# History

- First appeared in IBM mainframes in 1972
- Allowed multiple users to share a batch-oriented system
- Formal definition of virtualization helped move it beyond IBM
  1. A VMM provides an environment for programs that is essentially identical to the original machine
  2. Programs running within that environment show only minor performance decreases
  3. The VMM is in complete control of system resources
- In late 1990s Intel CPUs fast enough for researchers to try virtualizing on general purpose PCs
  - Xen and VMware created technologies, still used today
  - Virtualization has expanded to many OSes, CPUs, VMMs

# History(con..)

- VMware – Modern Virtual Machine System.
- Founded 1998, Mendel Rosenblum *et al.*
    - Research at Stanford University
- *VMware Workstation*
    - Separates *Host OS* from *virtualization layer*
    - Host OS may be Windows, Linux, etc.
    - Wide variety of Guest operating systems

# Virtual Machine

❧ :- A virtual machine (VM) is a software implementation of a machine that executes programs like a physical machine. It shares physical hardware resources with the other users but isolates the OS or application to avoid changing the end-user experience.

❧ Virtual Machine is the part of cloud computing.

# Why We Need VM?

ᴓ

- ❧ Where are your Machine?
- ❧ Who has access to these system?
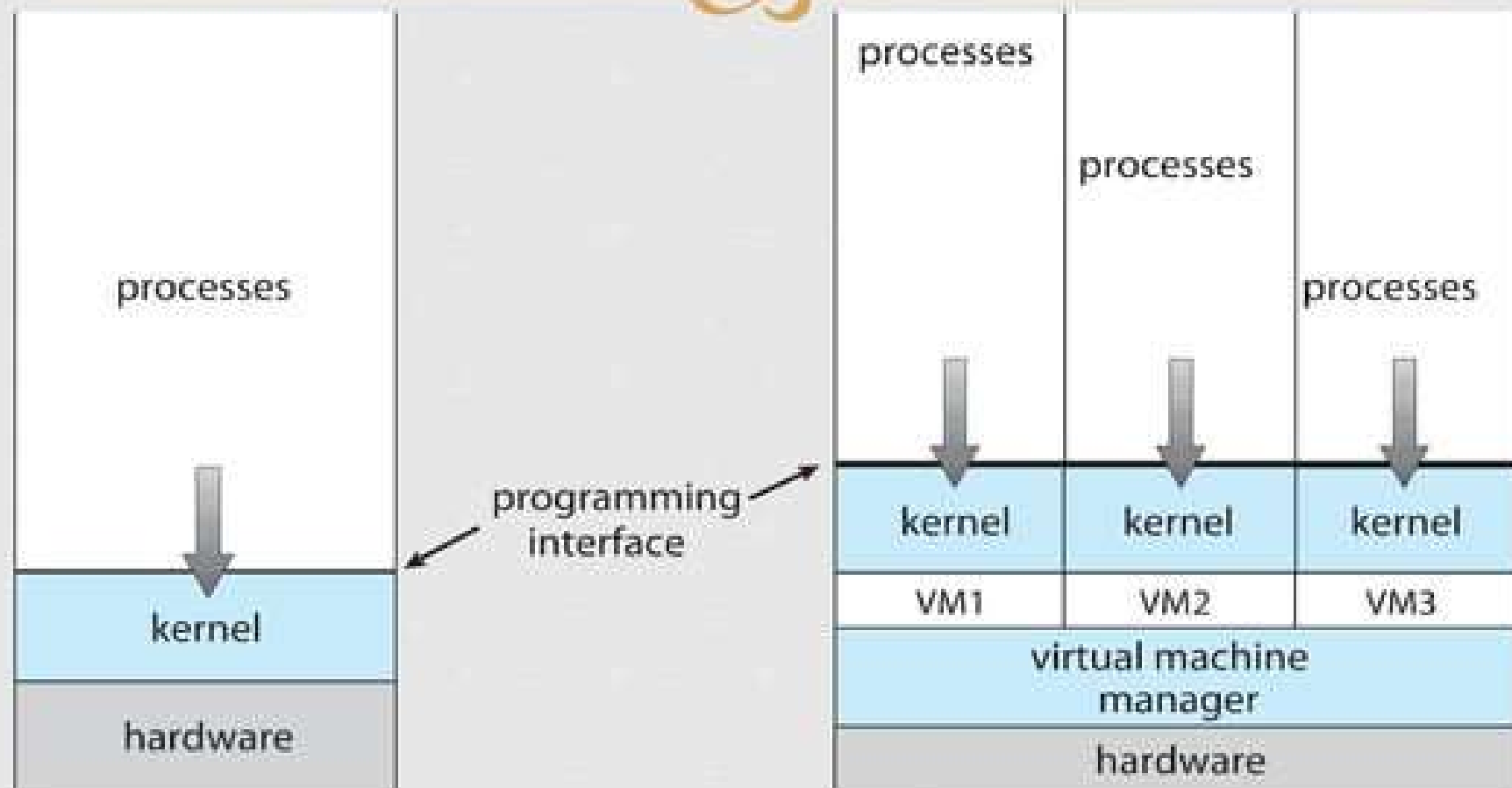- ❧ How are they managed?
- ❧ What security is in place?

