UNIT 1:
INTRODUCTION TO VIRTUALIZATION

Virtualization

- Virtualization is the technology that underlies the cloud.
- It is a broad term that refers to the abstraction of computer resources from their users, be they applications, or end users.

Virtualization

- This includes making a single physical resource (such as a server, an operating system, an application, or storage device) appear to function as multiple virtual resources;
- It can also include making multiple physical resources (such as storage devices or servers) appear as a single virtual resource.
- In layman's terms virtualization is often considered as:
  - The creation of many virtual resources from one physical resource.
  - The creation of one virtual resource from one or more physical resource.
- Concept of virtualization is realized through software named **Hypervisor/Virtual Machine Monitor (VMM)**.

Virtualization: Reasons for Renewed Interest

Virtualization technologies have gained a renewed interested recently due to the confluence of different phenomena:

- Increased performance and computing capacity: Almost all modern PCs have resources enough to host a virtual machine manager and execute a virtual machine with a by far acceptable performance.

Virtualization: Reasons for Renewed Interest

Virtualization technologies have gained a renewed interested recently due to the confluence of different phenomena:

- Underutilized hardware and software resources: Hardware and software underutilization is occurring due to (1) the increased performance and computing capacity, and (2) effect of limited or sporadic use of resources. Using these resources for other purposes after hours could improve the efficiency of the IT infrastructure. In order to transparently provide such a service, it would be necessary to deploy a separate environment, which can be achieved through virtualization.

Virtualization: Reasons for Renewed Interest

Virtualization technologies have gained a renewed interested recently due to the confluence of different phenomena:

- Lack of space: The continuous need for additional capacity, whether this is storage or compute power, makes data centers grow quickly. This condition along with hardware underutilization led to the diffusion of a technique called server consolidation, for which virtualization technologies are fundamental.

Virtualization: Reasons for Renewed Interest

Virtualization technologies have gained a renewed interested recently due to the confluence of different phenomena:

- Greening initiatives: Recently, companies are increasingly looking for ways to reduce the amount of energy they consume and to reduce their carbon footprint. Hence, reducing the number of servers through server consolidation will reduce the impact of cooling and power consumption of a data center. Virtualization technologies can provide an efficient way of consolidating servers.

Virtualization: Reasons for Renewed Interest

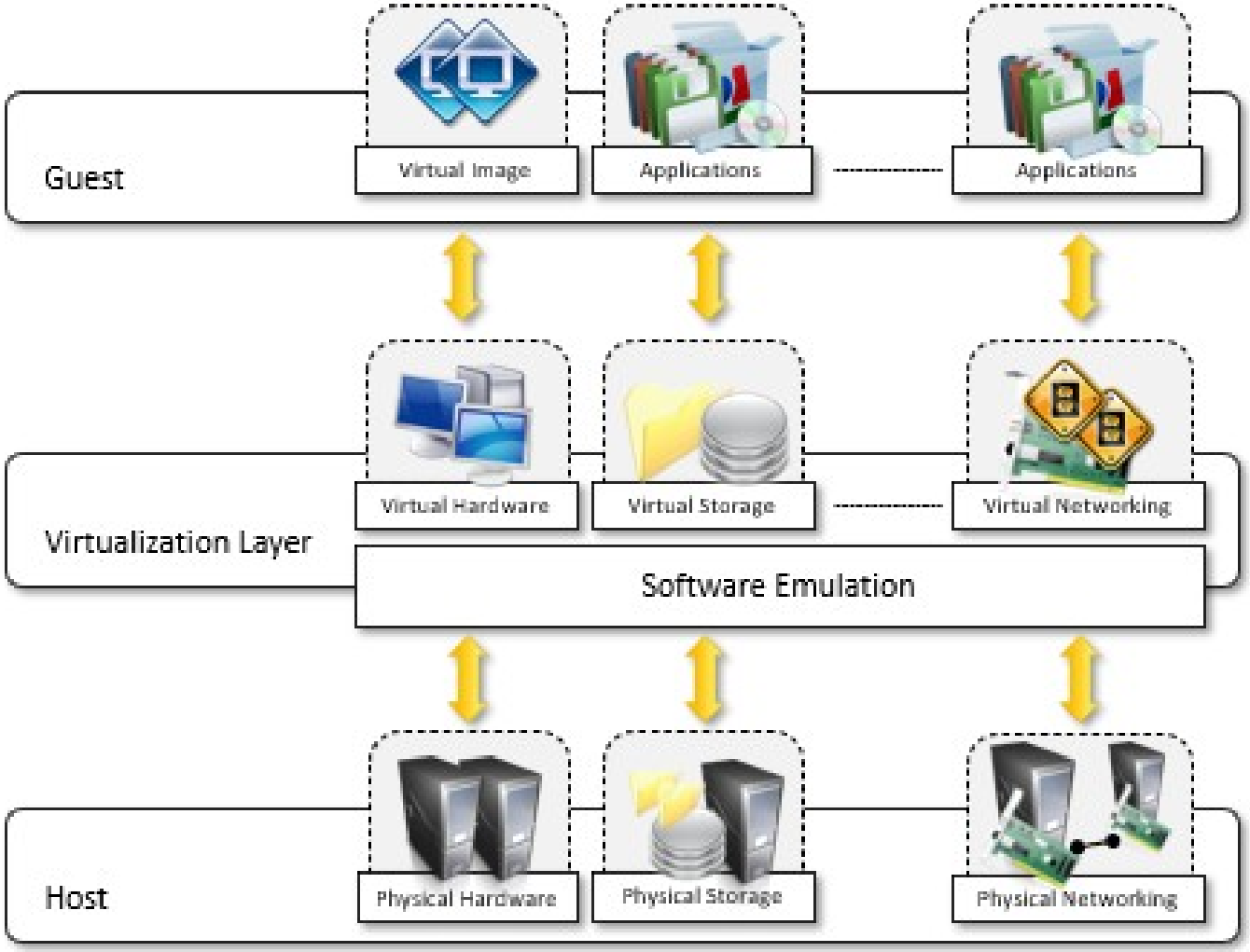
Virtualization technologies have gained a renewed interested recently due to the confluence of different phenomena:

- Rise of administrative costs: Power consumption and cooling costs have now become higher than the cost of the IT equipment. Virtualization can help in reducing the number of required servers for a given workload, thus reducing the cost of the administrative personnel.

Virtualization Reference Model

- Virtualization is a broad concept, and it refers to the creation of a virtual version of something, whether this is hardware, software environment, storage, or network.
- In a virtualized environment there are three major components: guest, host, and virtualization layer.
- The guest represents the system component that interacts with the virtualization layer rather than with the host as it would normally happen.
- The host represents the original environment where the guest is supposed to be managed.
- The virtualization layer is responsible for recreating the same or a different environment where the guest will operate.

