VIRTUALIZATION OF CPU, MEMORY, AND I/O DEVICES

To support virtualization, processors such as the x86 employ a special running mode and instructions, known as hardware-assisted virtualization. In this way, the VMM and guest OS run in different modes and all sensitive instructions of the guest OS and its applications are trapped in the VMM. To save processor states, mode switching is completed by hardware. For the x86 architecture, Intel and AMD have proprietary technologies for hardware-assisted virtualization.

1. Hardware Support for Virtualization

One or more guest OS can run on top of the hypervisor. KVM (Kernel-based Virtual Machine) is a Linux kernel virtualization infrastructure. KVM can support hardware-assisted virtualization and Para virtualization by using the Intel VT-x or AMD-v and VirtIO framework, respectively. The VirtIO framework includes a paravirtual Ethernet card, a disk I/O controller, a balloon device for adjusting guest memory usage, and a VGA graphics interface using VMware drivers.

Example 3.4 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 provides an overview of Intel's full virtualization techniques. For processor virtualization, Intel offers the VT-x or VT-i technique. 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, 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.

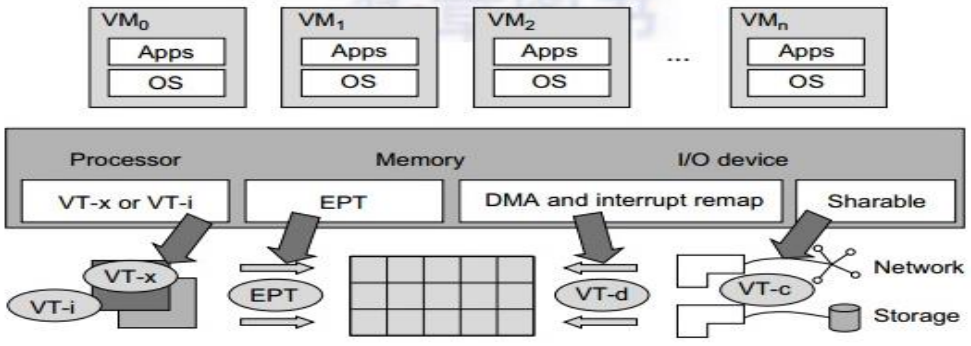


FIGURE 3.10
Intel hardware support for virtualization of processor, memory, and I/O devices.

2. 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. Other critical instructions should be handled carefully for correctness and stability.

The critical instructions are divided into three categories: privileged instructions, control-sensitive instructions, and behavior-sensitive instructions. 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 exe-cuted, 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. However, not all CPU architectures are virtualizable. RISC CPU architectures can be naturally virtualized because all control- and behavior-sensitive instructions are privileged instructions.

On the contrary, x86 CPU architectures are not primarily designed to support virtualization. This is because about 10 sensitive instructions, such as SGDT and SMSW, are not privileged instructions. When these instruc-tions execute in virtualization, they cannot be trapped in the VMM.

On a native UNIX-like system, a system call triggers the 80h interrupt and passes control to the OS kernel. The interrupt handler in the kernel is then invoked to process the system call.

On a para-virtualization system such as Xen, a system call in the guest OS first triggers the 80h interrupt nor-mally. Almost at the same time, the 82h interrupt in the hypervisor is triggered. Incidentally, control is passed on to the hypervisor as well. When the hypervisor completes its task for the guest OS system call, it passes control back to the guest OS kernel. Certainly, the guest OS kernel may also invoke the hypercall while it's running. Although paravirtualization of a CPU

lets unmodified applications run in the VM, it causes a small performance penalty.

2.1 Hardware-Assisted CPU Virtualization

This technique attempts to simplify virtualization because full or paravirtualization is complicated. Intel and AMD add an additional mode called privilege mode level (some people call it Ring-1) to x86 processors.

Therefore, operating systems can still run at Ring 0 and the hypervisor can run at Ring -1. All the privileged and sensitive instructions are trapped in the hypervisor automatically.

This technique removes the difficulty of implementing binary translation of full virtualization. It also lets the operating system run in VMs without modification.

Example 3.5 Intel Hardware-Assisted CPU Virtualization

Although x86 processors are not virtualizable primarily, great effort is taken to virtualize them. They are used widely in comparing RISC processors that the bulk of x86-based legacy systems cannot discard easily. Virtuali-zation of x86 processors is detailed in the following sections. Intel's VT-x technology is an example of hardware-assisted virtualization, as shown in Figure 3.11. Intel calls the privilege level of x86 processors the VMX Root Mode. In order to control the start and stop of a VM and allocate a memory page to maintain the

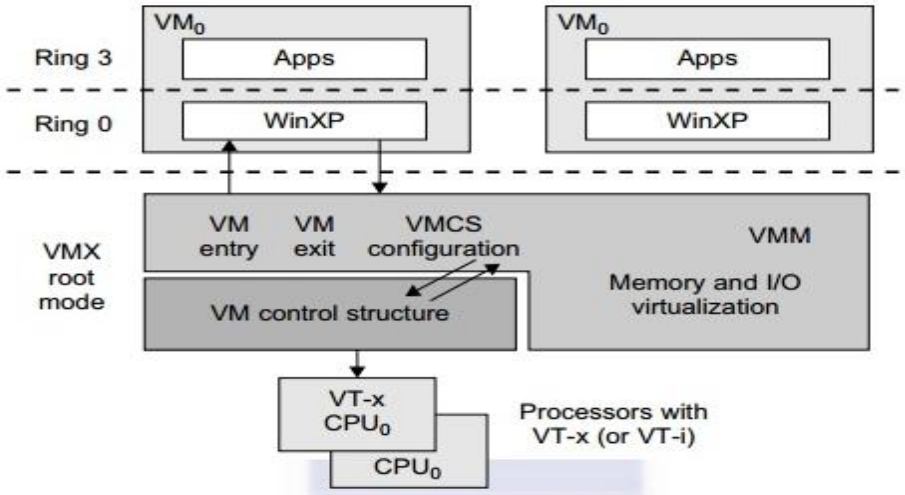


FIGURE 3.11
Intel hardware-assisted CPU virtualization.

CPU state for VMs, a set of additional instructions is added. At the time of this writing, Xen, VMware, and the Microsoft Virtual PC all implement their hypervisors by using the VT-x technology.

3. Memory Virtualization

Virtual memory virtualization is similar to the virtual memory support provided by modern operat-ing systems. In a traditional execution environment, the operating system maintains mappings of virtual memory to machine memory using page tables, 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 3.12 shows the two-level memory mapping procedure.

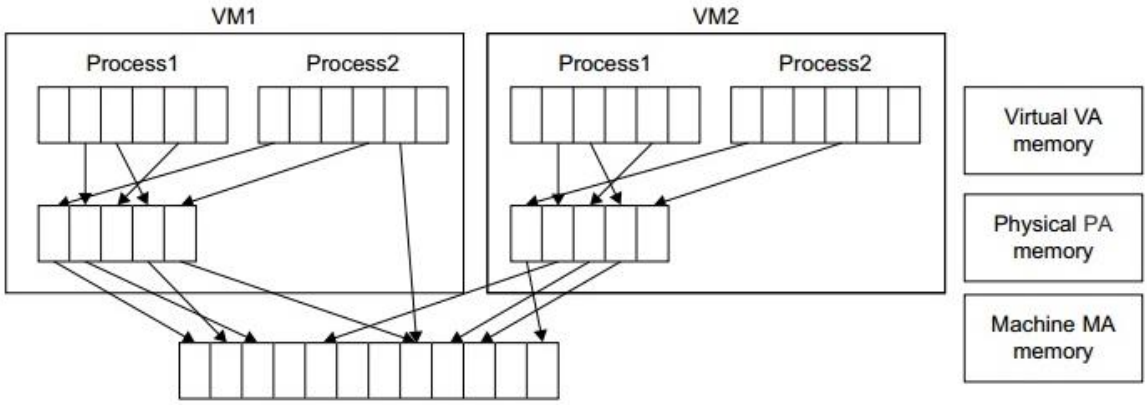


FIGURE 3.12

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. Nested page tables add another layer of indirection to virtual memory. 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 perfor-mance 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. The AMD Barcelona processor has featured hardware-assisted memory virtualization since 2007. It provides hardware assistance to the two-stage address translation in a virtual execution environment by using a technology called nested paging.

