Virtualization

Module 5
Open Source Software Development

Definition(s)

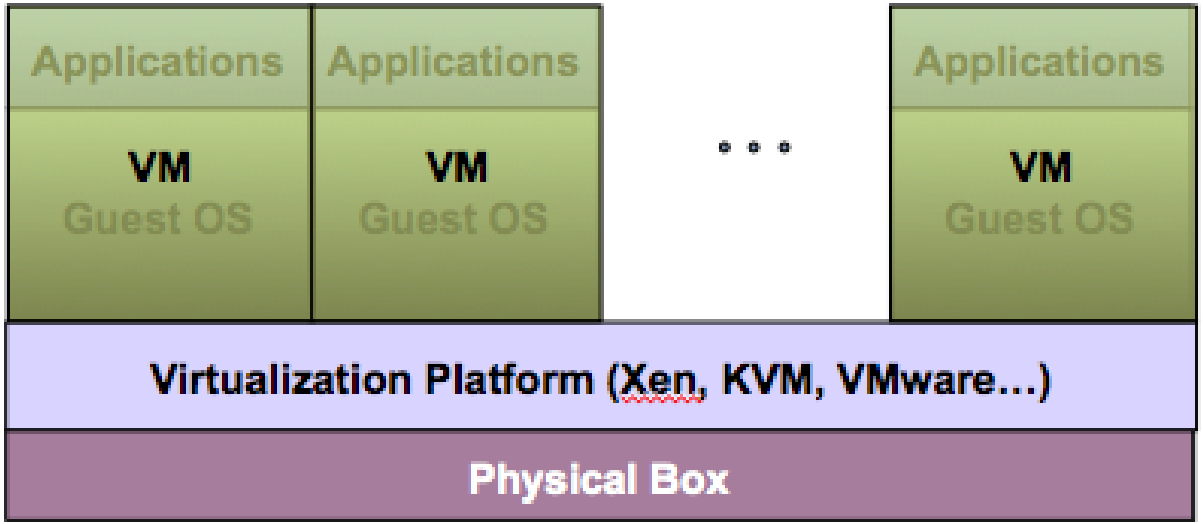
- Virtualization is the ability to run multiple operating systems on a single physical system and share the underlying hardware resources. *[VMWare White Paper]*
- Virtualization is a process that allows for more efficient utilization of physical computer hardware and is the foundation of cloud computing. *[IBM]*
- Virtualization is the process of running a virtual instance of a computer system in a layer abstracted from the actual hardware.*[opensource.com]*
- In computing, virtualization is the act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, storage devices, and computer network resources. *[wikipedia]*

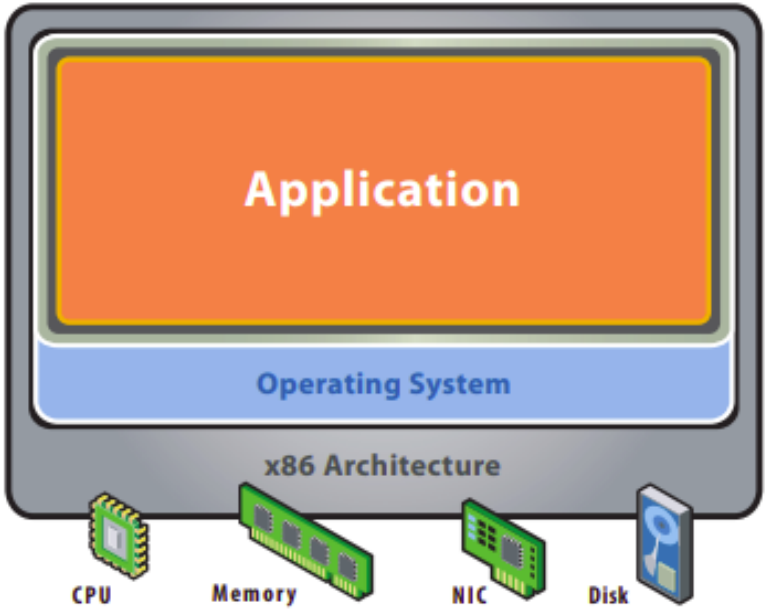
Introduction

- Virtualization uses software to create an abstraction layer over computer hardware that allows the hardware elements of a single computer — processors, memory, storage and more—to be divided into multiple virtual computers, commonly called virtual machines (VMs)
- Each VM runs its own operating system (OS) and behaves like an independent computer, even though it is running on just a portion of the actual underlying computer hardware.
- Virtualization is technology that lets you create useful IT services using resources that are traditionally bound to hardware. It allows you to use a physical machine’s full capacity by distributing its capabilities among many users or environments.

