Chap-02
Virtualization

- We can implement, test, and run various large sized applications with virtualization which is not possible to implement on physical hardware alone

- Each virtual machine contains its own virtual or software based hardware, including
  - a virtual CPU,
  - memory,
  - hard disk,
  - and network interface card.

- Virtualization technology allows the creation of virtual versions of
  - hardware platforms,
  - OS,
  - networking resources,
  - storage devices.

- It supports multiple guest OS's to be run on a single physical machine.

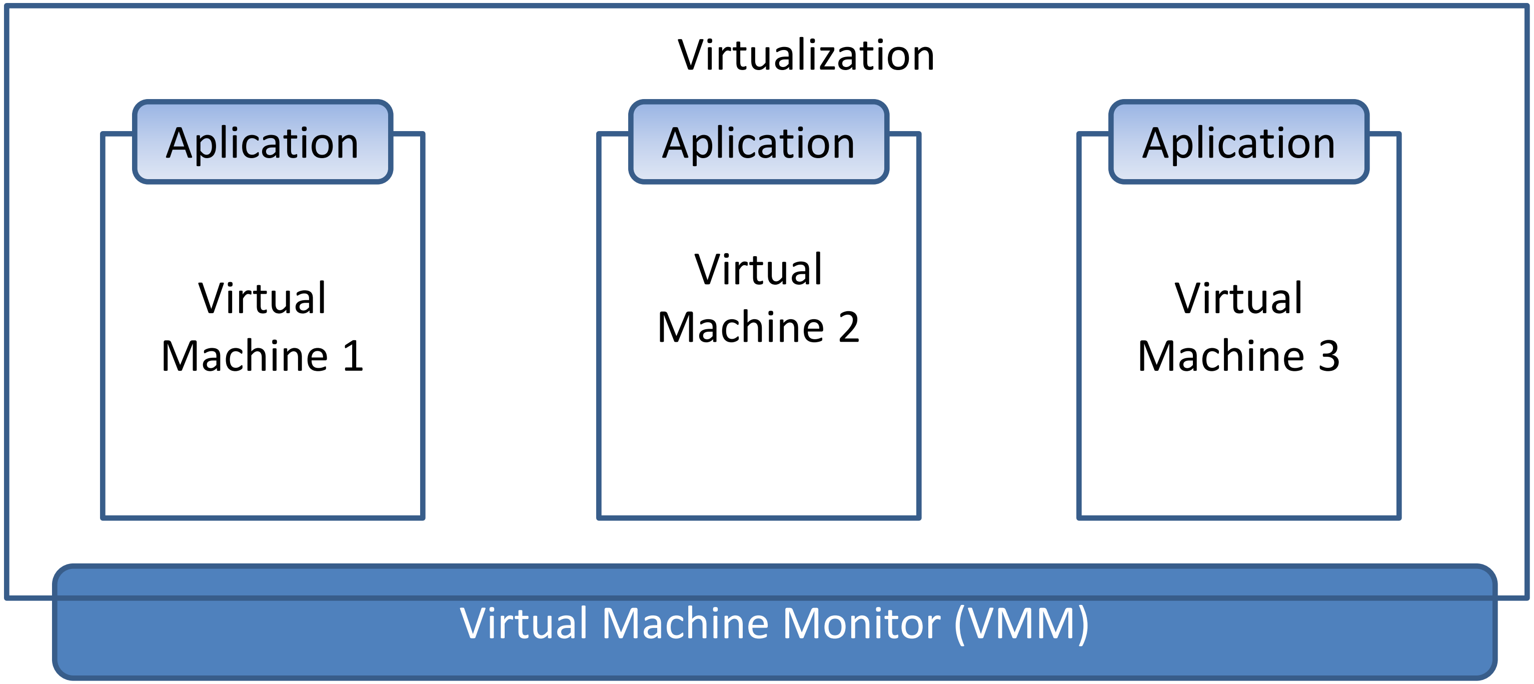
- **Virtualization**
- Virtualization technology separates the primary functions of computer like computing and technology implementation from,
  - physical infrastructure
  - and the hardware resources

  with the help of technology called Virtual Machine Monitor(VMM).

- Virtualization helps organizations save by removing the physical infrastructure to a large extent.

- Virtualization helps by taking care of capital costs that need to be invested in availing and maintaining the infrastructure.

- Typical virtualization Structure

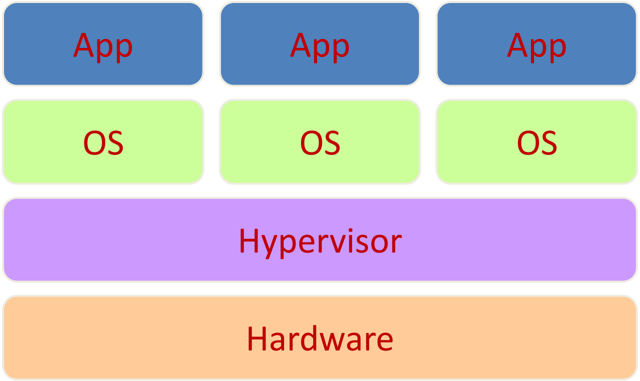


- **Hypervisor** is a software program that manages multiple operating systems (or multiple instances of the same operating system) on a single computer system.

- A **hypervisor**, sometimes referred to as a **virtualization** manager, is a program that allows multiple operating systems run on single computer system.

Virtualization

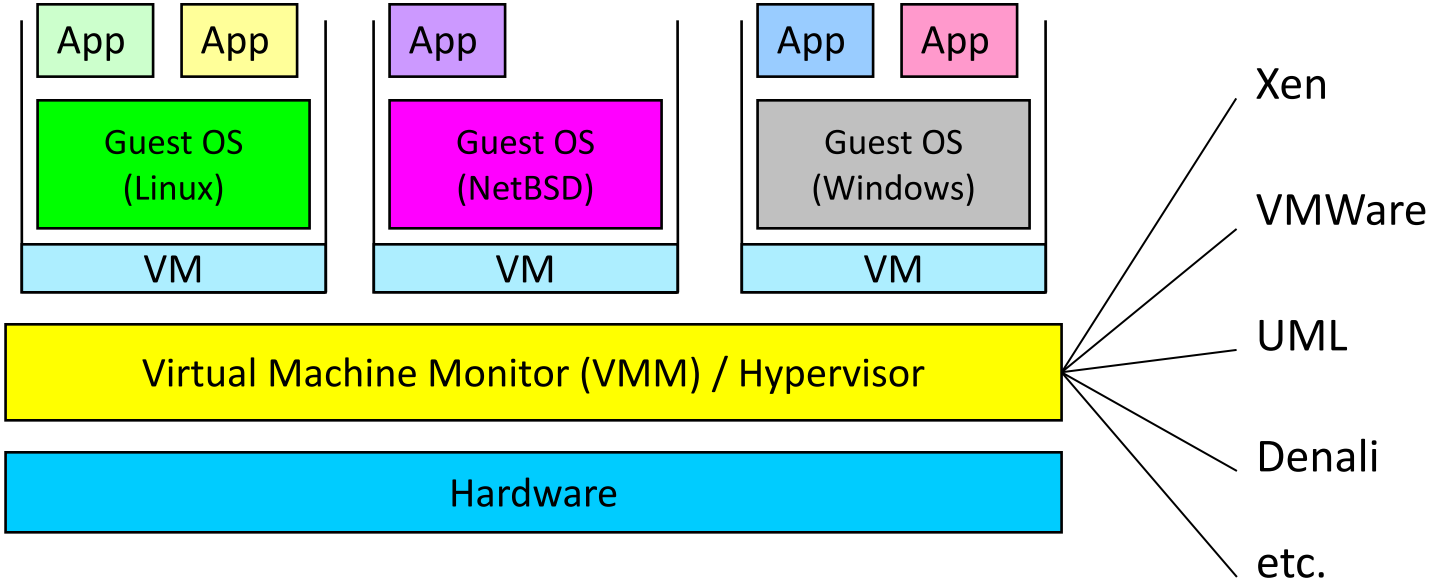
- Virtual workspaces:
  - An abstraction of an execution environment that can be made dynamically available to authorized clients by using well-defined protocols,
  - Resource quota (e.g. CPU, memory share),
  - Software configuration (e.g. O/S, provided services).
- Implement on Virtual Machines (VMs):
  - Abstraction of a physical host machine,
  - Hypervisor intercepts and emulates instructions from VMs, and allows management of VMs,
  - VMWare, Xen, etc.
- Provide infrastructure API:
  - Plug-ins to hardware/support structures



Virtualized Stack

Virtual Machines

- VM technology allows multiple virtual machines to run on a single physical machine.



Performance: Para-virtualization (e.g. Xen) is very close to raw physical performance!

What is the purpose and benefits?

- Cloud computing enables companies and applications, which are system infrastructure dependent, to be infrastructure-less.

- By using the Cloud infrastructure on "pay as used and on demand", all of us can save in capital and operational investment!

- Clients can:
  - Put their data on the platform instead of on their own desktop PCs and/or on their own servers.
  - They can put their applications on the cloud and use the servers within the cloud to do processing and data manipulations etc.

- **Virtualization benefits**

1) Maximizing Resources: Virtualization helps organizations utilize the maximum amount of required resources.

2) Reducing hardware cost: You do not require installing large servers huge disk space or expensive database because you can avail these services virtually, anytime.

3) Minimizing maintenance Requirement: Lesser is the hardware, lesser is the requirement for maintenance.

   Virtualization helps to run multiple OS on single hardware

4) Enjoying benefits of OS Services: Virtualization helps to take advantage of facility offered by OS.

   E.g. If you run OS on your personal computer but when need certain kind of service from another OS you can avail through virtualization

5) Using multiple systems: Use of multiple systems is made easy with the help of virtualization.

   VMM provides platform for more than one OS to work for the benefits of multiple computers through one.

6) Testing beta software and Maintaining Legacy Applications: Virtualization allows to install more than one OS.

   This test new release of software without running separate dedicated system for testing.

   For legacy system on which certain applications run and supported you can continue without requiring to port program to different OS

7) Increasing System Security: Individual systems that are run on virtual machines can be separated from each other.

   This helps avoid the requirement for different computers to be run on different level of security.

Virtualization

- Virtualization is process of creating virtual versions of Server, Desktop, a Storage device, an OS or Networking resource.
- A technique which allows to store a single physical instance of a resource or an application among multiple customers and/or organizations.
- It does by assigning a logical name to a physical storage and providing a pointer to that physical resource when demand
- Types
  - Hardware
  - OS
  - Server
  - Storage

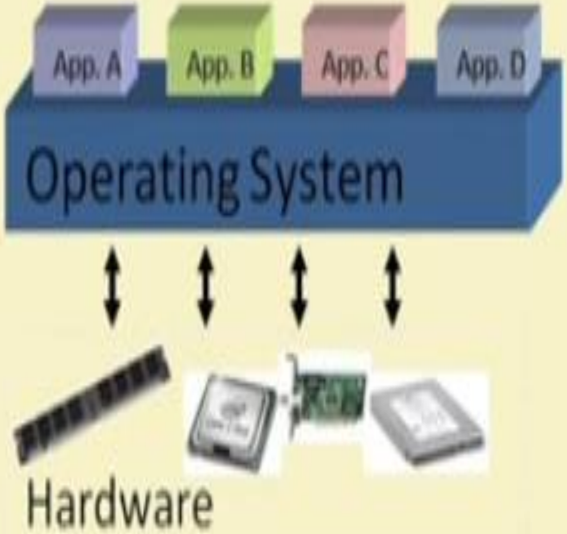
- The main goal of virtualization is to manage workloads by radically transforming traditional computing to make it more scalable.

- Virtualization has been a part of the IT landscape for decades now,

- and today it can be applied to a wide range of system layers, including
  - operating system-level virtualization,
  - hardware-level virtualization
  - and server virtualization.

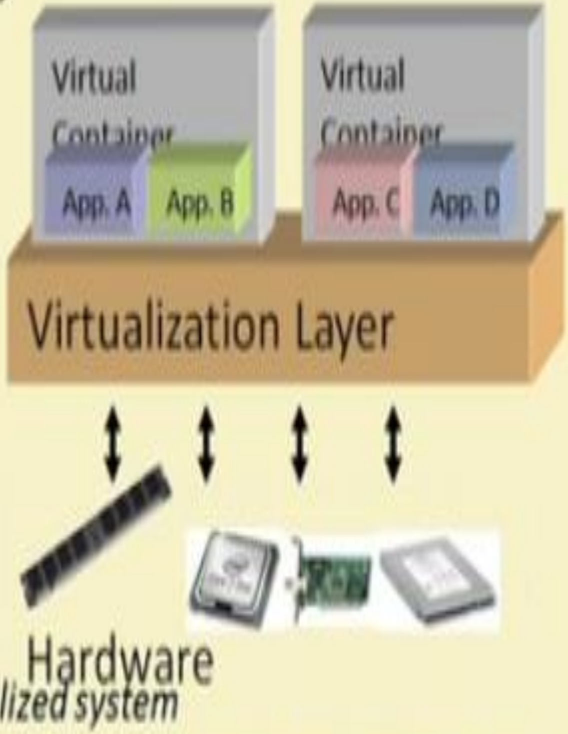
Virtualization

- Virtualization is a broad term (virtual memory, storage, network, etc)
- Focus: **Platform virtualization**
- Virtualization basically allows one computer to do the job of multiple computers, by sharing the resources of a single hardware across multiple environments



'Non-virtualized' system

A single OS controls all hardware platform resources



Virtualized system

It makes it possible to run multiple Virtual Containers on a single physical platform

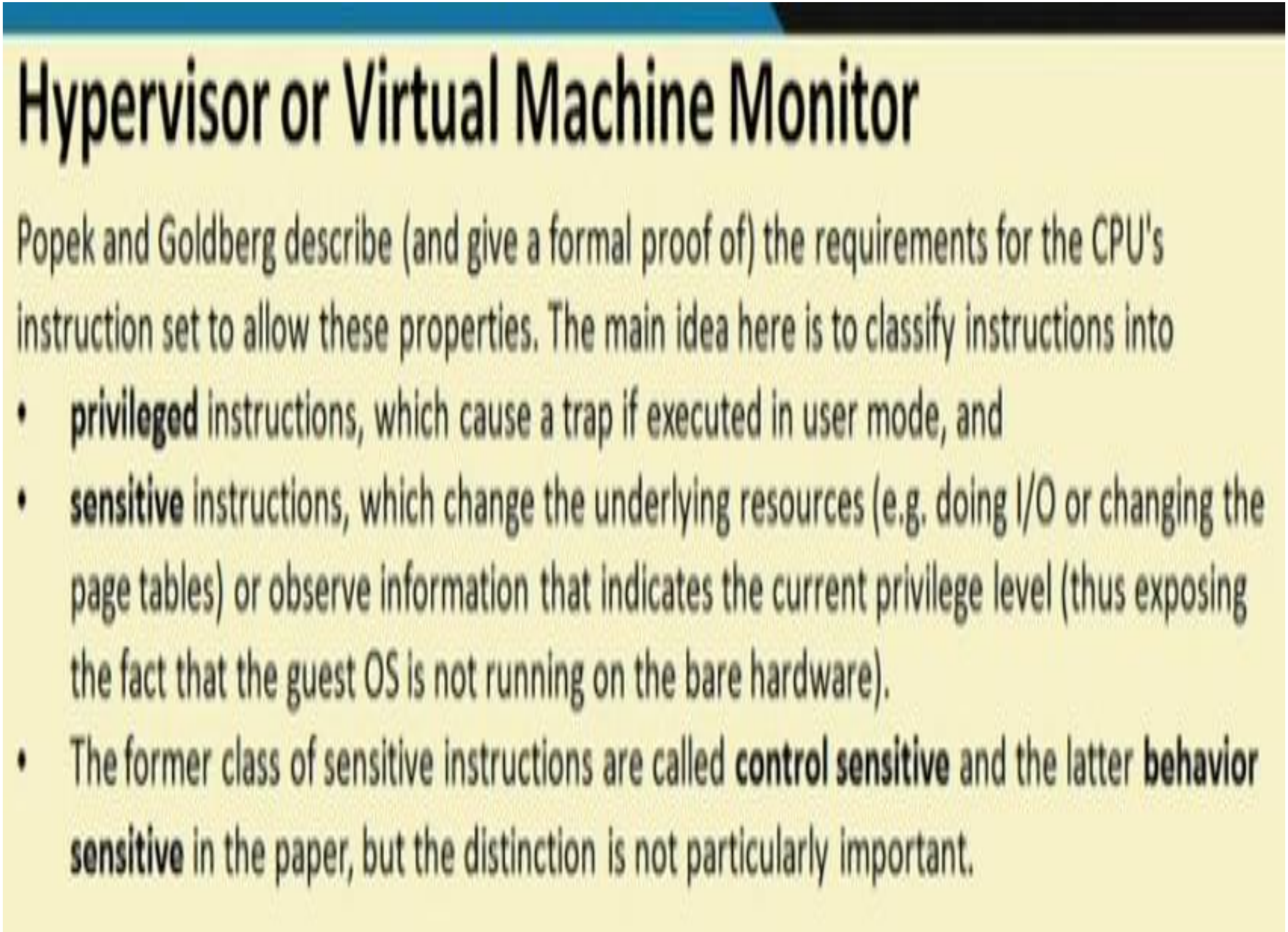
Source: www.dc.uba.ar/events/eci/2008/courses/n2/Virtualization-Introduction.ppt

Hypervisor or Virtual Machine Monitor

Research Paper :Popek and Goldberg, "Formal requirements for virtualizable third generation architectures", CACM 1974 (http://portal.acm.org/citation.cfm?doid=361011.361073).