# We Do Virtualization

ൟ

- ☞ VM share the centralized server processor, memory & storage.
- ☞ Earlier 800 Million unique server is working.
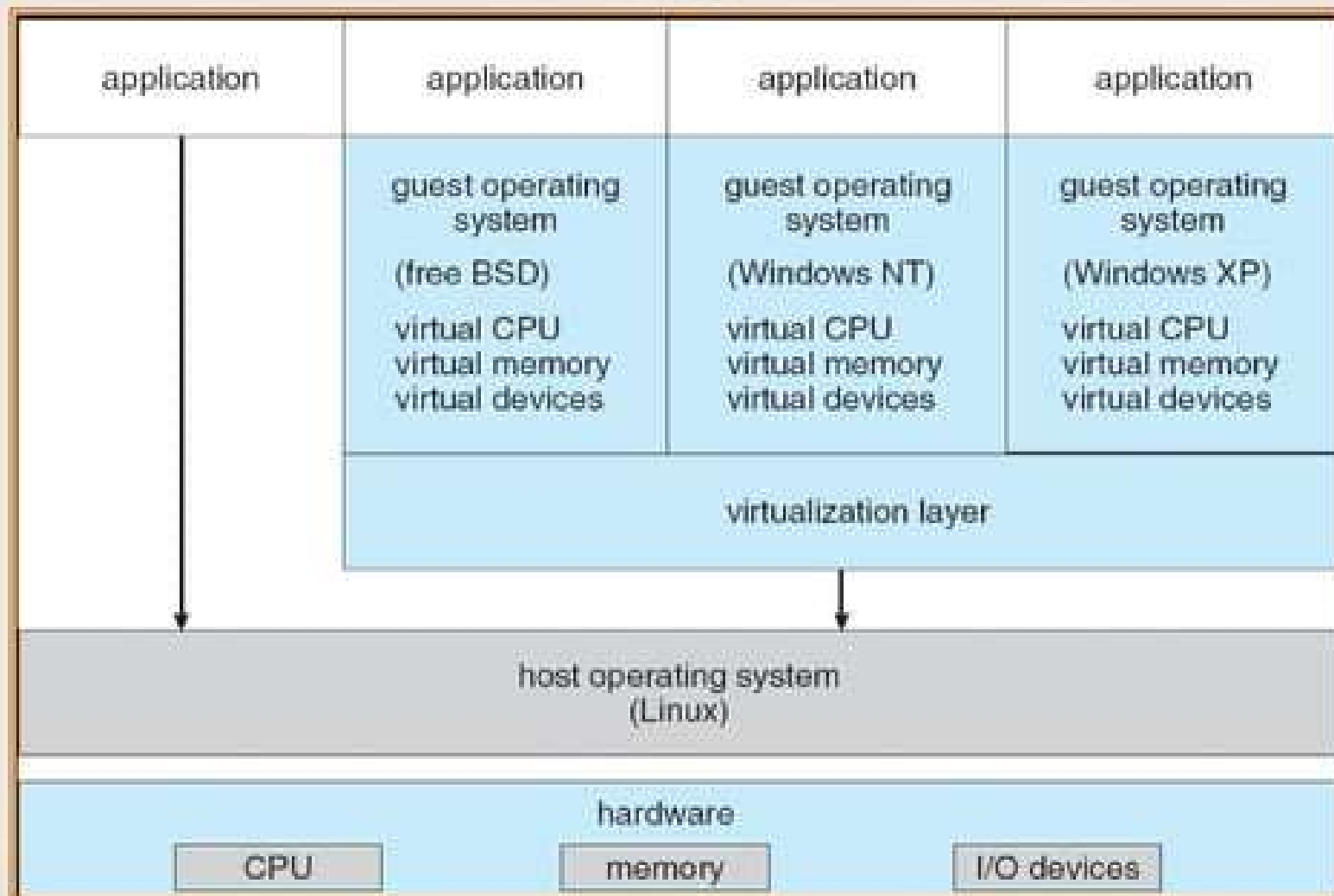- ☞ But now with VM only 60 Million server is doing the same work of 800 Million server.