Virtualization Reference Model

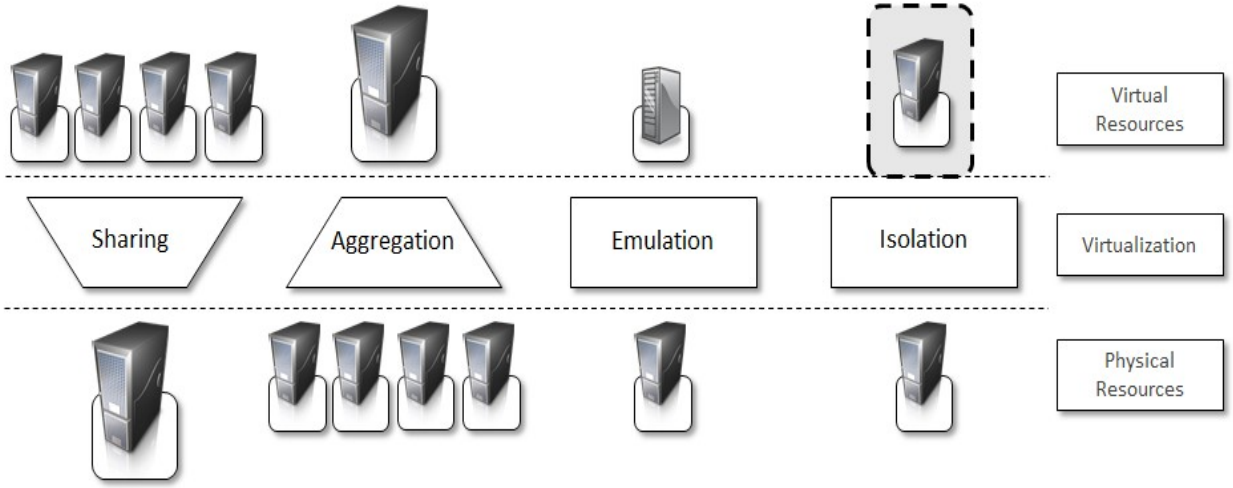


Characteristics of virtualized environments

- Increased Security
  - The ability to control the execution of a guest in a completely transparent manner opens new possibilities for delivering a secure, controlled execution environment.
    - The virtual machine represents an emulated environment in which the guest is executed. All the operations of the guest are generally performed on the virtual machine, which then translates and applies them to the host. This level of indirection allows the virtual machine manager to control and filter the activity of the guest, thus preventing some harmful operations from being performed.
    - Resources exposed by the host can then be hidden or simply protected from the guest. Moreover, sensitive information that is contained in the host can be naturally hidden without the need of installing complex security policies. Increased security is a requirement when dealing with untrusted code.

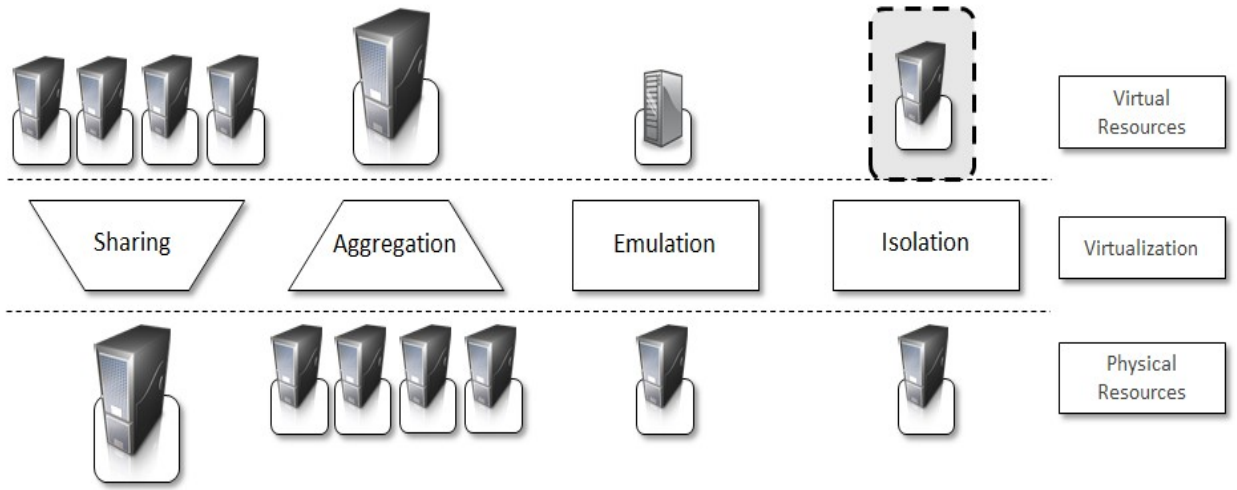
Characteristics of virtualized environments

- Managed Execution
  - Virtualization of the execution environment does not only allow the increased security, but a wider range of features can be implemented. Sharing, aggregation, emulation, and isolation are the most relevant.



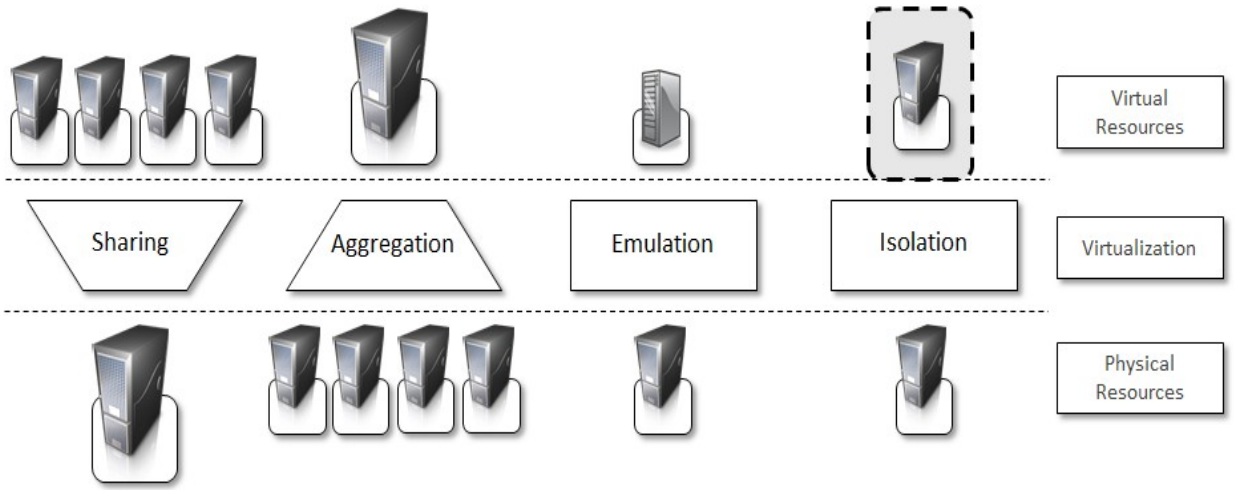
Characteristics of virtualized environments

- Sharing
  - Virtualization allows the creation of a separate computing environment within the same host. In this way it is possible to fully exploit the capabilities of a powerful guest, which would be otherwise underutilized.
  - Sharing is a particularly important feature in virtualized data centers, where this basic feature is used to reduce the number of active servers and limit power consumption.