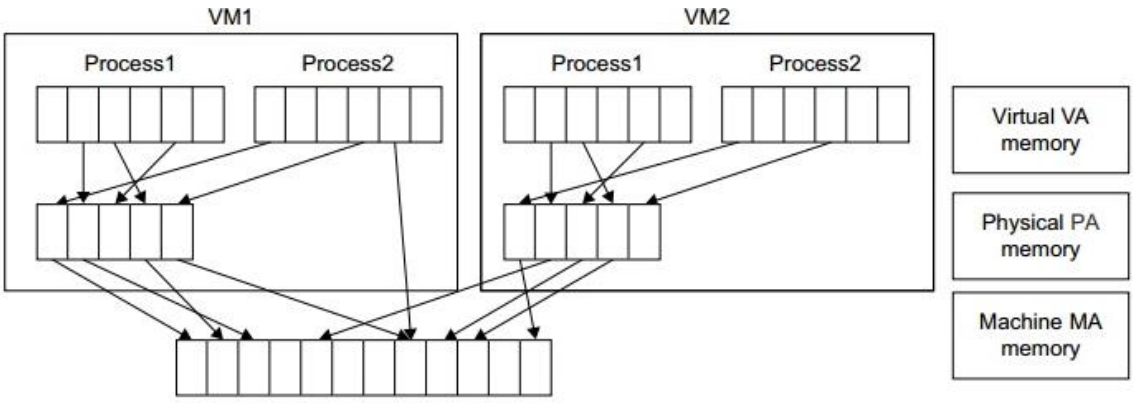


FIGURE 3.12

Two-level memory mapping procedure.

4. 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. Full device emulation is the first approach for I/O virtualization. Generally, this approach emulates well-known, real-world devices.

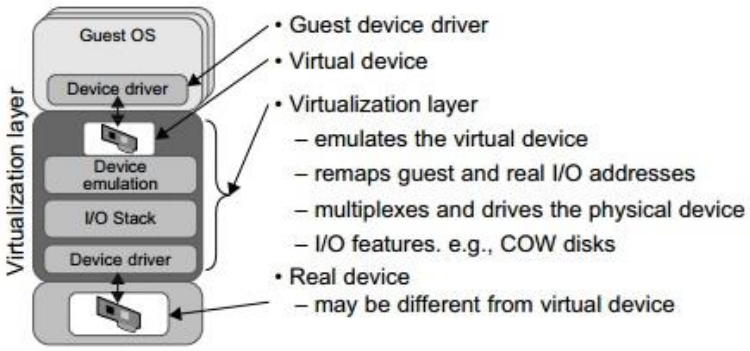


FIGURE 3.14

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.

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. The full device emulation approach is shown in Figure 3.14.

A single hardware device can be shared by multiple VMs that run concurrently. However, software emulation runs much slower than the hardware it emulates [10,15]. 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 dri-ver 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.

Example 3.7 VMware Workstation for I/O Virtualization

The VMware Workstation runs as an application. It leverages the I/O device support in guest OSes, host OSes, and VMM to implement I/O virtualization. The application portion (VMApp) uses a driver loaded into the host operating system (VMDriver) to establish the privileged VMM, which runs directly on the hardware.

A given physical processor is executed in either the host world or the VMM world, with the VMDriver facilitating the transfer of control between the two worlds. The VMware Workstation employs full device emulation to implement I/O virtualization. Figure 3.15 shows the functional blocks used in sending and receiving packets via the emulated virtual NIC.

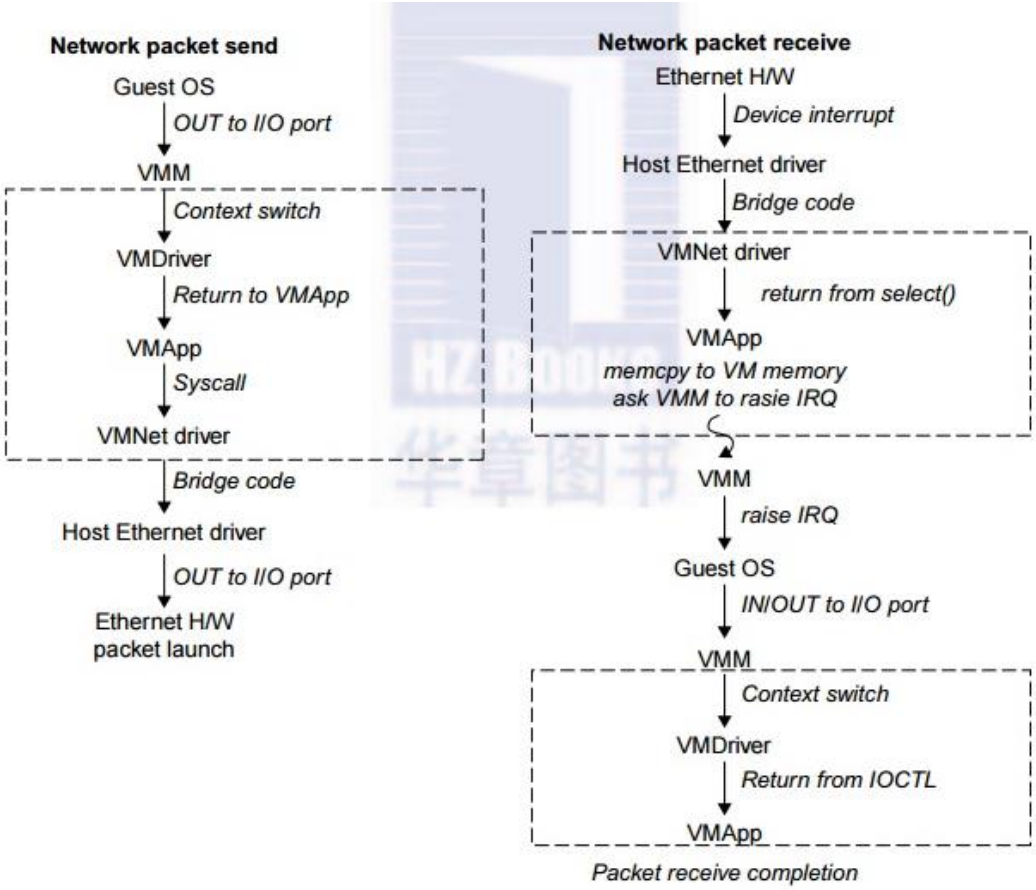


FIGURE 3.15

Functional blocks involved in sending and receiving network packets.

5. Virtualization in Multi-Core Processors

Virtualizing a multi-core processor is relatively more complicated than virtualizing a uni-core processor. Though multicore processors are claimed to have higher performance by integrating multiple processor cores in a single chip, muti-core virtualiuzation has raised some new challenges to computer architects, compiler constructors, system designers, and application programmers. There are mainly two difficulties: Application programs must be parallelized to use all cores fully, and software must explicitly assign tasks to the cores, which is a very complex problem.

Concerning the first challenge, new programming models, languages, and libraries are needed to make parallel programming easier. The second challenge has spawned research involving scheduling algorithms and resource management policies. Yet these efforts cannot balance well among performance, complexity, and other issues. What is worse, as technology scales,

a new challenge called dynamic heterogeneity is emerging to mix the fat CPU core and thin GPU cores on the same chip, which further complicates the multi-core or many-core resource management. The dynamic heterogeneity of hardware infrastructure mainly comes from less reliable transistors and increased complexity in using the transistors [33,66].

5.1 Physical versus Virtual Processor Cores

A multicore virtualization method to allow hardware designers to get an abstraction of the low-level details of the processor cores. This technique alleviates the burden and inefficiency of managing hardware resources by software. It is located under the ISA and remains unmodified by the operating system or VMM (hypervisor). Figure 3.16 illustrates the technique of a software-visible VCPU moving from one core to another and temporarily suspending execution of a VCPU when there are no appropriate cores on which it can run.

5.2 Virtual Hierarchy

The emerging many-core chip multiprocessors (CMPs) provides a new computing landscape. Instead of supporting time-sharing jobs on one or a few cores, we can use the abundant cores in a space-sharing, where single-threaded or multithreaded jobs are simultaneously assigned to separate groups of cores for long time intervals.

This idea was originally suggested by Marty and Hill [39]. To optimize for space-shared workloads, they propose using virtual hierarchies to overlay a coherence and caching hierarchy onto a physical processor. Unlike a fixed physical hierarchy, a virtual hierarchy can adapt to fit how the work is space shared for improved performance and performance isolation