Wasted Resources

# System Models



(a) Nonvirtual machine     (b) Virtual machine

# VMware Architecture

# Definitions

- *Host Operating System:*
  - The operating system actually running on the hardware
  - Together with *virtualization layer*, it simulates environment for …
- *Guest Operating System:*
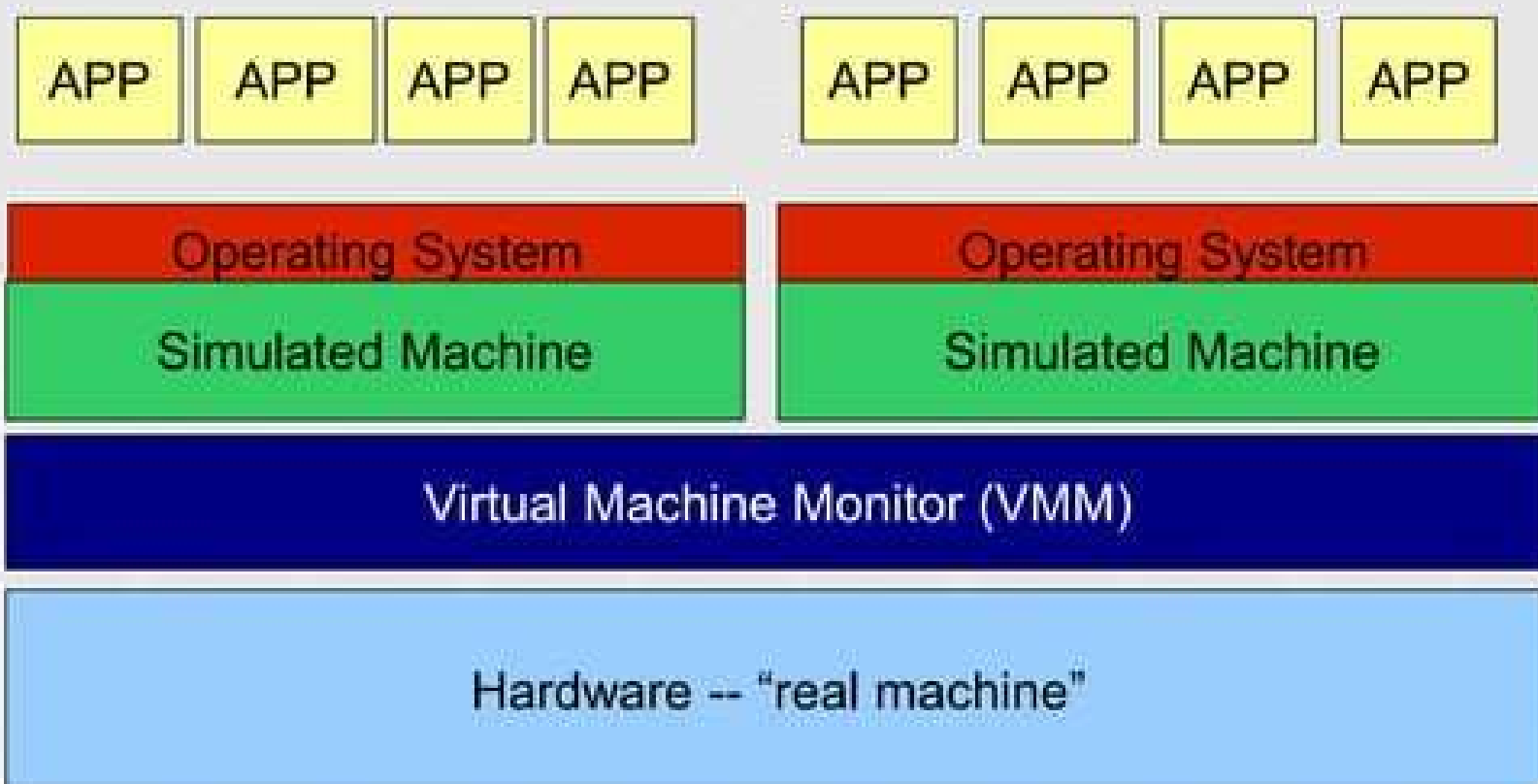  - The operating system running in the simulated environment
  - E.g., the one we are trying to debug

# Types of Virtual Machine

❧ System virtual machine:

❧ *system virtual machines* (also known as full virtualization VMs) provide a complete substitute for the targeted real machine and a level of functionality required for the execution of a complete operating system.

# System Virtual Machine

APP APP APP APP    APP APP APP APP

| Operating System | Operating System |
| Simulated Machine | Simulated Machine |

Virtual Machine Monitor (VMM)

Hardware -- "real machine"

# Process Virtual Machine

- *process virtual machines* are designed to execute a single computer program by providing an abstracted and platform-independent program execution environment. A process VM, sometimes called an *application virtual machine*, or *Managed Runtime Environment* (MRE).

- It is created when the process is started and destroyed when it exits.

- A process VM provides a high-level abstraction – that of a high-level programming language (compared to the low-level ISA abstraction of the system VM). Process VMs are implemented using an interpreter; performance comparable to compiled programming languages is achieved by the use of just-in-time compilation.