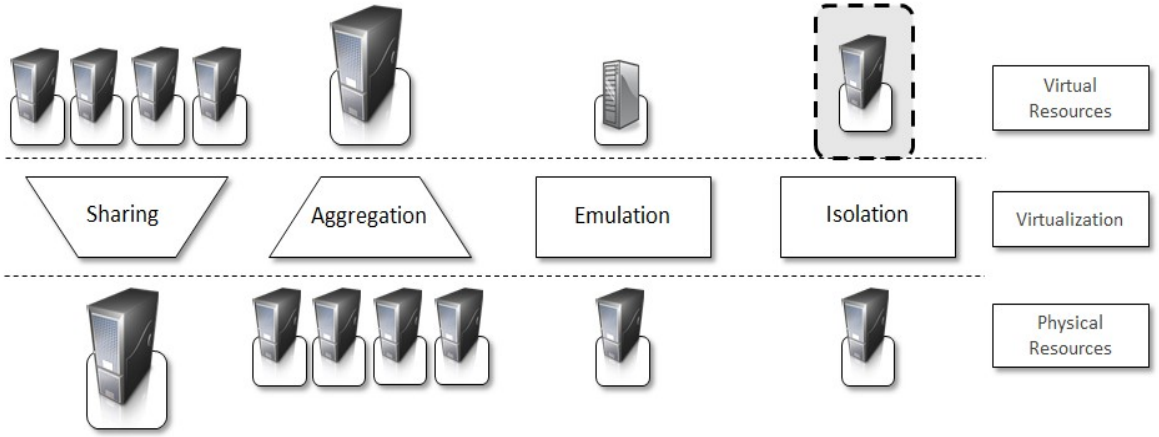
Characteristics of virtualized environments

- Aggregation
  - Virtualization also allows the aggregation, which is the opposite process. A group of separate hosts can be tied together and represented to guests as a single virtual host. This function is naturally implemented in middleware for distributed computing and a classical example is represented by cluster management software, which harnesses the physical resources of a homogeneous group of machines and represents them as a single resource.



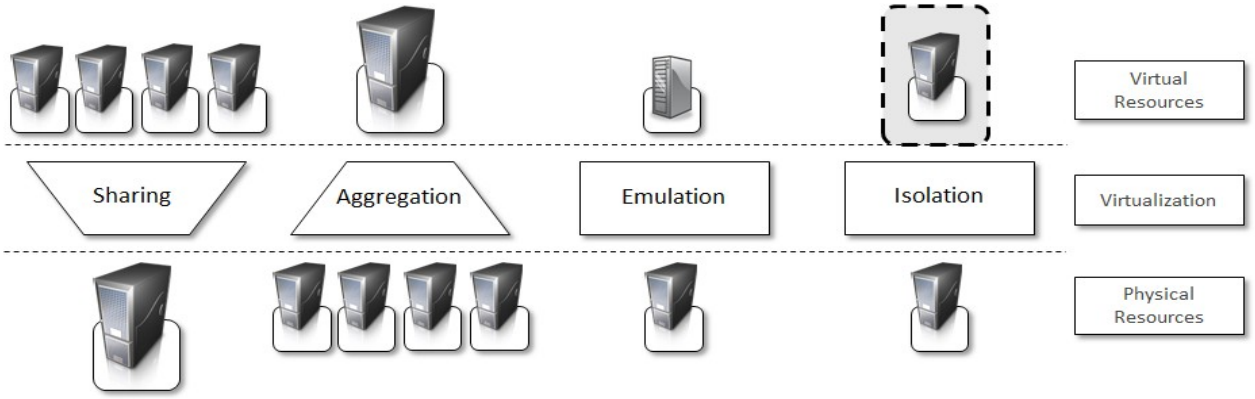
Characteristics of virtualized environments

- Emulation
  - Guests are executed within an environment that is controlled by the virtualization layer, which ultimately is a program. This allows for controlling and tuning the environment that is exposed to guests.
    - A completely different environment with respect to the host can be emulated, thus allowing the execution of guests requiring specific characteristics that are not present in the physical host.
    - Old and legacy software, which does not meet the requirements of current systems, can be run on emulated hardware without any need of changing their code.



Characteristics of virtualized environments

- Isolation
  - Virtualization allows providing guests—whether they are operating systems, applications, or other entities—with a complete separate environment, in which they are executed. The guest performs its activity by interacting with an abstraction layer, which provides access to the underlying resources.
    - Isolation allows multiple guests to run on the same host without each of them interfering with the other.
    - Isolation also provides a separation between the host and the guest. The virtual machine can filter the activity of the guest and prevent harmful operations against the host.



Characteristics of virtualized environments

- Performance Tuning
  - It becomes easier to control the performance of the guest by finely tuning the properties of the resources exposed through the virtual environment. This provides means to effectively implement a Quality-of-Service infrastructure that more easily fulfill the service level agreement established for the guest.
    - For instance, software implementing hardware virtualization solutions can expose to a guest operating system only a fraction of the memory of the host machine or to set the maximum frequency of the processor of the virtual machine.
    - Another advantage of managed execution is that sometimes it allows easily capturing the state of the guest, persisting it, and resuming its execution. This, for example, allows virtual machine managers such as Xen Hypervisor to stop the execution of a guest operating system, to move its virtual image into another machine, and to resume its execution in a completely transparent manner. This technique is called virtual machine migration and constitutes an important feature in virtualized data centers for optimizing their efficiency in serving applications demand.