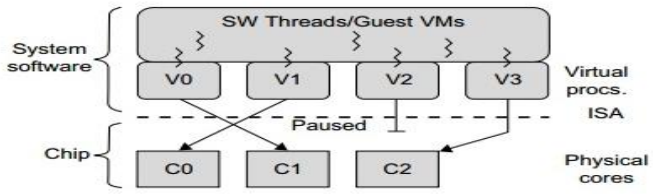
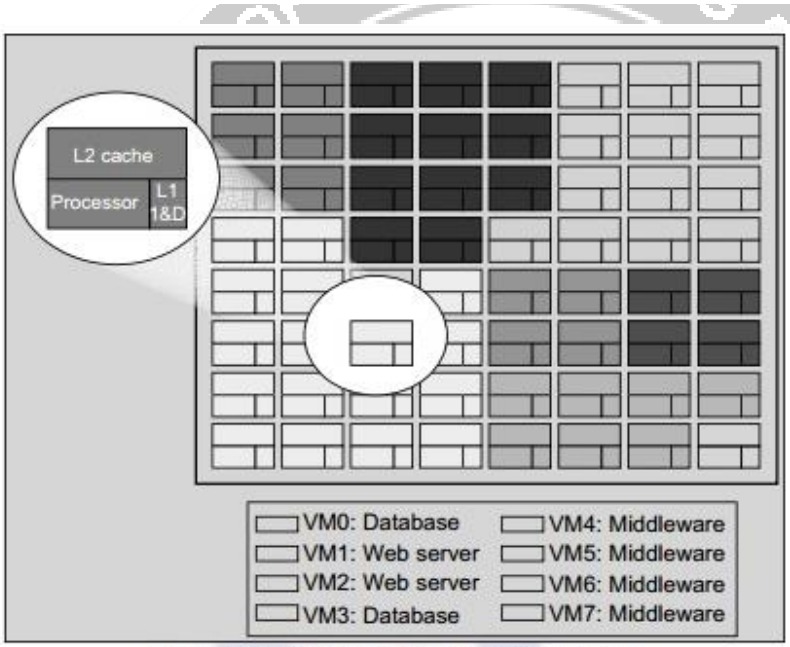
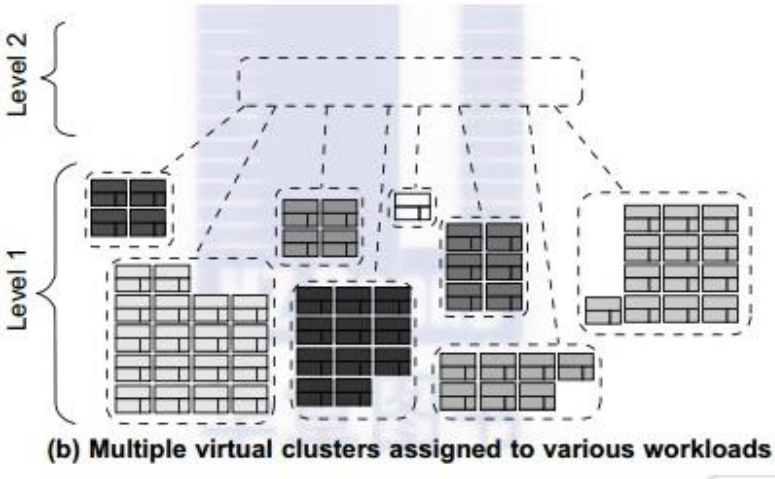


FIGURE 3.16
Multicore virtualization method that exposes four VCPUs to the software, when only three cores are actually present.

Today's many-core CMPs use a physical hierarchy of two or more cache levels that statically determine the cache allocation and mapping. A virtual hierarchy is a cache hierarchy that can adapt to fit the workload or mix of workloads [39]. The hierarchy's first level locates data blocks close to the cores needing them for faster access, establishes a shared-cache domain, and establishes a point of coherence for faster communication. When a miss leaves a tile, it first attempts to locate the block (or sharers) within the first level. The first level can also pro-vide isolation between independent workloads. A miss at the L1 cache can invoke the L2 access.



(a) Mapping of VMs into adjacent cores



(b) Multiple virtual clusters assigned to various workloads

CMP server consolidation by space-sharing of VMs into many cores forming multiple virtual clusters to execute various workloads.

CPU, Memory, and I/O Devices virtualization are three essential components of cloud computing that enable the efficient sharing of physical resources among multiple users.

1. CPU Virtualization: CPU virtualization is the process of dividing a physical CPU into multiple virtual CPUs. It allows multiple operating systems and applications to run on the same physical server without interfering with each other. Each virtual CPU is assigned a portion of the physical CPU's processing power, allowing multiple users to share the same physical hardware.
2. Memory Virtualization: Memory virtualization is the process of creating a virtual representation of a physical computer's memory. This allows multiple operating systems and applications to access the same physical memory, without interfering with each other. The virtual memory is managed by the hypervisor, which ensures that each virtual machine has the necessary amount of memory.
3. I/O Devices Virtualization: I/O devices virtualization is the process of creating a virtual representation of physical I/O devices, such as network adapters, storage devices, and other peripherals. This allows multiple virtual machines to access the same physical I/O devices without interference. The hypervisor acts as a virtual I/O controller, managing the access to physical I/O devices.

How it works: In cloud computing, virtualization software called a hypervisor is used to create and manage virtual machines (VMs) on a physical server. The hypervisor allows multiple VMs to share the same physical resources, such as CPU, memory, and I/O devices, by allocating a portion of these resources to each VM.

When a user requests a new virtual machine, the hypervisor creates a virtual representation of the hardware resources required by the VM. Each VM has its own operating system, applications, and data, but they share the same physical resources, allowing for efficient resource utilization.

Virtualization is a key technology that enables cloud computing by allowing multiple users to share the same physical resources. CPU, memory, and I/O devices virtualization are critical components of this technology, which enable efficient resource sharing, improved utilization, and cost savings.

2. Tabulate the difference between CPU, Memory, and I/O Devices virtualization in context with cloud computing list 10 points of each

Points	CPU Virtualization	Memory Virtualization	I/O Devices Virtualization
1. Definition	CPU virtualization refers to the process of creating multiple virtual CPUs out of a physical CPU, enabling multiple operating systems and applications to run on the same physical server.	Memory virtualization refers to the process of creating a virtual representation of physical memory, enabling multiple operating systems and applications to share the same physical memory.	I/O devices virtualization refers to the process of creating a virtual representation of physical I/O devices, enabling multiple virtual machines to access the same physical devices.
2. Function	It allows multiple users to share a single physical CPU while providing each user with an isolated virtual environment.	It enables multiple users to share a single pool of physical memory while providing each user with an isolated virtual environment.	It enables multiple users to share a single pool of physical I/O devices while providing each user with an isolated virtual environment.
3. Benefits	It increases the utilization of physical CPUs, reduces hardware costs, and provides a flexible environment for users.	It increases the utilization of physical memory, reduces hardware costs, and provides a flexible environment for users.	It increases the utilization of physical I/O devices, reduces hardware costs, and provides a flexible environment for users.
4. Requirements	Hardware support for virtualization, a hypervisor or virtual machine monitor (VMM), and virtualization-aware operating systems.	Hardware support for virtualization, a hypervisor or virtual machine monitor (VMM), and virtualization-aware operating systems.	Hardware support for virtualization, a hypervisor or virtual machine monitor (VMM), and virtualization-aware operating systems with virtual device drivers.
5. Examples	VMware ESXi, Microsoft Hyper-V, KVM, Xen.	VMware ESXi, Microsoft Hyper-V, KVM, Xen.	VMware ESXi, Microsoft Hyper-V, KVM, Xen.
6. Types	Full virtualization, Para-virtualization, Hardware-assisted virtualization.	Transparent memory page sharing, Memory ballooning, Memory over commitment.	Full device virtualization, Para-virtualization, Virtual I/O device (VirtIO).
7. Performance	The performance of virtual CPUs can be slightly lower than that of physical CPUs due to the overhead of virtualization.	The performance of virtual memory can be slightly lower than that of physical memory due to the overhead of virtualization.	The performance of virtual I/O devices can vary depending on the type of virtualization used and the workload characteristics.

8. Management	Virtual CPUs can be managed and allocated by the hypervisor to ensure that each virtual machine has the necessary resources.	Virtual memory can be managed and allocated by the hypervisor to ensure that each virtual machine has the necessary resources.	Virtual I/O devices can be managed and allocated by the hypervisor to ensure that each virtual machine has the necessary resources.
9. Security	Virtualization can improve security by isolating applications and operating systems, but it also introduces new security risks if not properly configured.	Virtualization can improve security by isolating applications and operating systems, but it also introduces new security risks if not properly configured.	Virtualization can improve security by isolating applications and operating systems, but it also introduces new security risks if not properly configured.
10. Limitations	Virtualization can introduce additional latency, and some workloads may not be suitable for virtualization due to performance requirements.	Memory virtualization may not be suitable for memory-intensive workloads or workloads with strict latency requirements.	Virtualizing certain types of I/O devices, such as graphics cards or storage controllers, can be challenging due to the complexity of their drivers and the need for low latency access.

Virtualization of I/O (input/output) devices is a technology that allows multiple virtual machines (VMs) to share access to physical I/O devices, such as network adapters, storage devices, and graphics cards. In other words, it allows multiple VMs to use the same physical I/O device as if each VM has its own dedicated device, even though they are sharing the same physical device.

In cloud computing, virtualization of I/O devices is critical for providing high-performance and scalable services to customers. The hypervisor manages the access to physical I/O devices and creates virtual devices for each VM. Each virtual device is assigned a portion of the physical device's capacity and can use it as if it was running on dedicated hardware.

Virtualization of I/O devices works by intercepting I/O requests from the VMs and redirecting them to the appropriate physical device. The hypervisor also ensures that each VM's I/O requests do not interfere with each other, by managing access to the physical device and implementing techniques like time slicing and queueing.

