

Cloud Computing

Prepared by Dr. Sherif Mostafa

IT381

Lecture 2



Virtualization

Virtualization

Layering and virtualization.

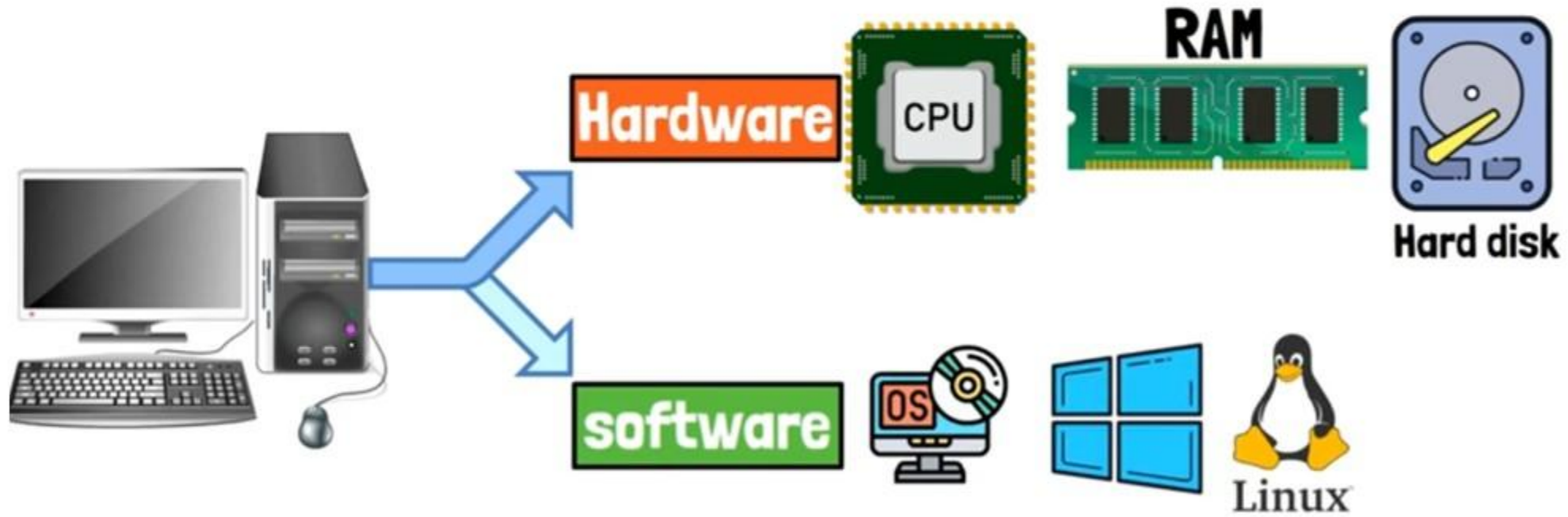
Virtual machine monitor.

Virtual machine.

x86 support for virtualization.

Full and paravirtualization.

Computer System

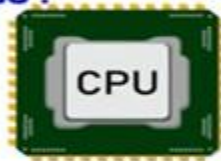


Computer Resources

1960

Computer Hardware resources :

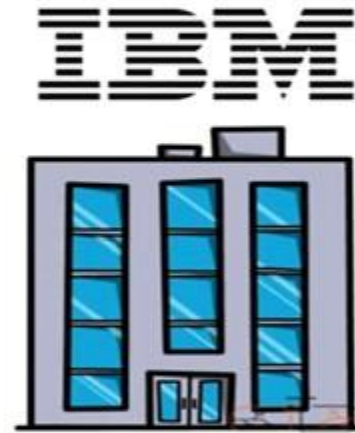
👉 CPU (Central Processing Unit)



👉 RAM (Random Access Memory)



👉 storage

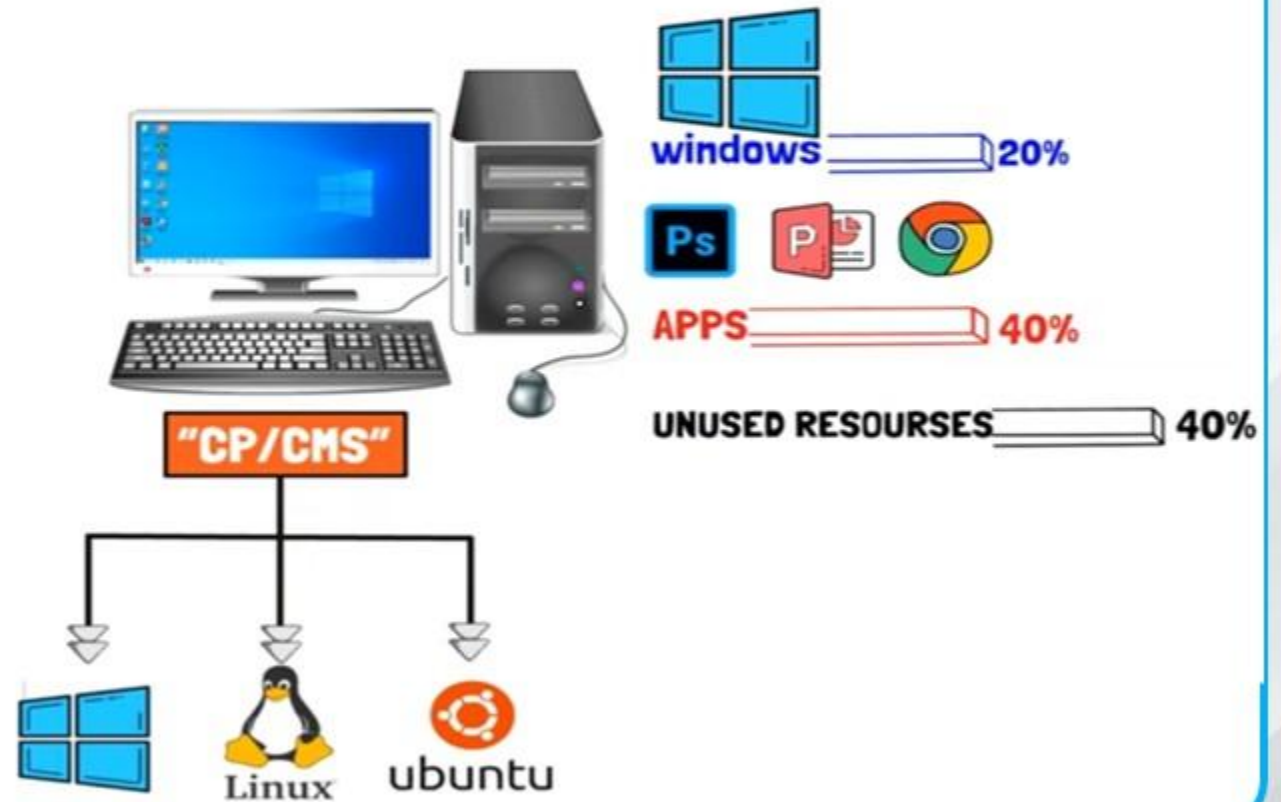


but in the sixties of the last century, a team from IBM thought that they wanted to exploit all the device's resources, which are, as we agreed, represented by the CPU, RAM, and storage, whether HDD or SSD