Characteristics of virtualized environments

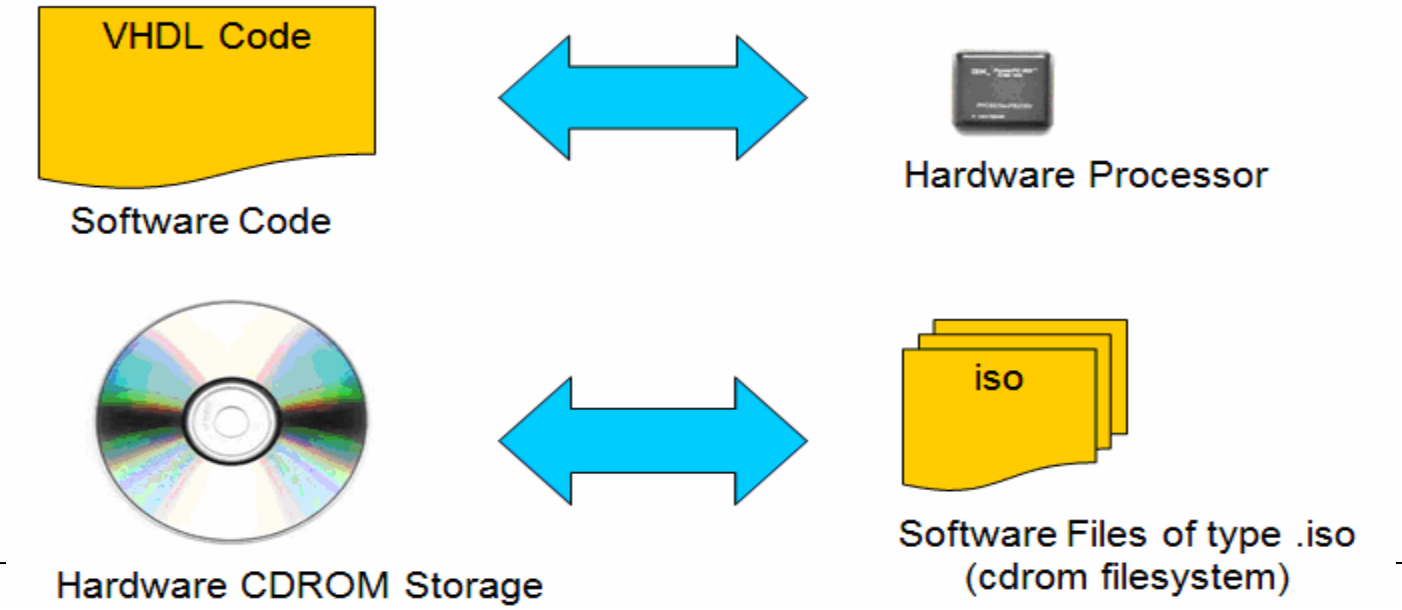
- Portability
  - The concept of portability applies in different ways according to the specific type of virtualization considered.
    - In the case of a hardware virtualization solution the guest is packaged into a virtual image that, in most of the cases, can be safely moved and executed on top of different virtual machines.
    - In the case of programming level virtualization, as implemented by the JVM or the .NET runtime, the binary code representing application components (jars or assemblies), can be run without any recompilation on any implementation of the corresponding virtual machine.
    - This makes the application development cycle more flexible and application deployment very straightforward: one version of the application, in most of the cases, can run on different platforms with no changes.

Principal of H/w-S/w Equivalence

- Hardware and software are logically interchangeable.
- Software written in logic can be converted to equivalent hardware and vice versa.
- Any existing hardware can be easily converted to software.

Principal of H/w-S/w Equivalence

- The hardware-software logical conversion is generally a tradeoff between performance and flexibility.
  - VHDL code is easier to modify and simulate than a physical processor.
  - A physical processor once fabricated cannot be changed.
  - However, the performance of simulation using VHDL is much slower than a physical processor.

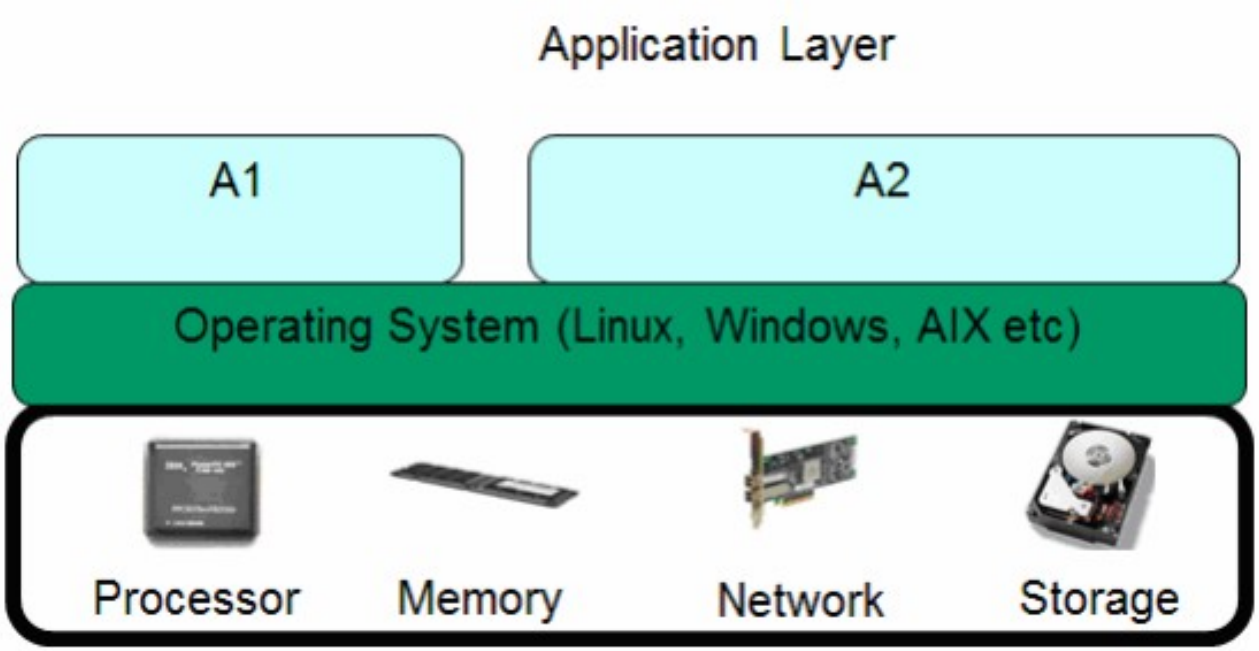


Principal of H/w-S/w Equivalence

- Based on the principle of hardware-software equivalence, it is possible to logically convert the system hardware components required to run an OS application, to their software equivalent or a virtual machine.
- The virtual machine can then be replicated to create multiple virtual machines each running its own instance of the OS and the application.
- A hypervisor layer is required to multiplex the actual real hardware resources among the virtual machines.
- The hypervisor is responsible for allocating memory, CPU resources, network and storage to each virtual machine.

Principal of H/w-S/w Equivalence

- The Operating System resides over the hardware.
- The application requires support from the Operating System and allied components.
- The applications are stacked over the operating system and use services provided by the OS.



Need for Virtualization / Virtualization Facilitates / Benefits of Virtualization

- Virtualization eliminates most of the inherent inflexibility of hardware systems.
- It allows for better manageability, resulting in better system utilization, more secure, and complete isolation.