Virtualization Architecture

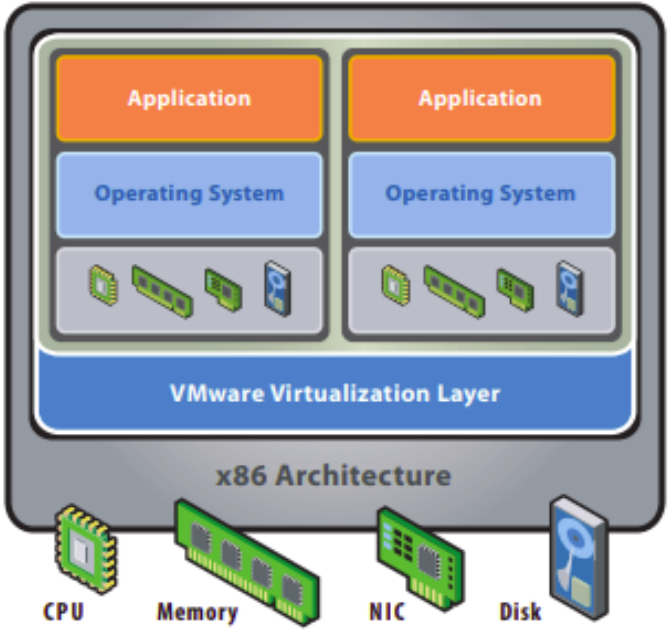
- A Virtual machine (VM) is an isolated runtime environment (guest OS and applications)
- Multiple virtual systems (VMs) can run on a single physical system





Before Virtualization:

- Single OS image per machine
- Software and hardware tightly coupled
- Running multiple applications on same machine often creates conflict
- Underutilized resources
- Inflexible and costly infrastructure



After Virtualization:

- Hardware-independence of operating system and applications
- Virtual machines can be provisioned to any system
- Can manage OS and application as a single unit by encapsulating them into virtual machines

Virtualization Benefits

- **Cost Reduction:** Sharing of resources helps cost reduction
- **Isolation:** Virtual machines are isolated from each other as if they are physically separated
- **Encapsulation:** Virtual machines encapsulate a complete computing environment
- **Hardware Independence:** Virtual machines run independently of underlying hardware
- **Portability:** Virtual machines can be migrated between different hosts.

Virtualization Benefits

- **Resource efficiency:** Virtualization lets you run several applications—each on its own VM with its own OS—on a single physical computer without sacrificing reliability.
- **Easier management:** Replacing physical computers with VMs makes it easier to use and manage policies written in software. Automated deployment and configuration tools enable administrators to define collections of virtual machines and applications as services, in software templates.
- **Minimal downtime:** One can run multiple redundant virtual machines alongside each other and failover between them when problems arise.
- **Faster provisioning:** Provided that the hardware is already in place, provisioning virtual machines to run all your applications is significantly faster. You can even automate it using management software and build it into existing workflows.

Types of Virtualization

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

Application virtualization or app virtualization

- Application virtualization or app virtualization is technology that allows users to access and use an application from a separate computer than the one on which the application is installed.
- Using application virtualization software, IT admin can set up remote applications on a server then deliver the apps to an end user’s computer.
- For the user, the experience of the virtualized app is the same as using the installed app on a physical machine.

https://www.citrix.com/en-in/glossary/what-is-application-virtualization.html

Application virtualization or app virtualization…

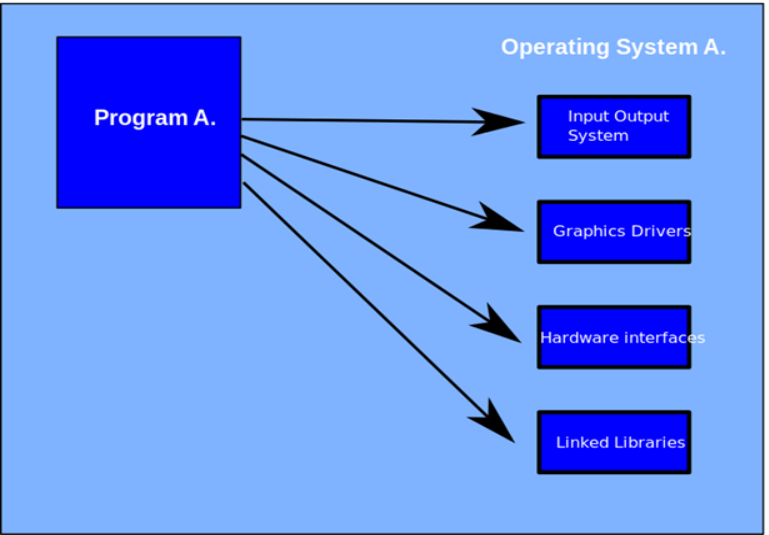
- How does application virtualization work?
  - The most common way to virtualize applications is the server-based approach.
  - IT administrator implements remote applications on a server inside an organization's data center or via a hosting service, then uses application virtualization software to deliver the applications to a user's desktop or other connected device.
  - The user can now access and use the application as though it were locally installed on their machine, and the user's actions are conveyed back to the server to be executed.
- Application virtualization is an important part of digital workspaces and desktop virtualization.