By allowing multiple VMs to share physical I/O devices, virtualization of I/O devices can significantly improve the performance and efficiency of cloud computing environments. With virtualization of I/O devices, cloud providers can consolidate multiple workloads onto a smaller number of physical servers, reducing hardware costs and improving resource utilization. It also makes it possible to dynamically allocate I/O resources to VMs based on their needs, enabling more efficient use of I/O resources and reducing wastage.

Virtualization of I/O devices can enhance the security and reliability of cloud computing environments. By isolating I/O access between VMs, it ensures that I/O errors or security breaches in one VM do not affect other VMs. This helps to prevent unauthorized access or attacks from spreading across the cloud infrastructure.
Virtualization of I/O devices is a crucial technology for enabling high-performance, scalable, and secure cloud computing environments.

There are three main types of virtualization in cloud computing: CPU virtualization, memory virtualization, and I/O virtualization. Each type of virtualization is used to create virtual machines (VMs) that operate as if they were running on dedicated hardware, even though they are sharing physical resources with other VMs.

- CPU virtualization is used to create virtual CPUs for each VM,
- memory virtualization is used to create virtual memory for each VM, and
- I/O virtualization is used to create virtual I/O devices for each VM.

1. **CPU virtualization:** As the name suggests, CPU virtualization is used to create virtual CPUs for each VM, enabling multiple operating systems to run simultaneously on a single physical machine. A hypervisor is used to manage access to the physical CPU, creating virtual CPUs for each VM and scheduling CPU resources to ensure that each VM has access to the necessary computing power.

Important points about CPU Virtualization:

- CPU virtualization is the process of dividing a physical CPU into multiple virtual CPUs.
- It allows multiple operating systems to run on a single physical machine.
- It enables improved resource utilization and reduces hardware costs.
- It provides isolation and security for each VM.
- It allows dynamic allocation of resources to VMs based on their needs.
- The hypervisor is responsible for creating virtual CPUs for each VM.
- CPU scheduling algorithms are used to ensure that each VM has access to the necessary computing power.
- The hypervisor intercepts CPU instructions and translates them to the appropriate physical CPU instruction.
- CPU virtualization can be done in two ways: full virtualization and paravirtualization.
- Full virtualization emulates an entire physical CPU, while paravirtualization provides a partially virtualized CPU that requires modifications to the guest OS.

2. **Memory virtualization:** Memory virtualization allows multiple VMs to share a physical memory resource while isolating each VM's memory space. The hypervisor creates virtual memory space for each VM, assigning a portion of the available physical memory to each VM. Memory virtualization also includes techniques like memory paging to move memory pages from physical memory to disk storage, freeing up physical memory for other VMs.

Important points about Memory Virtualization:

- Memory virtualization is the process of dividing physical memory into multiple virtual memory spaces.
- It allows multiple VMs to share a physical memory resource.
- It enables improved resource utilization and reduces hardware costs.
- It provides isolation and security for each VM.
- It allows dynamic allocation of resources to VMs based on their needs.
- The hypervisor is responsible for creating virtual memory space for each VM.
- Memory paging is used to move memory pages from physical memory to disk storage, freeing up physical memory for other VMs.
- Memory over commitment can be used to allocate more virtual memory to VMs than there is physical memory.
- Memory ballooning is a technique that enables the hypervisor to reclaim memory from VMs that are not using it.
- Memory virtualization can be done in two ways: hardware-assisted and software-based.

3. **I/O virtualization:** I/O virtualization allows multiple VMs to share access to physical I/O devices, such as network adapters, storage devices, and graphics cards. The hypervisor manages access to the physical devices and creates virtual devices for each VM, allowing each VM to use the I/O device as if it was running on dedicated hardware.

Important points about I/O Virtualization:

- I/O virtualization is the process of sharing physical I/O devices among multiple VMs.
- It allows multiple VMs to share access to network adapters, storage devices, and graphics cards.
- It enables improved resource utilization and reduces hardware costs.
- It provides isolation and security for each VM.
- It allows dynamic allocation of resources to VMs based on their needs.
- The hypervisor creates virtual I/O devices for each VM.
- I/O virtualization techniques include network virtualization, storage virtualization, and graphics virtualization.
- Time slicing and queueing are used to manage access to physical I/O devices among multiple VMs.
- Direct device assignment is a technique that allows a VM to access a physical I/O device directly without going through the hypervisor.
- I/O virtualization can be done in two ways: full virtualization and para-virtualization.

1. **CPU Virtualization Challenges:**

- **Performance overhead:** Virtualizing CPUs can introduce performance overhead due to the additional layer of abstraction introduced by the hypervisor. This overhead can be mitigated by using hardware-assisted virtualization, which offloads some of the virtualization tasks to the CPU hardware.

- **Resource contention:** When multiple VMs share a physical CPU, resource contention can occur, leading to performance degradation. This challenge can be addressed by using advanced CPU scheduling algorithms that prioritize the allocation of CPU resources based on the needs of each VM.

- **Compatibility issues:** Some operating systems or applications may not be compatible with certain CPU virtualization techniques. This challenge can be overcome by using para-virtualization, which requires the guest operating system to be modified to be aware of the hypervisor and the virtual environment.

2. **Memory Virtualization Challenges:**

- **Memory fragmentation:** When multiple VMs share a physical memory resource, memory fragmentation can occur, leading to inefficient use of memory. This challenge can be addressed by using memory ballooning techniques that enable the hypervisor to reclaim memory from VMs that are not using it.

- **Memory page swapping:** When memory resources are limited, the hypervisor may need to swap memory pages between physical memory and disk storage, which can introduce performance overhead. This challenge can be mitigated by using hardware-

assisted memory virtualization, which offloads some of the virtualization tasks to the CPU hardware.

- Memory overcommitment: Overcommitting memory can cause performance issues and stability problems. This challenge can be addressed by setting realistic memory limits for VMs and by monitoring memory usage closely to avoid overcommitment.

3. I/O Virtualization Challenges:

- **Device sharing:** Sharing I/O devices among multiple VMs can introduce performance overhead and contention. This challenge can be addressed by using dedicated hardware for each VM or by using advanced queuing and time slicing techniques to manage access to shared I/O devices.

- **Compatibility issues:** Some I/O devices may not be compatible with certain virtualization techniques, such as para-virtualization. This challenge can be addressed by using full virtualization techniques that emulate an entire physical device.

- **Direct device assignment:** Direct device assignment can improve I/O performance, but it can also introduce security risks by allowing VMs to access physical devices directly. This challenge can be addressed by using hardware-based security features that isolate VMs and prevent unauthorized access to physical devices.

DESKTOP VIRTUALIZATION

Desktop virtualization enables delivery of secure, full-fidelity desktop experiences to end users on any device. 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.

DESKTOP VIRTUALIZATION DEPLOYMENT MODELS

Virtual desktop infrastructure (VDI)

In VDI deployment model, 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—including drivers, CPUs and memory—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 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. 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 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.

Choosing a model

VDI is a popular choice because it offers a virtualized version of a familiar computing model, physical desktop computing. But implementing VDI requires you to manage all aspects of the infrastructure yourself, including the hardware, operating systems and applications, and hypervisor and associated software. This can be challenging if your VDI experience and expertise is limited. Purchasing all infrastructure components can require a larger upfront investment.

RDS/RDSH can be a solid choice if it supports the specific applications you need to run and your end users only need access to those applications, not full

Windows desktops. RDS offers greater end-user density per server than VDI, and systems are usually cheaper and more scalable than full VDI environments. Your staff does need the requisite skill set and experience to administer and manage RDS/RDSH technology, however.

DaaS is currently gaining in popularity as IT teams grow more comfortable with shared desktops and shared applications. Overall, it tends to be the most cost-effective option. It's also the easiest to administer, requiring little in-house expertise in managing infrastructure or VDI. It's readily scalable and involves operating expenditures rather than capital expenditures, a more affordable cost structure for many businesses.

Benefits of desktop virtualization

Virtualizing desktops provides many potential benefits that can vary depending upon the deployment model you choose.

Simpler administration

Desktop virtualization can make it easier for IT teams to manage employee computing needs. Your business can maintain a single VM template for employees within similar roles or functions instead of maintaining individual computers that must be reconfigured, updated or patched whenever software changes need to be made. This saves time and IT resources.

Cost savings

Many virtual desktop solutions allow you to shift more of your IT budget from capital expenditures to operating expenditures. Because compute-intensive applications require less processing power when they're delivered through VMs hosted on a data center server, desktop virtualization can extend the life of older or less powerful end-user devices. On-premise virtual desktop solutions may require a significant initial investment in server hardware, hypervisor software and other infrastructure, which makes cloud-based DaaS—wherein you simply pay a regular usage-based charge—a more attractive option.

Improved productivity

Desktop virtualization makes it easier for employees to access enterprise computing resources. They can work anytime, anywhere, from any supported device with an Internet connection.

Support for a broad variety of device types

Virtual desktops can support remote desktop access from a wide variety of devices, including laptop and desktop computers, thin clients, zero clients, tablets and even some mobile phones. You can use virtual desktops to deliver workstation-like experiences and access to the full desktop anywhere and anytime, regardless of the operating system native to the end user device.

