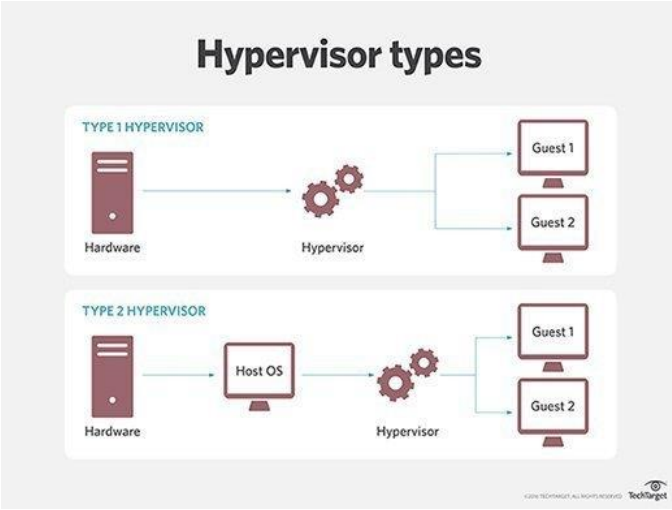




There are two types of hypervisors, according to their place in the server virtualization structure:

- **Type 1 Hypervisors**, also known as bare-metal or native.
- **Type 2 Hypervisors**, also known as hosted hypervisors.

The sections below explain both types in greater detail.

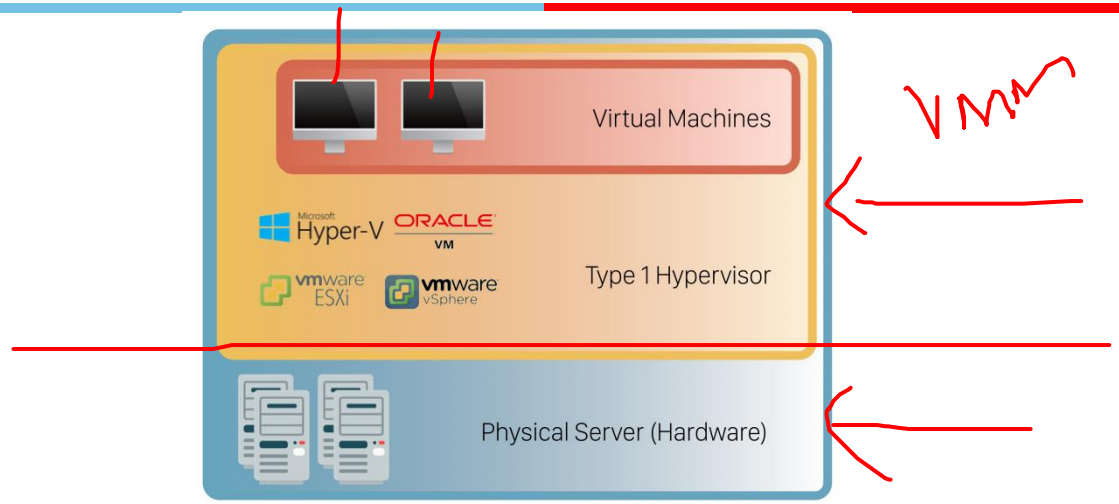


Type 1 Hypervisor

A Type 1 hypervisor is a layer of software installed directly on top of a physical server and its underlying hardware. Since no other software runs between the hardware and the hypervisor, it is also called the bare-metal hypervisor.

This hypervisor type provides excellent performance and stability since it does not run inside Windows or any other operating system. Instead, it is a simple operating system designed to run virtual machines. The physical machine the hypervisor runs on serves virtualization purposes only.

Type 1 hypervisors are mainly found in enterprise environments.



Type 1 Hypervisor – Pros

- **VM Mobility** - Type 1 hypervisors enable moving virtual machines between physical servers, manually or automatically. This move is based on the resource needs of a VM at a given moment and happens without any impact on the end-users. In case of a hardware failure, management software moves virtual machines to a working server as soon as an issue arises. The detection and restoration procedure takes place automatically and seamlessly.
- **Security** - The type 1 hypervisor has direct access to hardware without an additional OS layer. This direct connection significantly decreases the attack surface for potential malicious actors.
- **Resource Over-Allocation** - With type 1 hypervisors, you can assign more resources to your virtual machines than you have. For example, if you have 128GB of RAM on your server and eight virtual machines, you can assign 24GB of RAM to each. This totals 192GB of RAM, but VMs themselves will not consume all 24GB from the physical server. The VMs detect they have 24GB when they only use the amount of RAM they need to perform particular tasks.

Type 1 Hypervisor – Cons

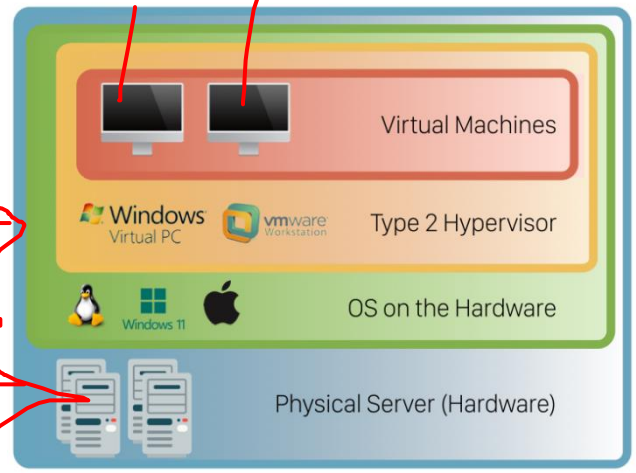
Cons

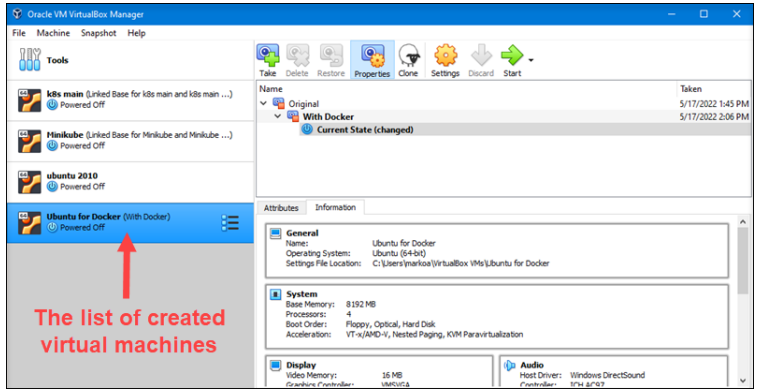
- **Limited functionality** - Type 1 hypervisors are relatively simple and do not offer many features. The functionalities include basic operations such as changing the date and time, IP address, password, etc.
- **Complicated management** - To create virtual instances, you need a management console set up on another machine. Using the console, you can connect to the hypervisor on the server and manage your virtual environment.
- **Price** - Depending on what functionalities you need, the license cost for management consoles varies substantially.

Type 2 Hypervisor

Type 2 hypervisors run inside the physical host machine's operating system, which is why they are called **hosted hypervisors**. Unlike bare-metal hypervisors that run directly on the hardware, hosted hypervisors have one software layer in between. The system with a hosted hypervisor contains:

- A physical machine.
- An operating system installed on the hardware (Windows, Linux, macOS).
- A type 2 hypervisor software within that operating system.
- Guest virtual machine instances.





Type 2 hypervisors are typically found in environments with a small number of servers.

What makes them convenient is that they do not need a management console on another system to set up and manage virtual machines. Everything is performed on the server with the hypervisor installed, and virtual machines launch in a standard OS window.

Hosted hypervisors also act as management consoles for virtual machines. Any task can be performed using the built-in functionalities. Below is one example of a type 2 hypervisor interface (VirtualBox by Oracle):

Type 2 Hypervisor - PROs

Pros

- **Easy to manage** - There is no need to install separate software on another machine to create and maintain your virtual environment. Install and run a type 2 hypervisor as any other application within your OS. Create snapshots or clone your virtual machines, import or export appliances, etc.
- **Convenient for testing** - Type 2 hypervisors are convenient for testing new software and research projects. It is possible to use one physical machine to run multiple instances with different operating systems to test how an application behaves in each environment or to create a specific network environment. You only need to ensure that there are enough physical resources to keep the host and virtual machines running.
- **Allows access to additional productivity tools** - The users of type 2 hypervisors can use the tools available on other operating systems alongside their primary OS. For example, Windows users can access Linux applications by creating a Linux virtual machine.