https://www.citrix.com/en-in/glossary/what-is-application-virtualization.html

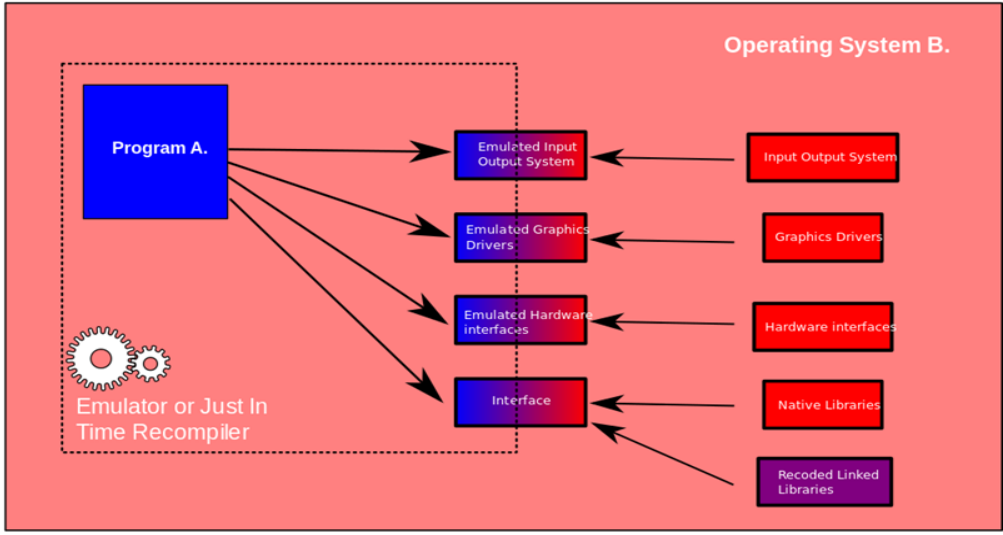
Application Virtualization…

Application virtualization helps a user to have remote access of an application from a server. The server stores all personal information and other characteristics of the application but can still run on a local workstation through the internet.

1. Application in Native Environment

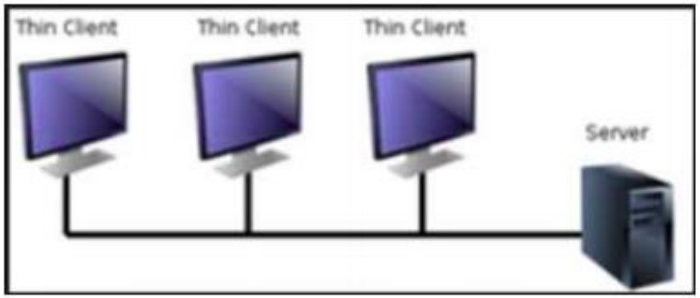


2. Application in Non-Native Environment



Desktop Virtualization

- Desktop virtualization creates a software-based (or virtual) version of an end user's desktop environment and operating system (OS) that is decoupled from the end user's computing device or client.
- This enables the user to access his or her desktop from any computing device.
- Main benefits of desktop virtualization are user mobility, portability, easy management of software installation, updates and patches.



Desktop Virtualization…

- There are three typical deployment models for desktop virtualization:
  - Virtual desktop infrastructure (VDI)
  - Remote desktop services (RDS)
  - Desktop-as-a-Service (DaaS)

https://www.ibm.com/cloud/learn/desktop-virtualization

Virtual desktop infrastructure (VDI)

- In virtual desktop infrastructure (VDI), the operating system runs on a virtual machine (VM) hosted on a server in a data center. The desktop image travels over the network to the end user's device, where the end user can interact with the desktop (and the underlying applications and operating system) as if they were local.
- VDI gives each user his or her own dedicated VM running its own operating system. The operating system resources—drivers, CPUs, memory, etc.—operate from a software layer called a hypervisor that mimics their output, manages the resource allocation to multiple VMs, and allows them to run side by side on the same server.
- A key benefit of VDI is that it can deliver the Windows 10 desktop and operating system to the end user's devices. However, because VDI supports only one user per Windows 10 instance, it requires a separate VM for each Windows 10 user.