A **hypervisor** or **virtual machine monitor** runs the guest OS directly on the CPU. (This only works if the guest OS uses the same instruction set as the host OS.) Since the guest OS is running in user mode, privileged instructions must be intercepted or replaced. This further imposes restrictions on the instruction set for the CPU, as observed in a now-famous paper by Popek and Goldberg identify three goals for a virtual machine architecture:

- *Equivalence*: The VM should be indistinguishable from the underlying hardware.
- *Resource control*: The VM should be in complete control of any virtualized resources.
- *Efficiency*: Most VM instructions should be executed directly on the underlying CPU without involving the hypervisor.



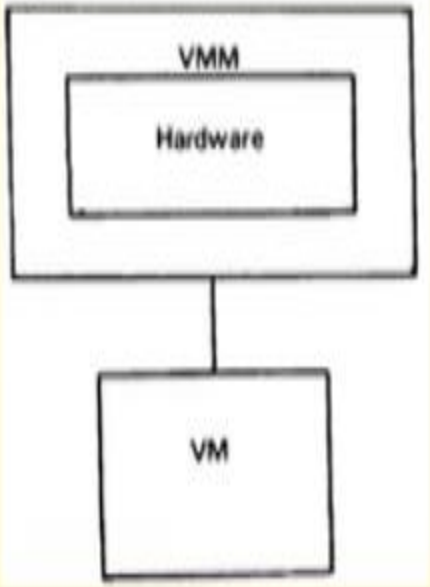
Hypervisor or Virtual Machine Monitor

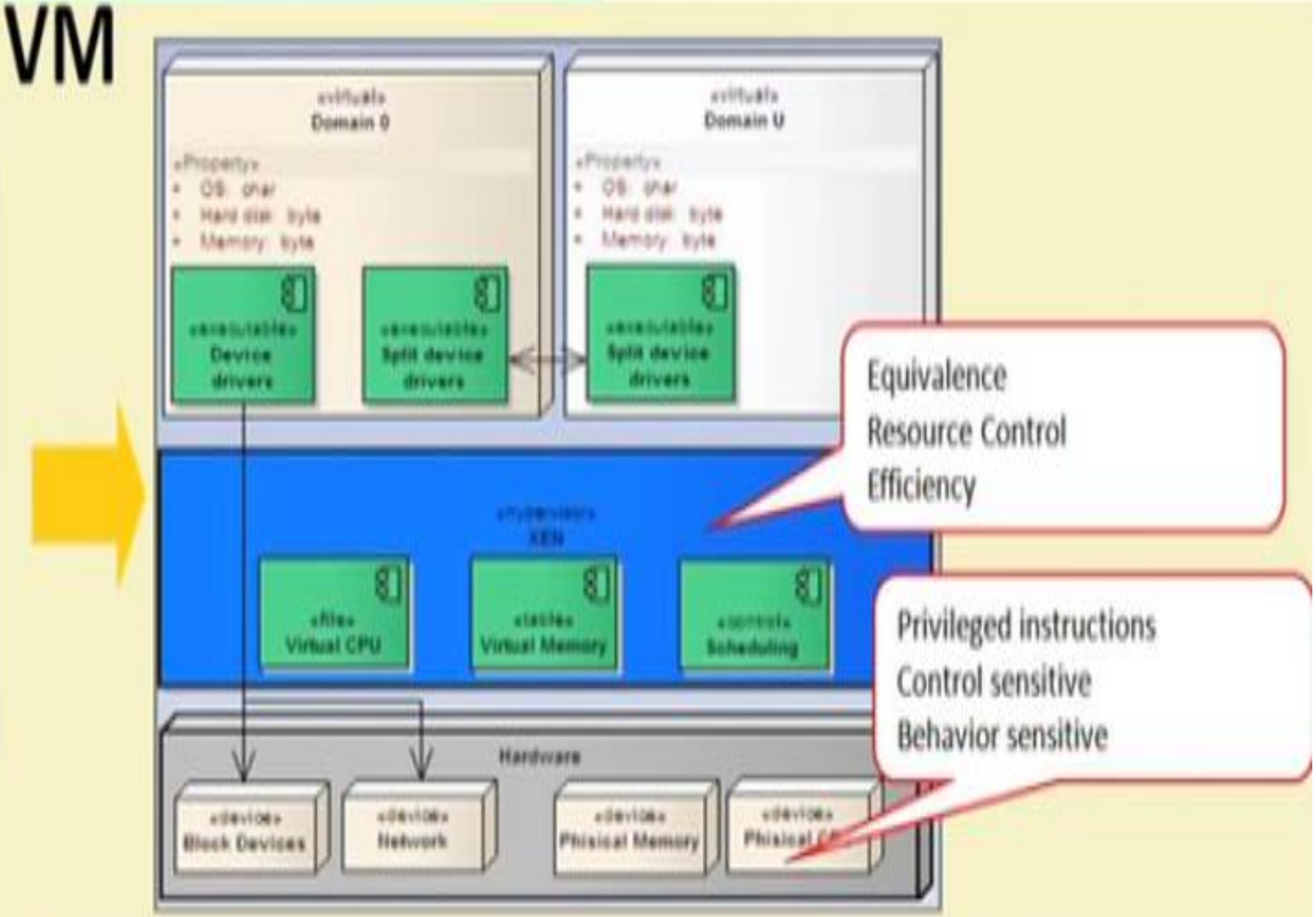
Popek and Goldberg describe (and give a formal proof of) the requirements for the CPU's instruction set to allow these properties. The main idea here is to classify instructions into

- **privileged** instructions, which cause a trap if executed in user mode, and
- **sensitive** instructions, which change the underlying resources (e.g. doing I/O or changing the page tables) or observe information that indicates the current privilege level (thus exposing the fact that the guest OS is not running on the bare hardware).
- The former class of sensitive instructions are called **control sensitive** and the latter **behavior sensitive** in the paper, but the distinction is not particularly important.

VMM and VM

Fig. 1. The virtual machine monitor.



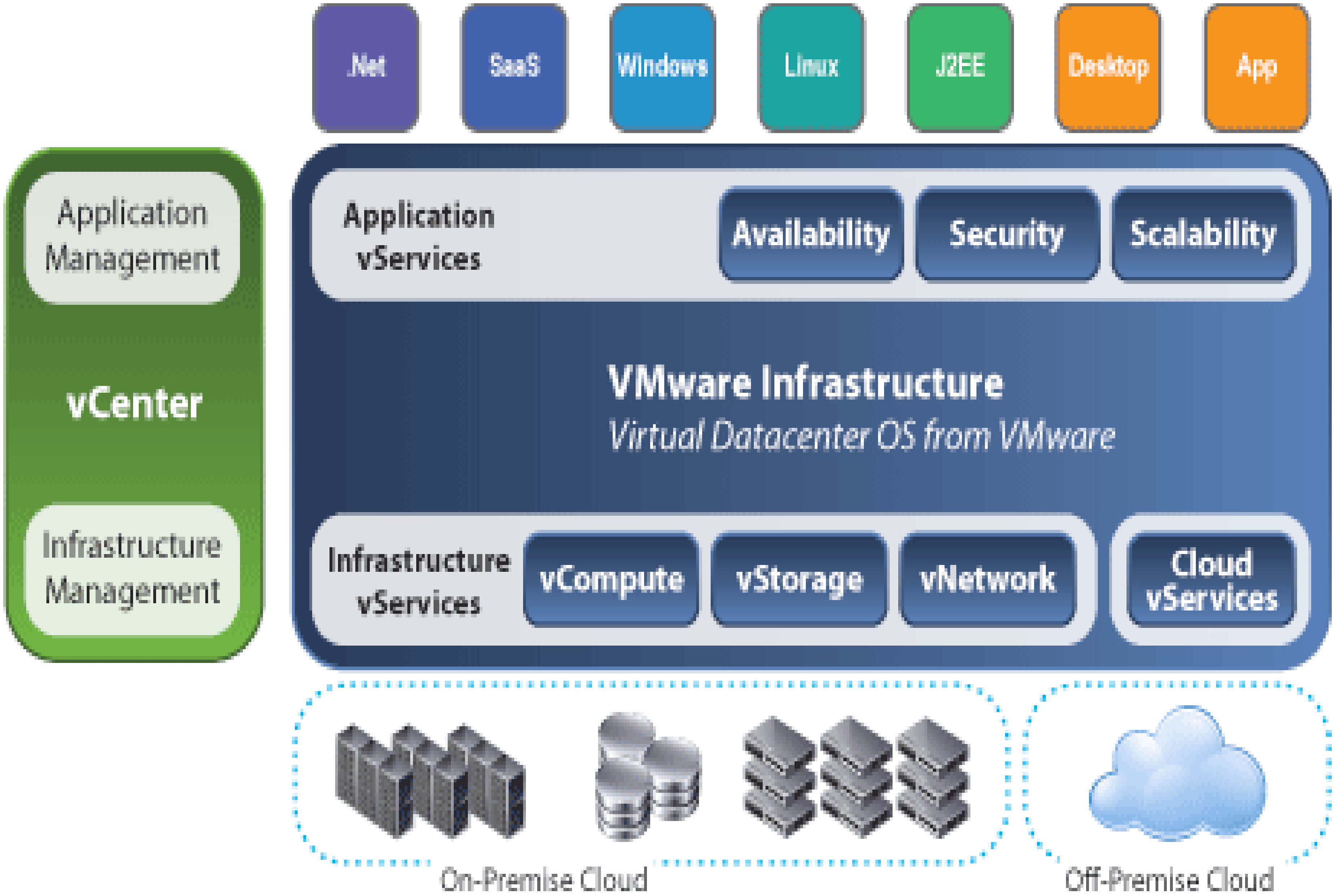


Equivalence
Resource Control
Efficiency

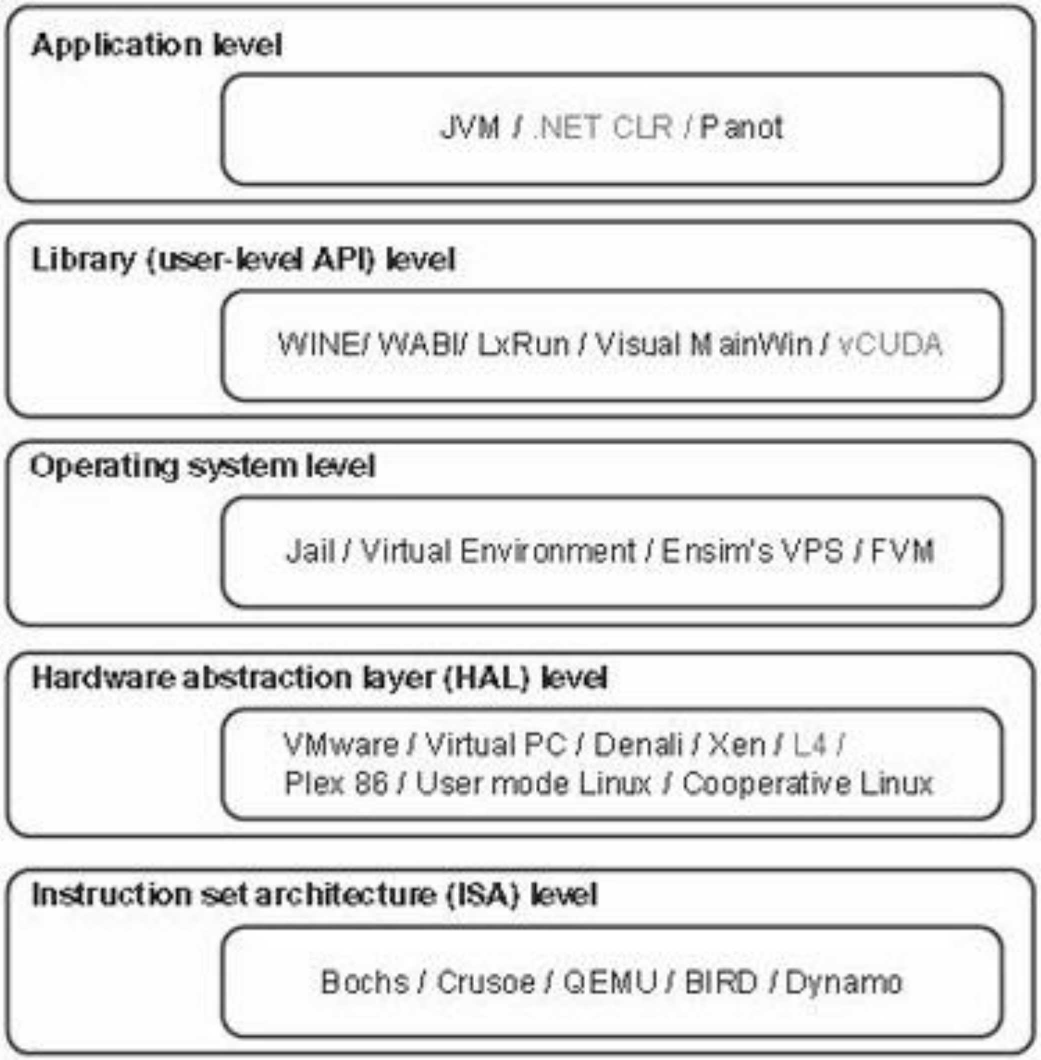
Privileged instructions
Control sensitive
Behavior sensitive

- For any conventional third generation computer, a VMM may be constructed if the set of sensitive instructions for that computer is a subset of the set of privileged instructions
- A conventional third generation computer is recursively virtualizable if it is virtualizable and a VMM without any timing dependencies can be constructed for it.

Source: www.dc.uba.ar/events/eci/2008/courses/n2/Virtualization-Introduction.ppt


.Net
SaaS
Windows
Linux
J2EE
Desktop
App
Application Management
vCenter
Infrastructure Management
Application vServices
Availability
Security
Scalability
VMware Infrastructure
Virtual Datacenter OS from VMware
Infrastructure vServices
vCompute
vStorage
vNetwork
Cloud vServices
On-Premise Cloud
Off-Premise Cloud

- Implementation Level of Virtualization



Application level

JVM / .NET CLR / Panot

Library (user-level API) level

WINE/ WABI/ LxRun / Visual MainWin / vCUDA

Operating system level

Jail / Virtual Environment / Ensim's VPS / FVM

Hardware abstraction layer (HAL) level

VMware / Virtual PC / Denali / Xen / L4 /
Plex 86 / User mode Linux / Cooperative Linux

Instruction set architecture (ISA) level

Bochs / Crusoe / QEMU / BIRD / Dynamo

FIGURE 3.2

Virtualization ranging from hardware to applications in five abstraction levels.

1. ISA Level: Emulator receives instructions convert to native instructions which are run on host machine.

2. HAL Level: Similarities between guest and host OS suitable for X86 architecture.

3. OS Level: Maintain API's and Libraries and environment and settings and other prerequisite for host and guest OS

4. Library level: Programming API and ABI (Application Binary Interface)

5. Application Level: User level program and OS are executed and apps behave like real machine. I/O map or I/O memory map is used to deal with hardware.