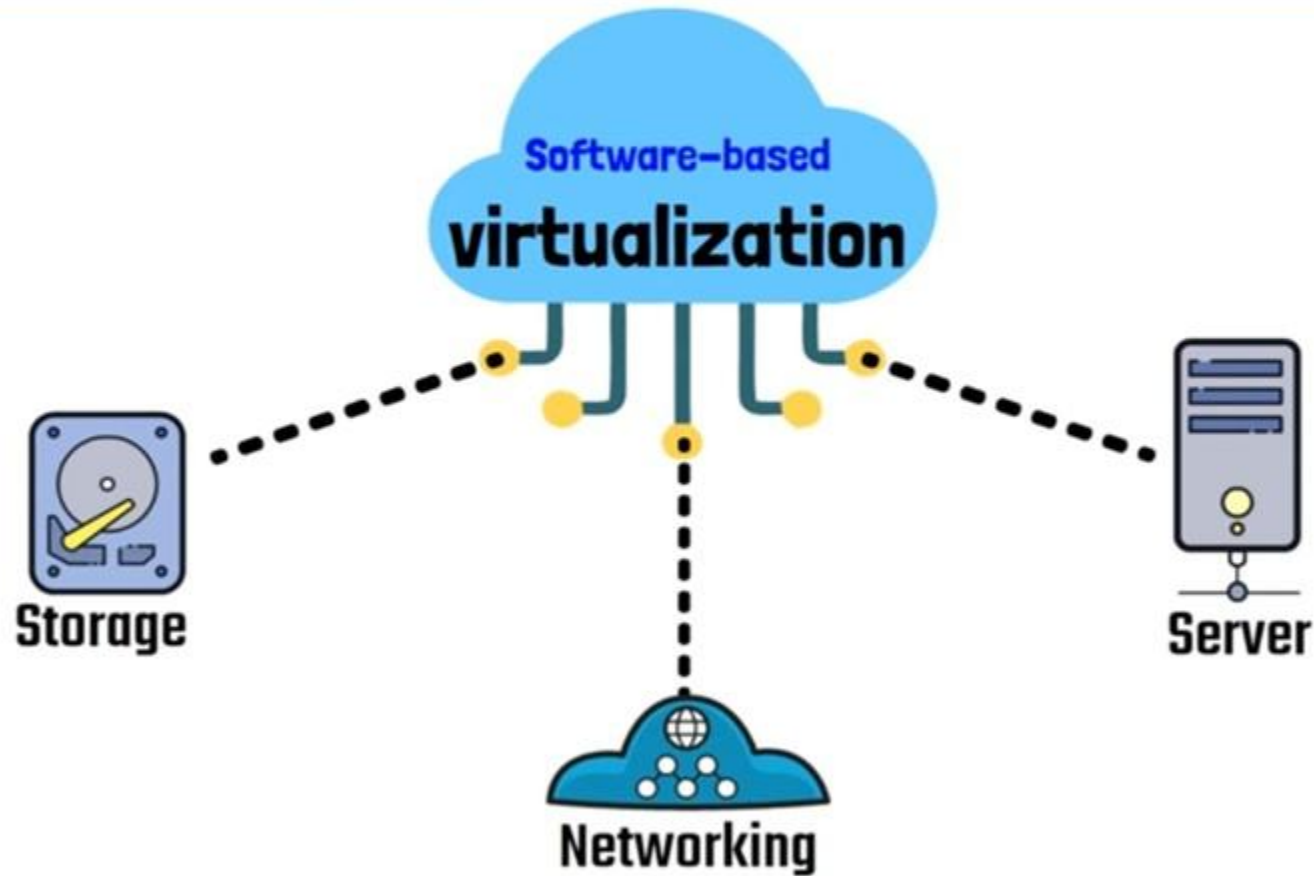
Computer Resources



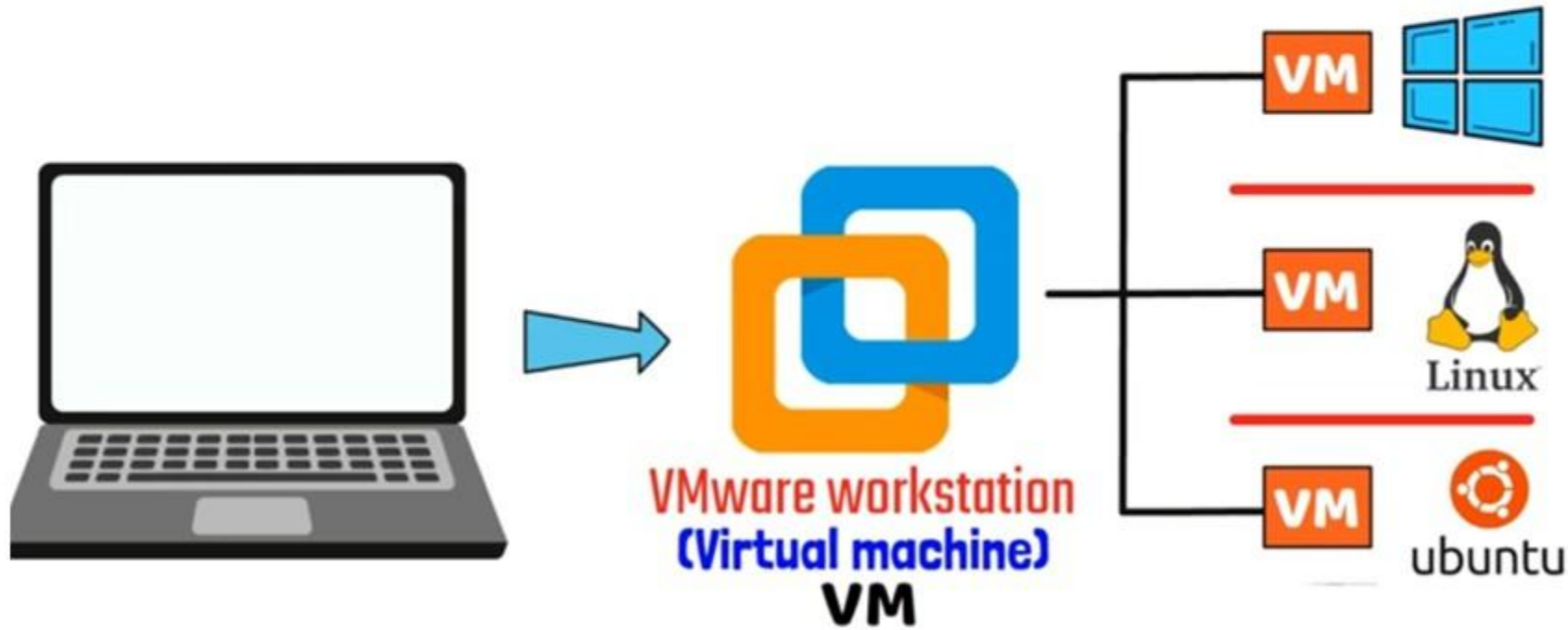
Server



Virtual Machine

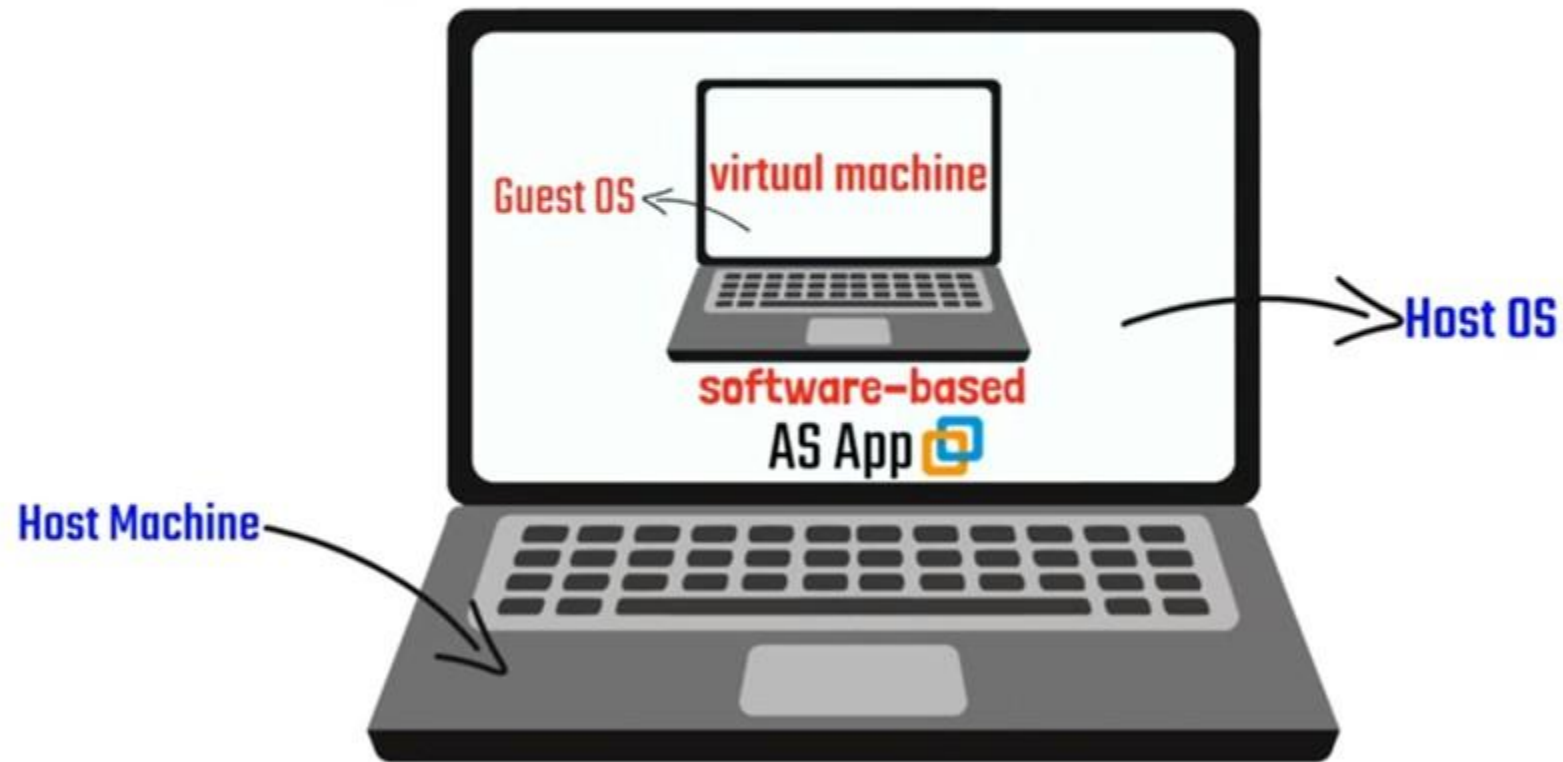


Virtual Machine



Virtual Machine

Virtual machine:



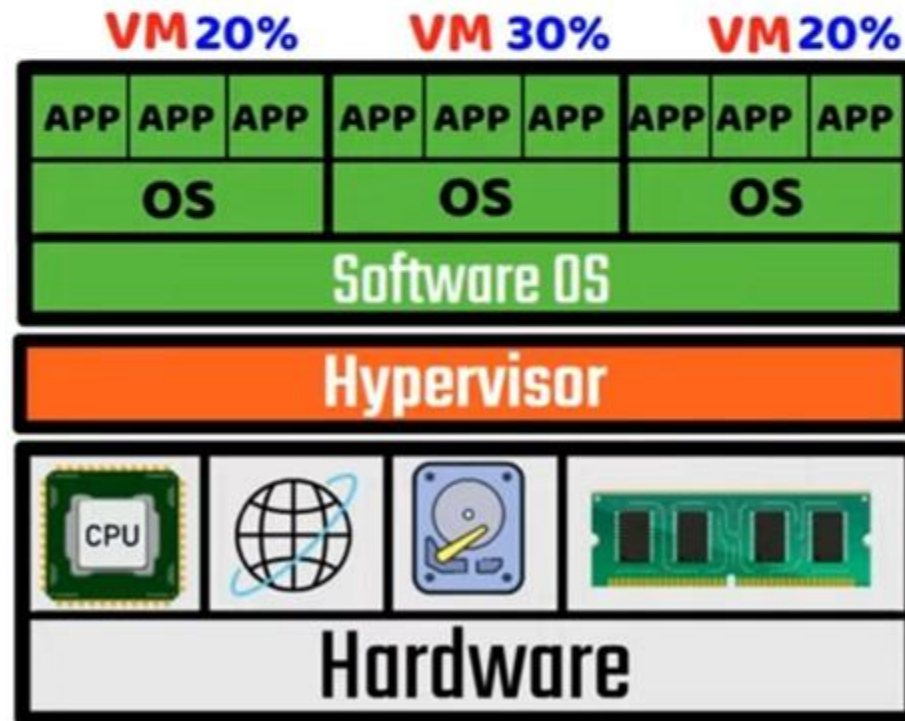
Hypervisor

Virtualization

Hypervisor:



Virtualization Software



100 %

Hypervisor (Type 1 & 2)

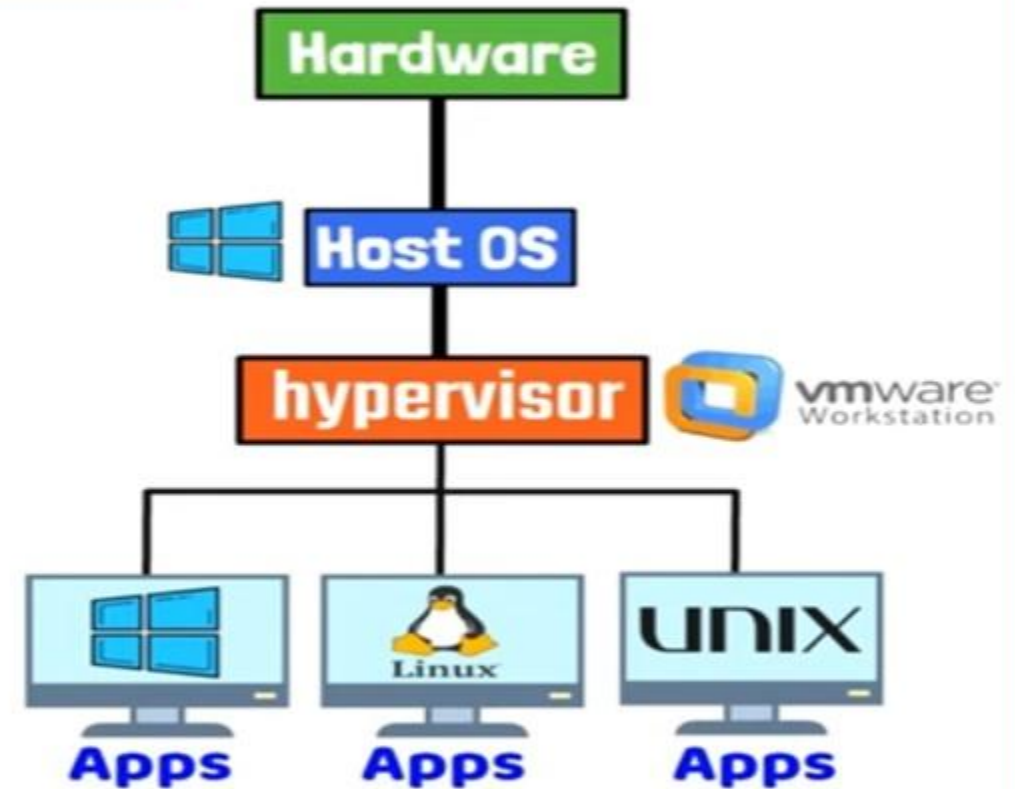
Type-1
(bare-metal hypervisor)

Examples



hypervisor

Type-2



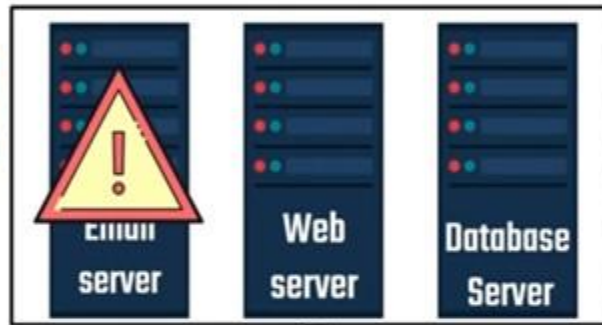
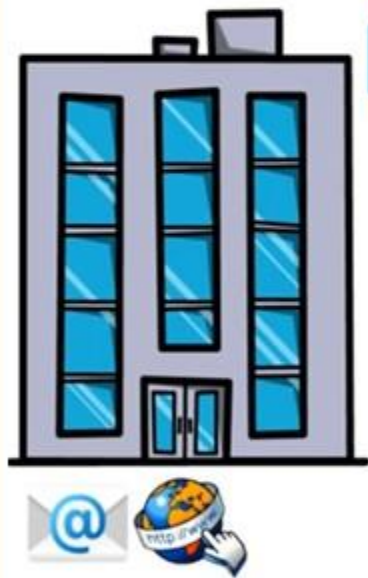
examples



Chapter

02

Hypervisor (Type 1): Example



3



UNIX



+3

Type-1
(bare-metal hypervisor)
one server

ESXi
hypervisor
vmware



+3

Hardware

hypervisor



Hypervisor (Type 1)

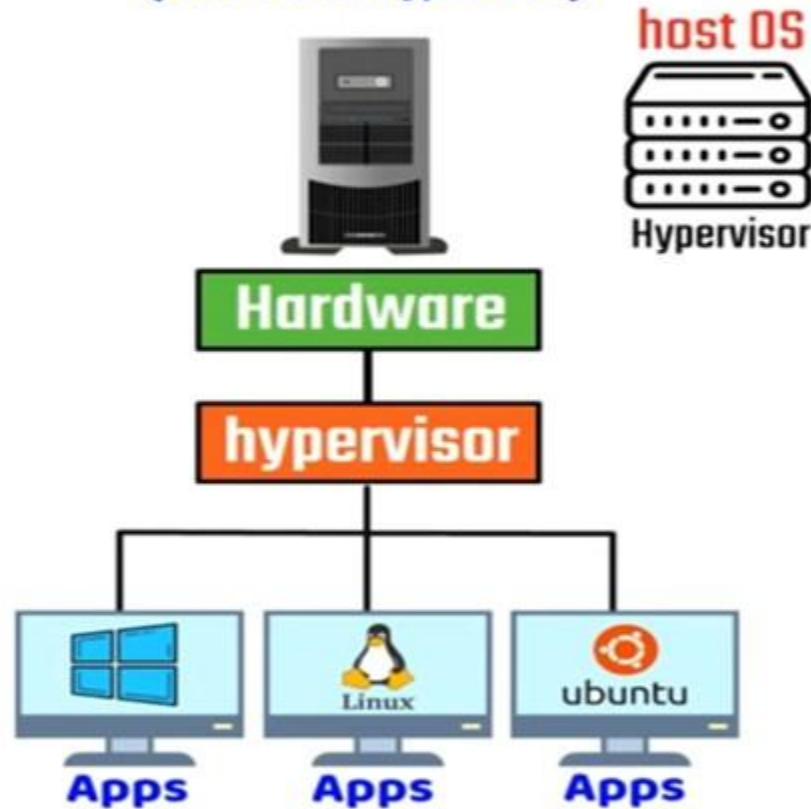
Advantages



Management

Here, of course, the company saved the money for the servers and saved the cooling and energy that they would have needed to operate. Also, the maintenance of the servers, in addition to the fact that their management will be much simpler.