Type 2 Hypervisor - CONs

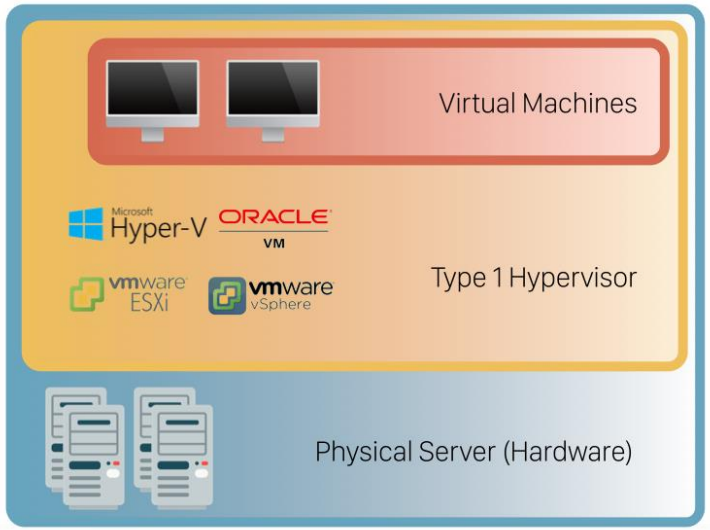
Cons

•**Less flexible resource management** - Allocating resources with this type of hypervisor is more difficult than with type 1. Bare-metal hypervisors can dynamically allocate available resources depending on the current needs of a particular VM. A type 2 hypervisor occupies whatever the user allocates to a virtual machine. When a user assigns 8GB of RAM to a VM, that amount will be taken up even if the VM is using only a fraction of it. If the host machine has 32GB of RAM and the user creates three VMs with 8GB each, they are left with 8GB of RAM to keep the physical machine running. Creating another VM with 8GB of ram would bring down the system.

•**Performance** - The host OS creates additional pressure on physical hardware, which may result in VMs having latency issues.

•**Security** - Type 2 hypervisors run on top of an operating system. This fact introduces a potential vulnerability since attackers may use potential vulnerabilities of the OS to gain access to virtual machines

Type 1 & 2 Hypervisor – At a Glance



← Type 1 or Bare Metal is installed directly on hardware

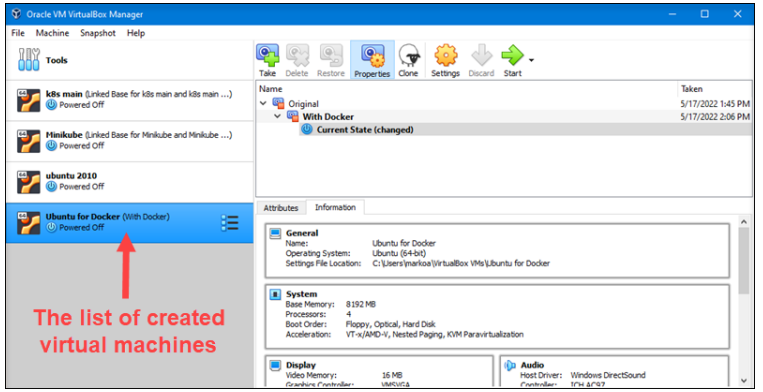
Type 2 is installed on top of an existing OS →

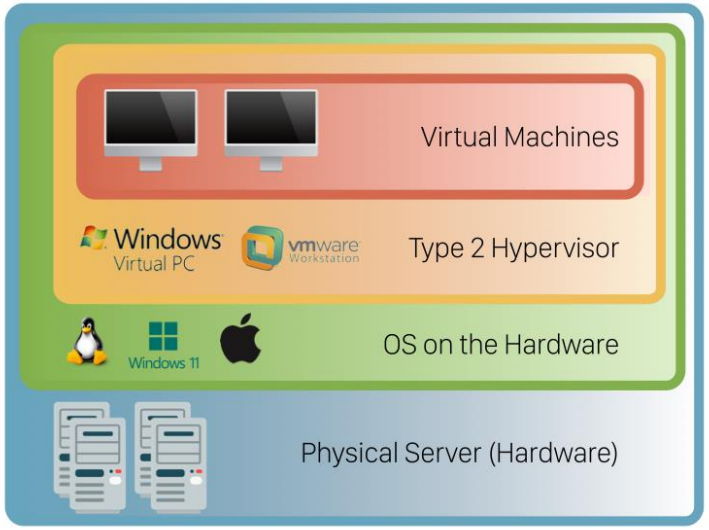
← Mostly used by Enterprise CSP

Mostly used by Medium & Medium Small scale enterprises →

← Need to have a separate VM Management console to monitor the VM

Has an in-built console →



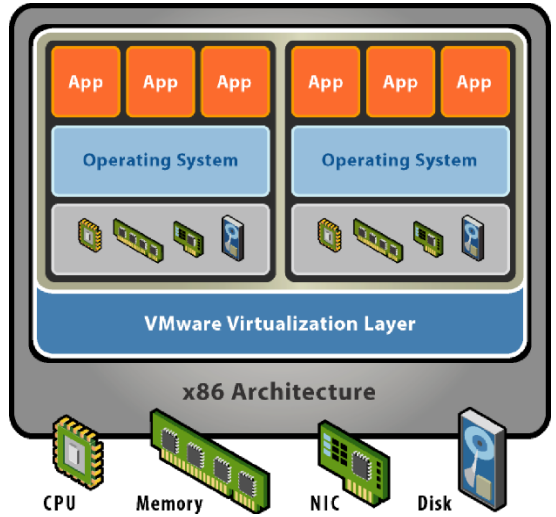


innovate achieve lead

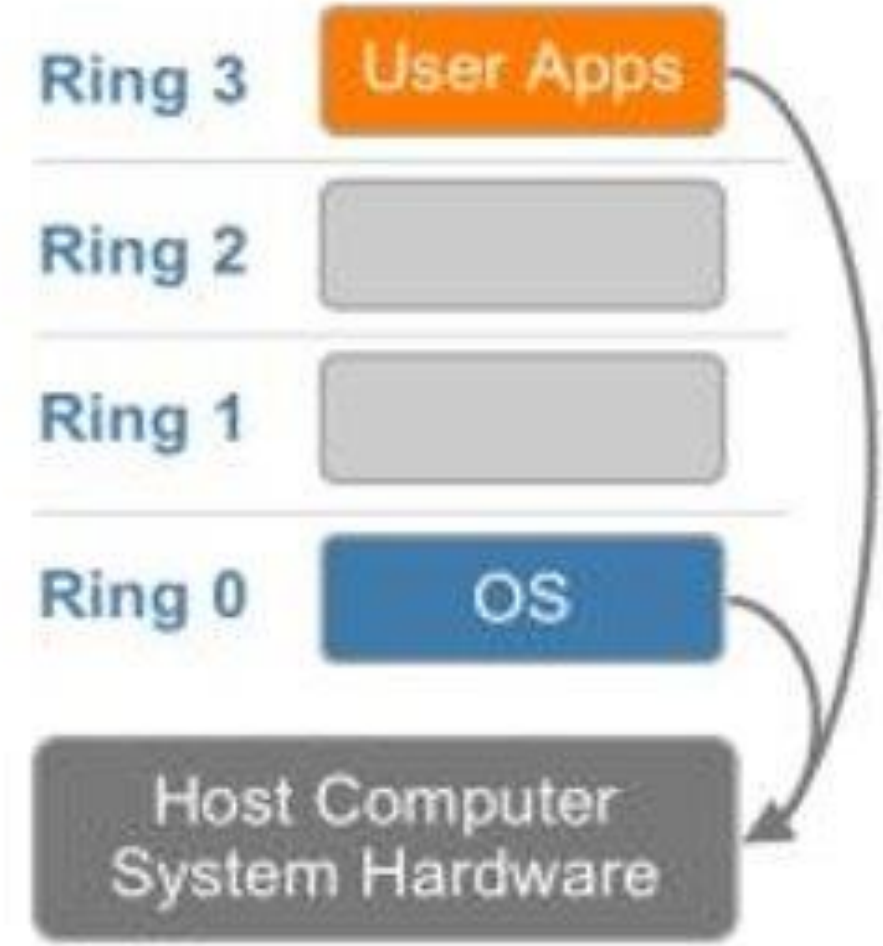
BITS Pilani

Pilani|Dubai|Goa|Hyderabad

Virtualization Evolution

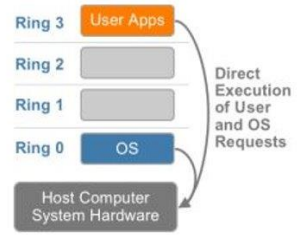


x86 Architecture




```
+---------------------------------+
|          Application            |
+---------------------------------+
|          OS Kernel              |
+---------------------------------+
|       Hardware Abstraction      |
+---------------------------------+
|       Microarchitecture         |
+---------------------------------+
|       Physical Hardware         |
+---------------------------------+
```

X86 Architecture



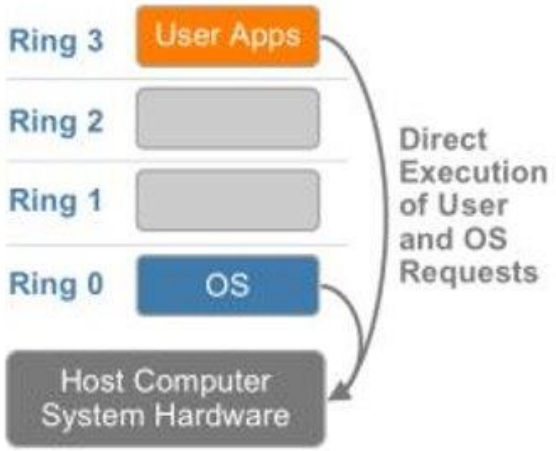
The x86 architecture uses a system of privilege rings to control access to sensitive areas of the computer's memory and resources. This system is known as Ring Architecture, and there are four privilege levels, or rings, numbered 0 through 3.