   Java or CLR used to write code

1) Virtualization at instruction set architecture (ISA)level :

- Virtualization is implemented at the level of instruction set architecture

- Transforms the physical architecture of the systems instruction set completely into software.

- Instructions are received by emulator ,transforms into a native instructions that are run on host machine hardware.

- Virtualization at ISA (Instruction Set Architecture) level:

- Emulating a given ISA by the ISA of the host machine.
  - e.g, MIPS binary code can run on an x-86-based host machine with the help of ISA emulation.

- Typical systems: Bochs, Crusoe, Quemu, BIRD, Dynamo

- Advantage:
  - It can run a large amount of legacy binary codes written for various processors on any given new hardware host machines
  - best application flexibility

- Shortcoming & limitation:
  - One source instruction may require tens or hundreds of native target instructions to perform its function, which is relatively slow.
  - V-ISA requires adding a processor-specific software translation layer in the complier.

- **Virtualization at Instruction Set Architecture(ISA) level**
- Every machine has an instruction set which is nothing but set is an interface between software and hardware.

- Using this instructions software can communicate with hardware.

- When virtualization is carried at this level, we create an emulator which receives all the instructions from the Virtual machines.

- for example if a virtual machine wants to access the printer then that instruction will be passed to this emulator.

- The emulator will then interpret what type of instruction it is

- Then map that instruction to the Host machine's instruction

- Then that instruction will be carried out on Host machine and the results will be passed to the emulator

- Finally emulator will return it to the virtual machine.

- This technique is simple to implement but as every instruction has to be interpreted before mapping it, too much time is consumed and performance becomes poor.

2) Virtualization at the Hardware abstraction Layer(HAL):

- Virtualization at HAL , the time spent in interpreting the instructions issued by the guest platform into the instructions of host platform is reduced,

- by taking advantage of the similarities that exist between the architectures of the system.

Virtualization at the Hardware abstraction Layer(HAL):

- Virtualization at Hardware Abstraction level:

  Virtualization is performed right on top of the hardware.
- It generates virtual hardware environments for VMs, and manages the underlying hardware through virtualization.
- Typical systems: VMware, Virtual PC, Denali, Xen

Advantage:

- Has higher performance and good application isolation

Shortcoming & limitation:

- Very expensive to implement (complexity)

- **Virtualization at Hardware Abstraction Layer(HAL) level**

- As in Virtualization at ISA level, performance is reduced due to interpretation of every instruction so to overcome that we have virtualization at HAL level.

- In this type we map the virtual resources with the physical resources.

- We don't interpret every instruction but we just check whether it is a privileged instruction or not.

- If the instruction is not privileged we simply allow normal execution because already virtual and physical resources are mapped so accessing is simple.

- But if the instruction is privileged, we pass the control to VMM(Virtual Machine Monitor) and it deals with it accordingly.

- There may be many Virtual machines running simultaneously on the same Host system
- so if privileged instructions like memory management or scheduling tasks aren't handled properly, system can crash.

- Even after many advancements still there are certain exceptions which cannot be caught by this method which is a drawback of this type of virtualization.

3) Virtualization at OS level:

- Virtualization at HAL level supports multiple OS's and applications to be run simultaneously.

- It does not require system reboot, or dual boot setup.

- It gives the appearance of having multiple separate machines each of which can be used as a normal system.

- HAL virtualization require a lot of time to be spent in the installation and administration of the virtual system before testing and running application.

- The whole process involves installation of OS, application suit, networking system etc.

- To overcome the issue of redundancy and time consumption we implement virtualization at higher level i.e. at the OS level.

- This technique include sharing of both the hardware and the OS.

- The physical machine is separated from the logical structure by separate virtualization layer.

- This layer is built on top of the base OS to enable the users to have access to multiple machines.

- The OS level virtualization technique keeps the environment required for proper running of application intact.

- It keeps the OS, the application specific data structure, the user level libraries, the environmental settings, and other requisites separately.

- The key idea behind implementing OS level virtualization is that the virtual environment (VE) remains indistinguishable from the real one.

- The virtualization layer replicates the operating environment to provide virtual environment (VE) for application by creating partitions for each virtual systems whenever demanded.

- The OS gets support from the virtualization layer

- The kind of support is also known as middleware support for virtualization.

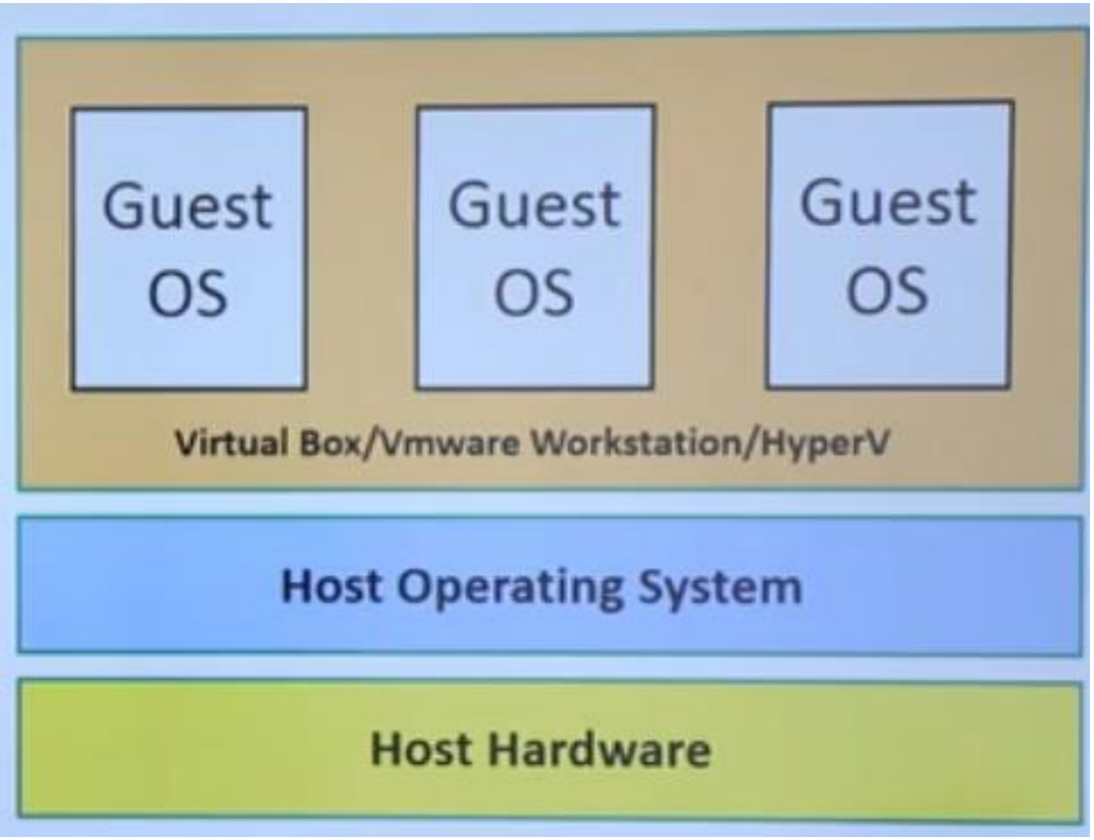
Virtualization at OS level:

- It is a type of Virtualization which work on OS layer.
- Here the kernel of OS allows more than one isolated user-space instances to exist.
- Such existence are called containers/ software containers or virtualization engines.
- In other word OS kernel will run a single OS and provide that OS functionality to replicate on each of the isolated partitions.
- Uses of OS level virtualization

Uses of OS level virtualization

- Used for virtual hosting environment.
- Used for securely allocation of finite hardware resources among a large number of distrusting users.
- System administrator uses it to integrate server hardware by moving services on separate hosts.
- To improve the security by separating several applications to several containers.
- These forms of virtualization don’t require hardware to work efficiently.

- How OS virtualization works
  - OS Virtualization done with three ways
    - Hypervisor
    - VMWare workstation
    - Virtual Box



- **Advantages of OS virtualization**
  - OS virtualization usually imposes little or no overhead
  - OS Virtualization is capable of live migration
  - It can also use dynamic load balancing of containers between nodes and a cluster.

- **Hypervisor**
  - Hypervisor is piece of computer software or hardware that creates and runs virtual machines
  - A hypervisor is a function which abstracts operating systems and applications from the underlying computer hardware
  - This abstraction allows host machine hardware to independently operate one or more virtual machines as guest.
  - Hpervisor use a thin layer of code in software or firmware to allocate resources in real time
  - We can think of hypervisor as the traffic cop that controls I/O and memory management.

- With Hypervisor we can run Linux, Windows, and macOS on a single physical x86 machine

- Each one of these virtual machines or operating systems you will be able to run its own program

- In reality it is a hypervisor that is allocating resources to the virtual machines

- Hypervisor allows you to have several virtual machines all working optimally on a single piece of computer hardware

- With Hypervisor we can run Linux, Windows, and macOS on a single physical x86 machine

- Each one of these virtual machines or operating systems you will be able to run its own program

- In reality it is a hypervisor that is allocating resources to the virtual machines

- Hypervisor allows you to have several virtual machines all working optimally on a single piece of computer hardware

- A computer on which a hypervisor runs one or more virtual machines is called a host machine and each virtual machine is called a guest machine

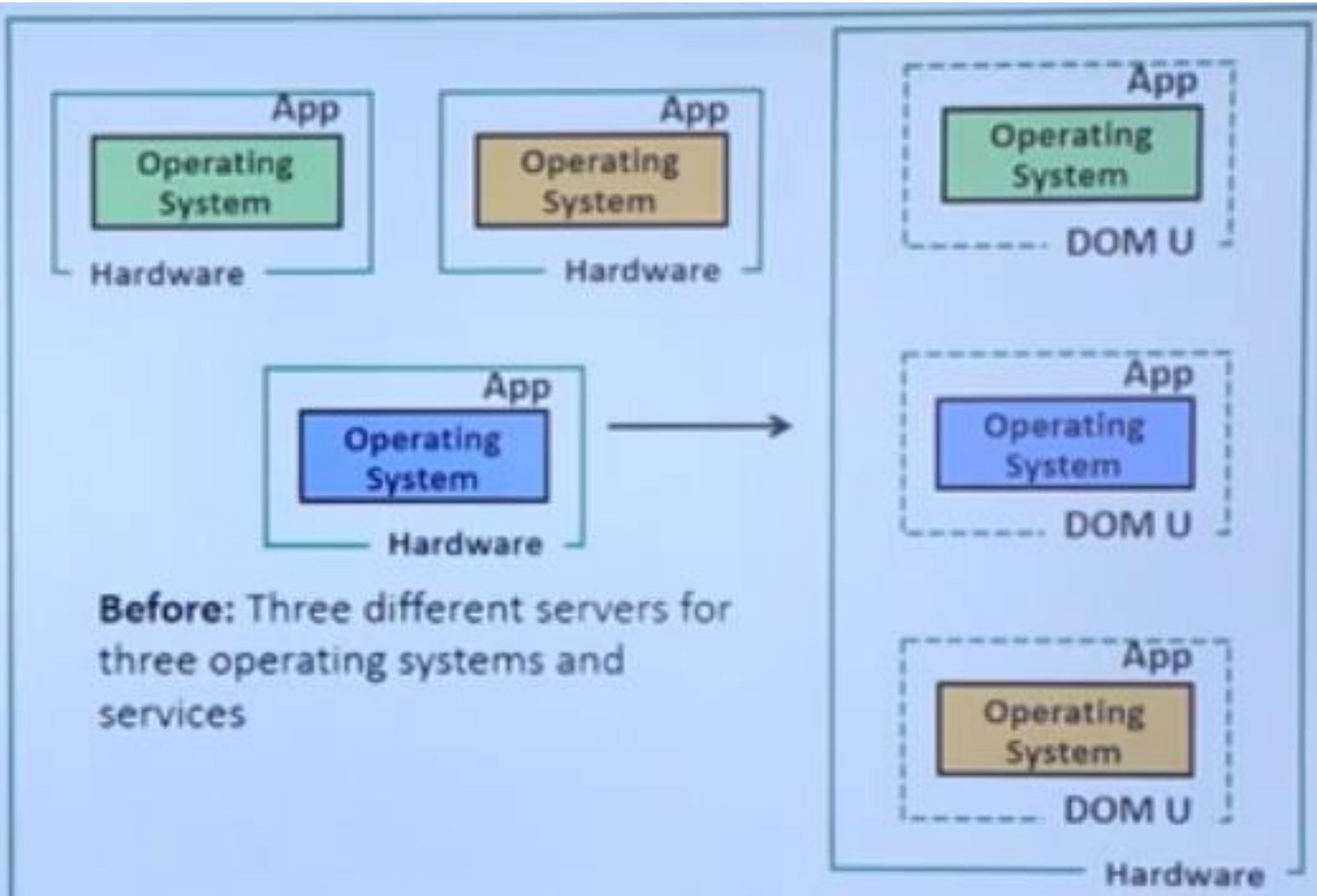
- The first hypervisors were introduced in the 1960s to allow fir different operating systems on a single mainframe computer

– It can access all physical devices residing on a server

– It can also access the memory and disk

– It can control all aspects and parts of a virtual machine

– The hypervisor is a software that can virtualize the hardware resources



Before: Three different servers for three operating systems and services

After: Only one server required for three different servers and operating systems

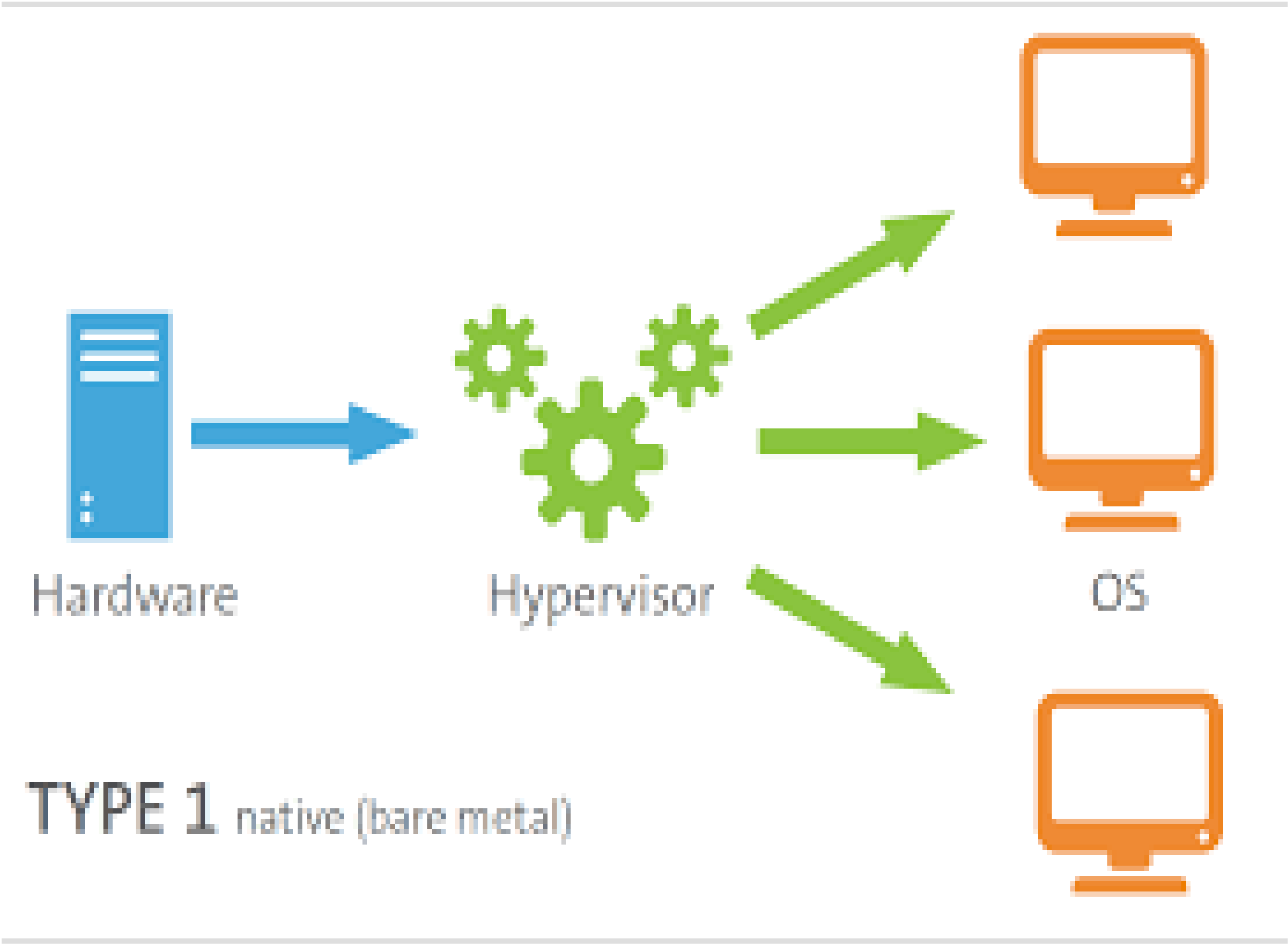
- Hypervisor Advantages
- Less Hardware
  - A hypervisor save your time and money by less hardware requirement and maintenance
  - This reduces the weight size and power consumption of the system.
- Maximum Efficiency
  - Its hard to switch between development environment for multiple works but hypervisors eliminate that need making it easier to develop as efficient as possible

- Highly Secure
  - Data Security is a major issue for everyone now a days, but virtualization provide airtight security
  - Users and software engineers can sleep easy knowing their work is safe.