Hypervisor (Type 2)

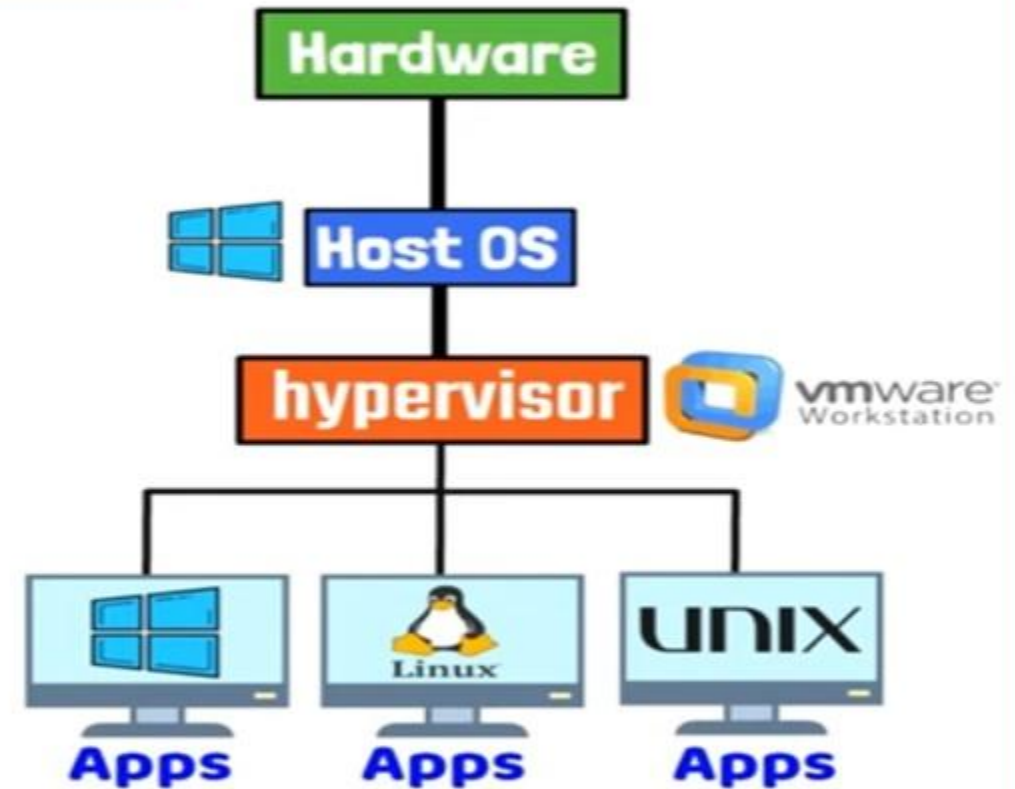
Type-1
(bare-metal hypervisor)

Examples



hypervisor

Type-2



examples



Hypervisor (Type 2)