1. **Ring 0**: Also known as the kernel mode, ring 0 is the most privileged level, and it has full access to all of the computer's resources, including memory and I/O devices. This is where the operating system kernel runs.
2. **Ring 1**: This ring is used for device drivers and other low-level system components. Ring 1 has access to more resources than ring 2 and ring 3, but it still has limited access compared to ring 0.
3. **Ring 2**: Ring 2 is not used in modern x86 systems, but it was used in the past for system extensions, such as device drivers and system libraries. Ring 2 has even less access to the computer's resources than ring 1.
4. **Ring 3**: Also known as user mode, ring 3 is the least privileged level, and it is where user applications run. Ring 3 has limited access to the computer's resources, and any access to sensitive areas of the system must be performed through system calls that are handled by the operating system kernel running in ring 0.

Each process running on an x86 system runs in a **specific ring,** and the ring level **determines the level of access the process has to the computer's resources**. This system of privilege rings provides a layer of security by ensuring that user applications cannot interfere with the operation of the operating system and other system components.

Challenges to Virtualization

- X86 operating systems are designed to run directly on the bare-metal hardware, so they naturally assume they fully 'own' the computer hardware.
- As shown, the 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.
- While user level applications typically run in Ring 3, the operating system needs to have direct access to the memory and hardware and must execute its privileged instructions in Ring 0.
- 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

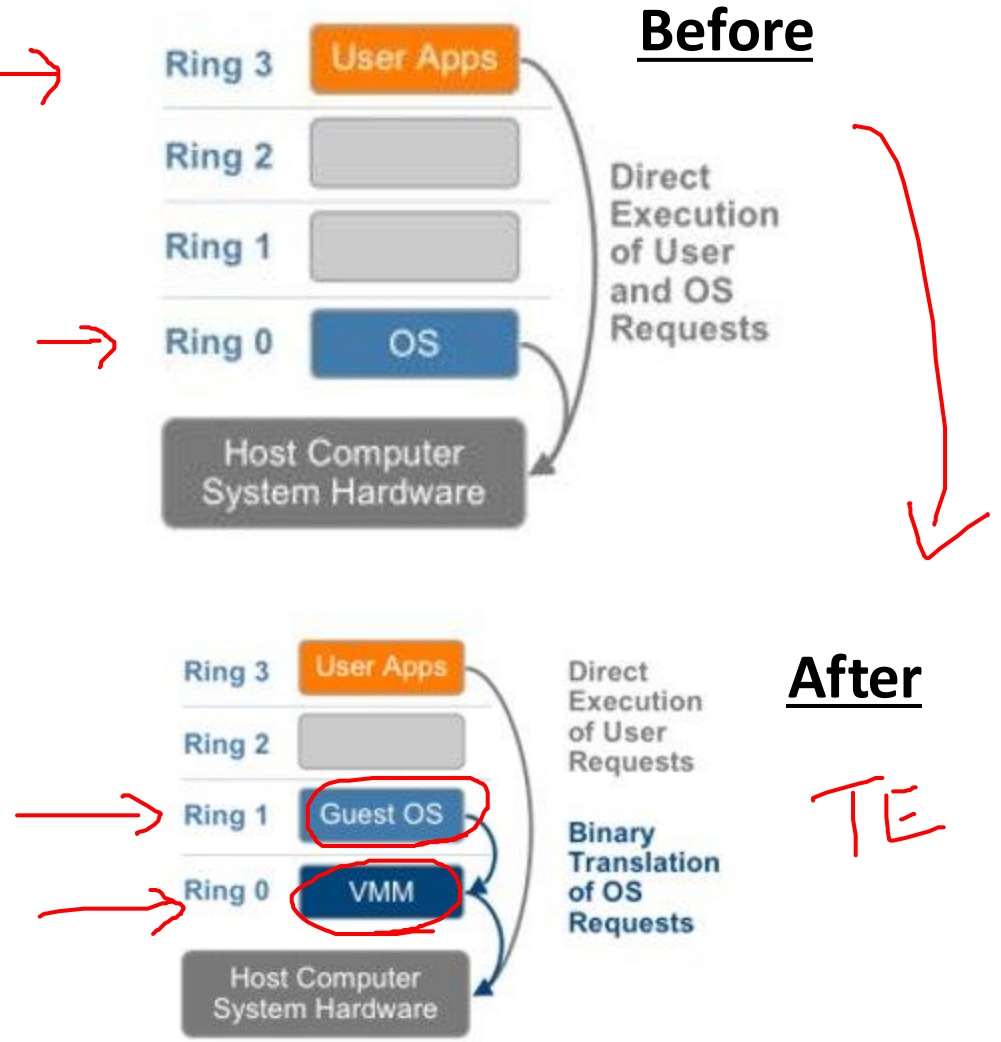


Further complicating the situation, some sensitive instructions can't effectively be virtualized as they have different semantics when they are not executed in Ring 0.

The difficulty in trapping and translating these sensitive and privileged instruction **requests at runtime was the challenge that originally made x86 architecture virtualization look impossible**

VMware's Solution

- VMware resolved the challenge in 1998, developing binary translation techniques that allow the VMM to run in Ring 0 for isolation and performance,
- The operating system was moved to a user level ring with greater privilege than applications in Ring 3 but less privilege than the virtual machine monitor in Ring 0.
- While VMware's full virtualization approach using binary translation is the de facto standard today the industry as a whole has not yet agreed on open standards to define and manage virtualization.
- Each company developing virtualization solutions is free to interpret the technical challenges and develop solutions with varying strengths and weaknesses.



Virtualization Evolution

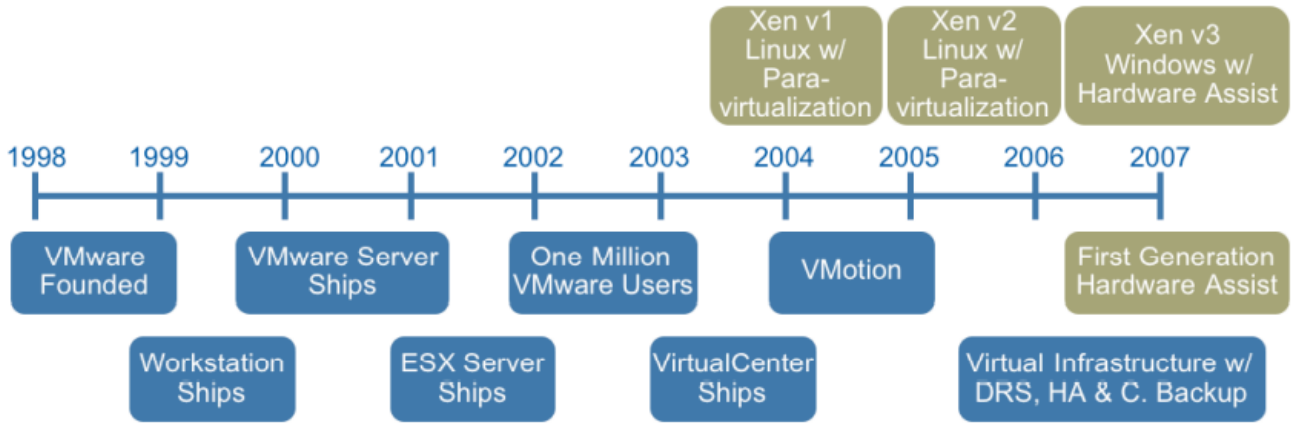


Figure 1 – Summary timeline of x86 virtualization technologies

Currently, the **most commonly used virtualization feature is hardware-assisted virtualization**. Hardware-assisted virtualization is a feature built into modern CPUs that provides improved performance and security for virtualization compared to traditional software-based virtualization technologies.
Hardware-assisted virtualization is supported by Intel's VT-x and AMD's AMD-V technology and is used by many popular virtualization platforms, such as VMware, Hyper-V, and Oracle VirtualBox. This technology allows the **virtualization software** to run **virtual machines** with **near-native performance**, as the **virtualization software** is able to directly access and use the CPU's hardware-assisted virtualization features

In 1998, VMware figured out how to virtualize the x86 platform, once thought to be impossible, and created the market for x86 virtualization.