- Safety
  - If one piece of software fails that is perfectly fine just lean on another instead. Redundancy is easy to built in with a hypervisor and when a partition fails it doesn’t affect any of other partitions.

- Reuse
  - When software or application is written for partition in a hypervisor that software can continue to be used with that hypervisor regardless of what hardware it is on.
  - So as long as the hypervisor works on the hardware you want it is easy to move that software to another hardware.
  - This simplifies the process of upgrading a system to new hardware.

- **Types of Hypervisor**
  - There are two types of Hypervisor
    - Type 1 Hypervisor
      - It directly run on system hardware
      - They are often referred to as a “native or “bare metal” or “embedded ” hypervisor in vendor literature
      - Type 1 Hypervisor act as an operating system and in addition it also host virtual machine
      - Type 1 Hypervisor are gaining popularity because built the hypervisor into the firmware is more efficient.
      - Examples VMware ESX and ESXi, Microsoft Hyper-V, Citrix XenServer, Oracle VM

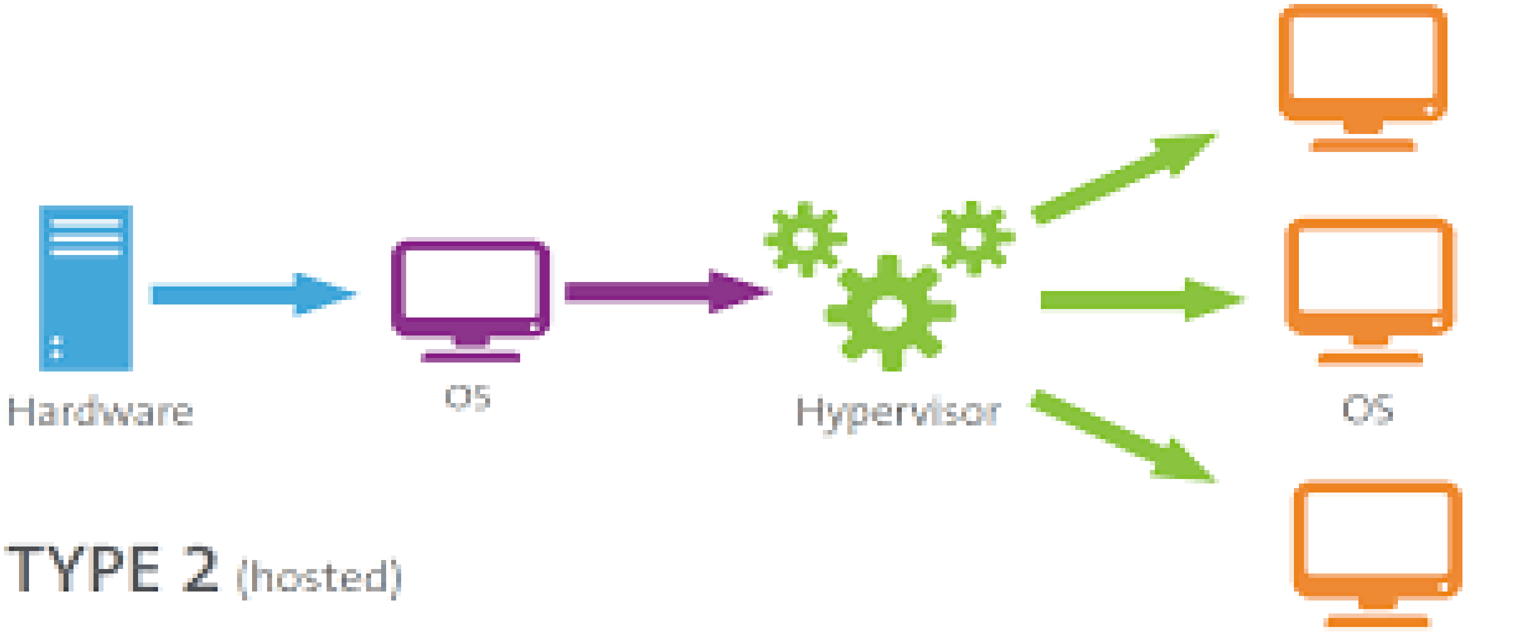


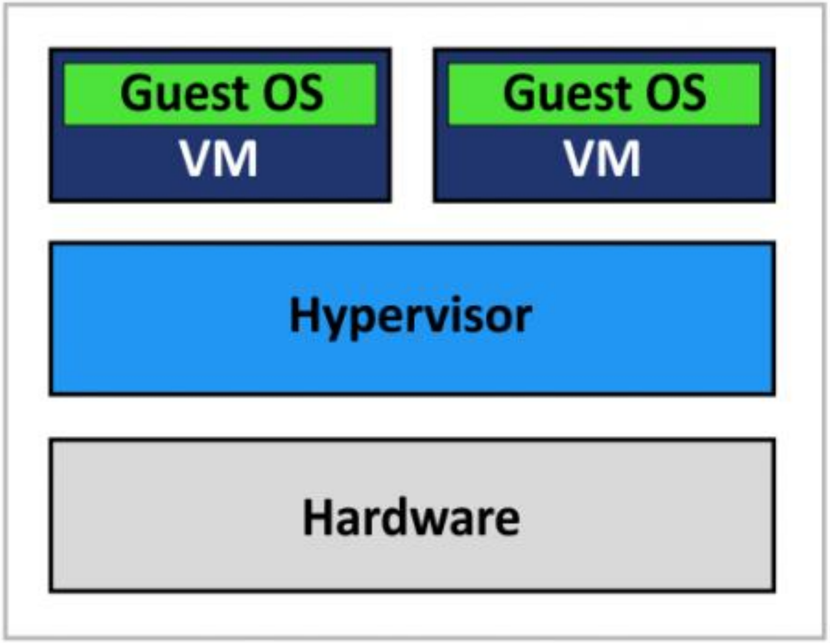
TYPE 1 native (bare metal)

- Type 2 Hypervisor
  - It runs on host server and then we can install different operating systems
  - It is also called Hosted Hypervisor
  - It run on host operating system that provides virtualization services such as I/O device support and memory management.
  - When the virtualization movement first began to take off Type 2 Hypervisor were most popular

- Administrator could buy the software and install it on a server they already had.

- Example VMware Workstation/Fusion/Player, VMware server, Microsoft virtual PC, Oracle VM virtual box, Red hat enterprise virtualization, KVM





Guest OS

VM

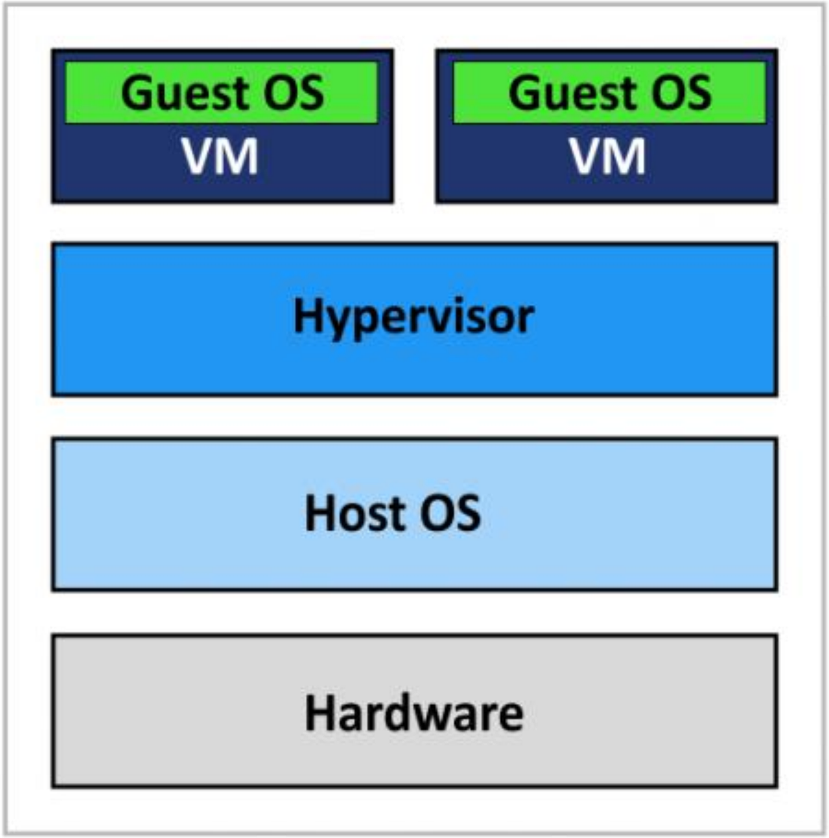
Guest OS

VM

Hypervisor

Hardware

Type 1 Hypervisor
(Bare-Metal Architecture)



Guest OS

VM

Guest OS

VM

Hypervisor

Host OS

Hardware

Type 2 Hypervisor
(Hosted Architecture)

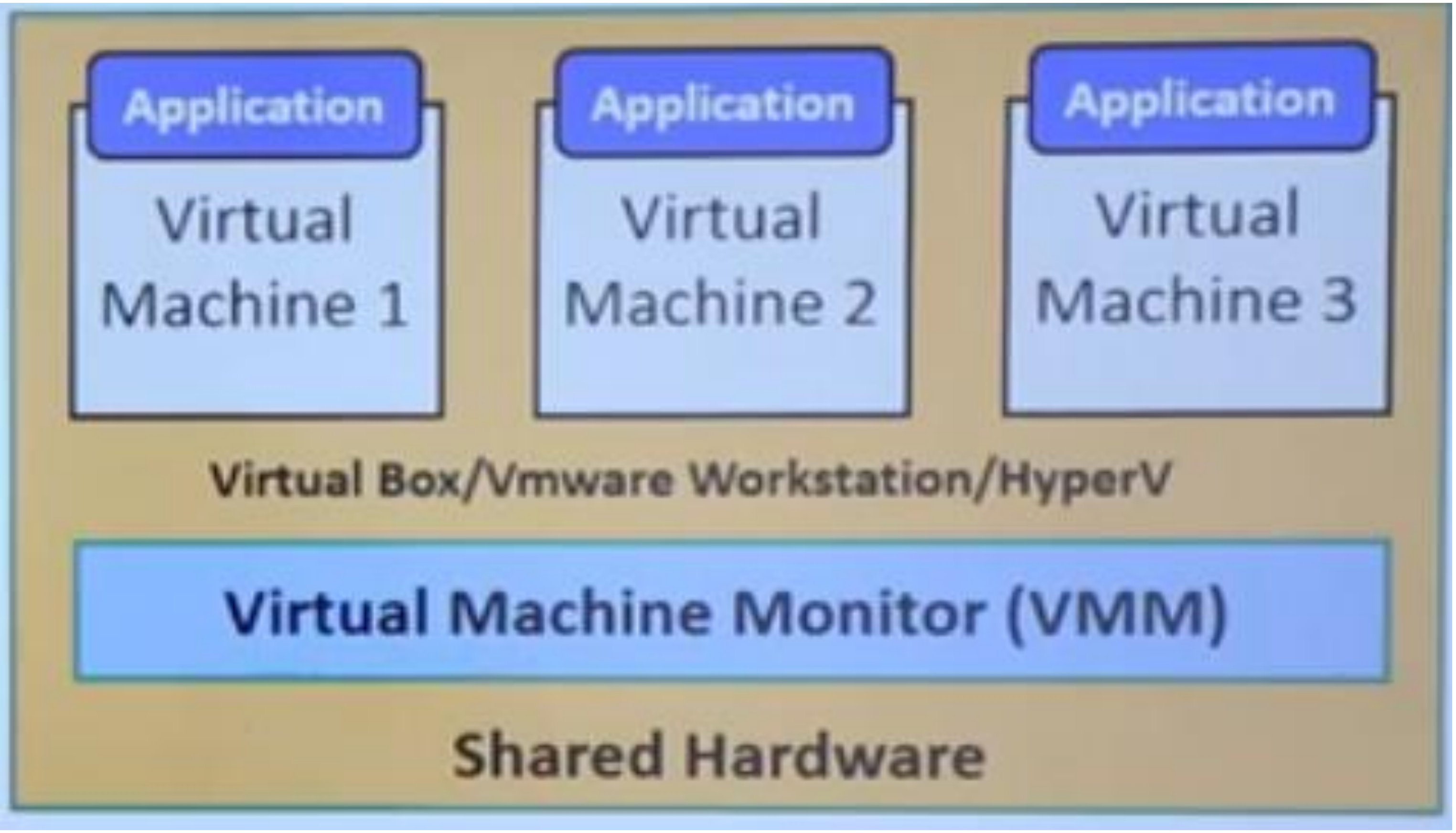
- **Hardware Virtualization**
  - It refer to creation of virtual versions of computers and Operating systems
  - This technology was developed by Intel and AMD for their server platforms and was designed to improve the performance of the processor
  - The term hardware virtualization is also known as hardware assisted virtualization.

- **Advantages of Hardware Virtualizations**
  - Hardware virtualization has many advantages because controlling virtual machines is much easier than controlling a physical machine

  - Lower cost
    - Because of server consolidation the cost decreases now it is possible for multiple OS to exist together in a single hardware. This decreases the quantity of rack space, reduces number of servers and eventually drops the power consumption.

- Efficient resource utilization
  - Physical resources can be shared among virtual machines.
  - The unused resources allocated by one virtual machine can be used by another virtual machine in case of any need.
- Increase IT flexibility
  - The quick development of hardware resources became possible using virtualization and the resources can be managed in a consistent way also.

- Types of hardware virtualization
  - Full Virtualization
  - Emulation Virtualization
  - Para Virtualization



5) Virtualization at the Application Level:

- Traditional machines execute instructions as per the definition of their ISA (instruction set architecture).

- In this technique user level programs and OS's are executed on applications that behave like real machines.

- I/O mapped input-output processing or memory mapped input-output processing is used to deal with the hardware.

- Thus an application may be taken simply as a block of instructions being executed on a machine.

- The arrival of JVM(java virtual machine) brought a new dimension to virtualization known as application level virtualization.

- The core concept behind this type of virtualization is to create virtual machine.