Uses

Testing
Virtualized environment



Type-2
Hypervisor



Hypervisor (Type 2): Example

 **Student**

Server administration
courses



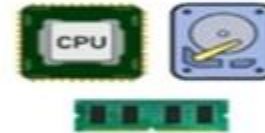
Hypervisor (Type 2): disadvantages

Type-2

Disadvantages



Slower than type-1



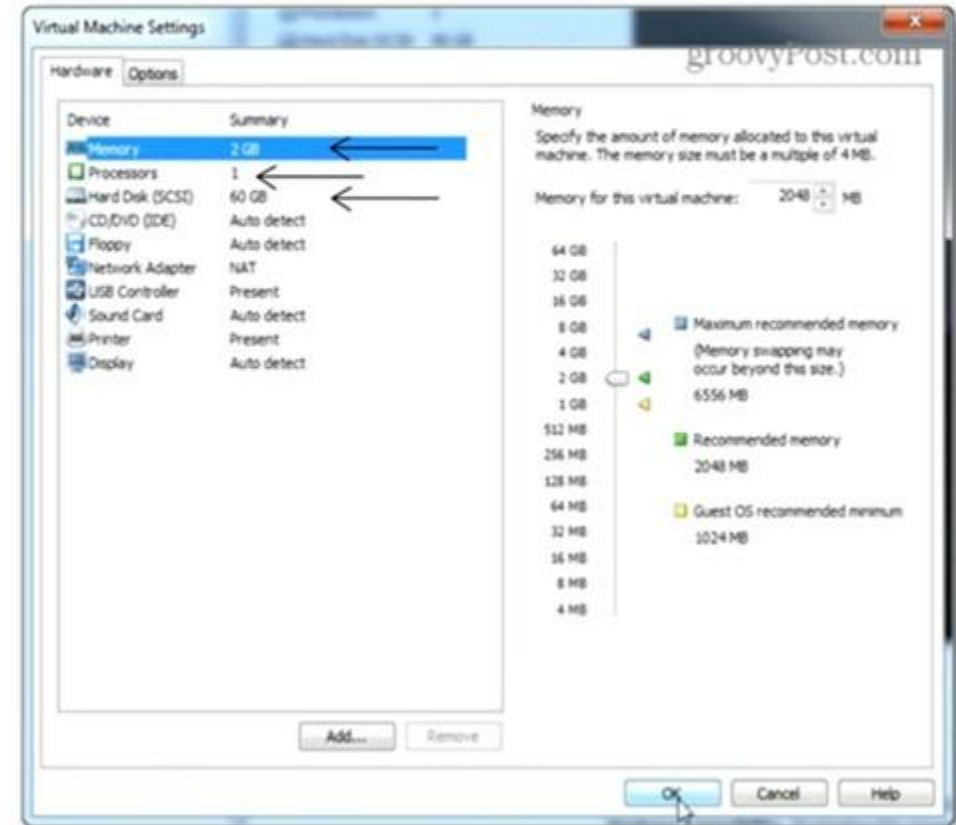
Unreliable



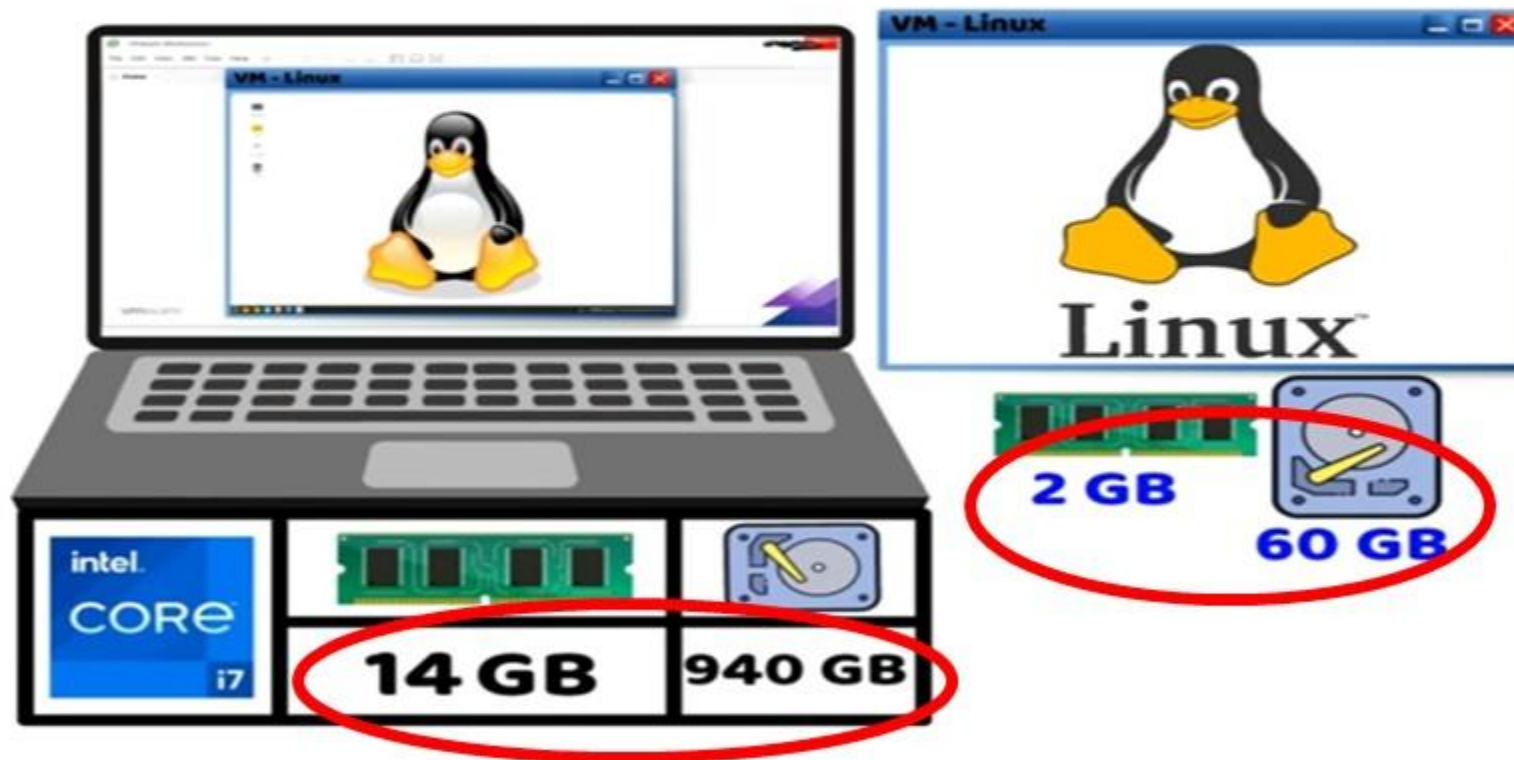
Chapter

02

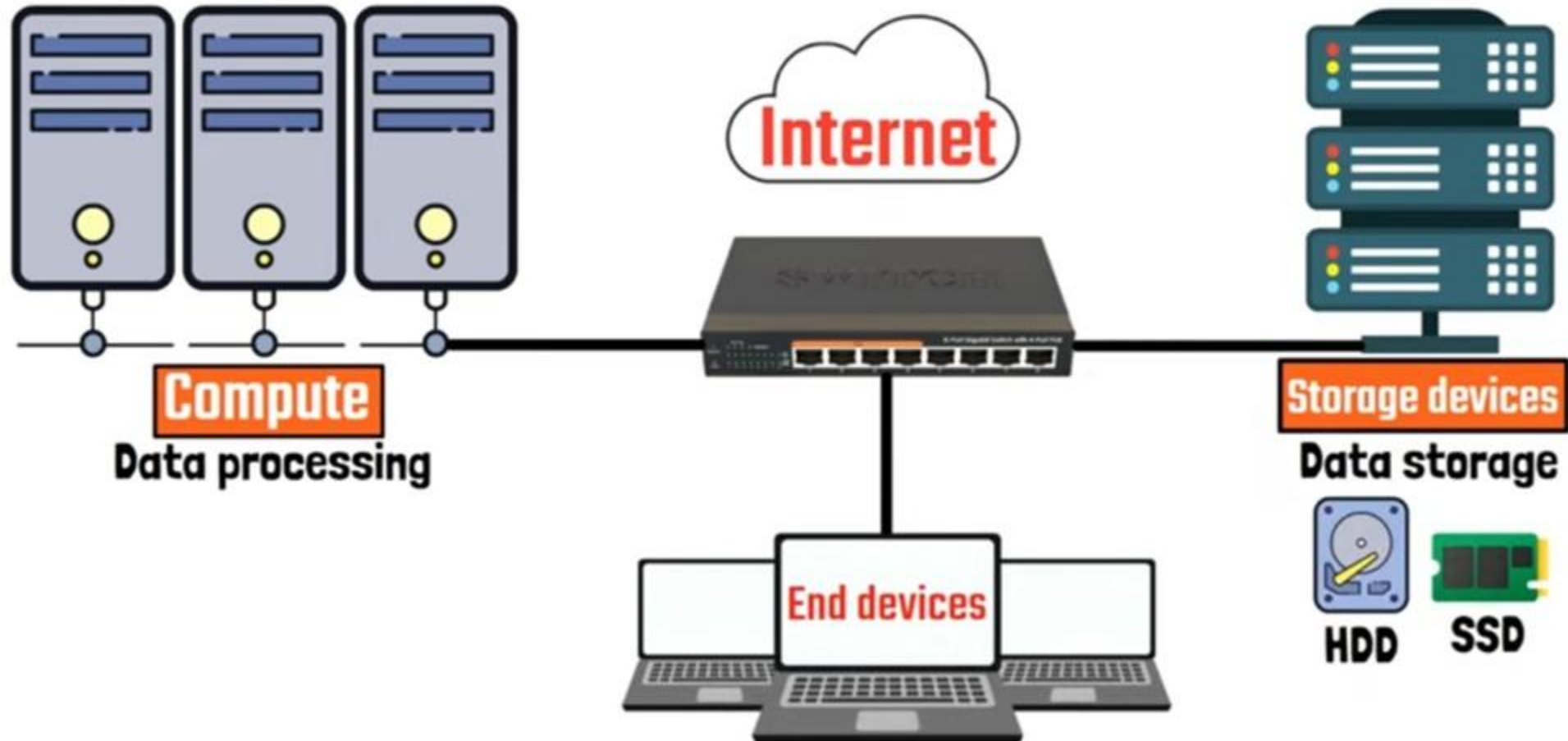
Example on Type-2



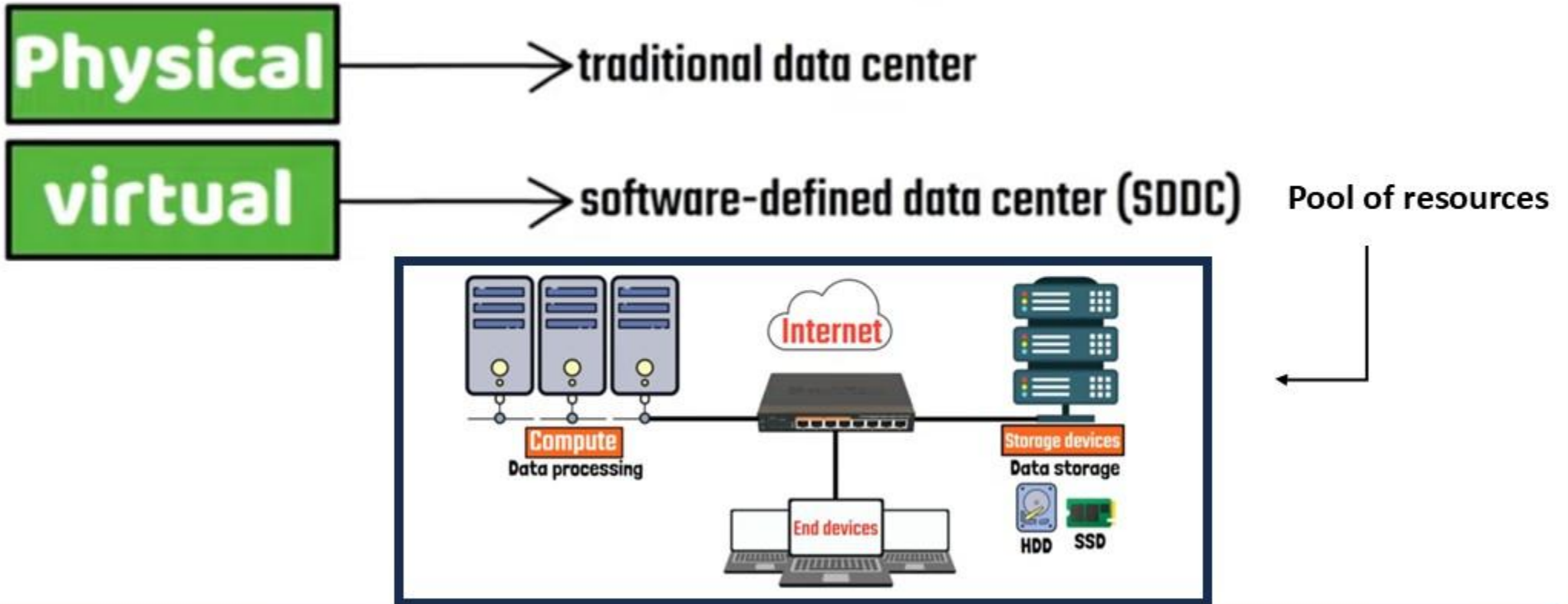
Example on Type-2



Datacenter

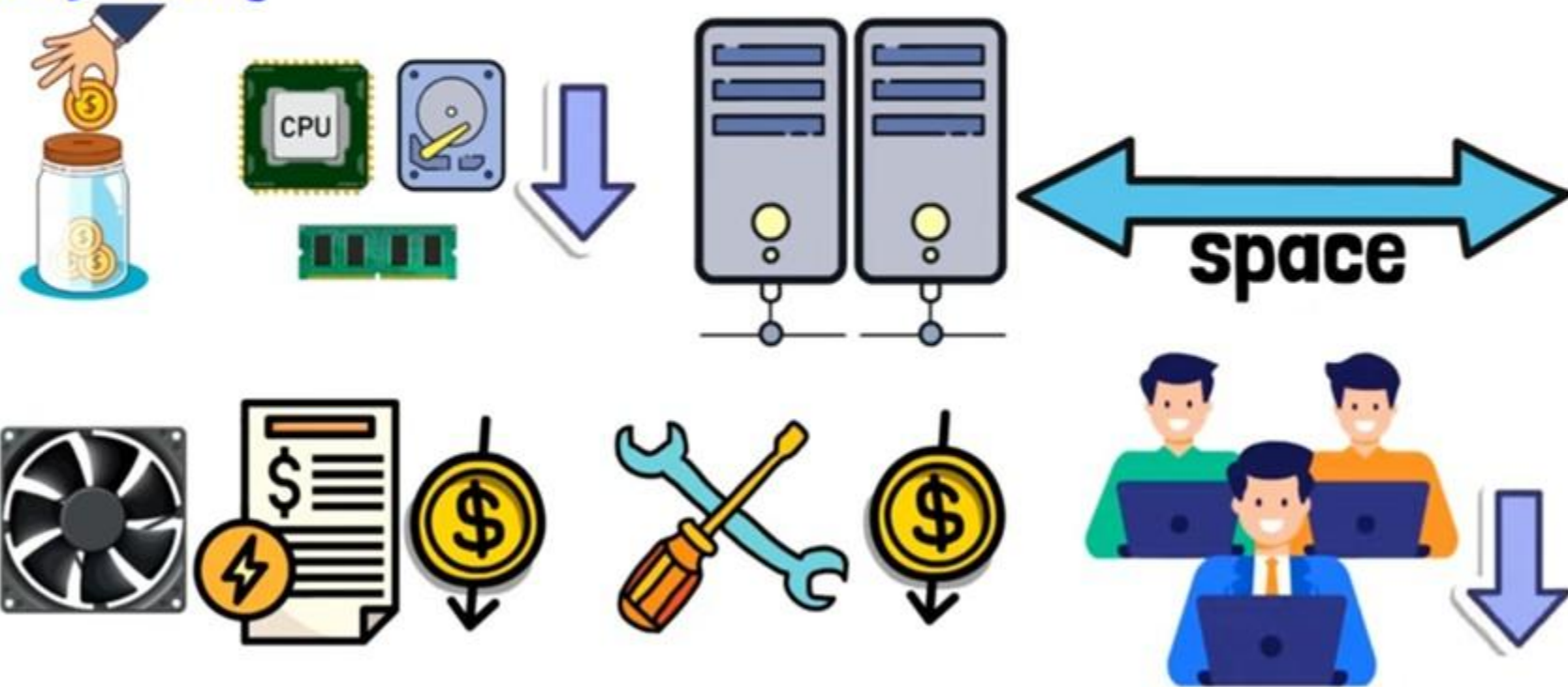


Motivation



Advantages

1- Money-saving

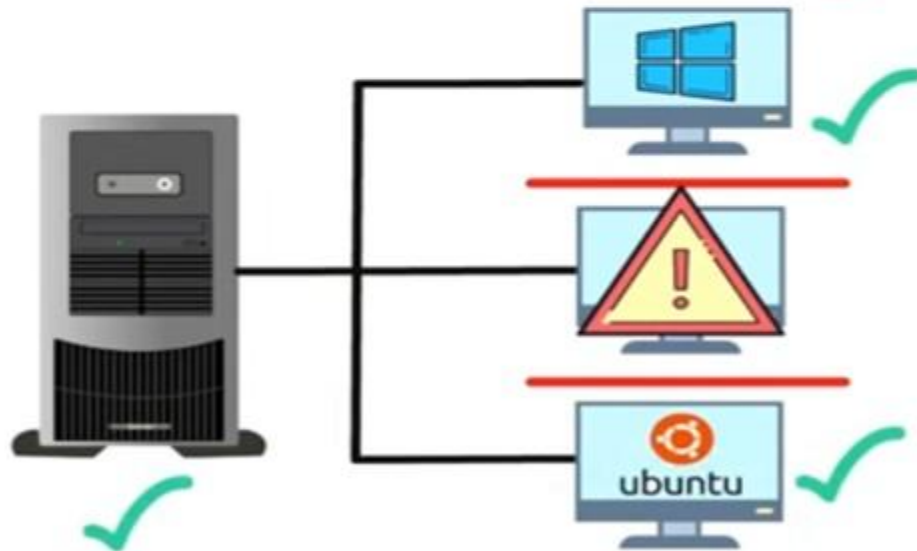


Advantages

2- Business Continuity



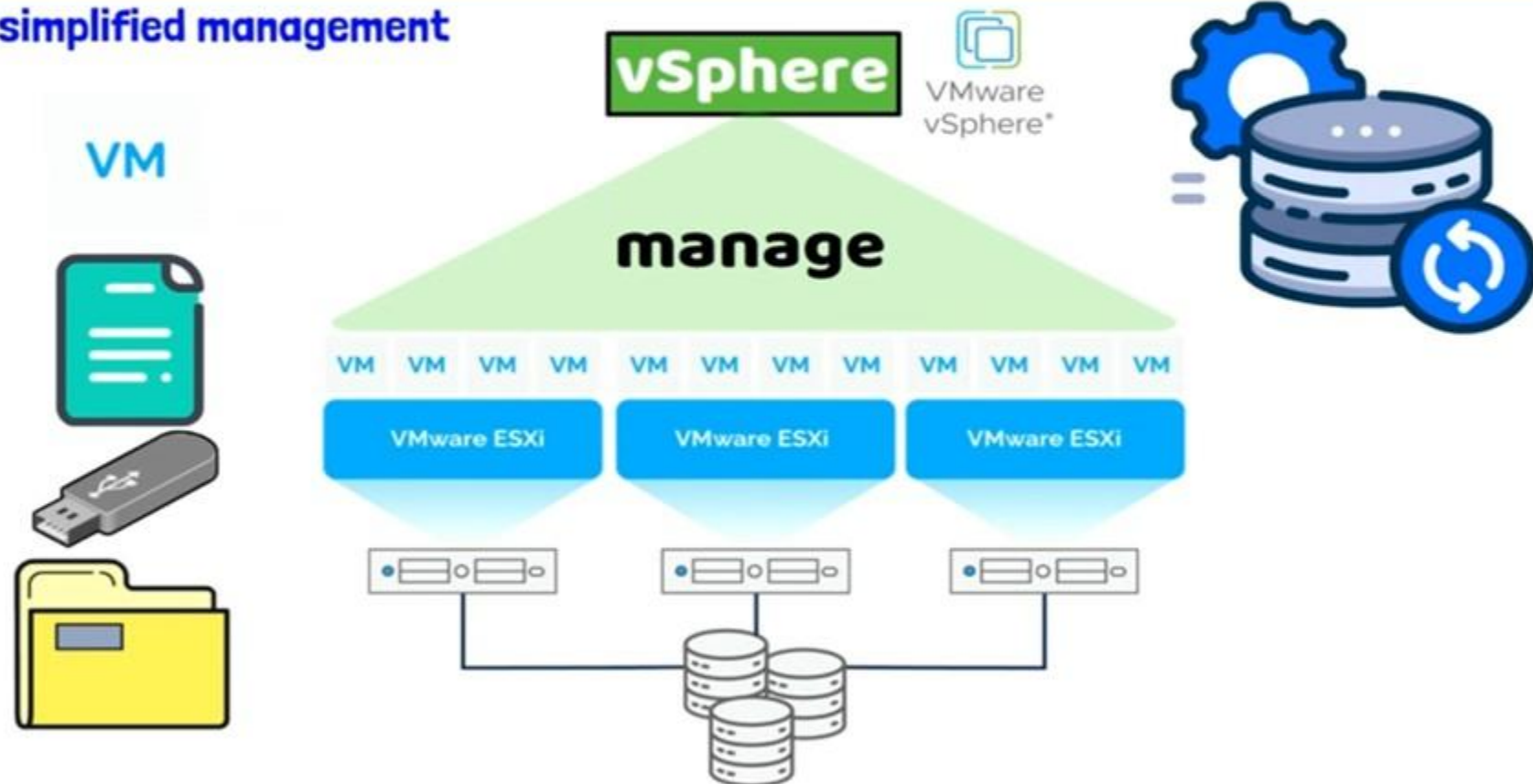
decrease down time



Chapter
02

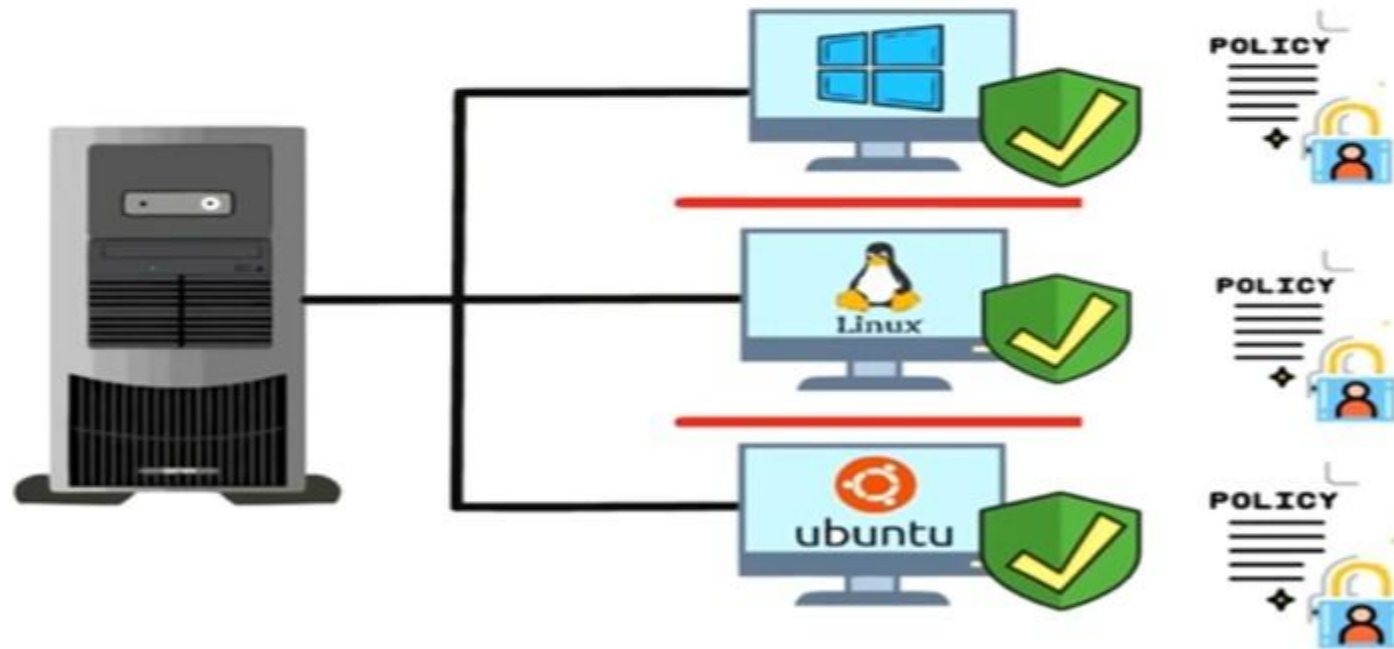
Advantages

3- simplified management



Advantages

4- security



Motivation

- Three fundamental abstractions are necessary to describe the operation of a computing systems:
(1) interpreters/processors, (2) memory, (3) communications links
- As the scale of a system and the size of its users grows, it becomes very challenging to manage its recourses.
- Resource management issues:
 - provision for peak demands → **overprovisioning**
 - **heterogeneity** of hardware and software
 - machine failures

Motivation

- **Virtualization is a basic enabler of Cloud Computing; it simplifies the management of physical resources for the three abstractions**
- For example, the state of a virtual machine (VM) running under a virtual machine monitor (VMM) can be saved and migrated to another server to balance the load
- For example, virtualization allows users to operate in environments they are familiar with, rather than forcing them to specific ones