# Process Virtual Machines

APP

APP

APP

OS 2

OS 3

APP

Virtual Machine

Operating System 1

Hardware
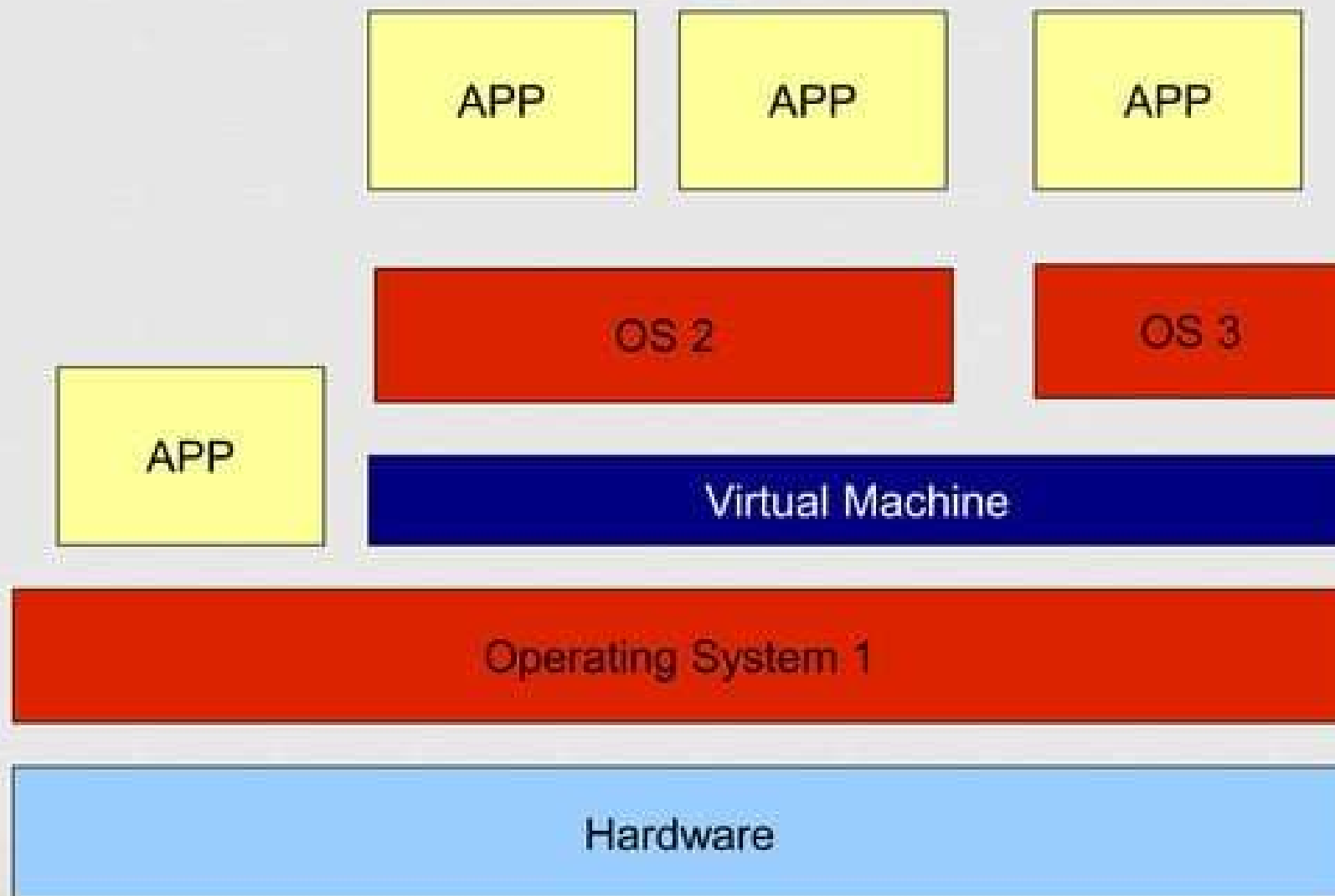
# Language Virtual Machine

❧

- ❧ This type of VM has become popular with the Java programming language, which is implemented using the Java virtual machine.

- ❧ The Parrot virtual machine, and the .NET Framework, which runs on a VM called the Common Language Runtime. All of them can serve as an abstraction layer for any computer language.

- ❧ Unlike other process VMs, these systems do not provide a specific programming language, but are embedded in an existing language; typically such a system provides bindings for several languages (e.g., C and FORTRAN).

- ❧ Examples are PVM (Parallel Virtual Machine) and MPI (Message Passing Interface). They are not strictly virtual machines, as the applications running on top still have access to all OS services, and are therefore not confined to the system model.