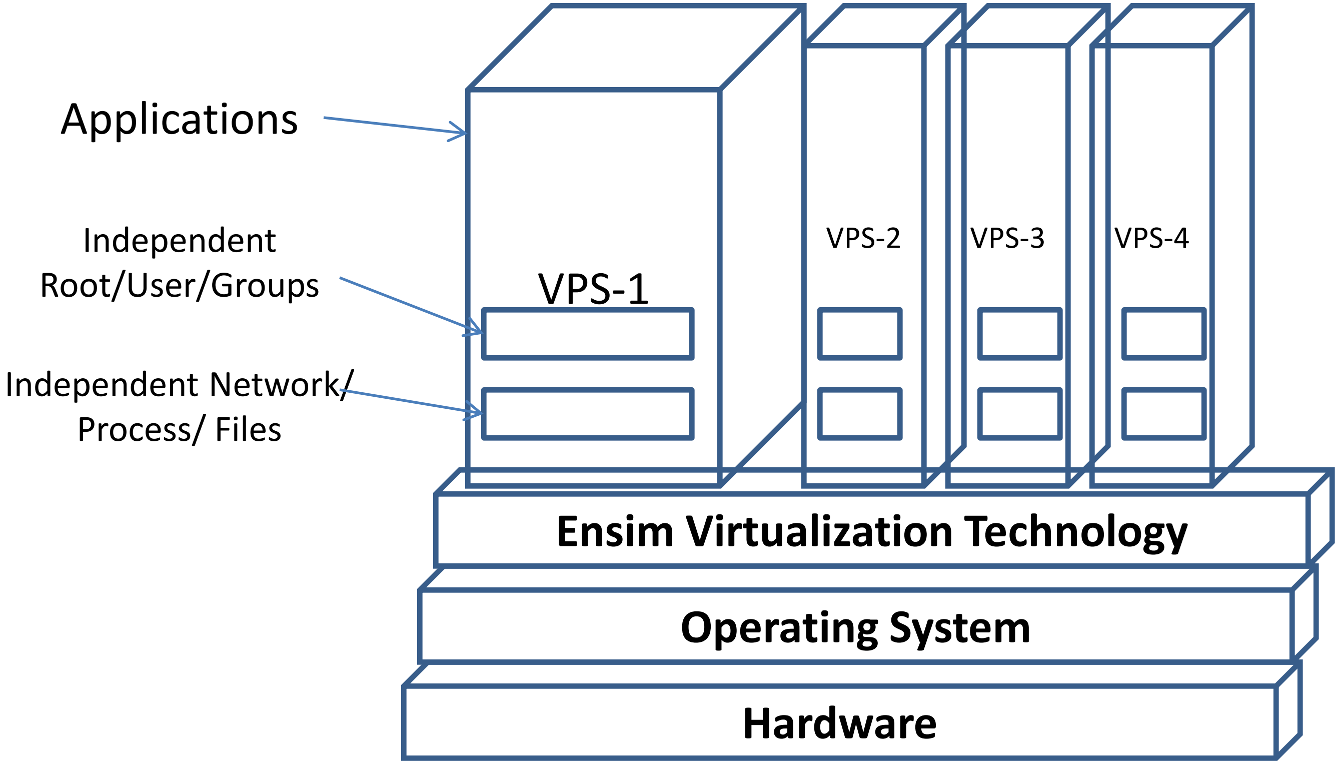
- The virtual machine, JVM, works separately at the application level.

- JVM operates in a manner similar, as a normal machine does to a set of application.

- The set of instructions for an application is defined by the machine specifically for itself.

- You can run your applications on these virtual machines as if you are running your applications on a physical machine.

- These machines must have as operating environment provides to the applications in the form of a hosted OS.
- or in the form of a separate environment of their own.


Applications
Independent Root/User/Groups
Independent Network/ Process/ Files
VPS-1
VPS-2
VPS-3
VPS-4
Ensim Virtualization Technology
Operating System
Hardware

Application Level Virtualization

- User-Application level:

  - It virtualizes an application as a virtual machine.
  - This layer sits as an application program on top of an operating system and exports an abstraction of a VM that can run programs written and compiled to a particular abstract machine definition.
  - Typical systems: JVM , NET CLI , Panot

- Advantage:
  - has the best application isolation

- Shortcoming & limitation:
  - low performance, low application flexibility and high implementation complexity.

- **Virtualization at Application Layer level**
- In this kind of virtualization Virtual machines run as an application on the Host operating system.

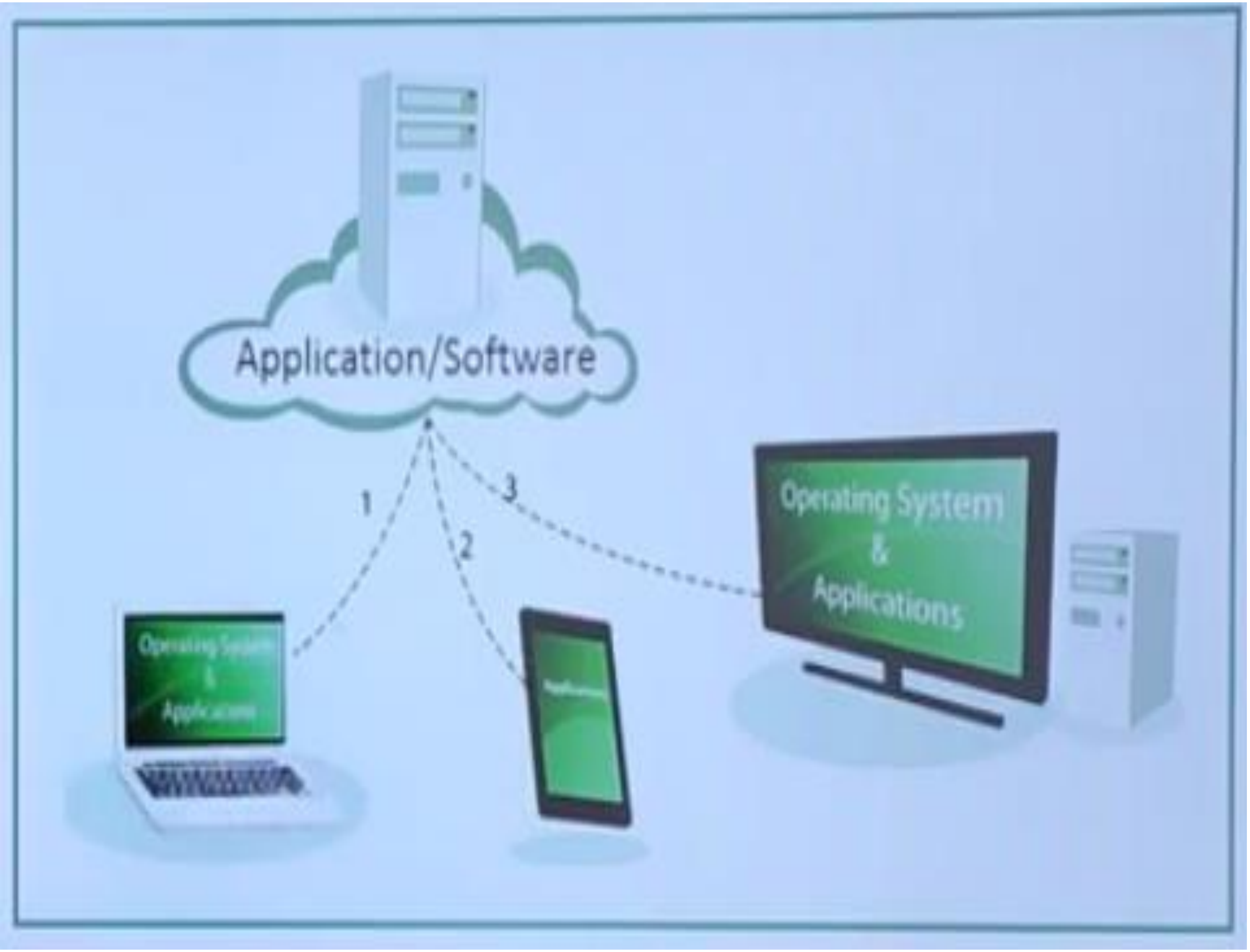
- We create a virtualization layer which is present above the Host Operating system and it encapsulates all the applications from the underlying O.S.

- While all the Applications are loaded, Host O.S. provides them with a Runtime environment.

- But virtualization layer replaces a part of this Runtime environment and gives a Virtual Environment to the Virtualized applications.

- **Application Virtualization**
  - It is also called software virtualization
  - Application virtualization is the practice of running software from a remote server
  - Software virtualization is similar to that of virtualization except that it is capable to abstract the software installation procedure and create virtual software installation.
  - Example Virtual Box, VMware etc.

- **Advantages of Application Virtualization**
  - Ease of client deployment
  - Virtual software makes it easy to link a file in a network or file copying to the workstation
  - Software migration
    - Before the concept of virtualization shifting from one software platform to another was time consuming and has a great impact on end system user. The software environment makes migration easier.
  - Easy to Manage
    - Application updates become a single task


Application/Software
1
2
3
Operating System
&
Applications
Applications
Operating System
&
Applications

- **Advantages of Virtualization**
  - Reduced Spending
  - For companies with fewer than 1000 employee up to 40 percent of an IT budget is spent on hardware
  - Purchasing multiple servers is often a good chunk of this cost
  - Virtualizing requires fewer servers and extends the lifespan of existing hardware

    This also means reduces energy cost.

- **Easier backup and disaster recovery**
  - Disasters are swift and unexpected
  - Virtualization makes recovery much faster and accurate with less manpower and a fraction of the equipment because it’s all virtual.

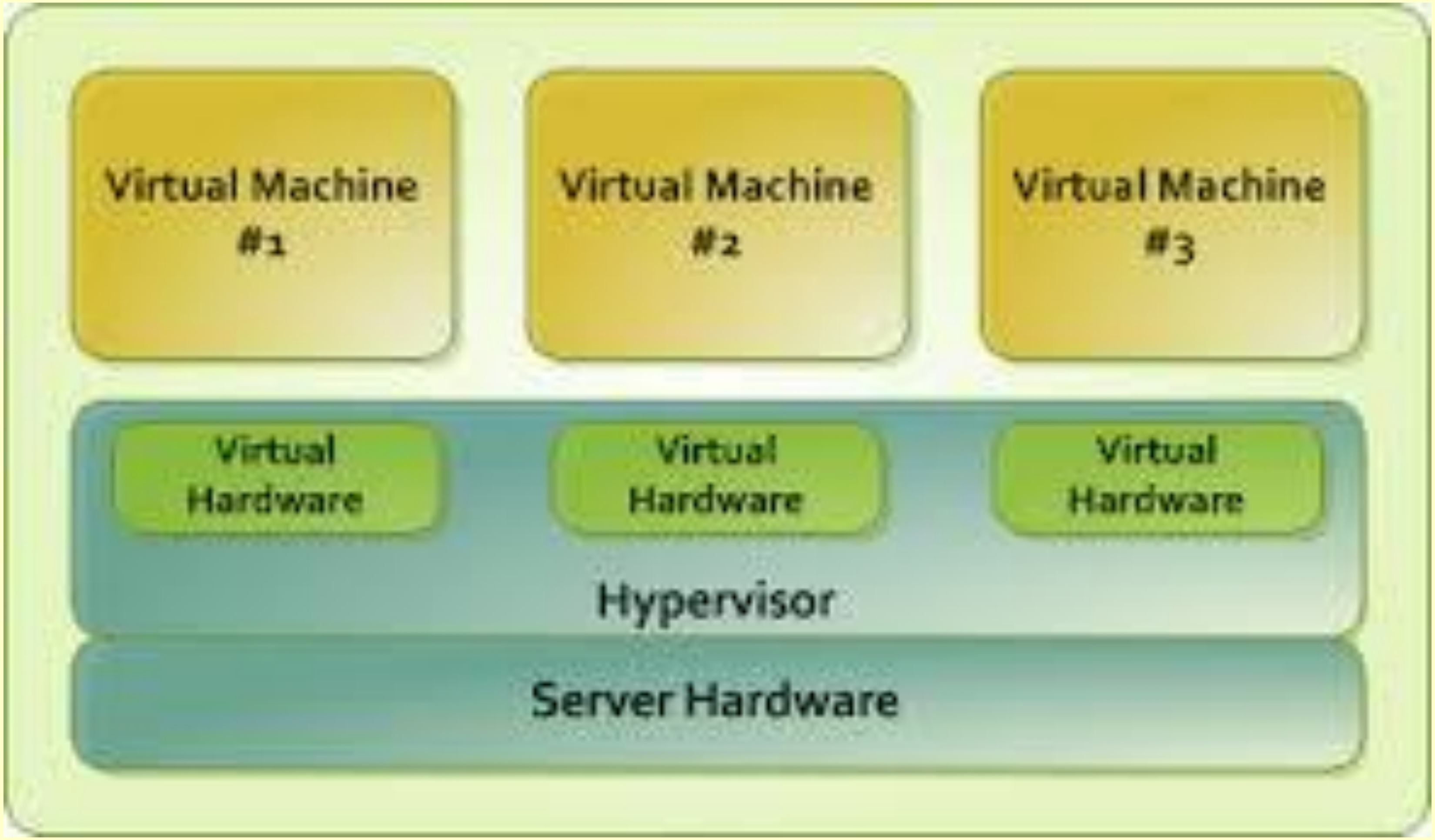
- **Better Business continuity**
  - With an increasingly mobile workforce having good business continuity is essential.

  - Virtualization gives employees access to software files and communications anywhere they are and can enable multiple people to access the same information for more continuity.

- **More efficient IT operations**
  - Going to a virtual environment can make everyone ‘s job easier especially the IT staff.
  - Virtualization provides an easier route for technicians to install and maintain software distribute updates and maintain a more secure network.
- **Software licensing**
- **Central location to manage all assets**
  - All of your virtual machines can be managed from one location.

- **Disadvantages of Virtualization**
  - It can have a high cost of implementation
    - The cost for the average individual or business when virtualization is being considered will be quite low
    - For the providers of a virtualization environment however the implementation cost can be quite high.
  - It creates a security risk
    - Information is our modern currency
    - If you have it you can make money.
    - Because data is crucial to the success of a business it is targeted frequently.
    - The average cost of a data security breach in 2017 was $3.62 million.

- It requires several links in a chain that must work together cohesively
  - If you have local equipment then you are in full control of what you can do. With virtualization, you lose that control because several links must work together to perform the same task.
  - Let’s using the example of saving a document file. With a local storage device like a flash drive or HDD you can save the file immediately and even create a backup.
  - Using virtualization your ISP connection would need to be valid. Your LAN or Wi-Fi would need to be working. Your online storage option would need to be available. If any of those are not working then you are not saving that file.
- Take Time
  - It costs user time over the long run when compared to local systems.


Virtual Machine #1
Virtual Machine #2
Virtual Machine #3
Virtual Hardware
Virtual Hardware
Virtual Hardware
Hypervisor
Server Hardware

- Virtualization at Operating System (OS) level:
  - It is an abstraction layer between traditional OS and user placations.
  - This virtualization creates isolated containers on a single physical server and the OS-instance to utilize the hardware and software in datacenters.
  - Typical systems: Jail / Virtual Environment / Ensim's VPS / FVM

- Advantage:
  - Has minimal startup/shutdown cost, low resource requirement, and high scalability; synchronize VM and host state changes.

- Shortcoming & limitation:
  - All VMs at the operating system level must have the same kind of guest OS
  - Poor application flexibility and isolation.

- **Virtualization at Operating System(O.S.) level**
- In virtualization at HAL level each virtual machine is built from scratch i.e. by installing O.S., application suites, networking systems, etc.
- In cloud sometimes we need to initialize 100 Virtual machines at a single time, If we use virtualization at Hardware abstraction layer(HAL) level this can take too much time.
- So to overcome this in Virtualization at Operating system level we share operating system between Virtual machines along with the hardware.

- So we keep the base O.S. same and install only the differences in each single Virtual machine.

- For example if we want to install different versions of windows on virtual machines(VM), you keep base O.S. of windows same and only install the differences among each VM.

- A drawback of this type is that you can install only those O.S. in VMs whose parent O.S. family is same like for example you can't install ubuntu on a VM whose base O.S. is windows.

4) Virtualization at the Programming Language Level or Library Level

- Programming the applications in most systems requires an extensive list of Application Program Interface (API).

- API's are exported by implementing various libraries at the user-level.

- These API's are used to save users from the minute details entailed with programming related to the OS.

- It enable programmers to write program easily.

- At the user level library implementation a different VE is provided in this kind of abstraction.

- This VE is created above the OS layer, which can expose a different class of binary interfaces altogether.

- **Virtualization at Library Level or Programming language level**
- When developers develop certain applications, they save the user from all the coding details by providing them Application User Interface(API).

- This has given a new opportunity for virtualization.

- In this type, we use Library Interfaces to provide a different Virtual Environment(VE) for that application.

- In short we provide user with an emulator with which user can run applications of different O.S.s.

- Example of this is the WINE tool which was used mostly by mac users to play Counter Strike 1.6 game which was only available for windows in the start.

– Techniques that implement virtualization at OS level

– Jail:

- A free BSD-based software capable of partitioning of OS environment.
- The scope of request made from users with privilege is limited to the jail itself.
- It allows the management capabilities to be delegated by the system administrator to each virtual machine environment.
- Process that runs in a partition is called “ in-jai-process”.
- No process would be in-jail process on a system boot after installing a system as fresh.

- A process and all it’s descendants would be in-jail after you place the process in jail.