- *"**Virtualization**, in computing, refers to the act of creating a **virtual (rather than actual) version of something**, including but not limited to a virtual computer hardware platform, operating system (OS), storage device, or computer network resources."* from Wikipedia
- Virtualization abstracts the underlying resources; simplifies their use; isolates users from one another; and supports replication which increases the elasticity of a system

Motivation

- Cloud resource virtualization is important for:
 - Performance isolation
 - as we can dynamically assign and account for resources across different applications
 - System security:
 - as it allows isolation of services running on the same hardware
 - Performance and reliability:
 - as it allows applications to migrate from one platform to another
 - The development and management of services offered by a provider

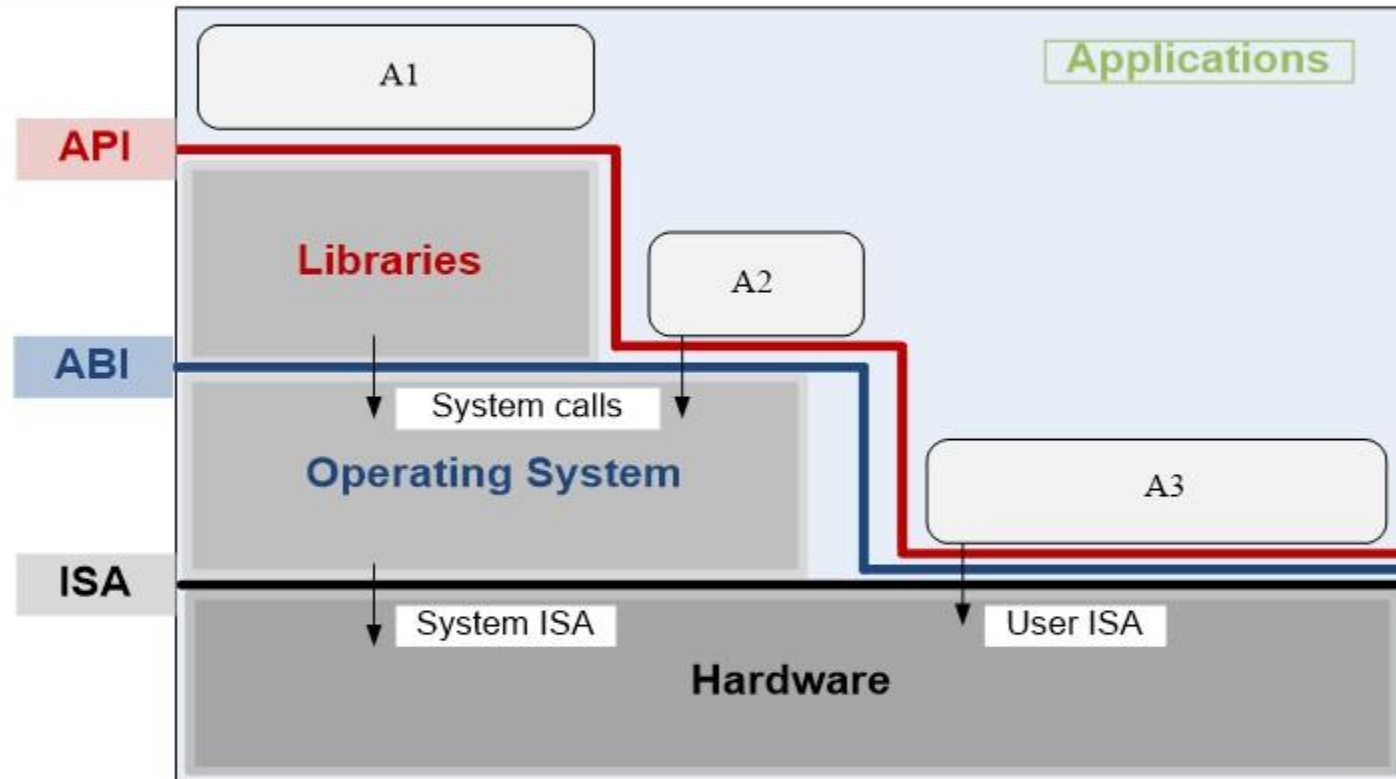
Virtualization

- Virtualization simulates the interface to a physical object by:
 - Multiplexing: creates multiple virtual objects from one instance of a physical object. Many virtual objects to one physical. Example - a processor is multiplexed among a number of processes or threads.
 - Aggregation: creates one virtual object from multiple physical objects. One virtual object to many physical objects. Example - a number of physical disks are aggregated into a RAID disk.
 - Emulation: constructs a virtual object of a certain type from a different type of a physical object. Example - a physical disk emulates a Random Access Memory (RAM).
 - Multiplexing and emulation. Examples - virtual memory with paging multiplexes real memory and disk; a virtual address emulates a real address.

Layering and Virtualization

- Layering – a common approach to manage system complexity:
 - Simplifies the description of the subsystems; each subsystem is abstracted through its interfaces with the other subsystems
 - Minimises the interactions among the subsystems of a complex system
 - With layering we are able to design, implement, and modify the individual subsystems independently
- Layering in a computer system:
 - Hardware
 - Software
 - Operating system
 - Libraries
 - Applications

Layering and Interfaces



Application Programming Interface (API), Application Binary Interface (ABI), and Instruction Set Architecture (ISA). An application uses library functions (A1), makes system calls (A2), and executes machine instructions (A3).

- **Instruction Set Architecture** (ISA) – at the boundary between hardware and software.
- ISA is the blueprint of a computer's CPU! It defines the instructions, data types, and registers that the processor can use to execute tasks.
- The ISA provides an interface between the hardware and software of a computer system.
- It defines the capabilities and limitations of the processor and determines the types of applications that can be executed on the computer.

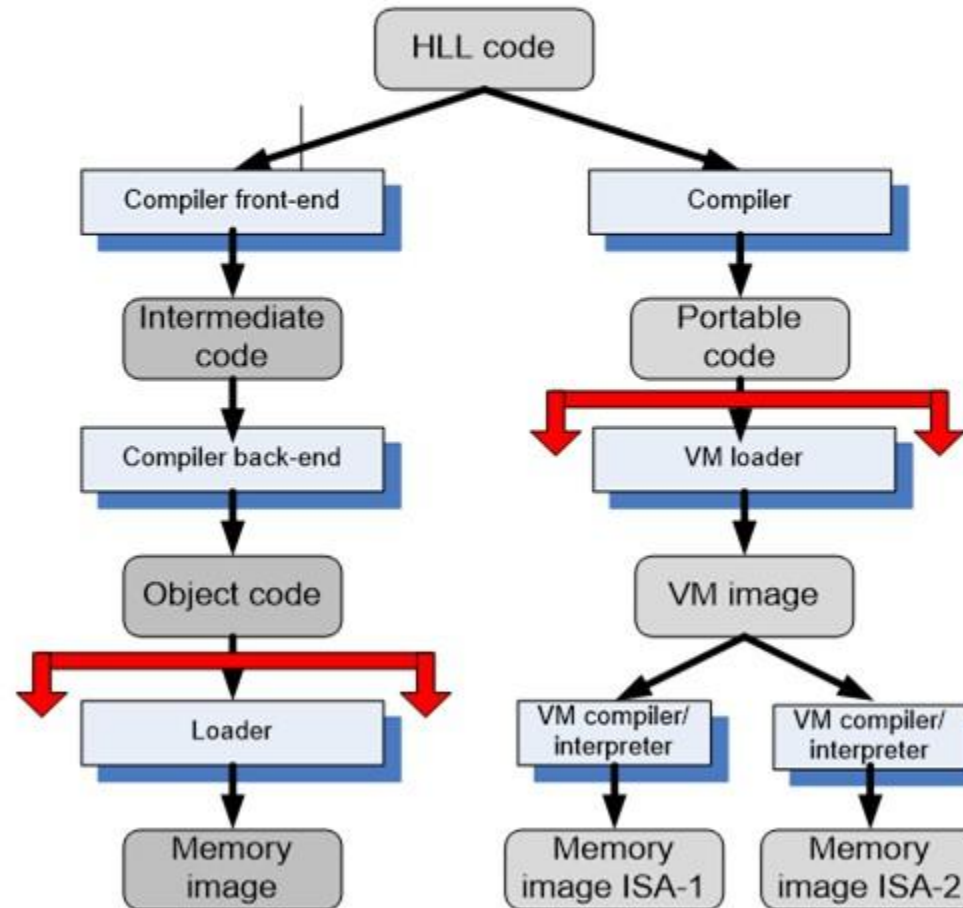
Interfaces

- **Application Binary Interface (ABI)** – is a set of rules and conventions that dictate how software components interact at the binary level. It defines the low-level interface between an application and the operating system or hardware. The ABI is crucial for enabling software to function correctly across various environments while maintaining performance and compatibility.
- **Application Program Interface (API)** – defines the set of instructions the hardware was designed to execute and gives the application access to the ISA; it includes high-level language (HLL) library calls which often invoke system calls.