Remote desktop services (RDS)

- In remote desktop services (RDS)—also known as Remote Desktop Session Host (RDSH)—users remotely access desktops and Windows applications through the Microsoft Windows Server operating system. Applications and desktop images are served via Microsoft Remote Desktop Protocol (RDP). Formerly known as Microsoft Terminal Server, this product has remained largely unchanged since its initial release.
- From the end user's perspective, RDS and VDI are identical. But because one instance of Windows Server can support as many simultaneous users as the server hardware can handle, RDS can be a more cost-effective desktop virtualization option. It's also worth noting applications tested or certified to run on Windows 10 may not be tested or certified to run on the Windows Server OS.

Desktop-as-a-Service (DaaS)

- In desktop as a service (DaaS), VMs are hosted on a cloud-based backend by a third-party provider. DaaS is readily scalable, can be more flexible than on-premise solutions, and generally deploys faster than many other desktop virtualization options.
- Like other types of cloud desktop virtualization, DaaS shares many of the general benefits of cloud computing, including support for fluctuating workloads and changing storage demands, usage-based pricing, and the ability to make applications and data accessible from almost any internet-connected device. The chief drawback to DaaS is that features and configurations are not always as customizable as required.

Server Virtualization

- It is virtualizing your server infrastructure where you do not have to use any more physical servers for different purposes.
- The physical server is divided into multiple different virtual servers by changing the identity number, processors. So, each system can operate its own operating systems in isolate manner.
- Each sub-server knows the identity of the central server. It causes an increase in the performance and reduces the operating cost by the deployment of main server resources into a sub-server resource.
- It’s beneficial in virtual migration, reduce energy consumption, reduce infrastructural cost etc.

Storage Virtualization

- Storage virtualization is an array of servers that are managed by a virtual storage system. The servers aren’t aware of exactly where their data is stored, and instead function more like worker bees in a hive.
- It makes managing storage from multiple sources to be managed and utilized as a single repository.
- Storage virtualization software maintains smooth operations, consistent performance and a continuous suite of advanced functions despite changes, break down and differences in the underlying equipment.

Network Virtualization

- The ability to run multiple virtual networks with each has a separate control and data plan. It co-exists together on top of one physical network. It can be managed by individual parties that potentially confidential to each other.

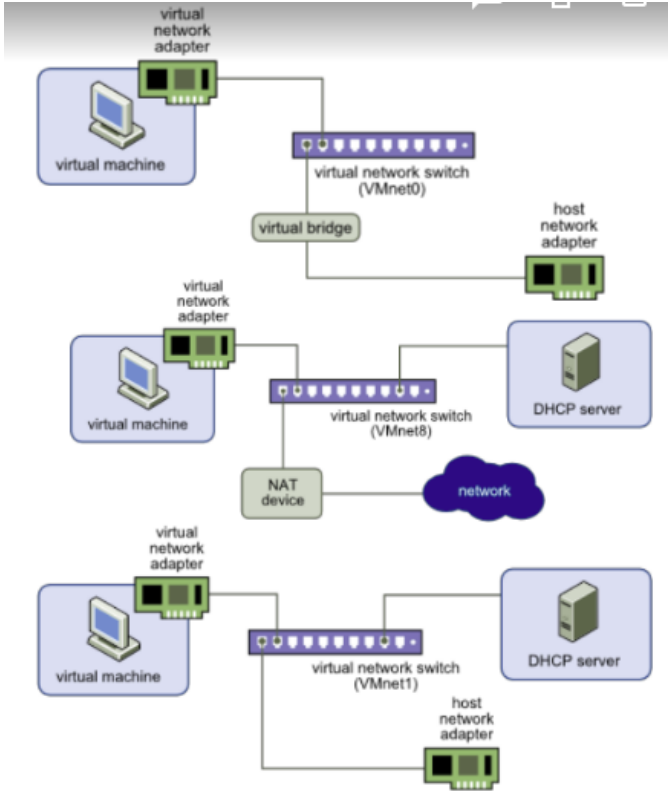
- Network virtualization provides a facility to create and provision virtual networks—logical switches, routers, firewalls, load balancer, Virtual Private Network (VPN), and workload security within days or even in weeks.