The solution was a combination of binary translation and direct execution on the processor that allowed multiple guest OSes to run in full isolation on the same computer with readily affordable virtualization overhead

Introduction

Evolution

1st Generation: Full virtualization (Binary rewriting)

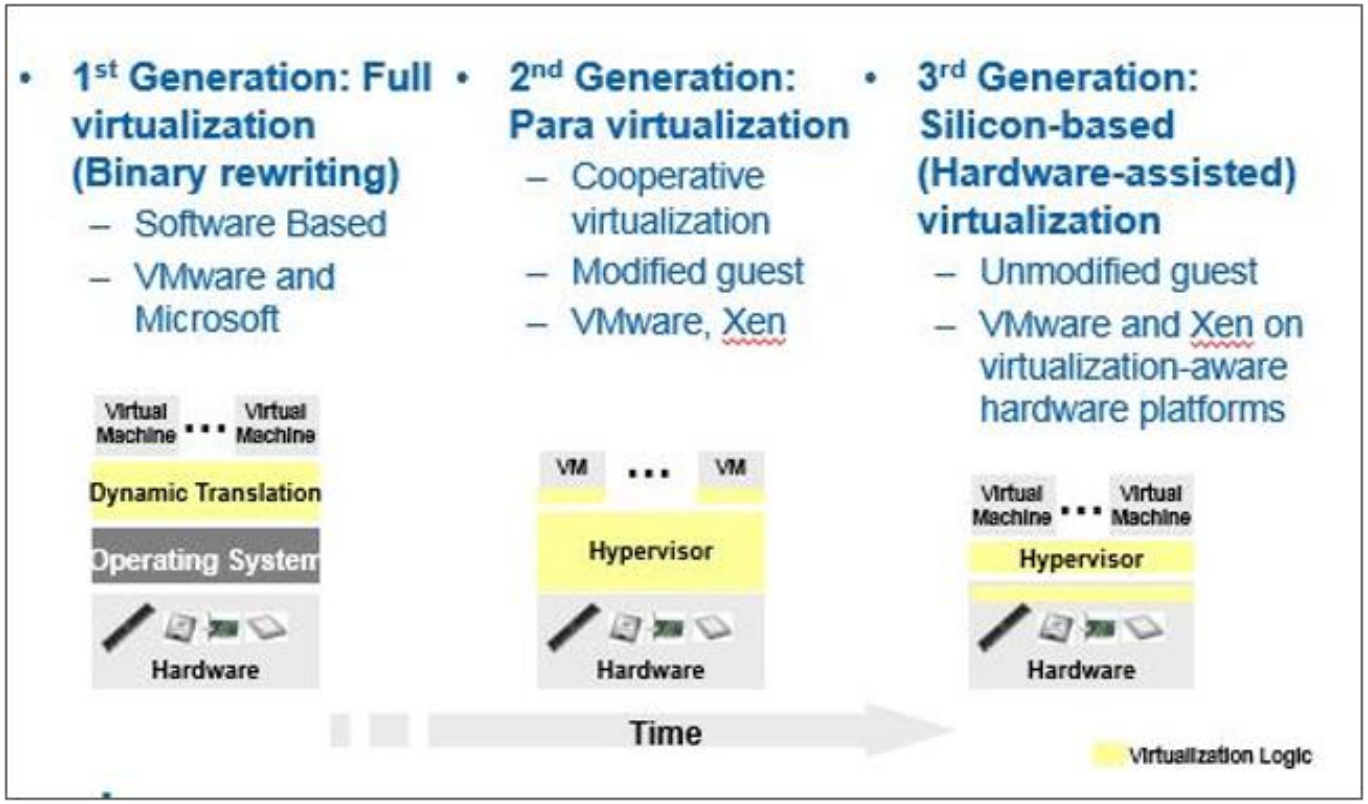
- Software Based
- VMware and Microsoft

2nd Generation: Para virtualization

- Cooperative virtualization
- Modified guest
- VMware, Xen

3rd Generation: Silicon-based (Hardware-assisted) virtualization

- Unmodified guest
- VMware and Xen on virtualization-aware hardware platforms

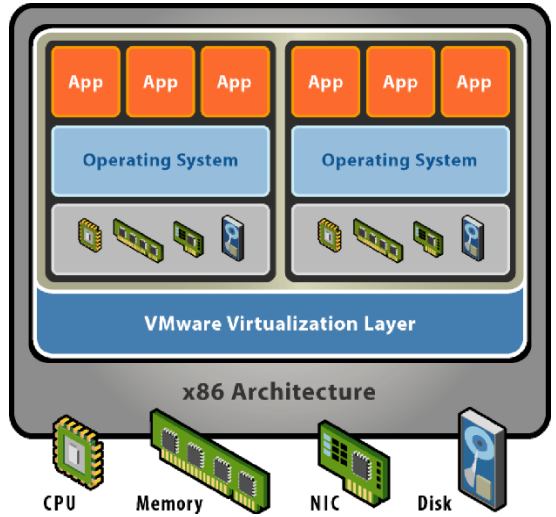


innovate achieve lead

BITS Pilani

Pilani|Dubai|Goa|Hyderabad

Virtualization Approaches



Emulation

Emulation is the process where the virtualizing software mimics that portion of hardware, which is provided to the guest operating system in the virtual machine. The presented emulated hardware is independent of the underlying physical hardware.

Emulation provides VM portability and wide range of hardware compatibility, which means the possibility of executing any virtual machine on any hardware, as the guest operating system interacts only with the emulated hardware.

In an emulated environment, both the application and guest operating system in virtual machines run in the user mode of base operating system. In simple terms, the behavior of the hardware is produced by a software program.

Emulation process involves only those hardware components so that user or virtual machines does not understand the underlying environment.

Emulation

Only CPU & memory are sufficient for basic level of emulation. Typically, emulation is implemented using interpretation. The emulator component takes each and every instruction of user mode and translates to equivalent instruction suitable according to the underlying hardware. This process is also termed as interpretation.

This means that the guest OS remains completely unaware of the virtualization. Also, in interpretation, each and every instruction issued by a VM is trapped in the VMM and interpreted for execution in the hardware. Goes without saying that computationally it is a very expensive method. However, in some cases,, it is needed to use an interpretation technique. However, due to the huge disadvantage of performance, emulation using interpretation is hardly used in virtualization.

Binary Translation / Full Virtualization

In its basic form known as “full virtualization” the hypervisor provides a fully emulated machine in which an operating system can run. VMWare is a good example.

The biggest advantage to this approach is its flexibility: one could run a RISC-based OS as a guest on an Intel-based host.

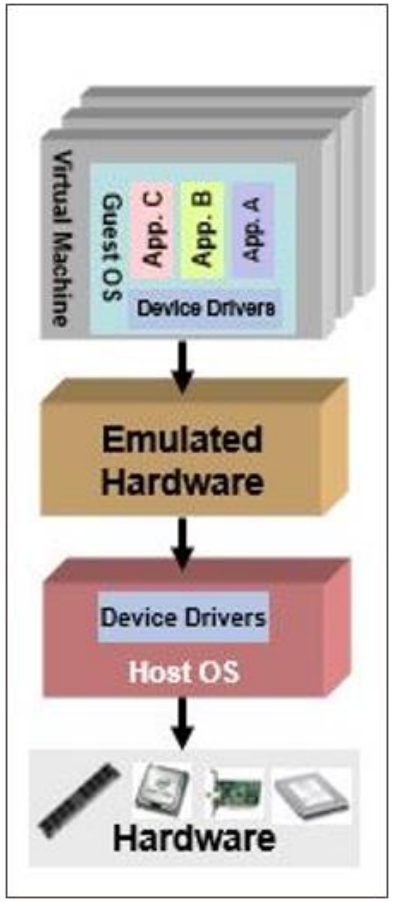
While this is an obvious approach, there are significant performance problems in trying to emulate a complete set of hardware in software.

1st Generation offering of x86/x64 server virtualization .

Dynamic binary translation.

The emulation layer talks to an operating system which talks to the computer hardware.

The guest OS doesn't see that it is used in an emulated environment.



Binary Translation / Full Virtualization - Pros

The emulation layer Isolates VMs from the host OS and from each other.

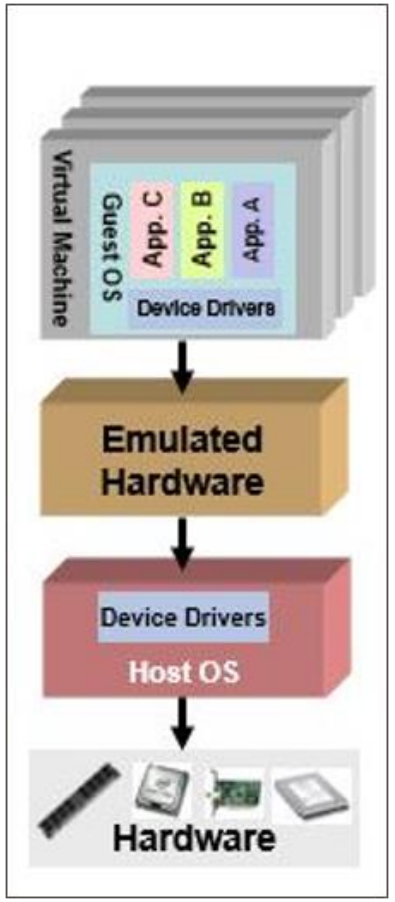
Controls individual VM access to system resources, preventing an unstable VM from impacting system performance.

Total VM portability.