Parameter	Traditional IT	Virtualization
Utilization	0-20%	Typically 60-70%
Provisioning	Typically takes 6-8 weeks	1 day
Monitoring	Usage of monitoring tools. However, need manual intervention to take care of any hardware failures	Comparative ease in monitoring using automated tools. However, need manual intervention to take care of any failures
Sizing	Sizing needs to be completed before deployment. Re-sizing involves procuring new hardware and planned downtimes	Easier to resize. However, manual intervention required to resize
Staff for Administration	Require larger number of Full Time employees to manage the infrastructure	Reduced number of Full Time employees
Cost	Upfront costs involved in outright purchase of hardware	Initial hardware cost reduced due to sharing of hardware assets and increased utilization. There is a typical reduction of 40% in hardware
Optimization	Difficult to do as there is no easy way to monitor and load balance across machines	Easy to share resources and re-balance loads on the virtual machines on the same host. However, re-balancing across physical hosts require advanced features and planned downtime

Emulation

- **Emulation** refers to the process of creating an environment that imitate the property of a system, either h/w or s/w inside a completely different one.
- **Emulator** is a computer program designed to imitate the properties of a ***guest system*** inside the ***host system***. In computing, the emulator is a hardware or software that enables one device (named Host) to function like other systems (named Guest).
- To properly achieve emulation, emulator relies on **Interpreter**. An interpreter is a computer program that reads the emulated guest system code instructions and then executes semantically equivalent operations on the host system.

Emulation

- Emulations is very popular for running programs in video games that have become obsolete in host systems that were originally built for another environment.
- BlueStacks is a popular and free emulator for running Android apps on a PC or Mac computer. BlueStacks creates a virtual version of an Android device that runs in a window on your computer.

Emulation vs Virtualization

- Emulation, in short, involves making one system imitate another. For example, if a piece of software runs on android OS and not on window OS, we make window OS system "emulate" the working of android OS. The software then runs on an emulation of android OS over windows OS.
- In this same example, virtualization would involve taking windows-based system and splitting it into two servers, android based and linux based. Both "virtual" servers are independent software containers, having their own access to software-based resources – CPU, RAM, storage and networking – and can be rebooted independently. They behave exactly like real hardware, and an application or another computer would not be able to tell the difference.

Emulation vs Virtualization

- In virtualization, guest system runs code directly on the host systems language while in emulation guest system needs a software bridge called an interpreter for translating its code into the host system language.
- In virtualization, the guest system gets direct access to the host allocated resources resulting in higher throughput and minimal overhead while in emulation, guest system does not have the direct access of the host physical hardware, so emulation is slower when compared to virtualization.

Execution Virtualization

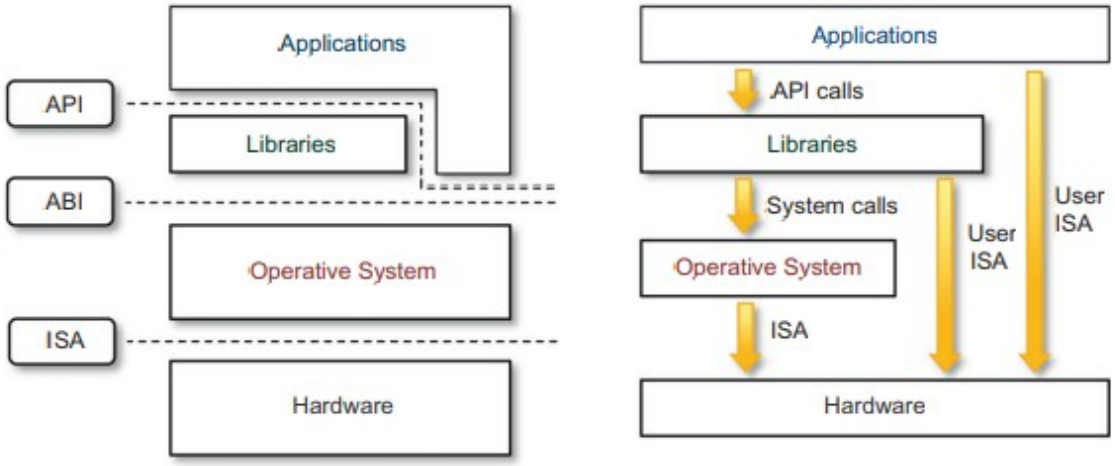
- Execution virtualization includes all those techniques whose aim is to emulate an execution environment that is separate from the one hosting the virtualization layer.
- All these techniques concentrate their interest on providing support for the execution of programs, whether these are the operating system, a binary specification of a program compiled against an abstract machine model, or an application.
- Therefore, execution virtualization can be implemented directly on top of the hardware, by the operating system, an application, or libraries dynamically or statically linked against an application image.

Machine Reference Model

- Virtualizing an execution environment at different levels of the computing stack requires the details of interfaces.
- The interfaces available between the different levels of abstractions, which hide implementation details will defined under machine reference model.
- From this perspective, virtualization techniques replace one of the layers and intercept the calls that are directed towards it.
- Therefore, a clear separation between layers simplifies their implementation, which only requires the emulation of the interfaces and a proper interaction with the underlying layer.

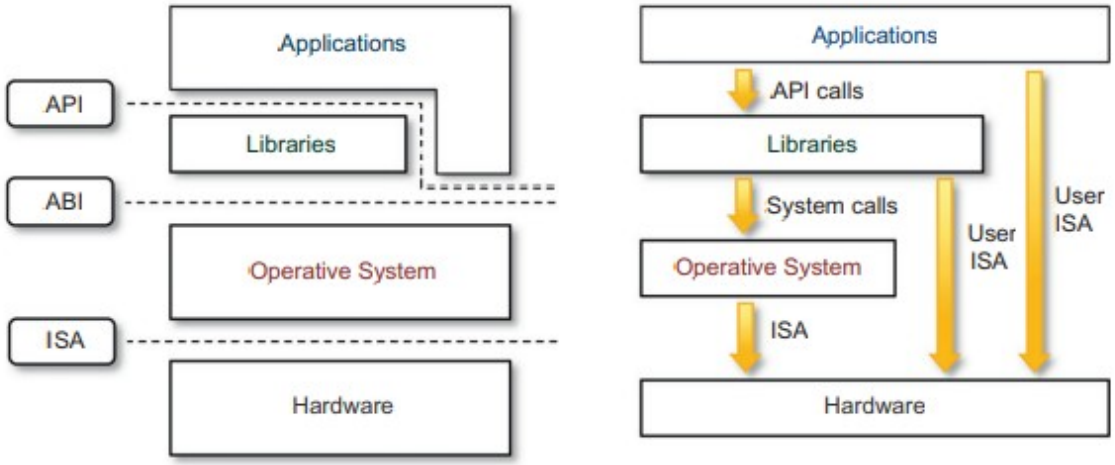
Machine Reference Model

- At the bottom layer, the model for the hardware is expressed in terms of the Instruction Set Architecture (ISA).
- ISA is the interface between hardware and software.
- ISA defines the instruction set for the processor, registers, memory, and interrupt management.
- It is important to the operating system (OS) developer (System ISA) and developers of applications (User ISA) that directly manage the underlying hardware.