Network Virtualization

- Physical components that make up the physical network are virtualized to create a virtual network
- **What is a vSwitch?**
  - Virtual switch that virtual devices can connect to in order to communicate with each other
- **What is a vLAN?**
  - Virtual Local Area Network that is segmented into groups of ports isolated from one another, creating different network segments

Types of Virtual Networks

- **Bridged Network:** The host server and the VM are connected to the same network, and the host shares its IP address with the VM.
- **NAT:** VMs use an IP translated from the host's IP (using NAT device) and communicate on a private network set up on the host computer
- **Host-only Network:** VMs use a private network but do not have translated IP addresses to connect to external network, therefore can only communicate to other VMs on the isolated host network.



Data Virtualization

The data is collected from various sources and managed that at a single place without knowing more about the technical information like how data is collected, stored & formatted then arranged that data logically so that its virtual view can be accessed by its interested people and stakeholders, and users through the various cloud services remotely. Many big giant companies are providing their services like Oracle, IBM, At scale, Cdata etc.

It can be used to performing various kind of tasks such as:

- Data-integration
- Business-integration
- Service-oriented architecture data-services
- Searching organizational data

Hypervisor

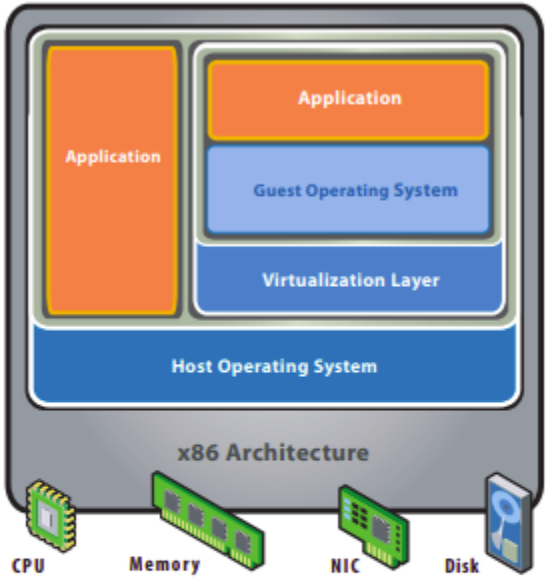
- A hypervisor, a.k.a. a virtual machine manager/monitor (VMM), or virtualization manager, is a program that allows multiple operating systems to share a single hardware host.

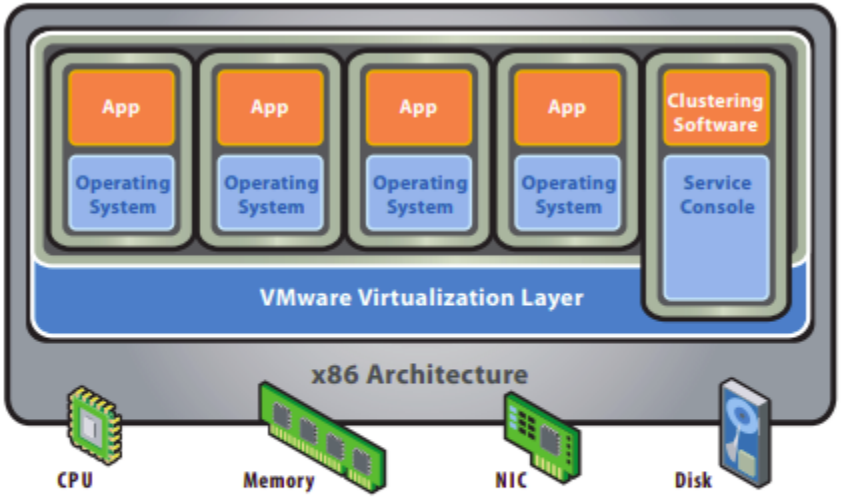
- Each guest operating system appears to have the host's processor, memory, and other resources all to itself. However, the hypervisor is actually controlling the host processor and resources, allocating what is needed to each operating system in turn and making sure that the guest operating systems (called virtual machines) cannot disrupt each other.

Hypervisor

- Hypervisor is a specialized firmware or software, or both, installed on single hardware that would allow you to host several virtual machines.
- It allows physical hardware to be shared across several virtual machines.
- A computer on which hypervisor runs one or more virtual machines is called a host machine.
- The virtual machine is called a guest machine.
- Basically, hypervisor allows physical host machine to run various guest machines.
- It helps in achieving maximum benefits from computing resources such as memory, network bandwidth, and CPU cycles.