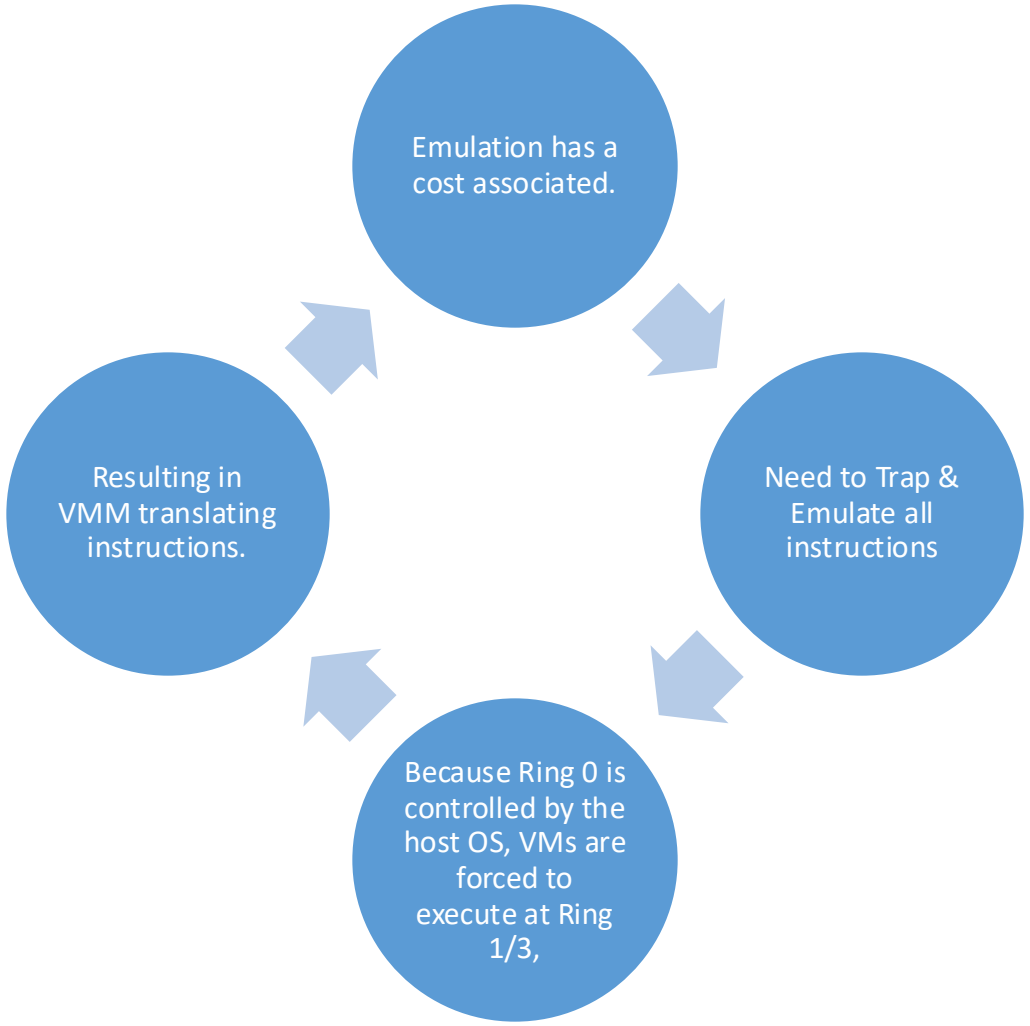
By emulating a consistent set of system hardware, VMs have the ability to transparently move between hosts with dissimilar hardware without any problems.

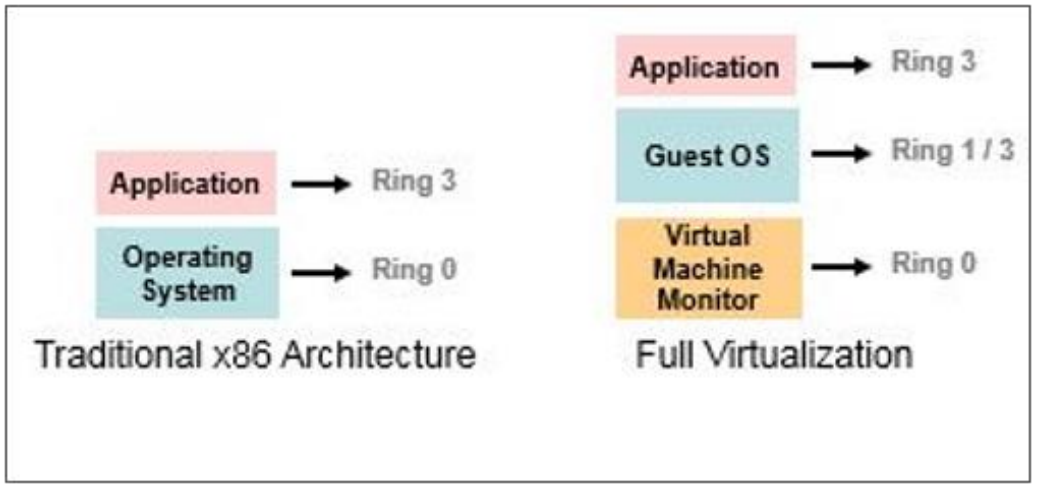
It is possible to run an operating system that was developed for another architecture on your own architecture.

A VM running on a Dell server can be relocated to a Hewlett-Packard server.



Binary Translation / Full Virtualization - Cons

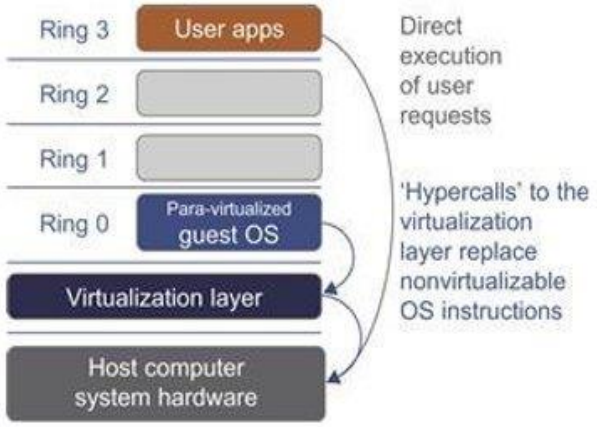




Para Virtualization

Paravirtualization," found in the XenSource, open source Xen product, attempts to reconcile these two approaches. Instead of emulating hardware, paravirtualization uses slightly altered versions of the operating system which allows access to the hardware resources directly as managed by the hypervisor.

The Guest OS is modified and thus run kernel level operations at Ring 1 (or 3)



the guest is fully aware of how to process privileged instructions

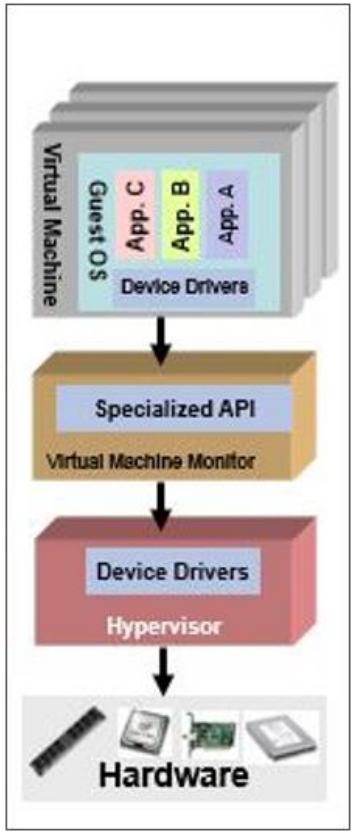
thus, privileged instruction translation by the VMM is no longer necessary

The guest operating system uses a specialized API to talk to the VMM and, in this way, execute the privileged instructions

The VMM is responsible for handling the virtualization requests and putting them to the hardware

Para Virtualization

- Today, VM guest operating systems are para virtualized using two different approaches:
- **Recompiling the OS kernel**
  - Para virtualization drivers and APIs must reside in the guest operating system kernel.
  - Modified operating system that includes this specific API, requiring a compiling operating systems to be virtualization aware.
  - Some vendors (such as Novell) have embraced para virtualization and have provided para virtualized OS builds, while other vendors (such as Microsoft) have not.
- **Installing para virtualized drivers**
  - In some operating systems it is not possible to use complete para virtualization, as it requires a specialized version of the operating system
  - To ensure good performance in such environments, para virtualization can be applied for individual devices
  - For example, the instructions generated by network boards or graphical interface cards can be modified before they leave the virtualized machine by using para virtualized drivers



H/W Assisted Virtualization

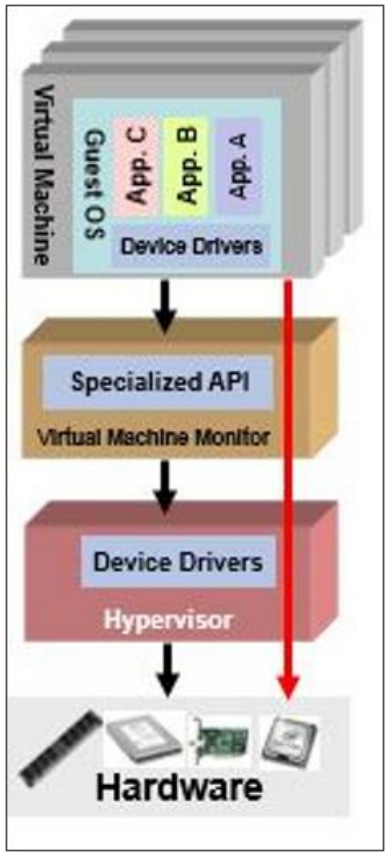
This technique attempts to simplify virtualization because full or paravirtualization is complicated.