- More than one jail does not access the same process.

- A privileged process creates the jail by invoking a special system called “jai(2)” .

- A new jail would be created on every system call to jail(2).

- A new process would enter the jail by only one process.

- i.e to have another process in the jail for inheriting access to the jail.

- The jail can never be left by processes that create the jail or that are created in a jail.

– Linux Kernel Mode Virtualization:

- A work similar to jail is the Linux VE system.
- The aim of this system is to allowing a computer to have multiple independent application environments run by the administrators.
- Proper boundaries are maintained within the environment.
- It also aims to improve the security of the system and enables application hosting.
- The administration of the environment from the inside is also permitted by this virtualization technique.
- Restricting the changes to be kept within the VE.
- The un-natural and not so suitable relationships between the file system roots and IP addresses, which are part of jail implementation avoided in Linux Kernel mode virtualization

– Ensim:

- To consolidate servers, reduce costs, and increase efficiency in managing and selling websites a similar type of technique is used by Ensim Virtual Private Server(VSP).

- The native OS of the server is virtualized by the Ensim VPS with the objective of partitioning the OS into separate environments that can be used for computational purposes.

- These separate environments are known as virtual private servers and the independent operation of these servers makes the complete Ensim VPS.

- The OS views the VPS as an application,

  where as the application view the VPS as the native OS resulting into the VPS appearing and operating in the form of a physical server for users.

- The Ensim VPS is implemented strongly than other two virtualization techniques because the VPS lets the administrator allocate the hardware resources as desired.

- The resources can also be adjusted and in case of requirement the VPS can transparently be moved to another physical machine .

- A seamless cross machine transfer can be accomplished by a centralized Ensim SeverXchange.

Comparison between the Implementation Levels of Virtualization

Relative Merits and Demerits of Different Levels of Virtualization Implementation				
Implementation Level	Performance	Application Flexibility	Implementation Complexity	Application Isolation
ISA	Very Poor	Excellent	Medium	Medium
HAL	Excellent	Medium	High	Very Good
OS-Level	Excellent	Low	Medium	Very Poor
Library-Level	Medium	Low	Low	Very Poor
Application Level	Poor	Low	High	Excellent

- Virtualization design Requirements
  - Equivalence Requirement
  - Efficiency Requirement
  - Resource Control Requirement

• Virtualization Structure

- Hypervisor
  - A hypervisor is a hardware virtualization technique allowing multiple operating systems, called guests to run on a host machine. This is also called the Virtual Machine Monitor (VMM).

- Type 1: bare metal hypervisor
  - sits on the bare metal computer hardware like the CPU, memory, etc.
  - All guest operating systems are a layer above the hypervisor.
  - The original CP/CMS hypervisor developed by IBM was of this kind.

- Type 2: hosted hypervisor
  - Run over a host operating system.
  - Hypervisor is the second layer over the hardware.
  - Guest operating systems run a layer over the hypervisor.
  - The OS is usually unaware of the virtualization

- Hypervisors are virtual machine monitor(VMM) that enables numerous virtual operating systems to simultaneously run on a computer system.

- These virtual machines are also referred as guest machines and they all share the hardware of the physical machine like memory, processor, storage and other related resources.

- This improves and enhances the utilization of the underlying resources.

- The hypervisor isolates the operating systems from the primary host machine.

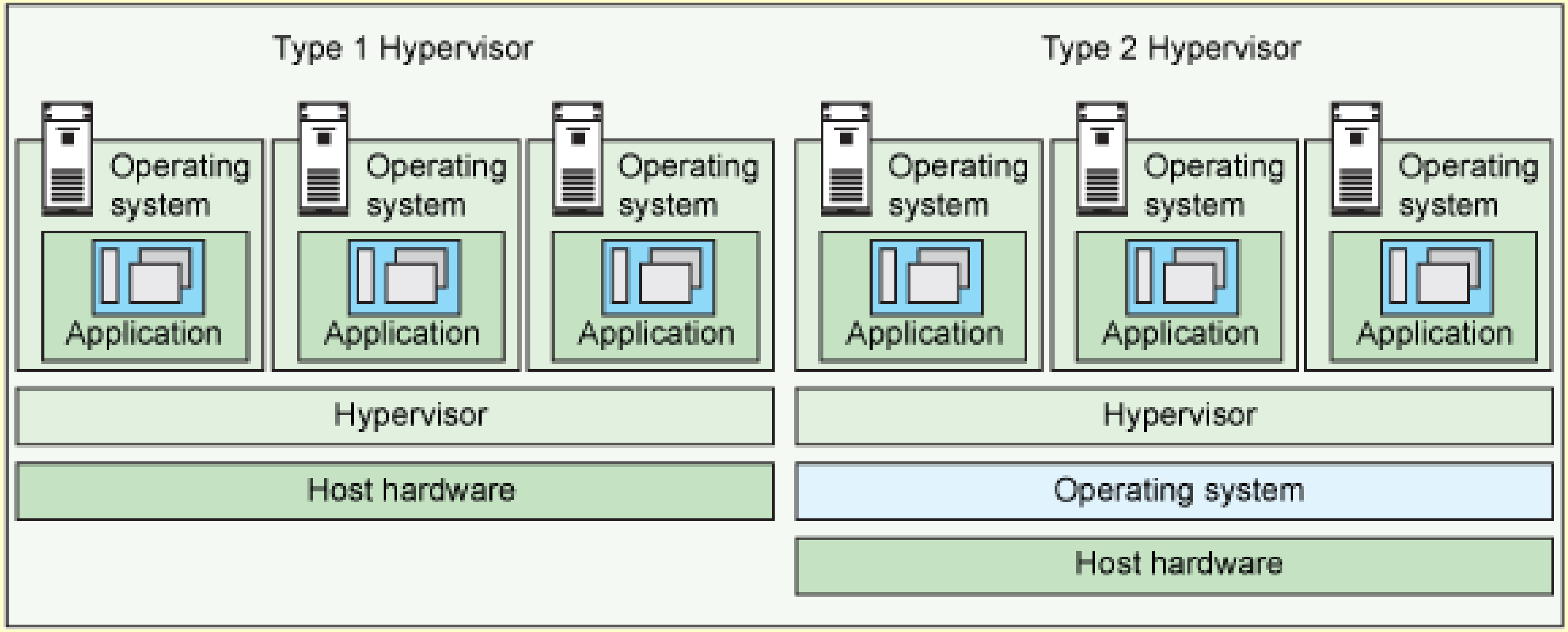
- The job of a hypervisor is to cater to the needs of a guest operating system and to manage it efficiently.

- Each virtual machine is independent and do not interfere with each another although they run on the same host machine.

- They are no way connected to one another.

- Even at times one of the virtual machines crashes or faces any issues, the other machines continue to perform normally.

- Hypervisors are divided into two types:
  - Type one is the bare-metal hypervisor that are deployed directly over the host's system hardware without any underlying operating systems or software.
    - Some examples of the type 1 hypervisors are Microsoft Hyper-V hypervisor, VMware ESXi, Citrix XenServer.
  - Type two is a hosted hypervisor that runs as a software layer within a physical operating system.
    - The hypervisor runs as a separate second layer over the hardware while the operating system runs as a third layer.
    - The hosted hypervisors include Parallels Desktop and VMware Player.


Type 1 Hypervisor
Operating system
Application
Operating system
Application
Operating system
Application
Hypervisor
Host hardware
Type 2 Hypervisor
Operating system
Application
Operating system
Application
Operating system
Application
Hypervisor
Operating system
Host hardware

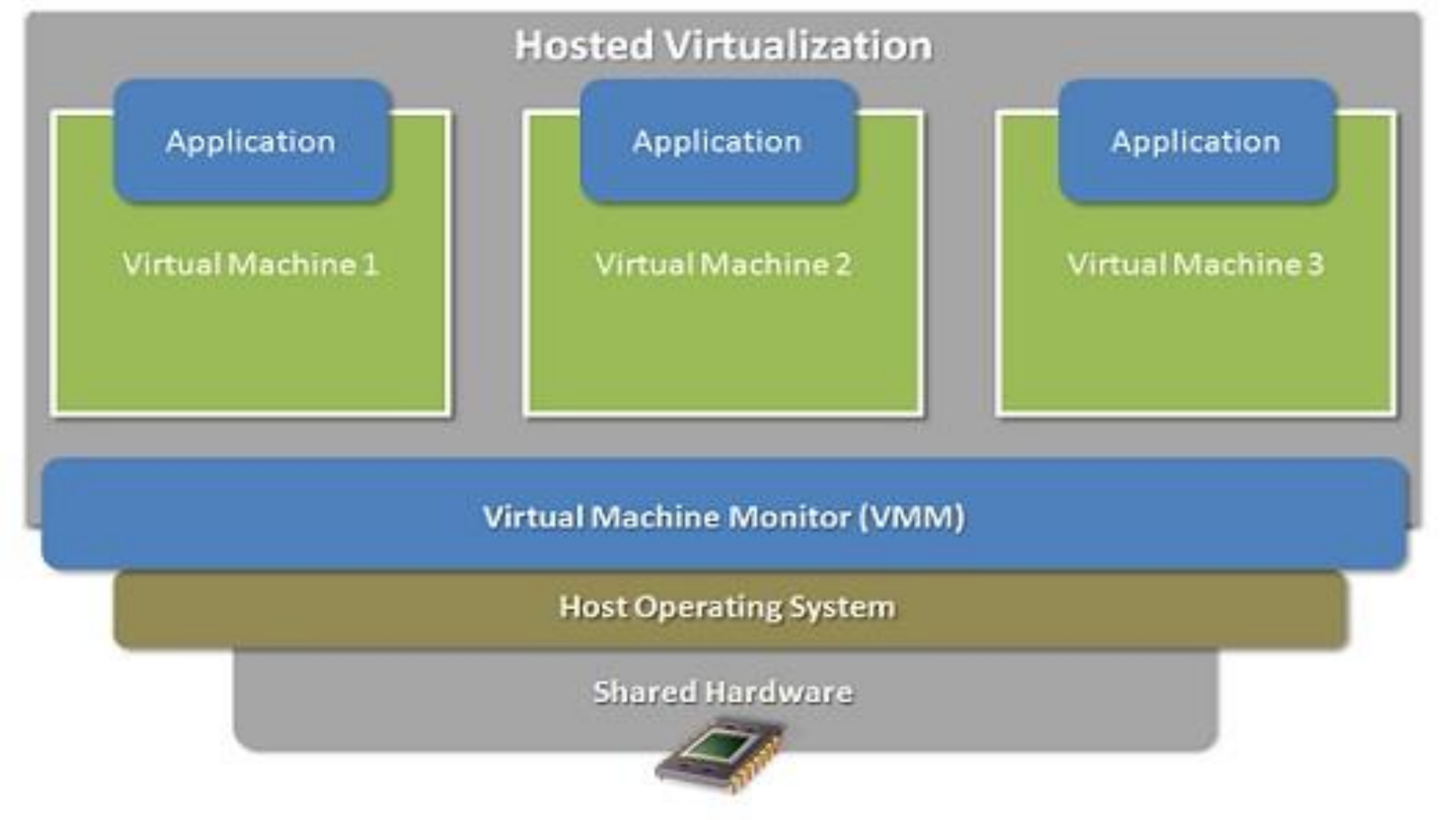
- Virtualization Structure
  - Virtualization is achieved through the software known as the Virtual Machine Monitor or the Hypervisor.

  - Virtualization Software is used in two ways
    - Hosted Virtualization (Type 1)
    - Bare-Metal Virtualization (Type 2)

1) Hosted Structure

- The Hosted Virtualization Structure enables you to run various guest application windows, of your own on top of a base OS with the help of the VMM.

- One of the most popular base OS's is the x86 OS of Windows.



Hosted Virtualization Structure

- I/O access in Hosted Structure

- The virtual OS in this virtualization structure have limited access to the I/O devices.

- You can use only a definite subset of I/O devices with your guest system.

- The I/O connections to a given physical system are owned by the host system.

- Only I/O's emulated view is presented by the VMM to every single guest machine running on the same base system.

- Emulation of only generic devices, such as Network Interface Card(NIC) and CD-ROM drivers, is possible in this structure.

- Non-generic devices like communication device, modem, do not update VMM about themselves

- Therefore it is not possible for the VMM to provide any view of non-generic devices to virtual machine.

- A pass through facility provided in hosted virtualization solution enables individual virtual machine to access the USB devices directly from the port.

- E.g. you can acquire data directly from your guest system by accessing an NI-USB data acquisition device.

- Hosted virtualization structure takes into account number software component for making the I/O access possible.

- E.g. Vmware workstation uses,
  - Low level VMM
  - Driver
  - VMApp
  - User application component

- Benefits and Drawbacks
- In hosted virtualization structure,
  - Multiple guest systems are easily installed, configured and run.

- Hosted Structure is incapable of providing a pass-through to many I/O devices.

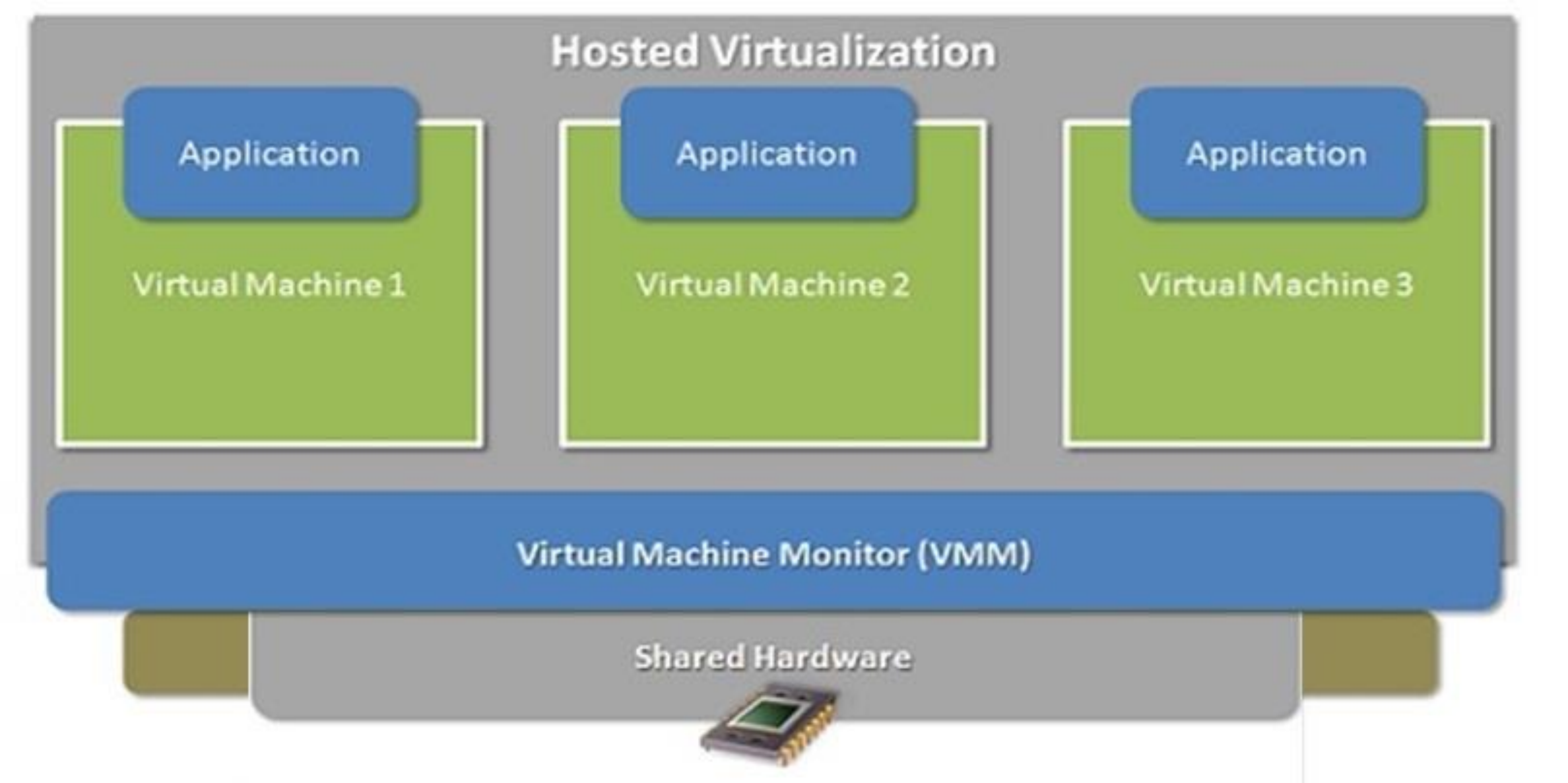
- The performance of the hosted system may be downgraded
- because the I/O requests made by the guest systems must be passed through a host OS.