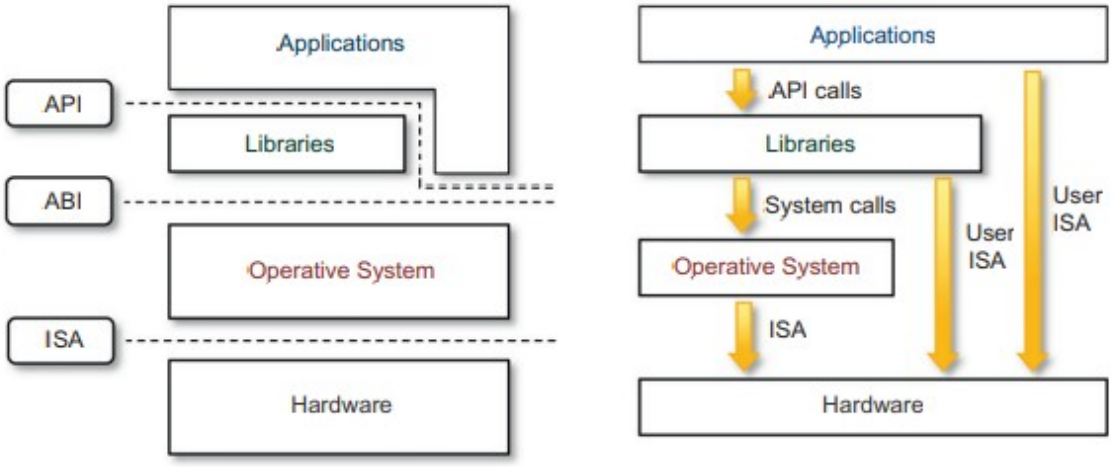
Machine Reference Model

- Application binary interface (ABI) separates the operating system layer from the applications and libraries, which are managed by the OS.
- ABI covers details such as low-level data types, alignment, and call conventions and defines a format for executable programs.
- System calls are defined at this level.
- This interface allows portability of applications and libraries across operating systems that implement the same ABI.



Machine Reference Model

- The highest level of abstraction is represented by the application programming interface (API), which interfaces applications to libraries and/or the underlying operating system.

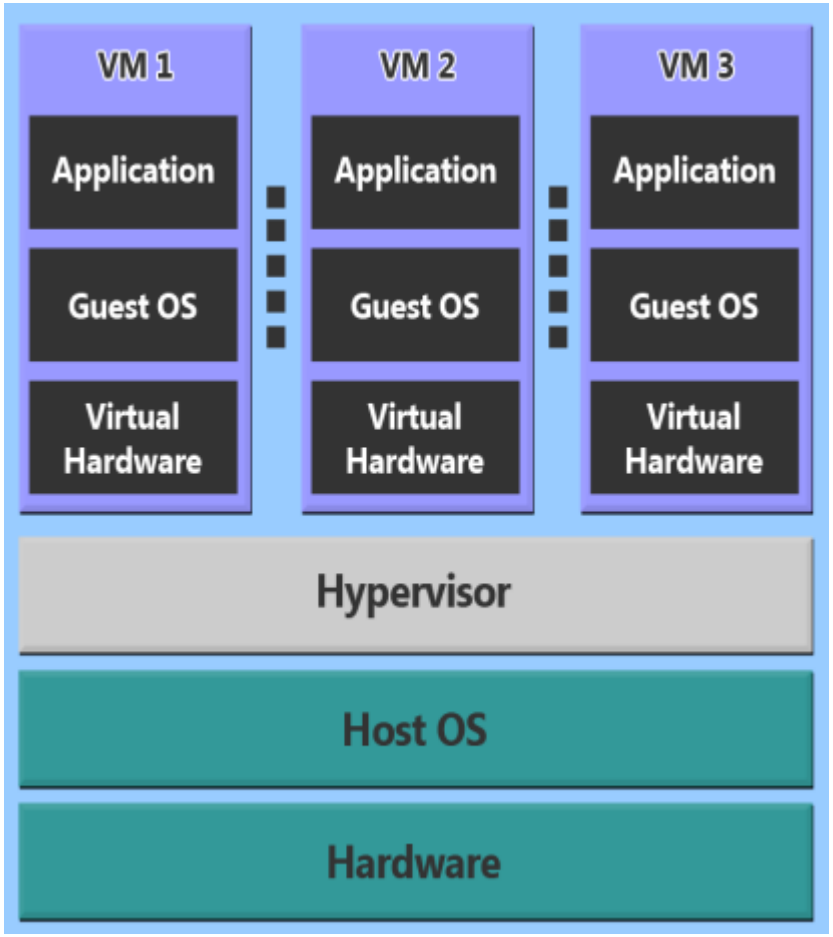


Virtual Machine (VM)

- A VM is a software-based computer.
- They run like a physical computer.
- They have operating system and applications.
- They are completely independent of one another.
  - So, we can run different operating systems on different VM's.
  - Independence also makes them portable. We can move VM from one hypervisor to another hypervisor on a completely different machine almost instantly. This will give us lot of flexibility and portability within the environment.
- We can run multiple VM on a single hypervisor.
- Hypervisor manages the resources allocated to these VM's from the underlying physical machine.

Hypervisor/Virtual Machine Manager

- Hypervisor is a piece of software that runs above the physical server or host OS.
- The primary purpose of a hypervisor is to share the underlying hardware resources by presenting a virtual hardware platform to the guest operating systems.
- The hypervisor also maintains strict isolation and ensures non-interference between virtual machines.
- **VMware, Microsoft, Citrix** are name of some of the companies whose hypervisor are available in the market.