# The Java Virtual Machine
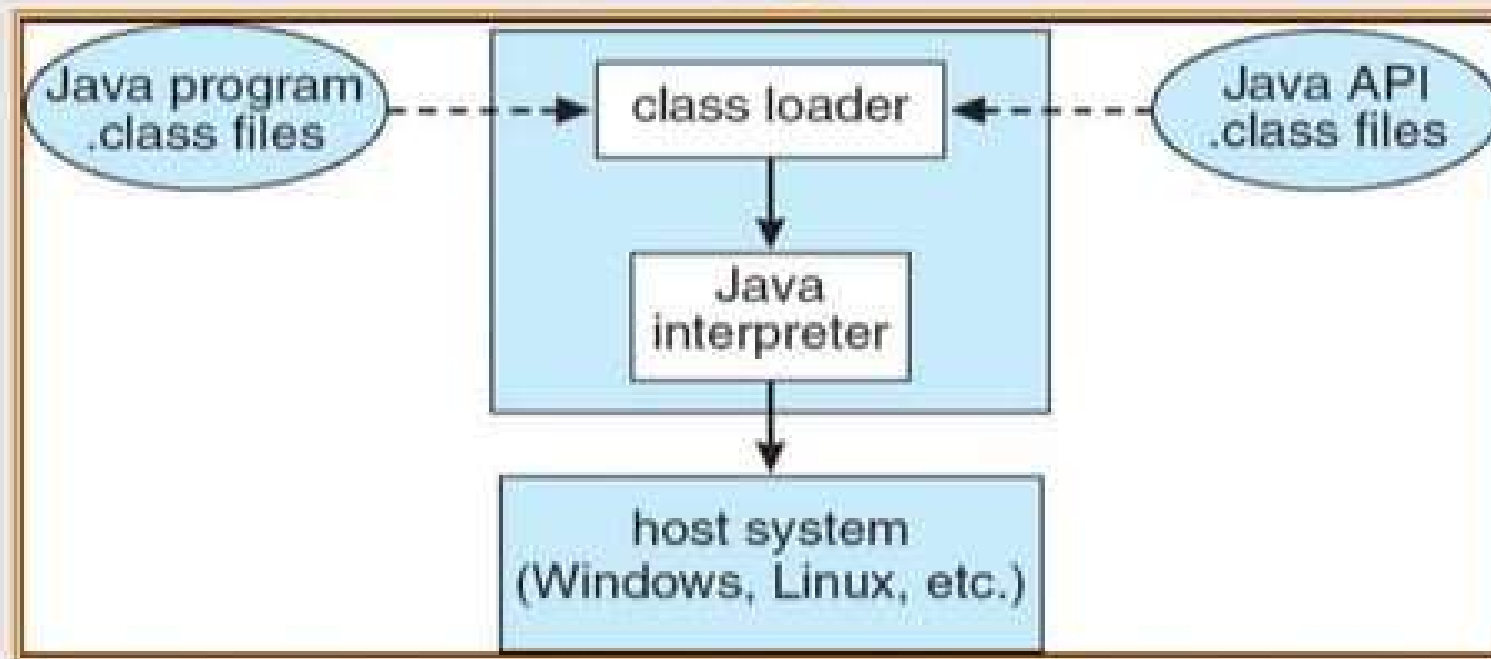


Own idealized architecture
Stylized machine language
*Byte codes*
Readily available
interpreter

# What is Hypervisor

෴

- A **hypervisor** or **virtual machine monitor (VMM)** is a piece of computer software, firmware or hardware that creates and runs **virtual machines**.

- A computer on which a hypervisor is running one or more virtual machines is defined as a *host machine*. Each virtual machine is called a *guest machine*

- Hypervisor is a bare metal approach; is installed on the bare metal and then the operating systems is installed ( para virtualized)
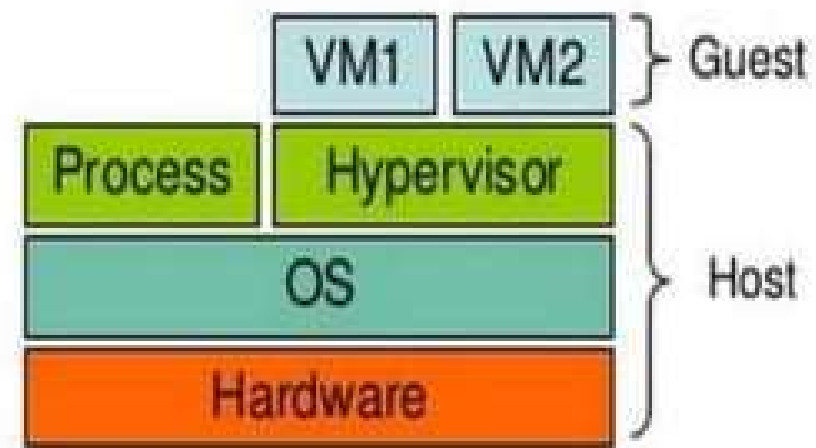
# Types of Hypervisor

- Bare-Metal

- Hosted



**Bare-Metal**

Guest { VM1 | VM2 }
Host { Hypervisor | Hardware }
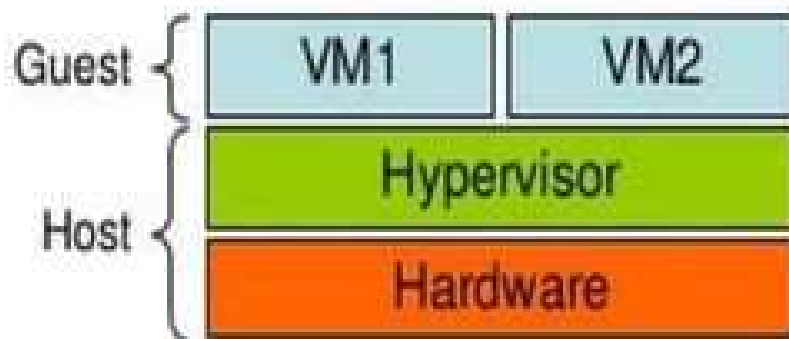
VMware ESX, Microsoft Hyper-V,
Citrix XenServer

**Hosted**

VM1 | VM2 } Guest
Process | Hypervisor
OS } Host
Hardware

VMware Workstation, Microsoft Virtual PC,
Sun VirtualBox, QEMU, KVM

# Implementation of VMMs

- Vary greatly, with options including:
  - **Type 0 hypervisors** - Hardware-based solutions that provide support for virtual machine creation and management via firmware
    - IBM LPARs and Oracle LDOMs are examples
  - **Type 1 hypervisors** - Operating-system-like software built to provide virtualization
    - Including VMware ESX, Joyent SmartOS, and Citrix XenServer
  - **Type 1 hypervisors** – Also includes general-purpose operating systems that provide standard functions as well as VMM functions
    - Including Microsoft Windows Server with HyperV and RedHat Linux with KVM
  - **Type 2 hypervisors** - Applications that run on standard operating systems but provide VMM features to guest operating systems
    - Includeing VMware Workstation and Fusion, Parallels Desktop, and Oracle VirtualBox