- A real time OS is not supported in this structure.

- The reason for this the full control of the host OS over scheduling amongst it’s applications and the VMM.

2) Bare-Metal Structure

- Bare-metal structure is the second common type of virtualization
- VMM is installed to establish direct communication with the hardware that is being used by the base system.
- VMM does not rely on the host system for pass-through permission.



- I/O Access
- In the bare metal virtualization technique you have several options to access I/O devices from the guest systems.
- The host OS is not relied upon, so the VMM can have direct communication with the I/O devices in the bare metal virtualization structure.
- The shared usage of I/O devices between the virtual systems require the hypervisor
- This hypervisor have a low level driver that will communicate with the device.
- Hypervisor is mandated to have the capability of emulating the shared devices for the guest virtual machines.

- Apart from direct accessing the I/O devices,
- Partitioning is another method through which I/O devices can be approached by the hypervisor.

- Partitioning involves assigning individual I/O devices to particular virtual machines.

- Partitioning helps largely to improve the performance of the I/O system.

- The VM intervention is also kept at a minimum,
- because guest systems access the partitioned I/O devices directly through their native drivers.

- Benefits and Drawbacks
- With the bare-metal virtualization,

- I/O performance improved by I/O device partitioning between separate virtual systems.

- Also you can run a real time OS on systems on bare-metal virtualization structure.

- The VMM of bare-metal type may be used for binding the interrupt latency
- and enabling deterministic performance, because the host OS is not relied upon.

- A single hardware platform can be used to run real time and general purpose OS’s in parallel with the bare-metal virtualization

- The drawbacks are associated with bare-metal virtualization structure as,
  - The hypervisor must include supporting drivers for hardware platform

  - It is harder to install the VMM’s in a bare-metal structure rather than in the hosted structure, because they are not installed on top of a base OS

- **Virtualization Mechanisms**
  - Binary Translation
    - Virtual machines issues privileged instructions, contained within their compiled code, for the VMM's to handle.
    - The VMM takes control on these instructions and changes the code which is under execution appropriately,
    - So that any impact on the state of the system can be avoided.
    - The binary translation method is used by the VMM.
    - This will directs I/O requests to the appropriate system thereby preventing individual virtual machines from causing any conflicts.
    - Binary translation is mainly used with a hosted virtualization structure. E.g. VMWare workstation.

- Switching the control between virtual machines and VMM’s result in a degradation in the performance.

- To overcome this the virtualization, software processes a group of instructions simultaneously.

- The impact on the performance of the system can be reduced by
  - reducing the number of times the VMM interferes with the virtual machine execution.

Binary Translation

- Binary translation is one specific approach to implementing full virtualization that does not require hardware virtualization features.

- It involves examining the executable code of the virtual guest for "unsafe" instructions, translating these into "safe“ equivalents, and then executing the translated code.

- A key thing to is that the goal that’s pursued by Vmware is to run unmodified guest operating systems.

- Meaning that we don’t need to install any special drivers, or policies or otherwise to change the guest OS in order to run in virtualized environment.

- As a startup they couldn’t tell Microsoft to modify Windows so that Vmware can improve its success rate.

- So this type of virtualization where the guest OS is not modified is called full virtualization.

- The basic approach consist of the instruction sequences that are about to be executed are dynamically captured from the VM binary

- and this is typically done at some meaningful granularity like a basic block such as a loop or a function.

- Now the reason that is done dynamically verses statically, is because the exact execution sequence may depend on the parameters that are available at run time.

- So its input dependent.

- So we can not really do all of this in an efficient way statically up front.

- Or in some cases we just can not do it all because we don’t have the input parameters.

- So then we dynamically capture these code blocks and then inspect them to see whether any of infamous instructions is about to be issued.

- If it turns out that the code block doesn't have any of these bad instructions its marked as save and allowed to execute natively at hardware speed.

- However if one of the bad instructions is found in the code block, then that particular instruction is translated into some other instruction sequence that avoids the undesired instruction and in some way emulates the desired behavior.

- This can possibly be achieved, even by bypassing a trap to the hypervisor.

- Certainly binary translation adds overheads and the number of mechanisms are incorporated
- Specifically in the viewer solutions in order to improve the efficiency of the process.
- These things include mechanisms such as caching code fragments that correspond to the translated basic blocks.
- So that the translation process can be avoided in the future.
- Also the steps like distinguishing which portions of the binary should be analyzed.
- For instance distinguishing between the kernel and the application code and making sure that the kernel code is the one that’s analyzed and various other optimizations.

- Hardware Assist

- The binary translation approach uses dynamic modification in the VM code during its execution.

- It causes the performance of the system being degraded.

- To improve on that aspect new virtualization approach, hardware assist technique is followed.

- This is a new processor technology that avoids change in the system state,
  - and calls the VMM directly as and when required.

- The hardware assisted VMM's interrupts the execution of the VM code only when the interruption is extremely necessary or can not be avoided

- Paravirtualization

- Paravirtualization makes the OS aware that it is being virtualized.

- To do so an explicit modification of the OS is done in this technique.

- So it is possible for OS to call the underlying VMM automatically, as and when the call is necessary.

- Calling the Hypervisors by the OS is known as hypercalls.

- The modification in the OS source code improves the virtual system performance greatly.

- Open Source Virtualization Technology
  - Open Source Technologies that provide virtualization support for the Linux Operating System.
    - Kernel-Based Virtual Machine (KVM)
    - Xen
  - KVM provides virtualization support for OS that are based on x86 hardware coupled with virtualization extension.
    - E.g. Intel VT and AMD-V.
  - KVM continues two modules
    - Loadable Kernel (kvm.ko)
    - Other is specific to processor (kvm-intel.co for Intel VT and kvm-amd.co for AMD-V)
  - The infrastructure for virtualization in KVM technology requires a modified Quick EMUlator(QEMU) for implementation of Virtualization.

- Kernel based virtual machine is used
  - to host multiple VM's that run Linux OS images or Windows OS images without modification

- Each of the VM has been provided with it's own set of virtualized hardware components that include
  - network card, disk, graphic adapter.

- Important features of KVM include
  - QEMU Monitor Protocol (QMP)
  - Kernel Samepage Merging (KSM)
  - KVM paravirtual clock
  - CPU hotplug support
  - PCI hotplug support
  - Vmchannel
  - Migration
  - Vhost
  - SCSI Disk Emulation
  - Virtion Devices
  - CPU Clustering

- Xen hypervisor is the only bare-metal hypervisor available as open source.
- Through Xen, a VM(or a host) can run a number of OS images or multiple different OS's in parallel.

- Xen hypervisor provides
  - server virtualization,
  - desktop virtualization,
  - security applications
  - IaaS
  - and embedded and hardware appliances.

- The key features of the Xen hypervisor include the following
  - Robustness and Security:
  - Scope for other OS:
  - Isolation of Drivers from the Rest of the System:
  - Support for Paravirtualization:

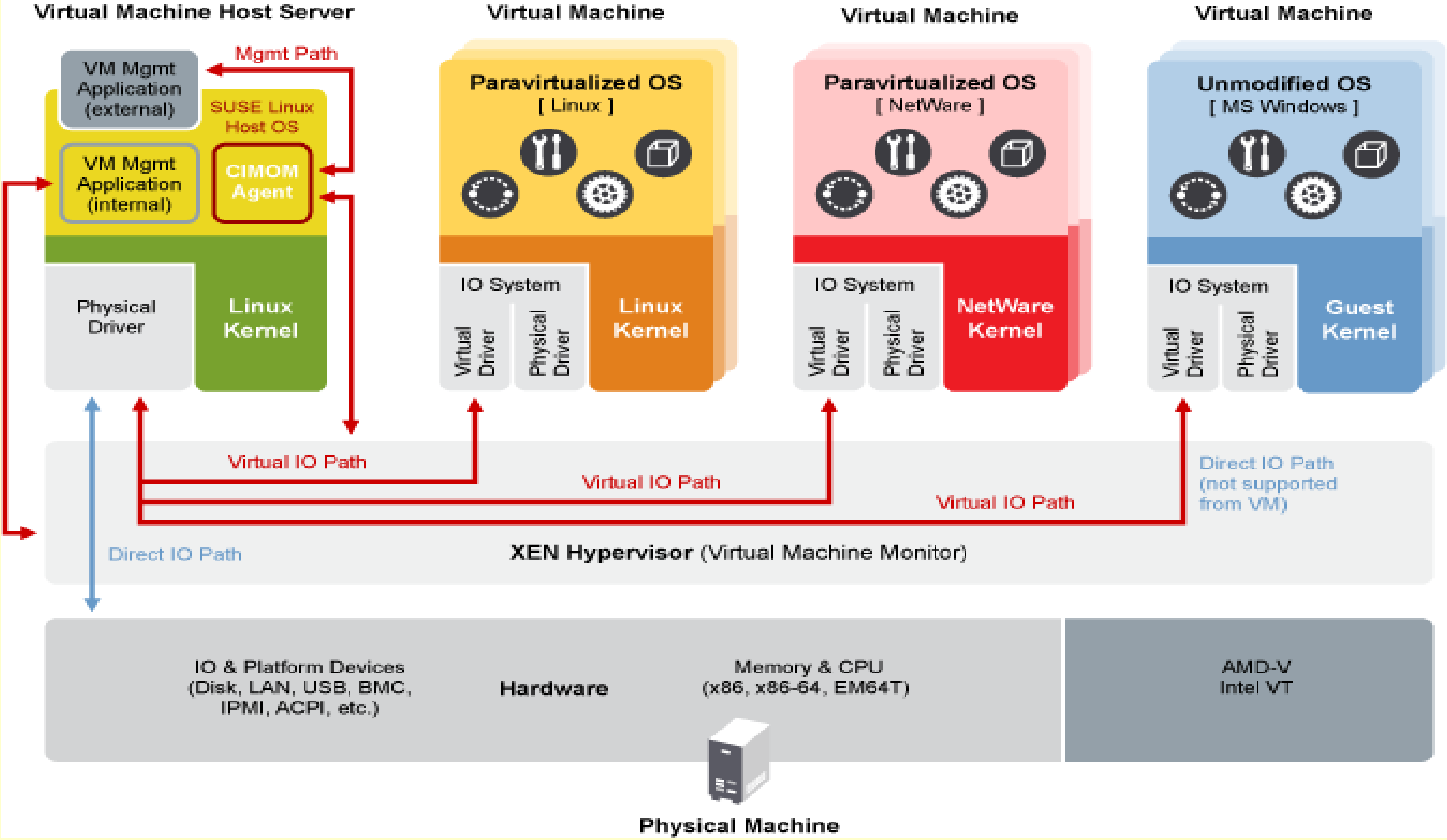
- Kernel Virtual Machine(KVM) v/s Xen Hypervisor
  - KVM and Xen hypervisor are similar, both are open-source technologies.
  - Xen hypervisor is type-1 hypervisor that provides isolation of the drivers from the rest of the system
  - The KVM is a type-2 virtualization mechanism in which the drivers cannot be isolated

- Xen Virtualization Architecture
- Xen hypervisor uses microkernel design

- Provides services for allowing a number of OS's being concurrently operated on a single hardware setup.

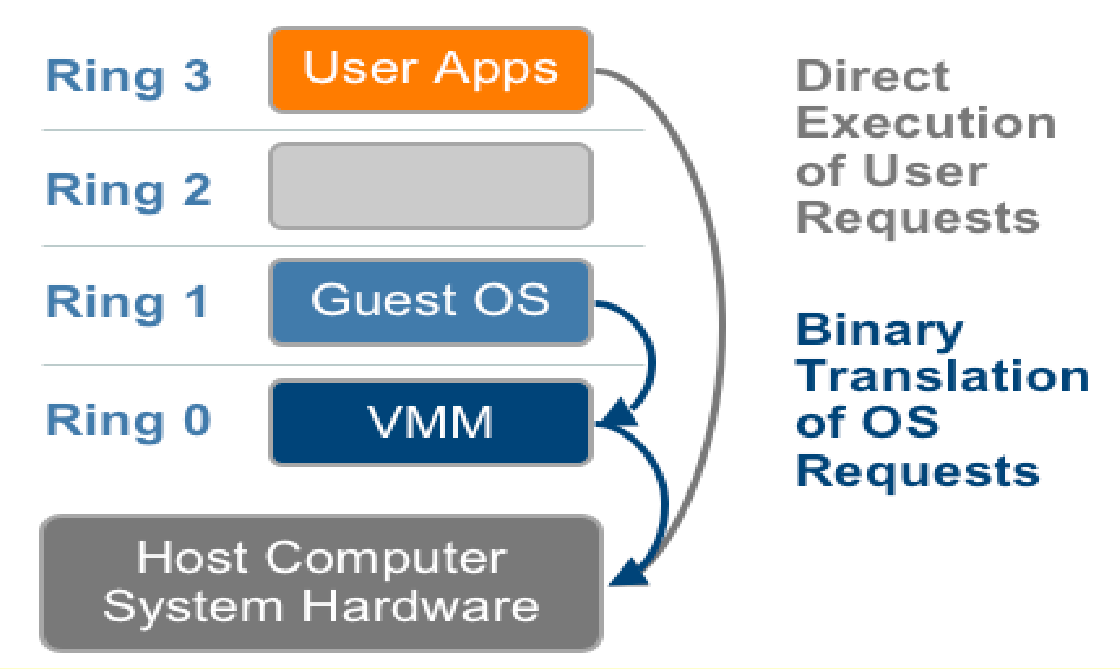
- The software is available for free and is maintained by the Xen community as an open-source system for virtualization.

- A host machine comprising four guest machine managed by a Xen hypervisor that is running directly on physical hardware as shown in fig.



- The controlling part is also a virtual machine having greater functionalities than the normal guest systems

- Binary Translation with full Virtualization
  - Binary translation in combination with the direct execution technique can be used by VMware to virtualize seemingly any OS on x86 platform
  - In this approach non-virtualized instructions are replaced by translating the kernel code into a new set of instructions
  - This could be used to affect the virtual hardware as intended by the programmer.


Ring 3
User Apps
Ring 2
Ring 1
Guest OS
Ring 0
VMM
Host Computer System Hardware
Direct Execution of User Requests
Binary Translation of OS Requests

- Meanwhile, we directly execute the user-level instructions on the processor for getting high performance virtualization.

- Every virtual machine is provided by each VMM with the physical systems services.

- These includes,
  - Virtualized memory management
  - Virtual devices
  - Virtual BIOS

- A full virtualization is provided by the combination.

- Because the virtualization layer fully abstracts the guest OS from the hardware on which the base OS is installed.

- Here no modification of the guest OS is required, because the OS has no awareness of being virtualized .

- All the instructions issued by the guest OS are translated by the hypervisor instantly and the result are cached for future use.

- The instructions at the user level are run at a native speed without being modified.

- In the full virtualization technique the isolation and security for virtual machines are offered at their best.

- While migration and portability are made simpler due to the same instance of the OS being run virtualized or on native hardware.

- Two examples that are utilizing full virtualization solution are
  - The virtualization products of VMware
  - and the Virtual Server of Microsoft

Paravirtualization with Compiler Support

- Paravirtualization is a technique in which the hypervisor communicates with the guest OS for improving the performance and efficiency of virtual systems.

- The non-virtualizable instructions are replaced by modification in the OS kernel through paravirtualzation as shown in figure.