Intel and AMD add an additional mode called privilege mode level (some people call it Ring-1) to x86 processors.

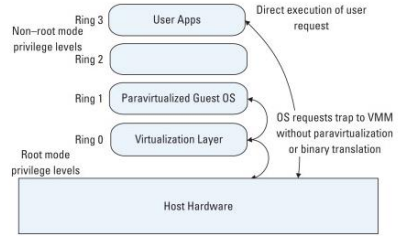
Therefore, operating systems can still run at Ring 0 and the hypervisor can run at Ring -1.

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 run without modification unlike para virtualization



H/W Assisted Virtualization



Hardware-assisted virtualization. This term refers to a scenario in which the hardware provides architectural support for building a virtual machine manager able to run a guest operating system in complete isolation.

This technique was originally introduced in the IBM System/370. At present, examples of hardware-assisted virtualization are the extensions to the x86-64 bit architecture introduced with *Intel VT* (formerly known as *Vanderpool*) and *AMD V* (formerly known as *Pacifica*).

These extensions, which differ between the two vendors, are meant to reduce the performance penalties experienced by emulating x86 hardware with hypervisors. Before the introduction of hardware-assisted virtualization, software emulation of x86 hardware was significantly costly from the performance point of view.

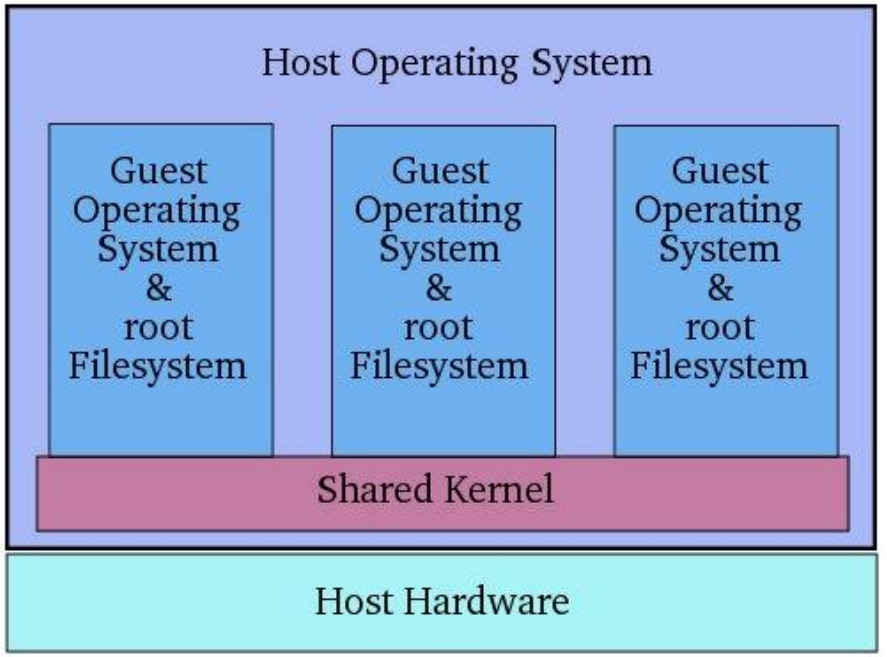
The reason for this is that by design the x86 architecture did not meet the formal requirements introduced by Popek and Goldberg, and early products were using binary translation to trap some sensitive instructions and provide an emulated version. Products such as VMware Virtual Platform, introduced in 1999 by VMware, which pioneered the field of x86 virtualization, were based on this technique. After 2006, Intel and AMD introduced processor extensions, and a wide range of virtualization solutions took advantage of them: Kernel-based Virtual Machine (KVM), VirtualBox, Xen, VMware, Hyper-V, Sun xVM, Parallels, and others.

SKI Virtualization

Instead of using a hypervisor, it runs a separate version of the Linux kernel and sees the associated virtual machine as a user-space process on the physical host. This makes it easy to run multiple virtual machines on a single host. A device driver is used for communication between the main Linux kernel and the virtual machine.

Processor support is required for virtualization (Intel VT or AMD – v). A slightly modified QEMU process is used as the display and execution containers for the virtual machines. In many ways, kernel-level virtualization is a specialized form of server virtualization.

Examples: User – Mode Linux(UML) and Kernel Virtual Machine(KVM) , Docker, LXC



Note:
SKI means Single Kernel Image, is the basis for Container Technologies

Virtualization Comparison

	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

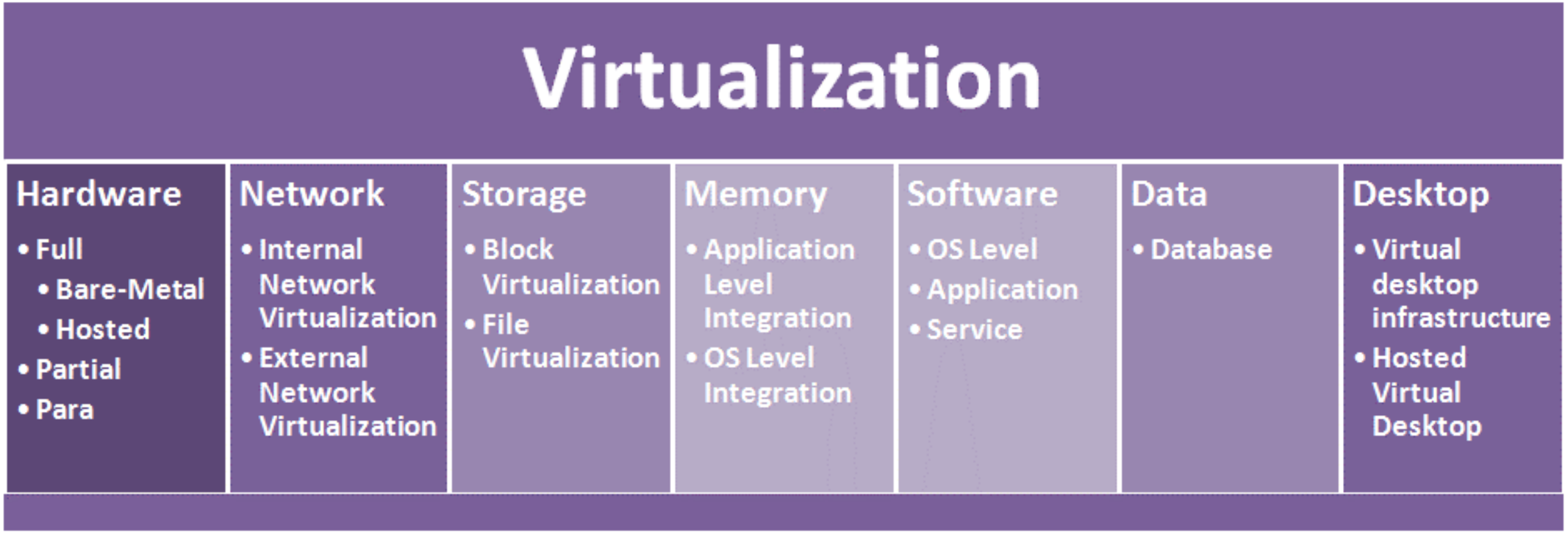
innovate achieve lead

BITS Pilani

Pilani|Dubai|Goa|Hyderabad

Virtualization Types

Virtualization



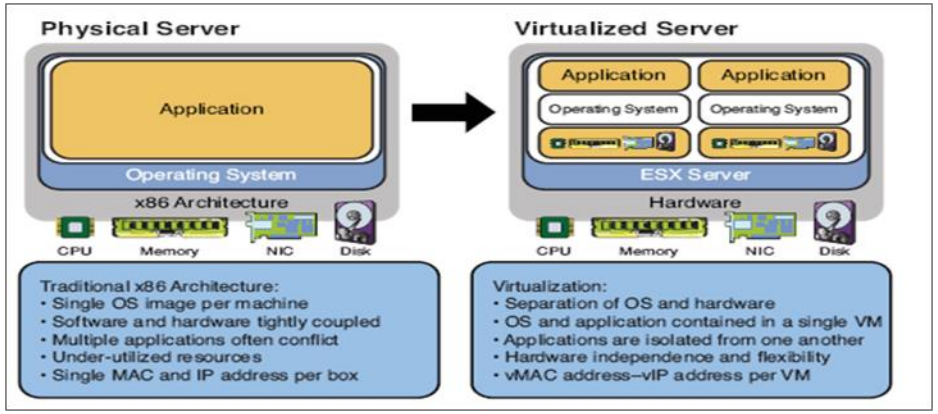
Server Virtualization

Abstracts the physical machine on which the software and operating system is running on and provides an illusion that the software is running on a virtual machine.

Enables Infrastructure as a service model.

A single physical machine can be used to create several VMs that can run several operating systems independently and simultaneously. VMs are stored as files, so restoring a failed system can be as simple as copying its file onto a new machine.

The hypervisor software enables the creation of a virtual machine (VM) that emulates a physical computer by creating a separate OS environment that is logically isolated from the host server.



Server Virtualization - Benefits

Partitioning

Run multiple operating systems on one physical machine.

Divide the physical system resources among virtual machines.