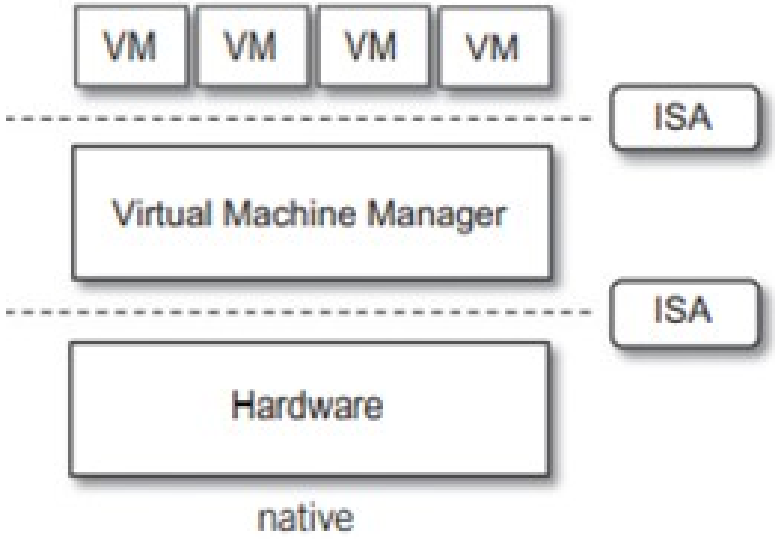


Hypervisor Types

- Hypervisors are classified as one of two types:
  - Type 1
  - Type 2

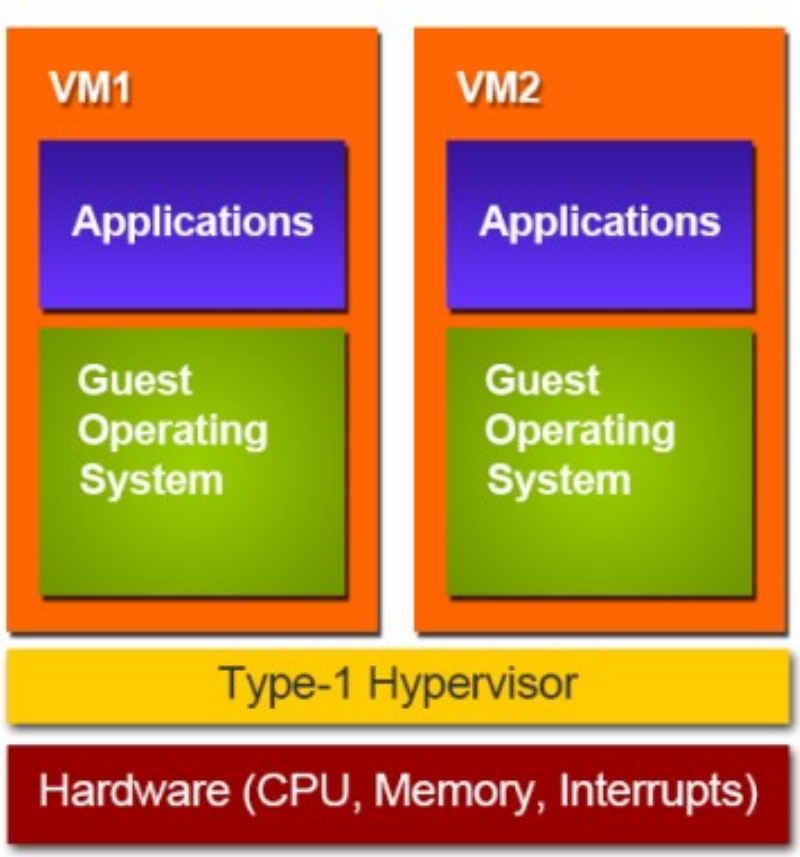
Type I Hypervisor

- Type 1 – This type of hypervisor is also known as native or bare-metal.
- They run directly on the hardware with guest operating systems running on top of them.
- They interact directly with the ISA interface exposed by the underlying hardware and emulate this interface in order to allow the management of guest operating systems.
- Examples include VMware **ESX**, Citrix **XenServer**, and Microsoft's **Hyper-V**.



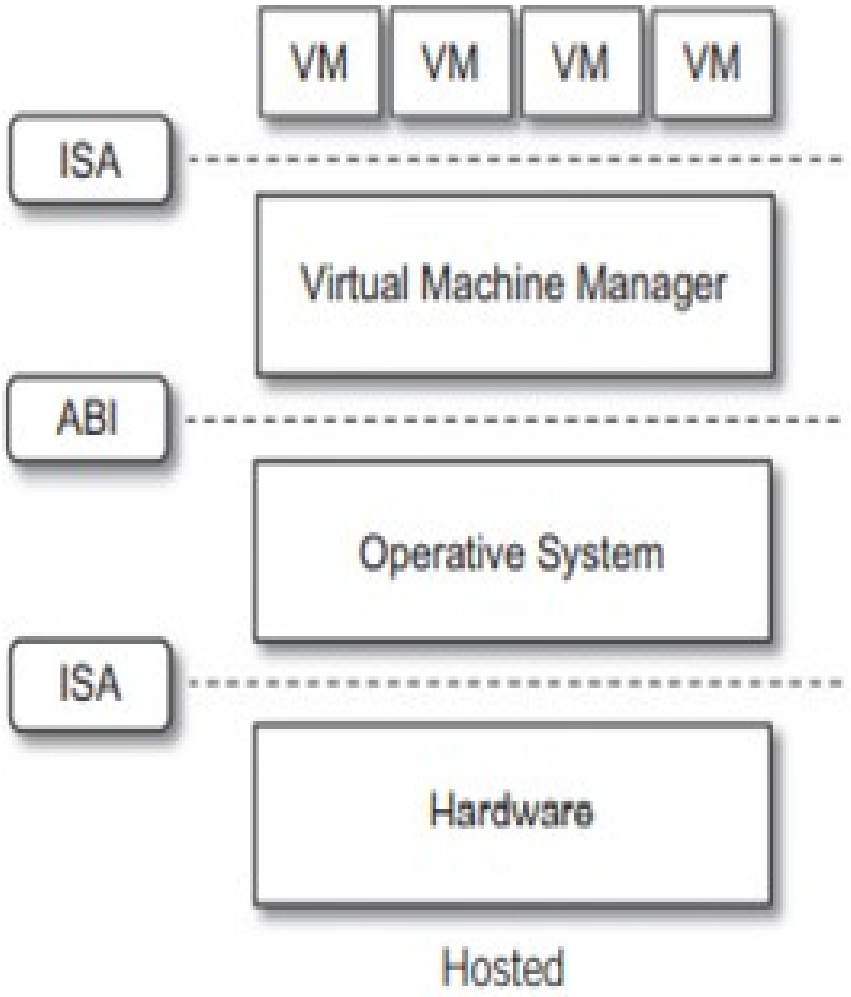
Type I Hypervisor

- Type 1 – This type of hypervisor is also known as native or bare-metal.
- They run directly on the hardware with guest operating systems running on top of them.
- They interact directly with the ISA interface exposed by the underlying hardware and emulate this interface in order to allow the management of guest operating systems.
- Examples include VMware **ESX**, Citrix **XenServer**, and Microsoft's **Hyper-V**.