Types of Hypervisors





Hosted Architecture

- Installs and runs as an application
- Relies on host OS for device support and physical resource management

Bare-Metal (Hypervisor) Architecture

- Lean virtualization-centric kernel
- Service Console for agents and helper applications

Type 1: Bare Metal or Native

- A bare-metal hypervisor (Type 1) is a layer of software we install directly on top of a physical server and its underlying hardware.
- There is no software or any operating system in between, hence the name *bare-metal hypervisor*. A Type 1 hypervisor is proven in providing excellent performance and stability since it does not run inside Windows or any other operating system.
- Type 1 hypervisors are an OS themselves, a very basic one on top of which you can run virtual machines. The physical machine the hypervisor is running on serves virtualization purposes only. You cannot use it for anything else.
- Type 1 hypervisors are mainly found in enterprise environments.

- Examples: VMware ESXi, Citrix XenServer and Microsoft Hyper-V hypervisor.

Type 2: Hosted or Embedded

- This type of hypervisor runs inside of an operating system of a physical host machine.
- This is why we call type 2 hypervisors – *hosted hypervisors*. As opposed to type 1 hypervisors that run directly on the hardware, hosted hypervisors have one software layer underneath. In this case we have:
  - A physical machine.
  - An operating system installed on the hardware (Windows, Linux, macOS).
  - A type 2 hypervisor software within that operating system.
  - The actual instances of guest virtual machines.

Examples: VMware Player, Parallels Desktop, Oracle Virtual Box etc.

Virtualization Approaches

- **Emulation**
  - Fully emulate the underlying architecture
- **Para virtualization**
  - Abstract the base architecture
- **Full virtualization**
  - Simulate the base hardware architecture
- **OS-level virtualization**
  - Shared kernel (and architecture)
  - Separate user spaces

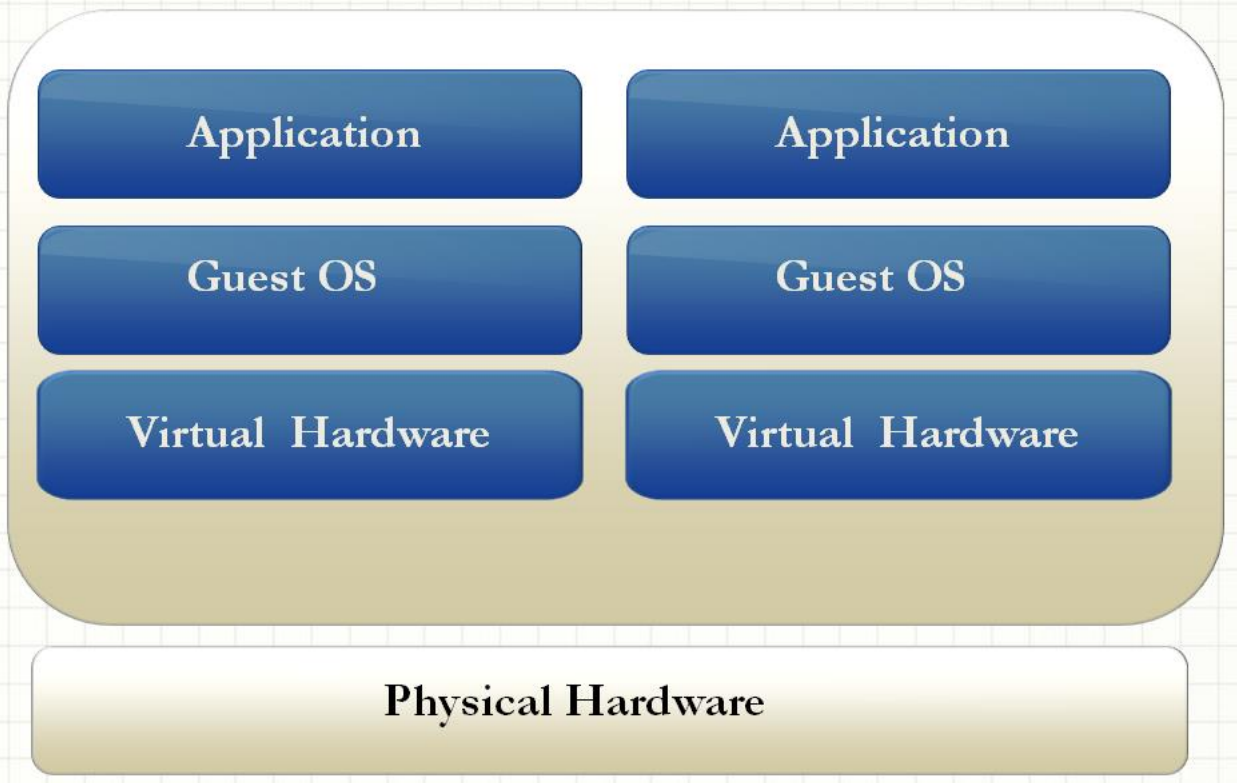
Emulation

- **Emulation**
  - Behavior of the computer hardware is copied to a software program
  - Emulation layer lies on the OS which lie on the Hardware
- **2 open source emulators**
  - QEMU
  - BOCHS
- **Advantage**
  - OS of another architecture can run on our own architecture
- **Disadvantage**
  - Virtualizing a complete CPU comes with heavy performance price