Stronger security

In desktop virtualization, the desktop image is abstracted and separated from the physical hardware used to access it, and the VM used to deliver the desktop image can be a tightly controlled environment managed by the enterprise IT department.

Agility and scalability

It's quick and easy to deploy new VMs or serve new applications whenever necessary, and it is just as easy to delete them when they're no longer needed.

Better end-user experiences

When you implement desktop virtualization, your end users will enjoy a feature-rich experience without sacrificing functionality they've come to rely on, like printing or access to USB ports.

Desktop virtualization software

The software required for delivering virtual desktops depends on the virtualization method you chose.With VDI, the desktop operating system (most commonly Microsoft Windows) runs and is managed in the data center. Hypervisor software runs on the host server, delivering access to a VM to each end user over the network. Connection broker software is required to authenticate users, connect each to a virtual machine, monitor activity levels, and reassign the VM when the connection is terminated. Connection brokers may be bundled with, or purchased separately from, the hypervisor.

RDS or RDSH can be implemented using utilities that are bundled with the Microsoft Windows Server operating system.

Desktop Virtualization

By far this can be looked at as the most common type of Virtualization in the IT industry. It is widely used in a workplace environment. In this system, every user's desktop is virtualized and stored on a common server which allows the user to **access his desktop virtually** from any device which is a part of the system.

This system also takes care of data security by ensuring that any kind of data transfers happen through secure protocols. This kind of virtualization comes with a number of advantages like it allows for portability and easy mobility of the user and an easy and efficient management of the software and its installation as well as the updates on it.

SERVER VIRTUALIZATION

Server Virtualization is most important part of Cloud Computing. So, Talking about Cloud Computing, it is composed of two words, cloud and computing.

Cloud means Internet and computing means to solve problems with help of computers. Computing is related to CPU & RAM in digital world.

Now Consider situation, You are using Mac OS on your machine but particular application for your project can be operated only on Windows. You can either buy new machine running windows or create virtual environment in which windows can be installed and used.

Second option is better because of less cost and easy implementation. This scenario is called **Virtualization**. In it, virtual CPU, RAM, NIC and other resources are provided to OS which it needed to run.

This resources is virtually provided and controlled by an application called Hypervisor. The new OS running on virtual hardware resources is collectively called **Virtual Machine (VM)**.

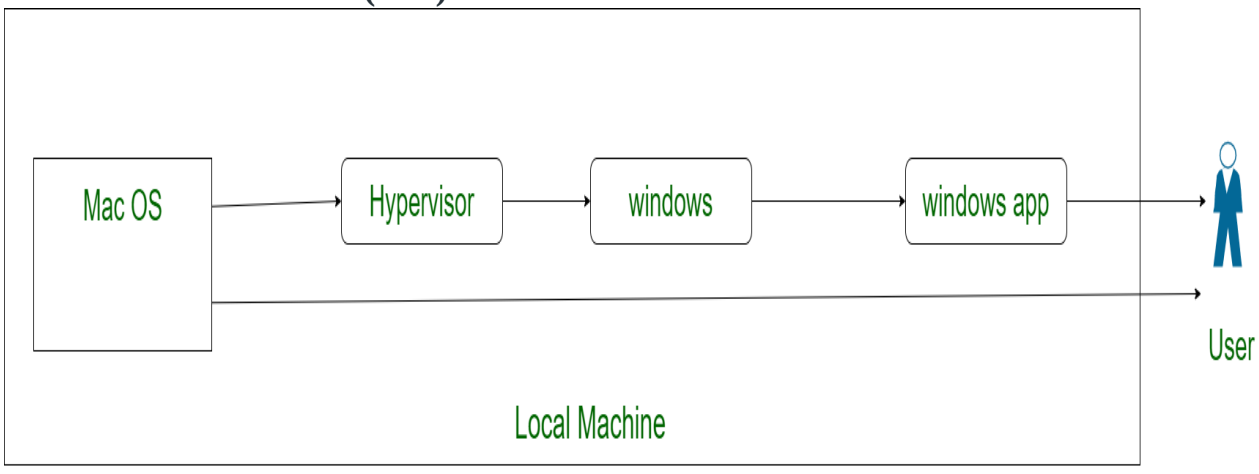


Figure - Virtualization on local machine

Now migrate this concept to data centers where lot of servers (machines with fast CPU, large RAM and enormous storage) are available.

Enterprise owning data centre provide resources requested by customers as per their need. Data centers have all resources and on user request, particular amount of CPU, RAM, NIC and storage with preferred OS is provided to users. This concept of virtualization in which services are requested and provided over Internet is called **Server Virtualization.**

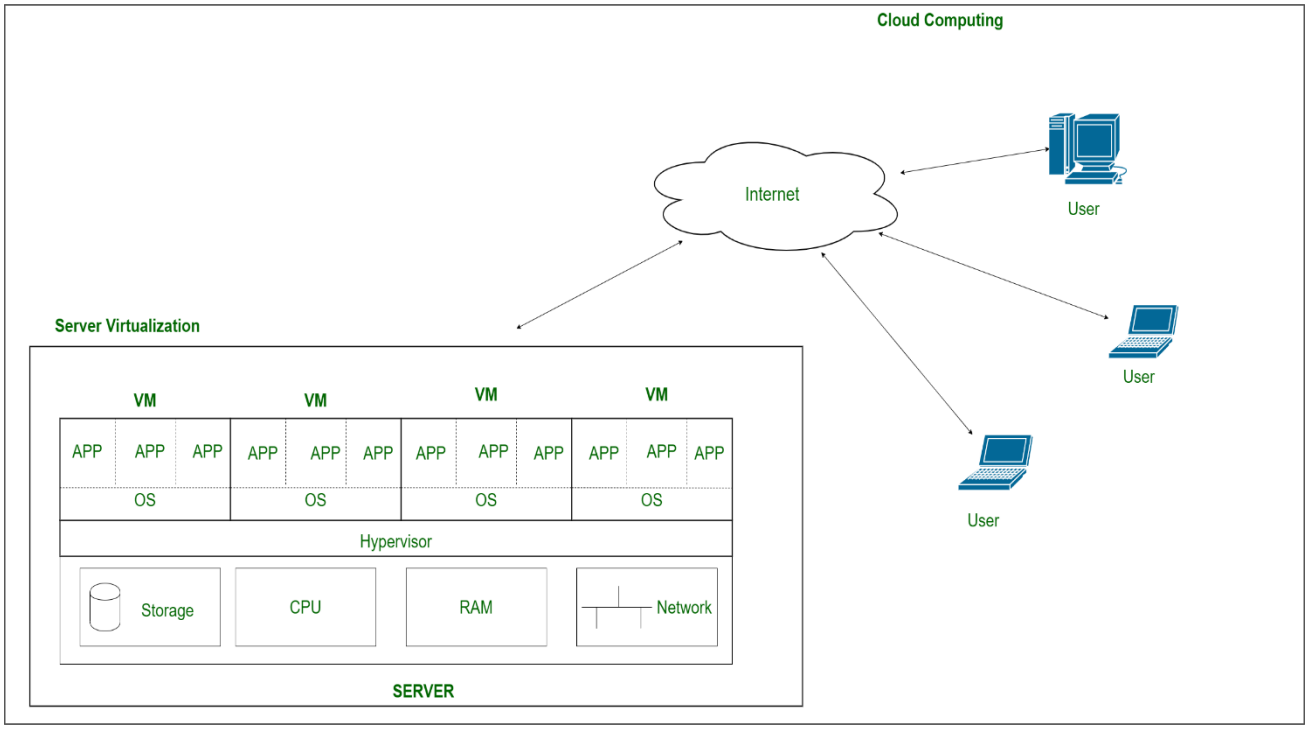


Figure - Server Virtualization

To implement Server Virtualization, hypervisor is installed on server which manages and allocates host hardware requirements to each virtual machine. This hypervisor sits over server hardware and regulates resources of each VM. A user can increase or decrease resources or can delete entire VM as per his/her need. This servers with VM created on them is called server virtualization and concept of controlling this VM by users through internet is called **Cloud Computing.**

Advantages of Server Virtualization:

- Each server in server virtualization can be restarted separately without affecting the operation of other virtual servers.
- Server virtualization lowers the cost of hardware by dividing a single server into several virtual private servers.
- One of the major benefits of server virtualization is disaster recovery. In server virtualization, data may be stored and retrieved from any location and moved rapidly and simply from one server to another.
- It enables users to keep their private information in the data centers.

Disadvantages of Server Virtualization:

- The major drawback of server virtualization is that all websites that are hosted by the server will cease to exist if the server goes offline.
- The effectiveness of virtualized environments cannot be measured.
- It consumes a significant amount of RAM.
- Setting it up and keeping it up are challenging.
- Virtualization is not supported for many essential databases and apps.