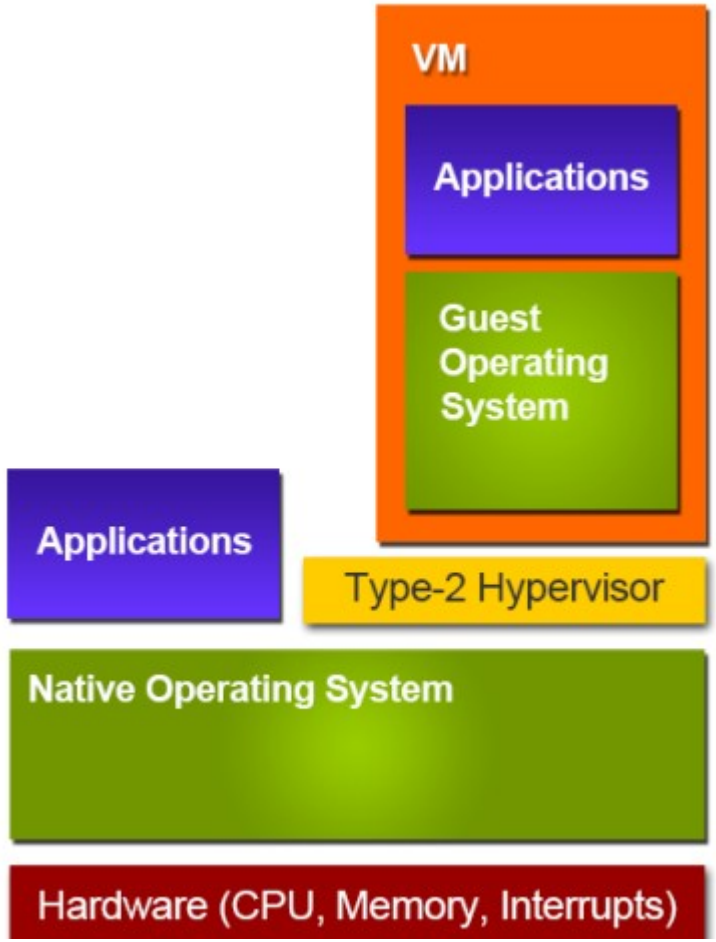
Type II Hypervisor

- Type II hypervisors require the support of an operating system to provide virtualization services.
- This means that they are programs **managed by the operating system**, which interact with it through the ABI and emulate the ISA of virtual hardware for guest operating systems.
- This type of hypervisors is also called hosted virtual machine, since it is hosted within an operating system.
- Examples include VMware **Workstation** and SWSoft's **Parallels Desktop**.



Type II Hypervisor

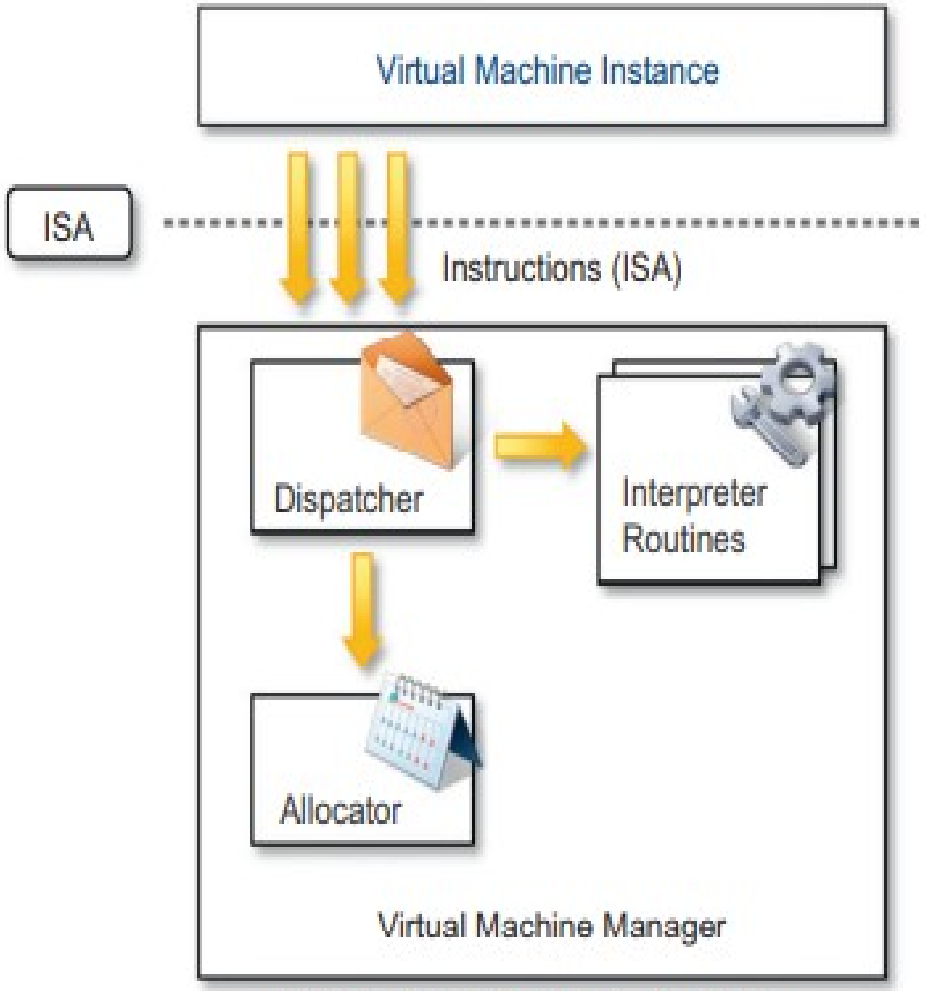
- Type II hypervisors require the support of an operating system to provide virtualization services.
- This means that they are programs **managed by the operating system**, which interact with it through the ABI and emulate the ISA of virtual hardware for guest operating systems.
- This type of hypervisors is also called hosted virtual machine, since it is hosted within an operating system.
- Examples include VMware **Workstation** and SWSoft's **Parallels Desktop**.



Hypervisor Reference Architecture

Three main modules of VMM are as follows:

- Dispatcher: The dispatcher constitutes the entry point of the VMM and reroutes the instructions issued by the virtual machine instance to one of the two other modules.
- Allocator: The allocator is responsible for deciding the system resources to be provided to the VM. Whenever a virtual machine tries to execute an instruction that results in changing the machine resources associated with that VM, the allocator is invoked by the dispatcher.
- Interpreter: The interpreter module consists of interpreter routines. These are executed whenever a virtual machine executes a privileged instruction: a trap is triggered, and the corresponding routine is executed.



A hypervisor reference architecture.

Hypervisor Reference Architecture

The criteria that need to be met by a virtual machine manager are as follows:

- Equivalence: A guest running under the control of a virtual machine manager should exhibit the same behavior that when it is executed directly on the physical host.
- Resource control: The virtual machine manager should be in complete control of virtualized resources.
- Efficiency: A statistically dominant fraction of the machine instructions should be executed without intervention from the virtual machine manager.