Full virtualization

- Virtual machine talks to VMM which communicates with the hardware platform
- CPU understands the unmodified instructions generated by Virtualized OS
- **Advantages**
  - Complete decoupling of the software from the hardware
  - Streamline the migration of applications
  - Complete isolation of different applications
- **Disadvantages**
  - Performance penalty
  - VMM should provide additionally virtual bios, virtual memory space and virtual devices
  - VMM creates and maintains data structures like shadow memory page table

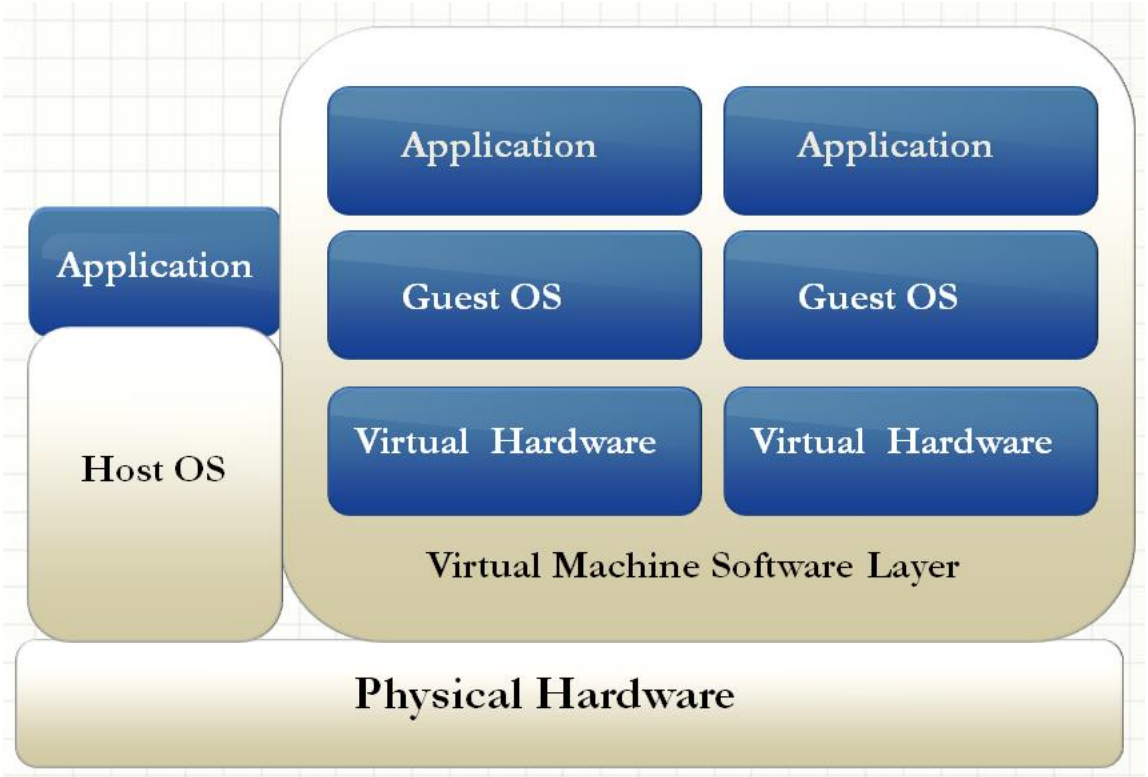
Full virtualization…



Para virtualization

- Guest OS uses specialized API that talks to the VMM which sends the virtualization requests to the hardware
- VMM does not need a resource intensive translation of instructions
- **Advantages**
  - near native performance
  - disaster recovery
  - Migration
- **Disadvantages**
  - several insecurities like guest OS cache data, unauthenticated connections
  - not applicable for Windows OS

Para virtualization…



File Format for Virtual Machines

1. **Open Virtualization Format (OVF):**
    - The OVF Specification provides a means of describing the properties of a virtual system.
    - It is XML based and has generous allowances for extensibility (with corresponding tradeoffs in actual portability).
    - Most commonly, an OVF file is used to describe a single virtual machine or virtual appliance.
    - It can contain information about the format of a virtual disk image file as well as a description of the virtual hardware that should be emulated to run the OS or application contained on such a disk image.
2. **Open Virtual Appliance (OVA):** An OVA is an OVF file packaged together with all of its supporting files (disk images, etc.).