Types of Server Virtualization

The types of server virtualization are as follows –

1. Full Virtualization

Full Virtualization uses a hypervisor to directly connect with the CPU and physical server. It supports the best isolation and security structure to the virtual machines. It is similar to Para-virtualization. Therefore, machine operation used by the operating system is used to implement input-output or change the system status.

The unmodified operating frameworks can run on top of the hypervisor. This is possible because of the operations, which are emulated in the application and the status programs are restored with what the real hardware can deliver.

2. Para Virtualization

In the Para virtualization model the simulation is trapping overhead in software virtualizations. It depends on the hypervisor and the guest operating system and changed entry compiled for installing it in a virtual device.

3. Operating System Virtualization

Operating system virtualization is also referred to as system-level virtualization. It is a server virtualization technology that divides one operating framework into multiple isolated user-space called a virtual environments. The main advantage of using server visualization is that it decreases the use of the physical area, so it will store money.

4. Hardware-Assisted Virtualization

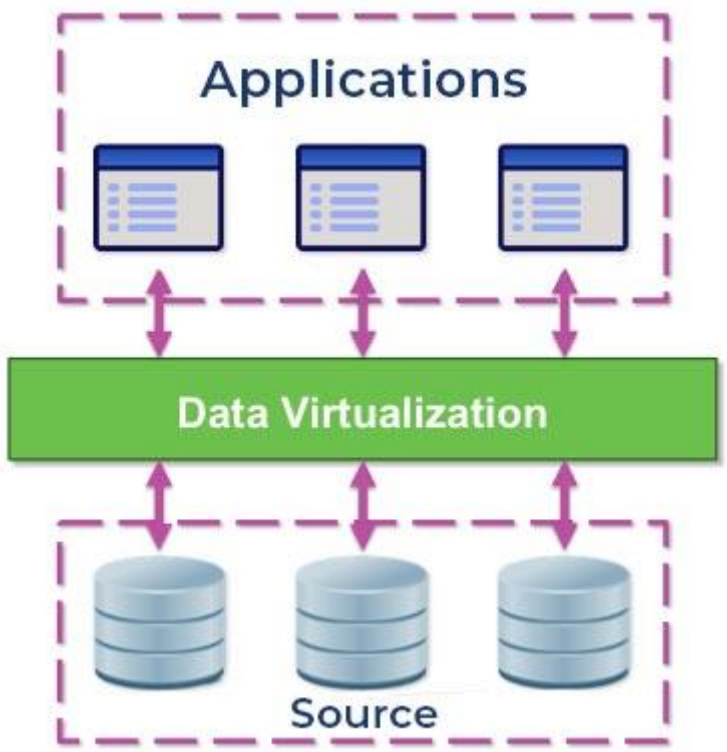
Hardware-Assisted Virtualization was given by AMD and Intel. It is also referred to as Hardware virtualization, AMD virtualization, and Intel virtualization. It is created to improve the performance of the processor. The benefit of using Hardware-Assisted Virtualization is that it needs less hypervisor overhead.

What is Data Virtualization?

Data virtualization is a data management technique in cloud computing that integrates data from different sources into a single virtual layer. It creates a single, logical, and virtual view of data. The virtual view of data can be

access by applications such as web portals, dashboards, e-commerce, mobile apps, etc. Data virtualization allows users to retrieve and manipulate data without knowing how and where it is stored.

In big organizations data are collected from different sources and stored in different format. To manipulate and analyze the data, it is required that the data is retrieved in a suitable format. Data virtualization creates a virtual layer between data sources and applications that need it.



Data Virtualization tools process the data request from the application and returns the result in a suitable format. It gives users a feel that all data are at a single place.

How Data Virtualization Works?

The working of data virtualization in cloud computing can be understood through the following steps –

- **Data Abstraction** – data from different sources is pulled together into a single virtual layer.
- **Data Integration** – Data from different systems is combined into a single view.
- **Querying and transformation** – Users can access and query the data from the source systems and perform different tasks such as data analysis, manipulation, etc.

Advantages of Data Virtualization

There are different benefits and advantages of the data virtualization, some are discussed as follows –

- Data virtualization integrates all your data sources and creates a single view and allows real-time access.
- It provides applications with real-time access to multiple data sources in a single view regardless of data source and format.
- It provides best resource utilization by running multiple virtual instances on a single physical server.
- Data virtualization solutions increase flexibility for data integration and support cross-functional data analysis.
- It reduces costs by creating multiple virtual instances onto fewer physical servers.
- It provided user friendly interface to analyze and manipulate data.
- It reduces latency as it eliminates the complex data movement.
- It used metadata and advance data query optimization to retrieve data from sources. It reduces the data integration cost.

Disadvantages of Data Virtualization

With several advantages, the data virtualization come with some drawbacks or disadvantages also. We have discussed some of them ass follows-

- It creates flexibility and portability issue as organizations may become dependent on third party providers.
- It requires high implementation cost.
- It causes issues with availability and scalability as it depends on third-party providers.
- It may introduce new security risks.

Use Cases of Data Virtualization

There are different uses of the data virtualization in cloud computing. Some of the use cases are as follows –

- **Real-time data integration** – It combines data from different systems, CRM, ERP, and external data sources to provide real-time insights for decision-making. Real-time data integration is critical for industries like finance and telecommunications that depend on real-time data for operations and customer services.
- **Cross-functional reporting** – It allows different business units to access relevant data from various data sources for reporting and analysis.
- **Business Intelligence and Reporting** – It create virtual data sets for quick analysis and reporting across different business units.
- **Application development** – It facilitates developers access to different data sources that enhances the application development speed.

Industries using Data Virtualization

The data virtualization is used in many industries. The following is a list of industries where data virtualization ins used −

- Healthcare
- Finance
- Telecommunications
- Government
- Manufacturing
- Retail

Data Virtualization Tools

The following are some data virtualization tools used by different organizations −

- IBM Cloud Pak for Data
- TIBCO Data Virtualization
- CData Software
- Informatica
- Red Hat JBoss Data Virtualization
- AtScale
- Stone Bond Technologies

GOOGLE APP ENGINE

A scalable runtime environment, Google App Engine is mostly used to run Web applications. These dynamic scales as demand change over time because of Google's vast computing infrastructure.

Because it offers a secure execution environment in addition to a number of services, App Engine makes it easier to develop scalable and high-performance Web apps. Google's applications will scale up and down in response to shifting demand. Croon tasks, communications, scalable data stores, work queues, and in-memory caching are some of these services.

The App Engine SDK facilitates the testing and professionalization of applications by emulating the production runtime environment and allowing developers to design and test applications on their own PCs.

When an application is finished being produced, developers can quickly migrate it to App Engine, put in place quotas to control the cost that is generated, and make the programmer available to everyone. Python, Java, and Go are among the languages that are currently supported.

The development and hosting platform Google App Engine, which powers anything from web programming for huge enterprises to mobile apps, uses the same

infrastructure as Google's large-scale internet services. It is a fully managed PaaS (platform as a service) cloud computing platform that uses in-built services to run your apps. You can start creating almost immediately after receiving the software development kit (SDK). You may immediately access the Google app developer's manual once you've chosen the language you wish to use to build your app.

After creating a Cloud account, you may Start Building your App

- Using the Go template/HTML package
- Python-based webapp2 with Jinja2
- PHP and Cloud SQL
- using Java's Maven

FEATURES OF APP ENGINE

Runtimes and Languages

To create an application for an app engine, you can use Go, Java, PHP, or Python. You can develop and test an app locally using the SDK's deployment toolkit. Each language's SDK and nun time are unique. Your program is run in a:

- Java Run Time Environment version 7
- Python Run Time environment version 2.7
- PHP runtime's PHP 5.4 environment
- Go runtime 1.2 environment

Generally Usable Features

These are protected by the service-level agreement and depreciation policy of the app engine. The implementation of such a feature is often stable, and any changes made to it are backward-compatible. These include communications, process management, computing, data storage, retrieval, and search, as well as app configuration and management. Features like the HRD migration tool, Google Cloud SQL, logs, datastore, dedicated Memcached, blob store, Memcached, and search are included in the categories of data storage, retrieval, and search.

Features in Preview

In a later iteration of the app engine, these functions will undoubtedly be made broadly accessible. However, because they are in the preview, their implementation may change in ways that are backward-incompatible. Sockets, MapReduce, and the Google Cloud Storage Client Library are a few of them.

Experimental Features

These might or might not be made broadly accessible in the next app engine updates. They might be changed in ways that are irreconcilable with the past. The "trusted tester" features, however, are only accessible to a limited user base and require registration in order to utilize them. The experimental features include Prospective Search, Page Speed, OpenID, Restore/Backup/Datastore Admin, Task Queue Tagging, MapReduce, and Task Queue REST API. App metrics analytics, datastore admin/backup/restore, task queue tagging, MapReduce, task queue REST API, OAuth, prospective search, OpenID, and Page Speed are some of the experimental features.