Code portability

- Binaries created by a compiler for a specific ISA and a specific operating systems are not portable
- It is possible, though, to compile a HLL program for a virtual machine (VM) environment where portable code is produced and distributed and then converted by binary translators to the ISA of the host system
- A **dynamic binary translation** converts blocks of guest instructions from the portable code to the host instruction and leads to a significant performance improvement, as such blocks are cached and reused

HLL Language Translations



History of Virtualization

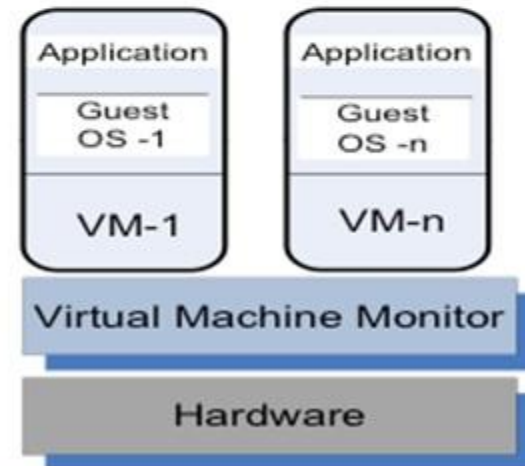
- **1960's, IBM: CP/CMS** control program: a virtual machine operating system for the IBM System/360 Model 67
- **2000, IBM: z-series** with 64-bit virtual address spaces and backward compatible with the System/360
- **1974: Popek and Golberg** from UCLA published "*Formal Requirements for Virtualizable Third Generation Architectures*" where they listed the conditions a computer architecture should satisfy to support virtualization efficiently. The popular x86 architecture that originated in the 1970s did not support these requirements for decades.

History of Virtualization

- **1990's, Stanford researchers, VMware:** Researchers developed a new hypervisor and founded VMware, the biggest virtualization company of today's. First virtualization solution was in 1999 for x86.
- Today many virtualization solutions: Xen from Cambridge, KVM, Hyper-V, ...
- IBM was the first to produce and sell virtualization for the mainframe. But, VMware popularized virtualization for the masses.

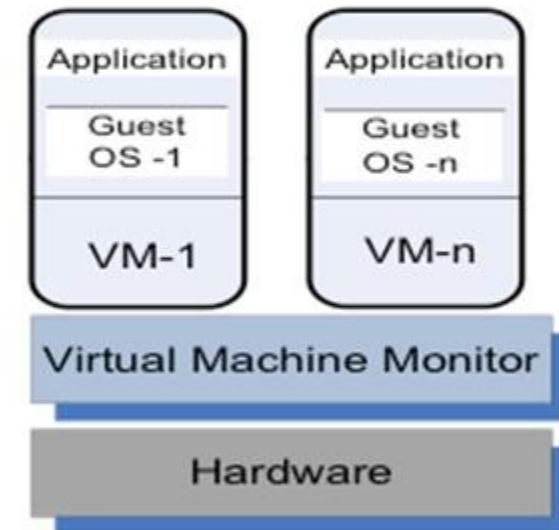
Virtual Machine Monitor (VMM / Hypervisor)

- A **virtual machine monitor (VMM/hypervisor)** partitions the resources of computer system into one or more **virtual machines (VMs)**. Allows several operating systems to run concurrently on a single hardware platform
- A VM is an execution environment that runs an OS
- VM – an isolated environment that appears to be a whole computer, but only has access to a portion of the computer resources



Virtual Machine Monitor (VMM / Hypervisor)

- A VMM allows:
 - Multiple services to share the same platform
 - Live migration - the movement of a server from one platform to another
 - System modification while maintaining backward compatibility with the original system
 - Enforces isolation among the systems, thus security
- A **guest operating system** is an OS that runs in a VM under the control of the VMM.

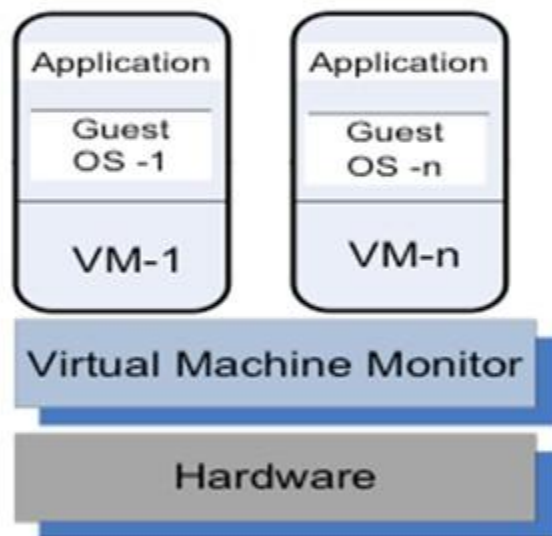


VMM Virtualizes the CPU and the Memory

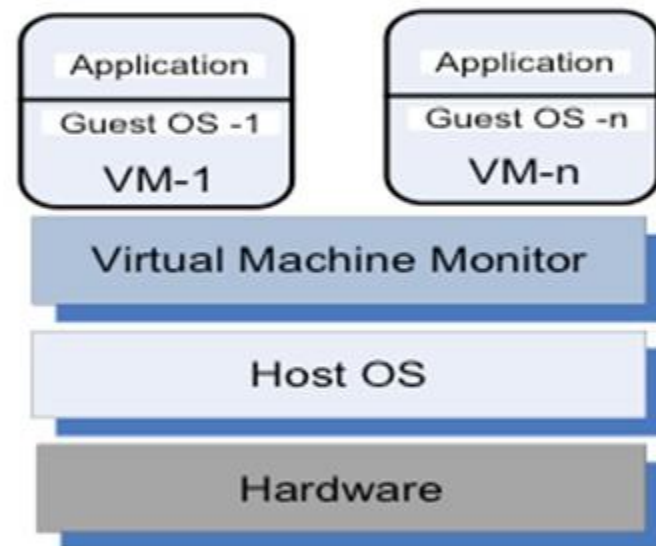
- A VMM (also hypervisor) (how to):
 - Traps the privileged instructions executed by a guest OS and enforces the correctness and safety of the operation
 - Traps interrupts and dispatches them to the individual guest operating systems
 - Controls the virtual memory management
 - Maintains a shadow page table for each guest OS and replicates any modification made by the guest OS in its own shadow page table. This shadow page table points to the actual page frame, and it is used by the Memory Management Unit (MMU) for dynamic address translation.
 - Monitors the system performance and takes corrective actions to avoid performance degradation. For example, the VMM may swap out a VM to avoid thrashing.

Type 1 and 2 Hypervisors

Type 1 Hypervisor



Type 2 Hypervisor



Taxonomy of VMMs:

1. Type 1 Hypervisor (bare metal, native): supports multiple virtual machines and runs directly on the hardware (e.g., VMware ESX, Xen, Denali)
2. Type 2 Hypervisor (hosted) VM - runs under a host operating system (e.g., user-mode Linux)

Examples of Hypervisors

Name	Host ISA	Guest ISA	Host OS	guest OS	Company
Integrity VM	<i>x86-64</i>	<i>x86-64</i>	HP-Unix	Linux, Windows HP Unix	HP
Power VM	Power	Power	No host OS	Linux, AIX	IBM
z/VM	z-ISA	z-ISA	No host OS	Linux on z-ISA	IBM
Lynx Secure	<i>x86</i>	<i>x86</i>	No host OS	Linux, Windows	LinuxWorks
Hyper-V Server	<i>x86-64</i>	<i>x86-64</i>	Windows	Windows	Microsoft
Oracle VM	<i>x86, x86-64</i>	<i>x86, x86-64</i>	No host OS	Linux, Windows	Oracle
RTS Hypervisor	<i>x86</i>	<i>x86</i>	No host OS	Linux, Windows	Real Time Systems
SUN xVM	<i>x86, SPARC</i>	same as host	No host OS	Linux, Windows	SUN
VMware EX Server	<i>x86, x86-64</i>	<i>x86, x86-64</i>	No host OS	Linux, Windows Solaris, FreeBSD	VMware
VMware Fusion	<i>x86, x86-64</i>	<i>x86, x86-64</i>	MAC OS <i>x86</i>	Linux, Windows Solaris, FreeBSD	VMware
VMware Server	<i>x86, x86-64</i>	<i>x86, x86-64</i>	Linux, Windows	Linux, Windows Solaris, FreeBSD	VMware
VMware Workstation	<i>x86, x86-64</i>	<i>x86, x86-64</i>	Linux, Windows	Linux, Windows Solaris, FreeBSD	VMware
VMware Player	<i>x86, x86-64</i>	<i>x86, x86-64</i>	Linux Windows	Linux, Windows Solaris, FreeBSD	VMware
Denali	<i>x86</i>	<i>x86</i>	Denali	ILVACO, NetBSD	University of Washington
Xen	<i>x86, x86-64</i>	<i>x86, x86-64</i>	Linux Solaris	Linux, Solaris NetBSD	University of Cambridge

Performance and Security Isolation

- The run-time behavior of an application is affected by other applications running concurrently on the same platform and competing for CPU cycles, cache, main memory, disk and network access. Thus, it is difficult to predict the completion time!
- Performance isolation - a critical condition for QoS guarantees in shared computing environments
- A VMM is a much simpler and better specified system than a traditional operating system. Example - Xen has approximately 60,000 lines of code; Denali has only about half: 30,000 (Xen is a type-1 hypervisor that enables virtualization; Denali is a research hypervisor designed to experiment with new virtualization techniques)
- The security vulnerability of VMMs is considerably reduced as the systems expose a much smaller number of privileged functions. For example, Xen VMM has 28 hyper calls while Linux has 100s of system calls



Thank You