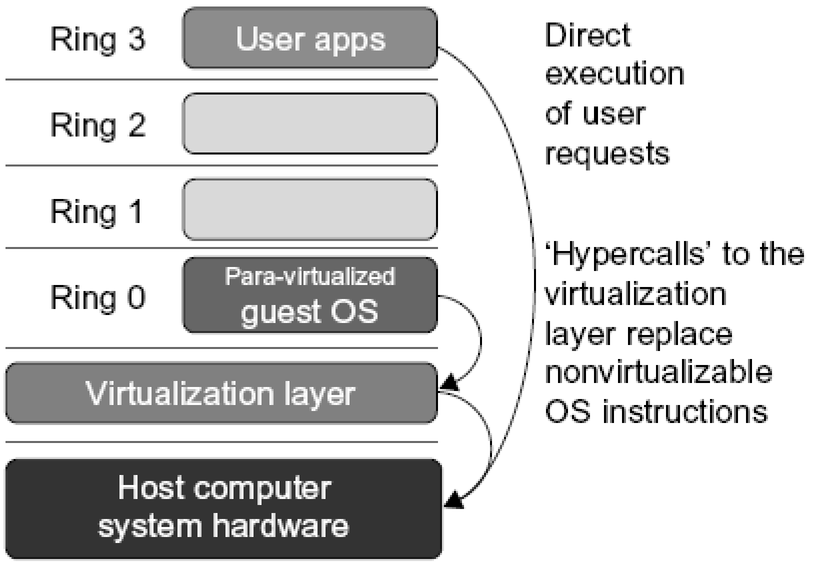

Ring 3
User apps
Ring 2
Ring 1
Ring 0
Para-virtualized guest OS
Virtualization layer
Host computer system hardware
Direct execution of user requests
'Hypercalls' to the virtualization layer replace nonvirtualizable OS instructions

FIGURE 3.8

The Use of a para-virtualized guest OS assisted by an intelligent compiler to replace nonvirtualizable OS instructions by hypercalls.

- The technique uses hypercalls for communicating with the virtualization layer hypervisor.

- Providing the hypercall interface for other critical operations being performed by the kernel is also done by the hypervisor.

- These critical operations may include managing the memory, handling the interrupts, and keeping the time.

- Full virtualization and paravirtualization are two different techniques.

- In full virtualization technique the OS has no awareness of being virtualizad

- In the paravirtualization involves making the OS aware of virtualization being applied on it.

- In the full virtualization technique binary translation traps the sensitive OS calls.

- The propagation value of paravirtualization is in the lower overhead of virtualization.

- Paravirtualization technique advantage of performance over full virtualization is largely determined by the workload.

- On the point of compatibility and portability paravirtualization proves far from preferable because of no support for unmodified OS kernel.

- In addition to compatibility and portability concerns, paravirtualization involves significant issues regarding support and maintainability.

- The reason for the issue is the requirement for deep modification in the kernel of an OS

- Paravirtualization by the way of modified OS is easier than building the sophisticated binary translation support for full virtualization.

- Para-virtualization needs to modify the guest operating systems.

- A para-virtualized VM provides special APIs requiring substantial OS modifications in user applications.

- Performance degradation is a critical issue of a virtualized system.

- No one wants to use a VM if it is much slower than using a physical machine.

- The virtualization layer can be inserted at different positions in a machine soft-ware stack.

- However, para-virtualization attempts to reduce the virtualization overhead, and thus improve performance by modifying only the guest OS kernel.

- The figure illustrates the concept of a paravirtualized VM architecture.

- The guest operating systems are para-virtualized.

- They are assisted by an intelligent compiler to replace the nonvirtualizable OS instructions by hypercalls as illustrated in Figure.

- The traditional x86 processor offers four instruction execution rings: Rings 0, 1, 2, and 3.

- The lower the ring number, the higher the privilege of instruction being executed.

- The OS is responsible for managing the hardware and the privileged instructions to execute at Ring 0, while user-level applications run at Ring 3.

- The best example of para-virtualization is the KVM to be described below.

- KVM (Kernel-Based VM)

- This is a Linux para-virtualization system—a part of the Linux version 2.6.20 kernel.

- Memory management and scheduling activities are carried out by the existing Linux kernel.

- The KVM does the rest, which makes it simpler than the hypervisor that controls the entire machine.

- KVM is a hardware-assisted para-virtualization tool, which improves performance and supports unmodified guest OSes such as Windows, Linux, Solaris, and other UNIX variants.

- Para-Virtualization Architecture
- When the x86 processor is virtualized, a virtualization layer is inserted between the hardware and the OS.
- According to the x86 ring definition, the virtualization layer should also be installed at Ring 0.
- Different instructions at Ring 0 may cause some problems.
- In Figure, we show that para-virtualization replaces nonvirtualizable instructions with hypercalls that communicate directly with the hypervisor or VMM.
- However, when the guest OS kernel is modified for virtualization, it can no longer run on the hardware directly.

- Unlike the full virtualization architecture which intercepts and emulates privileged and sensitive instructions at runtime, para-virtualization handles these instructions at compile time.

- The guest OS kernel is modified to replace the privileged and sensitive instructions with hypercalls to the hypervisor or VMM.

- The guest OS running in a guest domain may run at Ring 1 instead of at Ring 0.

- This implies that the guest OS may not be able to execute some privileged and sensitive instructions.

- The privileged instructions are implemented by hypercalls to the hypervisor.

- After replacing the instructions with hypercalls, the modified guest OS emulates the behavior of the original guest OS.

- Virtualization of CPU, Memory, and I/O Devices
- Hardware Support for Virtualization in the Intel x86 Processor
- Since software-based virtualization techniques are complicated and incur performance overhead,
- Intel provides a hardware-assist technique to make virtualization easy and improve performance.

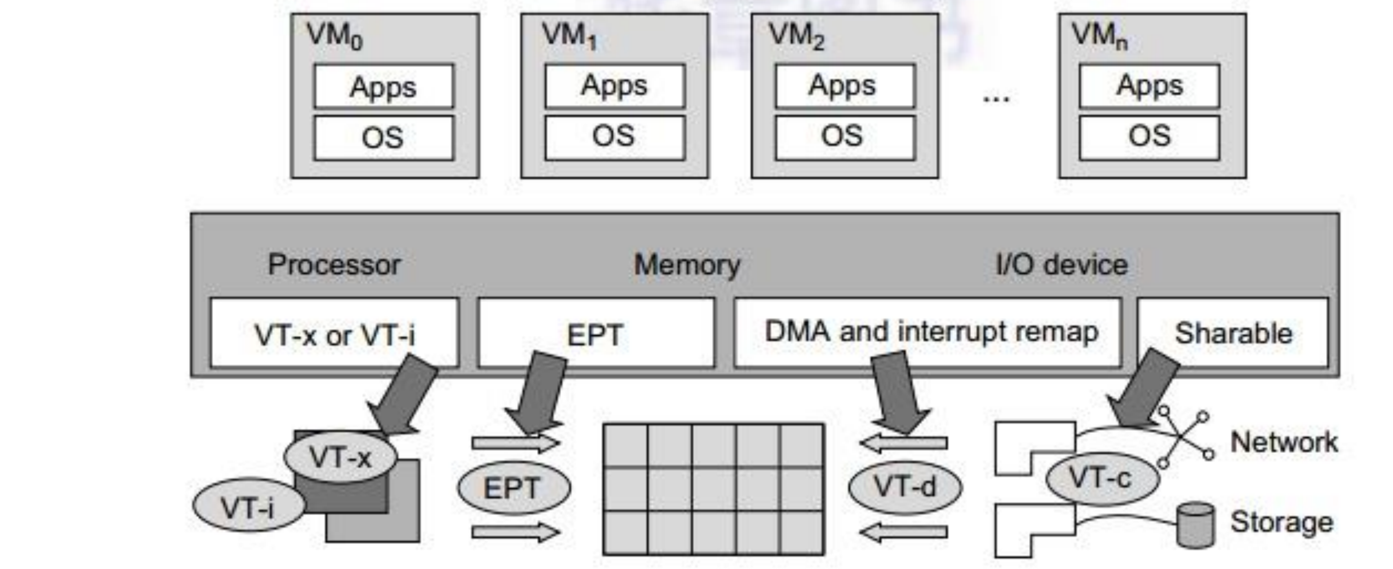


FIGURE 3.10

Intel hardware support for virtualization of processor, memory, and I/O devices.

- Above fig. provides an overview of Intel’s full virtualization techniques.

- For processor virtualization, Intel offers the VT-x or VT-i technique.

  (The Core i3, i5 and i7 series microprocessors bring **VT**-x virtualization technology to desktop PCs)

  - VT-x adds a privileged mode (VMX Root Mode) and some instructions to processors.

  - This enhancement traps all sensitive instructions in the VMM automatically.

- For memory virtualization, Intel offers the EPT (Extended Page Tables), which translates the virtual address to the machine’s physical addresses to improve performance.

- For I/O virtualization, Intel implements VT-d and VT-c to support this.

- **CPU Virtualization**
- A VM is a duplicate of an existing computer system in which a majority of the VM instructions are executed on the host processor in native mode.

- Thus, unprivileged instructions of VMs run directly on the host machine for higher efficiency.
- The critical instructions are divided into three categories:
  - privileged instructions: execute in a privileged mode and will be trapped if executed outside this mode

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

  - Behavior-sensitive instructions : have different behaviors depending on the configuration of resources, including the load and store operations over the virtual memory.

- A CPU architecture is virtualizable if it supports the ability to run the VM’s privileged and unprivileged instructions in the CPU’s user mode while the VMM runs in supervisor mode.

- When the privileged instructions including control- and behavior-sensitive instructions of a VM are executed, they are trapped in the VMM.

- In this case, the VMM acts as a unified mediator for hardware access from different VMs to guarantee the correctness and stability of the whole system

Memory Virtualization

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

- In a traditional execution environment, the operating system maintains mappings of virtual memory to machine memory using page tables,

- In which is a one-stage mapping from virtual memory to machine memory.

- All modern x86 CPUs include a memory management unit (MMU) and a translation lookaside buffer (TLB) to optimize virtual memory performance.

- However, in a virtual execution environment, virtual memory virtualization involves sharing the physical system memory in RAM and dynamically allocating it to the physical memory of the VMs.

- That means a two-stage mapping process should be maintained by the guest OS and the VMM, respectively:
  - virtual memory to physical memory and physical memory to machine memory.

- Furthermore, MMU virtualization should be supported, which is transparent to the guest OS.

- The guest OS continues to control the mapping of virtual addresses to the physical memory addresses of VMs.

- But the guest OS cannot directly access the actual machine memory.

- The VMM is responsible for mapping the guest physical memory to the actual machine memory.
- Figure shows the two-level memory mapping procedure.

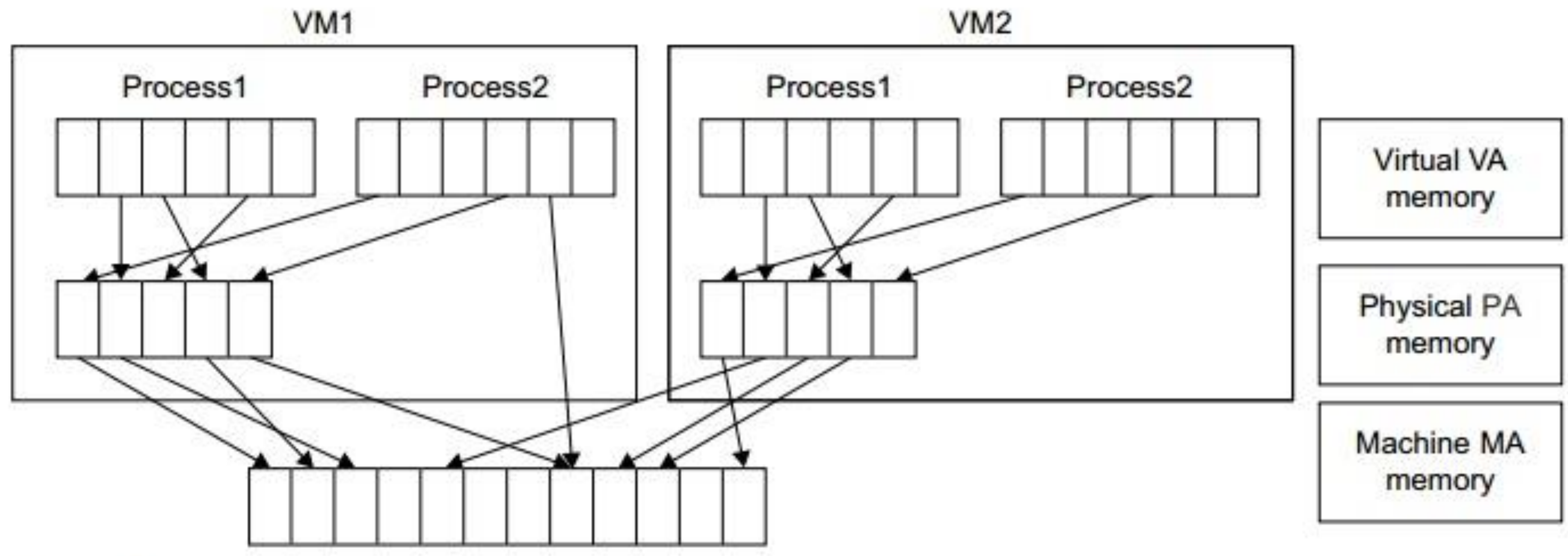


FIGURE 3.12

Two-level memory mapping procedure.

- Since each page table of the guest OSes has a separate page table in the VMM corresponding to it, the VMM page table is called the shadow page table.

- The MMU already handles virtual-to-physical translations as defined by the OS.

- Then the physical memory addresses are translated to machine addresses using another set of page tables defined by the hypervisor.

- Since modern operating systems maintain a set of page tables for every process, the shadow page tables will get flooded.

- Consequently, the performance overhead and cost of memory will be very high.

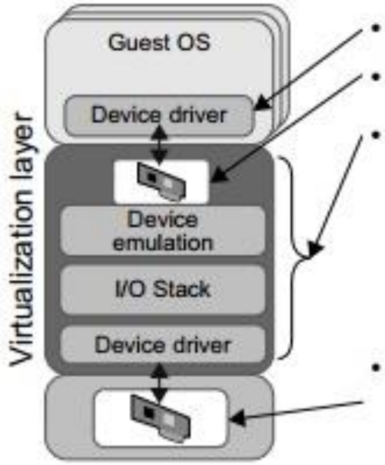
- VMware uses shadow page tables to perform virtual-memory-to-machine-memory address translation.

- Processors use TLB hardware to map the virtual memory directly to the machine memory to avoid the two levels of translation on every access.

- When the guest OS changes the virtual memory to a physical memory mapping, the VMM updates the shadow page tables to enable a direct lookup.

I/O Virtualization

- I/O virtualization involves managing the routing of I/O requests between virtual devices and the shared physical hardware.
- At the time of this writing, there are three ways to implement I/O virtualization:
  - full device emulation,
  - para-virtualization,
  - and direct I/O.



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

FIGURE 3.14

Device emulation for I/O virtualization implemented inside the middle layer that maps real I/O devices into the virtual devices for the guest device driver to use.

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

- para-virtualization
- The para-virtualization method of I/O virtualization is typically used in Xen.
- It is also known as the split driver model consisting of a frontend driver and a backend driver.
- The frontend driver is running in Domain U and the backend driver is running in Domain 0.
- They interact with each other via a block of shared memory.
- The frontend driver manages the I/O requests of the guest OSes
- and the backend driver is responsible for managing the real I/O devices and multiplexing the I/O data of different VMs.
- Although para-I/O-virtualization achieves better device performance than full device emulation, it comes with a higher CPU overhead.

- direct I/O
- Direct I/O virtualization lets the VM access devices directly.
- It can achieve close-to-native performance without high CPU costs.
- There are a lot of challenges for commodity hardware devices.

- For example, when a physical device is reclaimed (required by workload migration) for later reassignment, it may have been set to an arbitrary state that can function incorrectly or even crash the whole system.
- e.g., DMA to some arbitrary memory locations.

- Since software-based I/O virtualization requires a very high overhead of device emulation, hardware-assisted I/O virtualization is critical.

- **Hardware Support for Virtualization**
- Modern operating systems and processors permit multiple processes to run simultaneously.

- If there is no protection mechanism in a processor, all instructions from different processes will access the hardware directly and cause a system crash.

- Therefore, all processors have at least two modes, user mode and supervisor mode,
  - to ensure controlled access of critical hardware.

- Instructions running in supervisor mode are called privileged instructions.

- Other instructions are unprivileged instructions.

- In a virtualized environment, it is more difficult to make OSes and applications run correctly because there are more layers in the machine stack.

- The VMware Workstation is a VM software suite for x86 and x86-64 computers.

- This software suite allows users to set up multiple x86 and x86-64 virtual computers and to use one or more of these VMs simultaneously with the host operating system.

- http://www.brainkart.com/article/Hardware-Support-for-Virtualization_11338/