Third-Party Services

As Google provides documentation and helper libraries to expand the capabilities of the app engine platform, your app can perform tasks that are not built into the core product you are familiar with as app engine. To do this, Google collaborates with other organizations. Along with the helper libraries, the partners frequently provide exclusive deals to app engine users.

Advantages of Google App Engine

The Google App Engine has a lot of benefits that can help you advance your app ideas. This comprises:

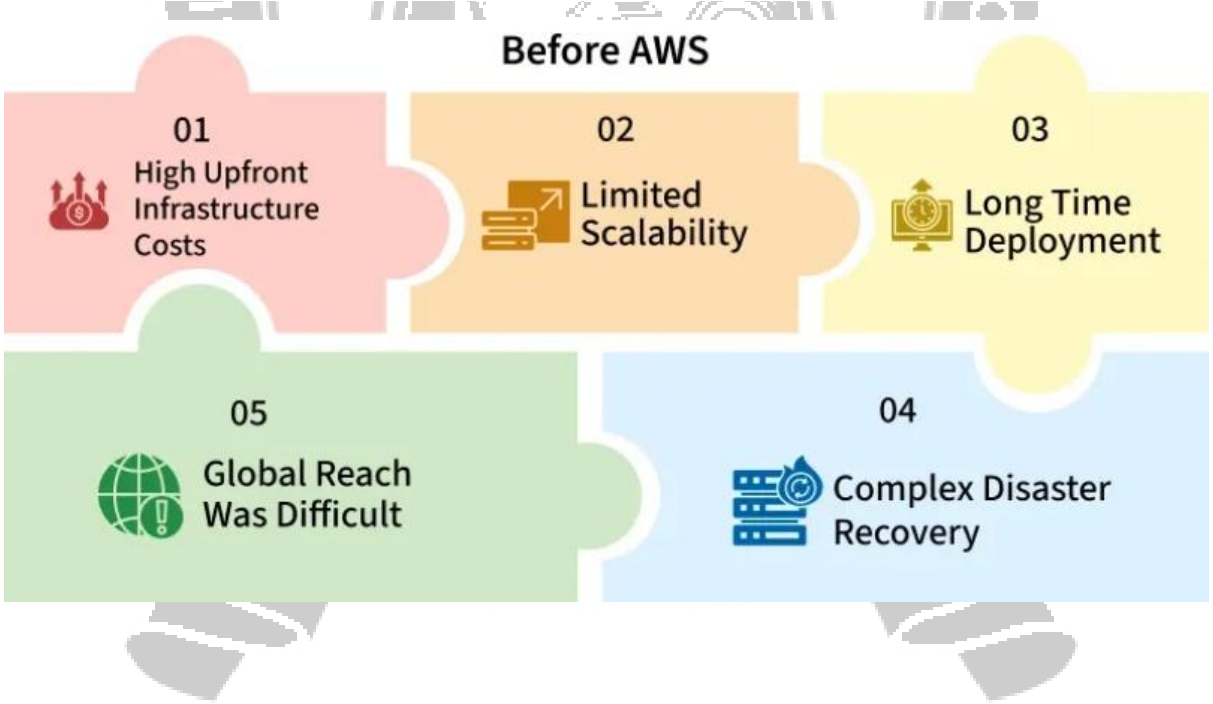
1. **Infrastructure for Security:** The Internet infrastructure that Google uses is arguably the safest in the entire world. Since the application data and code are hosted on extremely secure servers, there has rarely been any kind of illegal access to date.
2. **Faster Time to Market:** For every organization, getting a product or service to market quickly is crucial. When it comes to quickly releasing the product, encouraging the development and maintenance of an app is essential. A firm can grow swiftly with Google Cloud App Engine's assistance.
3. **Quick to Start:** You don't need to spend a lot of time prototyping or deploying the app to users because there is no hardware or product to buy and maintain.
4. **Easy to Use:** The tools that you need to create, test, launch, and update the applications are included in Google App Engine (GAE).
5. **Rich set of APIs & Services:** A number of built-in APIs and services in Google App Engine enable developers to create strong, feature-rich apps.
6. **Scalability:** This is one of the deciding variables for the success of any software. When using the Google app engine to construct apps, you may access technologies like GFS, Big Table, and others that Google uses to build its own apps.

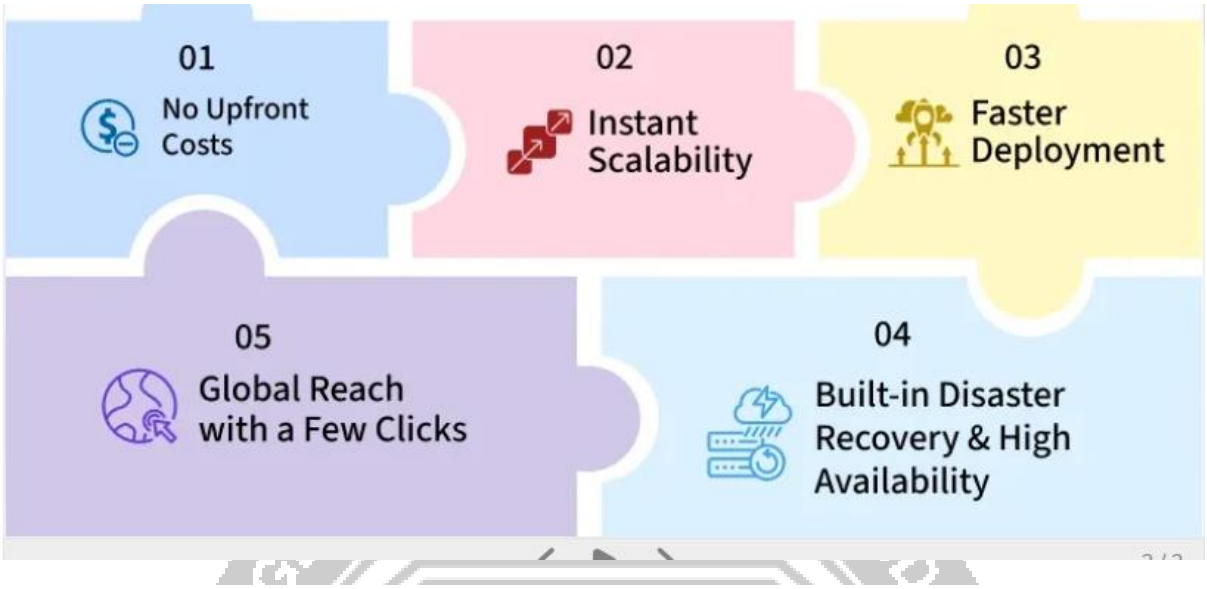
7. **Performance and Reliability:** Among international brands, Google ranks among the top ones. Therefore, you must bear that in mind while talking about performance and reliability.
8. **Cost Savings:** To administer your servers, you don't need to employ engineers or even do it yourself. The money you save might be put toward developing other areas of your company.
9. **Platform Independence:** Since the app engine platform only has a few dependencies, you can easily relocate all of your data to another environment.

AMAZON AWS

Amazon Web Services (AWS) is the world's most comprehensive and broadly adopted cloud platform. Started in 2006, it allows individuals, companies, and governments to access technology services such as computing power, storage, and databases on an on-demand basis.

Instead of buying, owning, and maintaining physical data centers and servers, you can access technology services from AWS and pay only for what you use. This is akin to flipping a light switch: you pay for the electricity (compute power) when you need it, and turn it off when you don't.





The AWS Global Infrastructure

The backbone of AWS is its massive global infrastructure. Understanding this is key to architecting resilient applications.

1. Regions: A Region is a physical location in the world where AWS clusters data centers. (e.g., us-east-1 in N. Virginia or ap-south-1 in Mumbai).

- Why it matters: You choose a region to optimize latency (closeness to users), minimize costs (prices vary by region), and meet data sovereignty/compliance laws.

2. Availability Zones (AZs): Each Region consists of multiple, isolated locations known as Availability Zones.

- An AZ is one or more discrete data centers with redundant power, networking, and connectivity.
- Why it matters: If you run your app in just one data center and it floods, your app dies. If you run it across multiple AZs, your app stays online even if one building fails.

3. Edge Locations: These are smaller data centers located in major cities worldwide, used primarily for Amazon CloudFront (CDN) to cache content closer to end-users to reduce latency.

The Shared Responsibility Model:

Security is the top priority at AWS. To maintain it, AWS operates under a **Shared Responsibility Model**.

- **Security OF the Cloud (AWS Responsibility):** AWS protects the infrastructure that runs all the services offered in the AWS Cloud. This includes the hardware, software, networking, and facilities.
- **Security IN the Cloud (Customer Responsibility):** You are responsible for your data. This includes encryption, network configuration (firewalls), identity management (IAM), and OS patching.

Core AWS Services

AWS offers over 200 services. Here are the fundamental ones you must know to get started.