One VM does not know the presence of the other.

Management

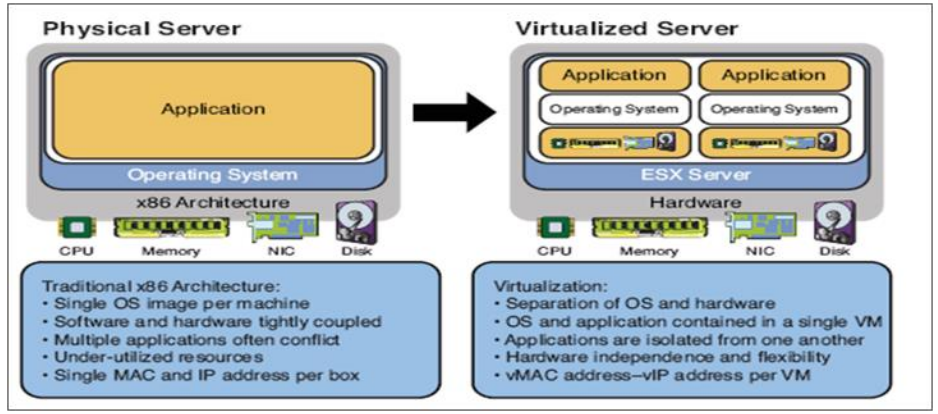
Failure of one VM does not affect other VMs.

Management agents can be run on each VM separately to determine the individual performance of the VM and the applications that are running on the VM.

Encapsulation

The entire VM state can be saved in a file.

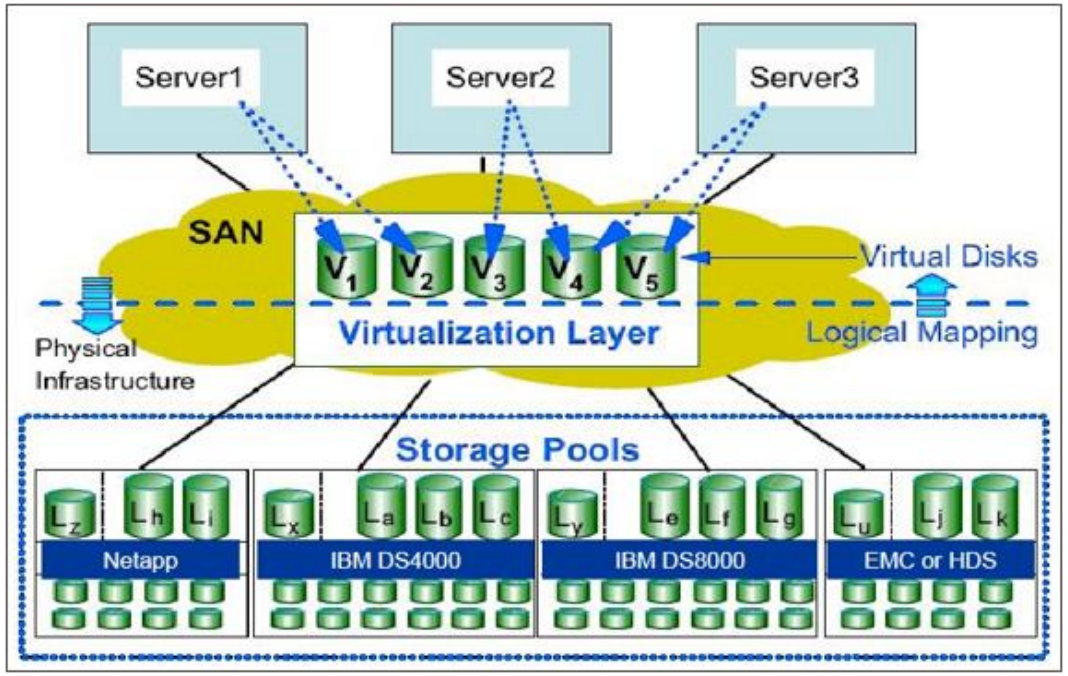
Moving and copying VM information is as easy as copying files.



Server virtualization is a key driving force in reducing the number of physical servers and hence the physical space, cooling, cabling, and capital expenses in any data center consolidation projects

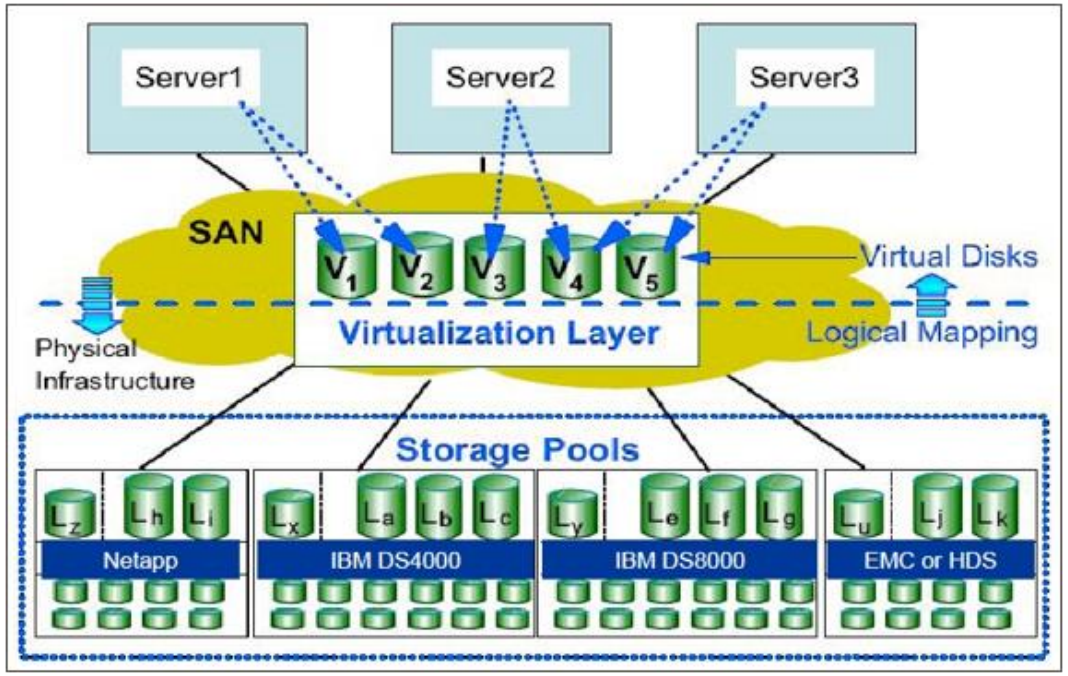
Storage Virtualization

- Storage virtualization refers to providing a logical, abstracted view of physical storage devices.
- It provides a way for many users or applications to access storage without being concerned with where or how that storage is physically located or managed.
- It enables physical storage in an environment to be shared across multiple application servers, and physical devices behind the virtualization layer to be viewed and managed as if they were one large storage pool with no physical boundaries.
- The storage virtualization hides the fact there are separate storage devices in an organization by making all the devices appear as one device.
- Virtualization hides the complex process of where the data needs to be stored and bringing it back and presenting it to the user when it is required.



Storage Virtualization - Benefits

- Typically, the benefits are :-
- **Resource optimization** : Storage virtualization enables you to obtain the storage space on an as-needed basis without any wastage, and it allows organizations to use existing storage assets more efficiently without the need to purchase additional assets.
- **Cost of operation**: Storage virtualization enables adding storage resources without regard to the application, and storage resources can be easily added to the pool by a drag-and-drop method using a management console by the operations people.
- **Increased availability:** Storage virtualization provisions the new storage resources in a minimal amount of time, improving the overall availability of resources
- **Improved performance**

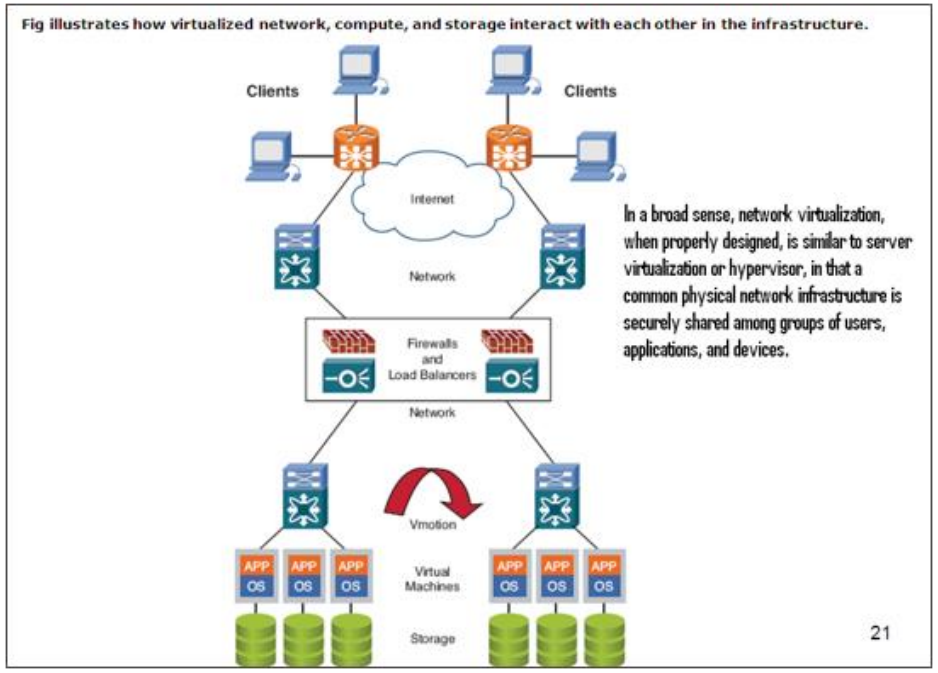


Examples of Storage Virtualization:

- SAN ➔ Storage area Network
- VDA ➔ Virtual Disk Array

Network Virtualization

- Network virtualization might be the most ambiguous virtualization of all virtualization types. Several types of network virtualization exist, as briefly described here:
- A VLAN is a simple example of network virtualization. VLANs allow logical segmentation of a LAN into several broadcast domains. VLANs are defined on a switch on a port-by-port basis. That is, you might choose to make ports 1–10 part of VLAN 1 and ports 11–20 part of VLAN 2. There's no need for ports in the same VLAN to be contiguous. Because this is a logical segmentation and not physical, workstations connected to the ports do not have to be located together, and users on different floors in a building or different buildings can be connected together to form a LAN.
- Virtual Routing and Forwarding (VRF), commonly used in Multi-Protocol Label Switching (MPLS) networks, **allows multiple instances of a routing table to coexist within the same router at the same time.**



Examples
VLAN ➔ Virtual Local Area Network
SDN ➔ S/W Defined Network
NFV ➔ Network Function Virtualization

Memory Virtualization

Beyond CPU virtualization, the next critical component is memory virtualization.

This involves sharing the physical system memory and dynamically allocating it to virtual machines.

Virtual machine memory virtualization is very similar to the virtual memory support provided by modern operating systems.

Applications see a contiguous address space that is not necessarily tied to the underlying physical memory in the system.

The operating system keeps mappings of virtual page numbers to physical page numbers stored in page tables. All modern x86 CPUs include a memory management unit (MMU) and a translation lookaside buffer (TLB) to optimize virtual memory performance

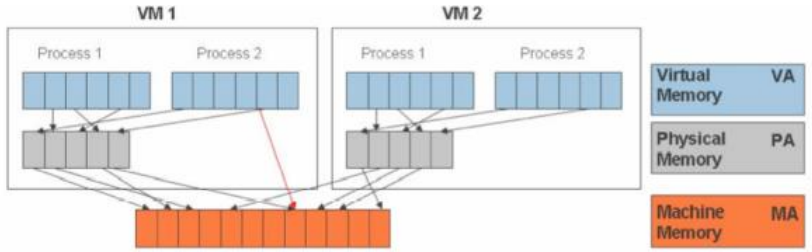


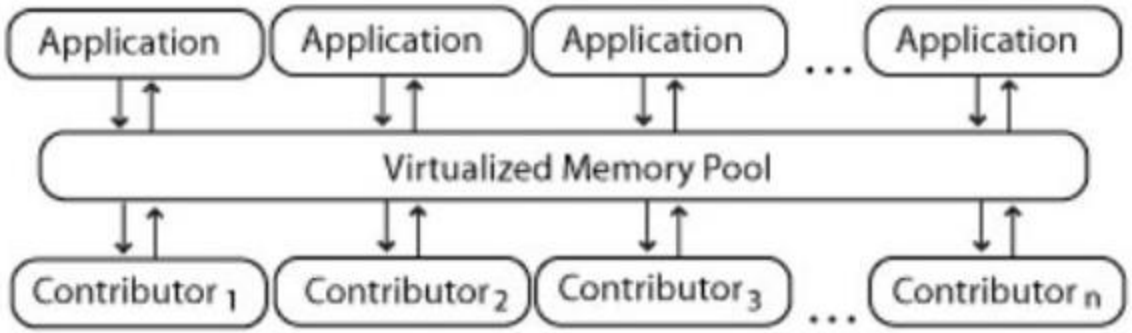
Figure 8 – Memory Virtualization

Memory Virtualization

It introduces a way to decouple memory from the server to provide a shared, distributed or networked function. It enhances performance by providing greater memory capacity without any addition to the mai n memory. That's why a portion of the disk drive serves as an extension of the main memory.

Application level integration – Applications running on connected computers directly connect to the memory pool through an API or the file system.

Operating System Level Integration – The operating system first connects to the memory pool, and makes that pooled memory available to applications.



Device Virtualization

It provides the work convenience and security.

As one can access remotely, you are able to work from any location and on any PC. It provides a lot of flexibility for employees to work from home or on the go.

It also protects confidential data from being lost or stolen by keeping it safe on central servers.

This involves managing the routing of I/O requests between virtual devices and the shared physical hardware.

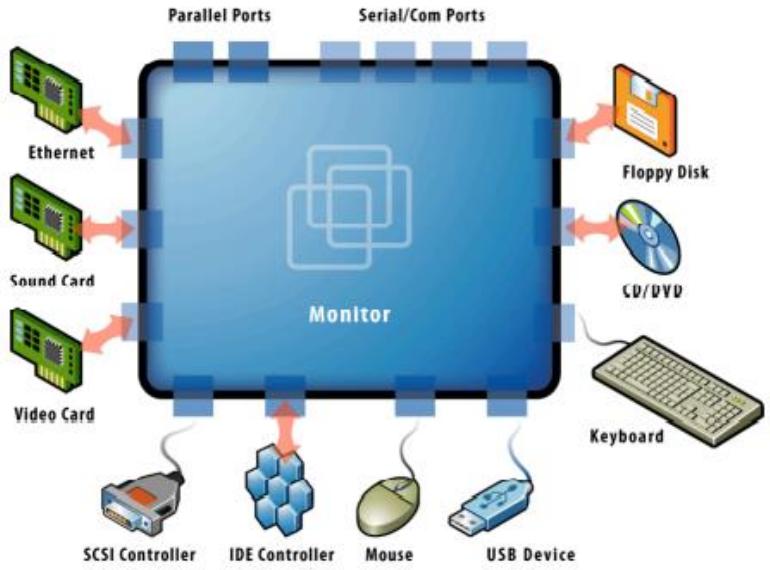


Figure 9 – Device and I/O virtualization

Virtualization Advantages

- 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

Virtualization Summary

	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 I lypervisor VMI-Linux is Hypervisor agnostic

Virtualization Advantages

Security: by compartmentalizing environments with different security requirements in different virtual machines one can select the guest operating system and tools that are more appropriate for each environment. For example, we may want to run the Apache web server on top of a Linux guest operating system and a backend MS SQL server on top of a guest Windows XP operating system, all in the same physical platform. A security attack on one virtual machine does not compromise the others because of their isolation.

Virtualization Advantages

Reliability and availability: A software failure in a virtual machine does not affect other virtual machines.

Cost: It is possible to achieve cost reductions by consolidation smaller servers into more powerful servers. Cost reductions stem from hardware cost reductions (economies of scale seen in faster servers), operations cost reductions in terms of personnel, floor space, and software licenses. VMware cites overall cost reductions ranging from 29 to 64%

Virtualization Advantages

Adaptability to Workload Variations: Changes in workload intensity levels can be easily taken care of by shifting resources and priority allocations among virtual machines. Autonomic computing-based resource allocation techniques, such as the ones in can be used to dynamically move processors from one virtual machine to another.

Load Balancing: Since the software state of an entire virtual machine is completely encapsulated by the VMM, it is relatively easy to migrate virtual machines to other platforms in order to improve performance through better load balancing

Virtualization Advantages

Legacy Applications: Even if an organization decides to migrate to a different operating system, it is possible to continue to run legacy applications on the old OS running as a guest OS within a VM. This reduces the migration cost.

Points to Note

- **Software licensing**

  One of the most significant virtualization-related issues to be aware of is software licensing. Virtualization makes it easy to create new servers, but each VM requires its own separate software license. Organizations using expensive licensed applications could end up paying large amounts in license fees if they do not control their server sprawl.

- **IT training**

  IT staff used to dealing with physical systems will need a certain amount of training in virtualization. Such training is essential to enable the staff to debug and troubleshoot issues in the virtual environment, to secure and manage VMs, and to effectively plan for capacity.

- **Hardware investment**

  Server virtualization is most effective when powerful physical machines are used to host several VMs. This means that organizations that have existing not-so-powerful hardware might still need to make upfront investments in acquiring new physical servers to harvest the benefits of virtualization.

Application of Virtualization

- Today, virtualization can apply to a range of system layers, including hardware-level virtualization, operating system-level virtualization, and high-level language virtual machines.

- **Maximize resources** — Virtualization can reduce the number of physical systems you need to acquire, and you can get more value out of the servers. Most traditionally built systems are underutilized. Virtualization allows maximum use of the hardware investment.
-
- **Multiple systems** — With virtualization, you can also run multiple types of applications and even run different OS for those applications on the same physical hardware.

- **IT budget integration** — When you use virtualization, management, administration and all the attendant requirements of managing your own infrastructure remain a direct cost of your IT operation.

Technology Trends

- 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

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.