Format for Disk Images

- **VDI** – VirtualBox's internal default disk image format is VDI. Nevertheless, this is not what is used by Vagrant boxes.
- **VMDK** – One of the most common formats. VMWare's products use various versions and variations of VMDK disk images. Several versions and variations exist, so it's very important to understand which one you're working with and where it can be used.
- **VHD** – Commonly used by Microsoft (e.g. for Microsoft Virtual PC).
- **raw (.img, .raw, etc.)** – Without compression or thin provisioning, disk images can be very large, but sometimes converting to raw disk images might make sense as an intermediate step or in certain scenarios for better performance at the cost of space.

Snapshot

- Working on a VM and need to save progress or state
- Snapshots are saved as files in the VM folder
- What is saved by a snapshot?
  - State of VM disks
  - Contents of VM memory
  - VM settings

What Files Make Up a Virtual Machine?

- A virtual machine typically is stored on the host computer in a set of files, usually in a directory created by Workstation for that specific virtual machine.
- The key files are listed here by extension. In these examples, <vmname> is the name of your virtual machine

Extension	File Name	Description
.log	<vmname>.log or vmware.log	This is the file that keeps a log of key VMware Workstation activity. This file can be useful in troubleshooting if you encounter problems. This file is stored in the directory that holds the configuration (.vmx) file of the virtual machine.
.nvram	<vmname>.nvram or nvram	This is the file that stores the state of the virtual machine's BIOS.

.vmdk	<vmname>.vmdk	This is a virtual disk file, which stores the contents of the virtual machine's hard disk drive. A virtual disk is made up of one or more .vmdk files. If you have specified that the virtual disk should be split into 2GB chunks, the number of .vmdk files depends on the size of the virtual disk. As data is added to a virtual disk, the .vmdk files grow in size, to a maximum of 2GB each. (If you specify that all space should be allocated when you create the disk, these files start at the maximum size and do not grow.) Almost all of a .vmdk file's content is the virtual machine's data, with a small portion allotted to virtual machine overhead. If the virtual machine is connected directly to a physical disk, rather than to a virtual disk, the .vmdk file stores information about the partitions the virtual machine is allowed to access. Earlier VMware products used the extension .dsk for virtual disk files.

<diskname>-<###>.vmdk	This is a redo-log file, created automatically when a virtual machine has one or more snapshots. This file stores changes made to a virtual disk while the virtual machine is running. There may be more than one such file. The ### indicates a unique suffix added automatically by VMware Workstation to avoid duplicate file names.

.vmem	<uuid>.vmem	The virtual machine's paging file, which backs up the guest main memory on the host file system. This file exists only when the virtual machine is running, or if the virtual machine has crashed.
	<snapshot_name_and_number>	Each snapshot of a virtual machine that is powered on has an associated `.vmem` file, which contains the guest's main memory, saved as part of the snapshot.
`.vmsd`	<vmname>.vmsd	This is a centralized file for storing information and metadata about snapshots.
`.vmsn`	<vmname>-Snapshot.vmsn	This is the snapshot state file, which stores the running state of a virtual machine at the time you take that snapshot
	<vmname>-Snapshot<###>.vmsn	This is the file which stores the state of a snapshot
`.vmss`	<vmname>.vmss	This is the suspended state file, which stores the state of a suspended virtual machine

`.vmss`	<vmname>.vmss	This is the suspended state file, which stores the state of a suspended virtual machine .Some earlier VMware products used the extension `.std` for suspended state files
`.vmtm`	<vmname>.vmtm	This is the configuration file containing team data.
`.vmx`	<vmname>.vmx	This is the primary configuration file, which stores settings chosen in the New Virtual Machine Wizard or virtual machine settings editor. If you created the virtual machine under an earlier version of VMware Workstation on a Linux host, this file may have a .cfg extension
`.vmxf`	<vmname>.vmxf	This is a supplemental configuration file for virtual machines that are in a team. Note that the `.vmxf` file remains if a virtual machine is removed from the team

Virtualization in Cloud Computing

Cloud computing takes virtualization one step further:

- You don’t need to own the hardware
- Resources are rented as needed from a cloud
- Various providers allow creating virtual servers:
  - Choose the OS and software each instance will have
  - The chosen OS will run on a large server farm
  - Can instantiate more virtual servers or shut down existing ones within minutes
- You get billed only for what you used