# Implementation of VMMs (cont.)

ભ

- ભ Other variations include:
- ભ **Para virtualization** - Technique in which the guest operating system is modified to work in cooperation with the VMM to optimize performance
- ભ **Programming-environment virtualization** - VMMs do not virtualize real hardware but instead create an optimized virtual system
  - ભ Used by Oracle Java and Microsoft.Net
- ભ **Emulators** – Allow applications written for one hardware environment to run on a very different hardware environment, such as a different type of CPU
- ભ **Application containment** - Not virtualization at all but rather provides virtualization-like features by segregating applications from the operating system, making them more secure, manageable
  - ભ Including Oracle Solaris Zones, BSD Jails, and IBM AIX WPARs
- ભ Much variation due to breadth, depth and importance of virtualization in modern computing

# What is Virtualization

Virtualization is defined as the abstraction of objects ("things"), creating a virtual (rather than actual) version of objects such as a server or storage device.

For example, when you partition a hard drive into two partitions - C and D, say - you create virtual drives but the physical hard drive has not changed.

OR

Virtualization is a technology to run multiple same or different operating systems which is completely isolated from each other . "Ex: Run both Windows and Linux on the same machine.

# Types of Virtualization

- Server Virtualization
- Desktop Virtualization
- Application Virtualization
- Memory
- Storage
- Data
- Network

# Server Virtualization

- *Hardware virtualization* or *platform virtualization* refers to the creation of a virtual machine that acts like a real computer with an operating system. Software executed on these virtual machines is separated from the underlying hardware resources.
- Different types of hardware virtualization include:
- Full virtualization – almost complete simulation of the actual hardware to allow software, which typically consists of a guest operating system, to run unmodified.
- Partial virtualization – some but not all of the target environment is simulated. Some guest programs, therefore, may need modifications to run in this virtual environment.
- Para virtualization – a hardware environment is not simulated; however, the guest programs are executed in their own isolated domains, as if they are running on a separate system. Guest programs need to be specifically modified to run in this environment.
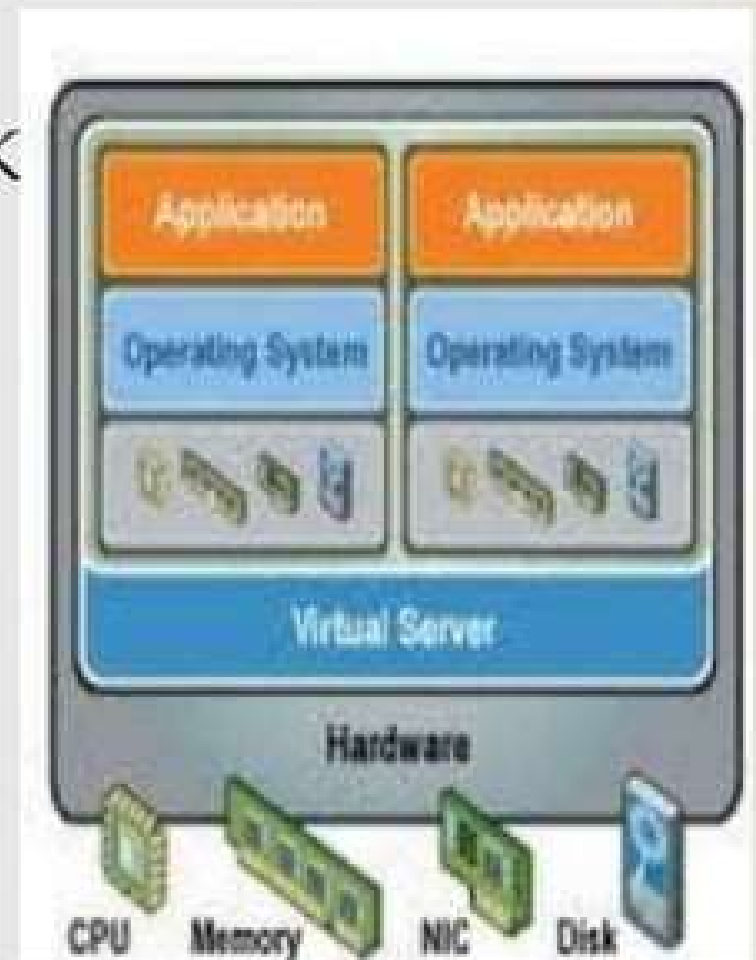
# Normal Server Issue

ଔ Backup / Restore
ଔ Re-build OS
ଔ Data recovery
ଔ Disaster Recovery
ଔ Same Vendor
ଔ Same Model

# Solution

- Consolidation
- Shared CPU, Memory, NIC ,DISK
- Maximum utilization – 90-95%
- Centralized Management
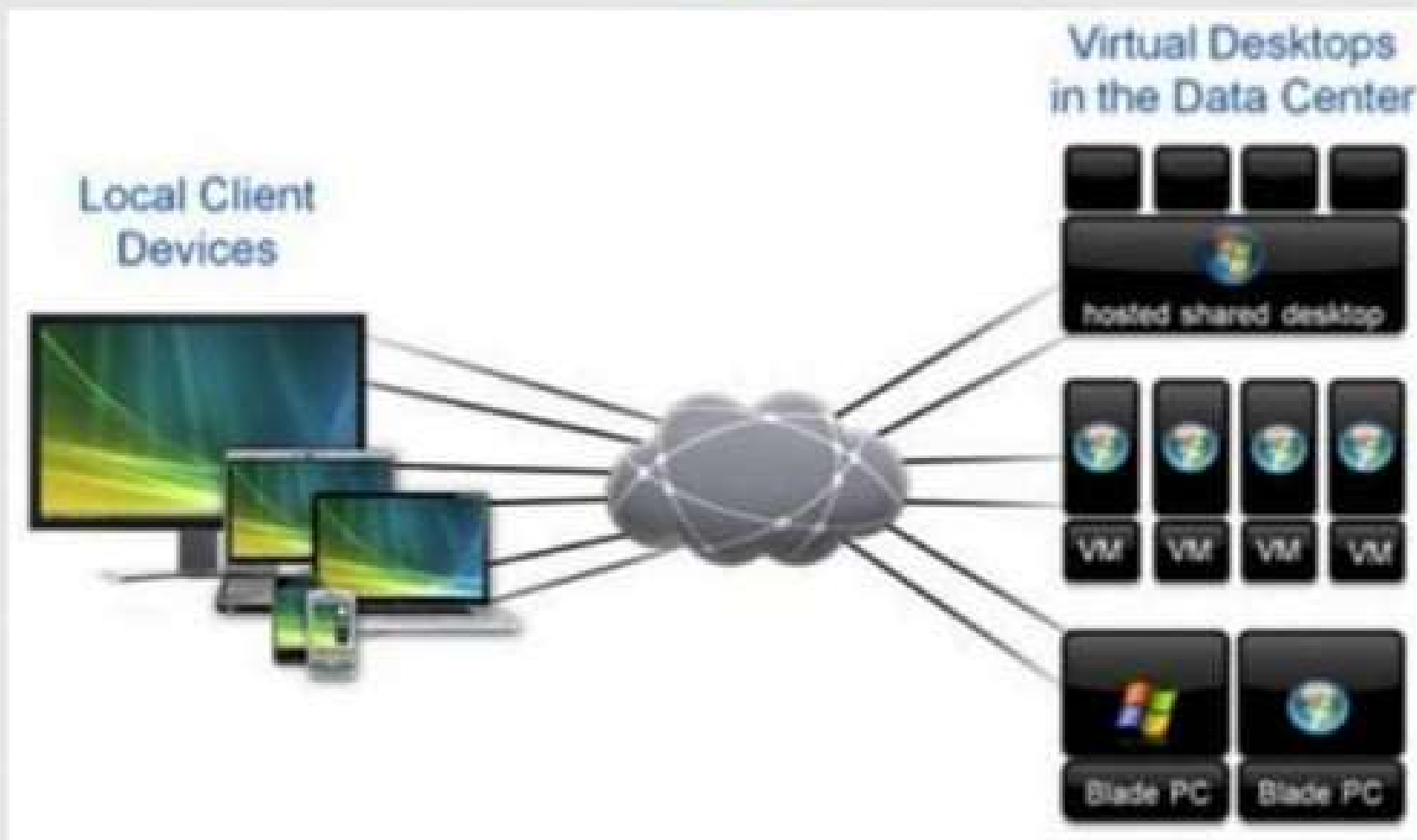- Migration
- Less space

# Server Virtualization- Benefits

- Host servers connected over SAN
- Hardware fail tolerance
- High availability
- Online live migration
- Easy management
- Disaster Recovery
- Easily scalable, Adjustable
- Different Vendor, Type, Model servers controlled by Hypervisor
- Less hardware management / maintenance cost
- Better efficiency with less cost
- Maintain average utilization %

# Desktop Virtualization

ß

ﾒ Same as server Virtualization

ﾒ User's desktop runs on data servers as Virtual machines

ﾒ Secured by design

ﾒ Multiple OSes runs on same Hypervisor host

ﾒ Centralized management

ﾒ Shared resources with maximum utilization

ﾒ Supports multimedia / memory intensive apps on VM

# Desktop Pool

Virtual Desktops in the Data Center

Local Client Devices

hosted shared desktop

VM   VM   VM   VM

Blade PC   Blade PC

# Desktop Virtualization - Benefits

- Centralized Service pack upgrade / OS patch installation
- OS upgrade from one version to another
- High availability
- Fault tolerance
- Thin / Thick provisioned storage
- Accessed via LAN, WAN, Internet & Mobile
- Offline use with help of VM Player / XenClient on Desktop, Laptop and ThinClients.

# Application Virtualization

- Business applications runs on servers located in datacenter
- Secured by design
- One / more apps runs on same set of servers
- Centralized application management
- Shared resources with maximum utilization
- Supports multimedia / memory intensive apps
- Reduces licensing cost
- Role based user access controlled by policies

# How it Work?
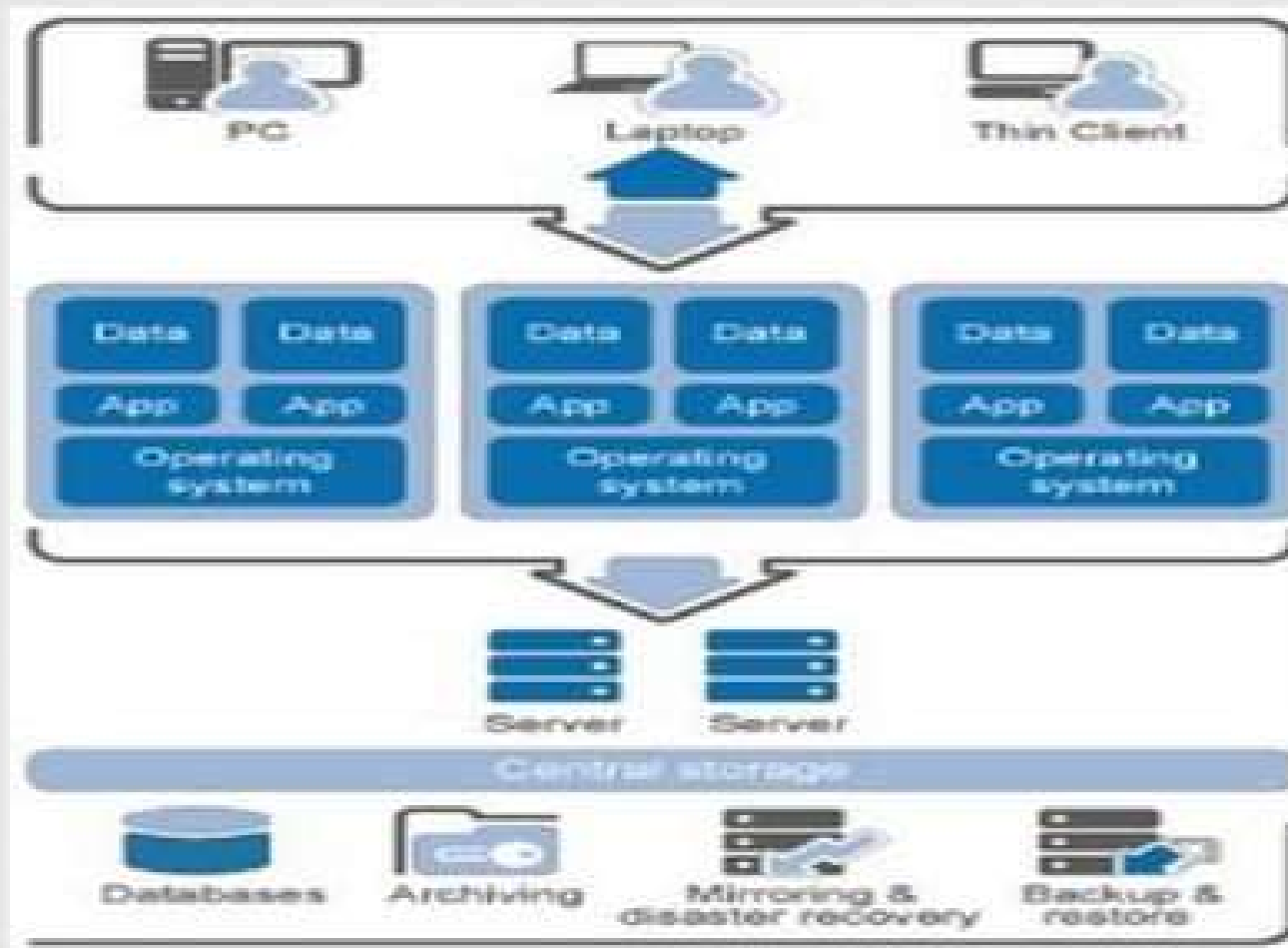
# Application Virtualization - Benefits

- Applications delivered on-demand
- No need to install and manage on every user desktop
- Increased performance
- Easily upgraded and deliver instantly
- Published either locally installed or steamed to server

# Application Virtualization - Benefits

❧

- ✧ Offline access via streamed on client machines with help of offline plug-in
- ✧ Role based application functionality
- ✧ Multiple version of same application accessibility
- ✧ Application isolation
- ✧ Accessed via LAN, WAN, Internet & Mobile
- ✧ Content Redirection
- ✧ Server to client
- ✧ Client to server

# Top 10 Virtualization Technology Companies

- VMware
- Citrix
- Oracle
- Microsoft
- Red hat
- Amazon
- Google
- Virtual bridges
- Proxmox
- Parallels

# Summary

ଓ Virtual machines are a number of discrete identical execution environments on a single computer, each of which runs an operating system. This can allow applications written for one OS to be executed on a machine which runs a different OS which provide a greater level of isolation between processes than is achieved when running multiple processes on the same instance of an OS.