1. Compute Services

- **Amazon EC2 (Elastic Compute Cloud):** Resizable virtual servers. You choose the OS (Linux/Windows) and the hardware power (CPU/RAM). It is the workhorse of AWS.
- **AWS Lambda:** Serverless compute. You upload your code, and Lambda runs it only when triggered (by an event like a file upload). You don't manage any servers.
- **AWS Elastic Beanstalk:** Platform as a Service (PaaS) for deploying web apps. You upload code, and AWS handles the deployment (capacity provisioning, load balancing, auto-scaling).

2. Storage Services

- **Amazon S3 (Simple Storage Service):** Object storage for files (images, videos, backups). It allows you to store and retrieve any amount of data from anywhere on the web.
- **Amazon EBS (Elastic Block Store):** Block storage (virtual hard drives) that you attach to EC2 instances. It is persistent storage for your virtual servers.
- **Amazon Glacier:** Extremely low-cost storage for data archiving and long-term backup.

3. Database Services

- **Amazon RDS (Relational Database Service):** Managed SQL databases. It supports engines like MySQL, PostgreSQL, Oracle, SQL Server, and **Amazon Aurora** (AWS's high-performance proprietary engine).
- **Amazon DynamoDB:** A managed NoSQL database service that provides fast and predictable performance with seamless scalability.

4. Networking Services

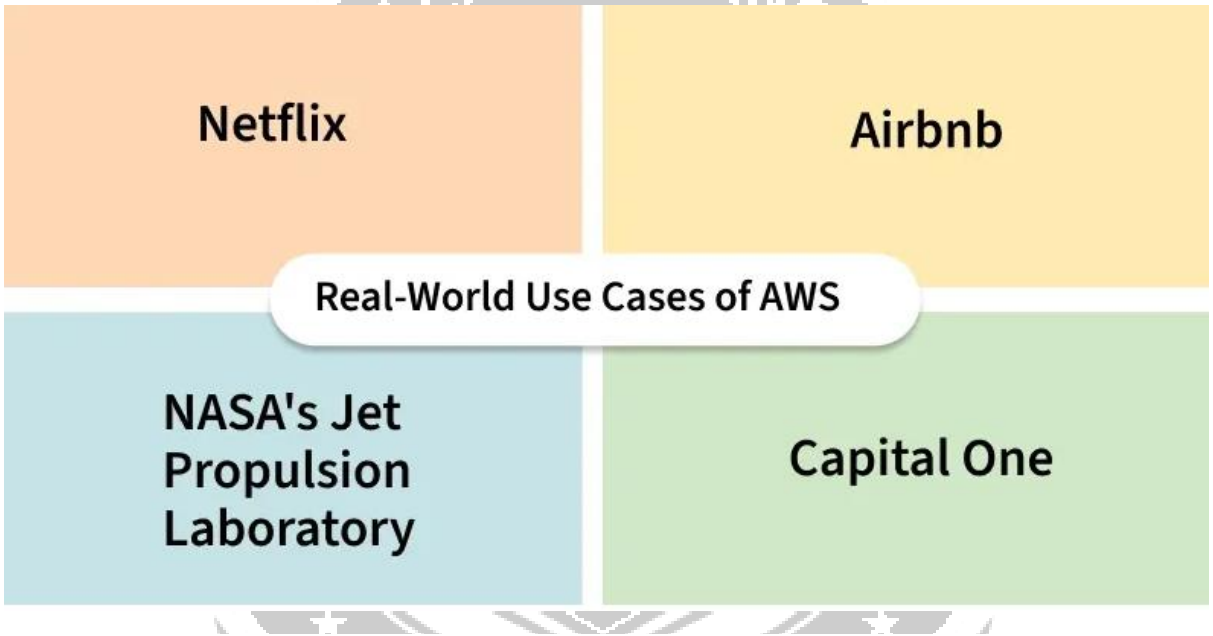
- **Amazon VPC (Virtual Private Cloud):** Lets you provision a logically isolated section of the AWS Cloud where you can launch resources in a virtual network you define.
- **Amazon CloudFront:** A Content Delivery Network (CDN) that speeds up distribution of your static and dynamic web content to users.

- **Amazon Route 53:** A highly available and scalable cloud Domain Name System (DNS) web service.

From startups to large enterprises like **Netflix**, **Airbnb**, and **NASA**, AWS is widely adopted for its flexibility, scalability, and security.

Real-World Use Cases of AWS

AWS services are used by both startups and large enterprises based on their specific needs. Startups use AWS to overcome hardware infrastructure costs and deploy applications efficiently. Whereas large scale companies are using AWS cloud services for the management of their Infrastructure to completely focus on the development of products widely.



Here are some real-world use cases of AWS services:

- **Netflix:** The large streaming giant using AWS for the storage and scaling of the applications for ensuring seamless content delivery with low latency without interruptions to millions of users globally.
- **Airbnb:** By utilizing AWS, Airbnb manages the various workloads and provides scalable and reliable infrastructure for its virtual marketplace and lodging offerings.
- **NASA's Jet Propulsion Laboratory:** It takes the help of AWS services to handle and analyze large-scale volumes of data related to vital scientific research missions and space exploration.
- **Capital One:** A financial Company that is utilizing AWS for its security and compliance while delivering innovative banking services to its customers.

AWS (Amazon Web Services) follows a pay-as-you-go pricing model, offering flexibility and scalability for businesses of all sizes. Pricing varies depending on the services you use, and AWS provides multiple options to help optimize costs. Here's an overview of key AWS pricing features and models:

1. Pay-as-You-Go Pricing

AWS charges for usage-based billing, meaning you only pay for what you use. This pricing model is based on factors like:

- Compute (e.g., EC2 instances)
- Storage (e.g., S3)
- Data transfer
- Requests and service usage (e.g., Lambda invocations)

This model is ideal for businesses with variable workloads.

2. On-Demand Instances

On-Demand Instances let you pay for compute capacity by the hour or second (depending on the instance type) with no long-term commitments or upfront payments. This option is ideal for:

- Applications with short-term, irregular, or unpredictable workloads
- First-time AWS users testing the platform
- Projects that cannot be interrupted

These instances offer maximum convenience and are perfect for development, testing, and prototyping workloads.

3. Reserved Pricing

For predictable usage, you can commit to a long-term contract (1 or 3 years) with reserved instances for services like EC2, RDS, and Redshift. This offers:

- Up to 75% cost savings compared to on-demand pricing.
- Flexible payment options (All upfront, Partial upfront, or No upfront).

4. Spot Instances

Spot Instances allow you to bid on unused EC2 capacity. Prices fluctuate based on supply and demand, and you can save up to 90% compared to on-demand prices.

- Great for batch processing, data analysis, or flexible workloads.

5. Free Tier

AWS offers a Free Tier for new users, providing access to a limited set of services for free, such as:

- 750 hours/month of EC2 (t2.micro instance) for the first 12 months.
- 5GB of standard S3 storage.
- 1 million Lambda requests/month.

This is an excellent way for businesses to explore AWS without incurring costs.

To Know How to Set up Free Tier Account on Amazon Web Services *Click Here*

6. AWS Pricing Calculator

AWS provides a Pricing Calculator to estimate costs based on your specific usage. It helps you project the total cost of your cloud infrastructure by selecting services and configurations relevant to your business.

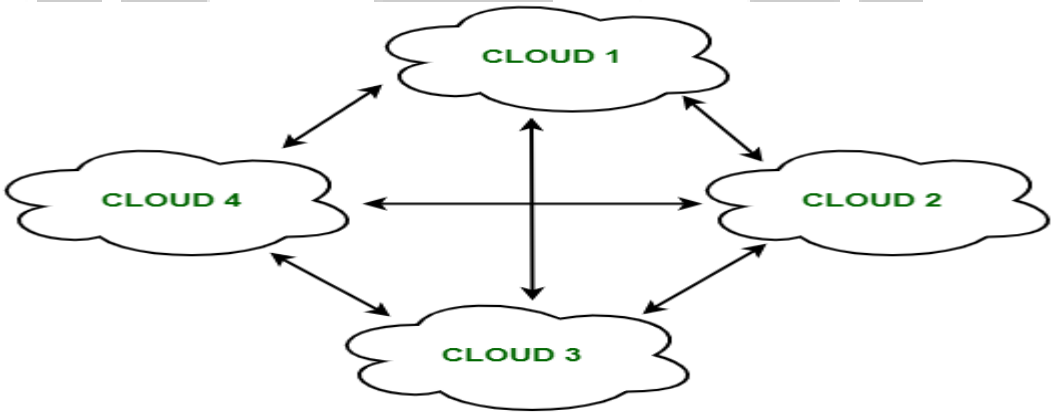
7. Cost Management and Optimization

AWS offers tools like AWS Cost Explorer and AWS Budgets to:

- Track usage and manage expenses.
- Set custom budgets and receive alerts when approaching limits.

FEDERATION IN THE CLOUD. 4.1

Cloud Federation, also known as Federated Cloud is the deployment and management of several external and internal cloud computing services to match business needs. It is a multi-national cloud system that integrates private, community, and public clouds into scalable computing platforms. Federated cloud is created by connecting the cloud environment of different cloud providers using a common standard.



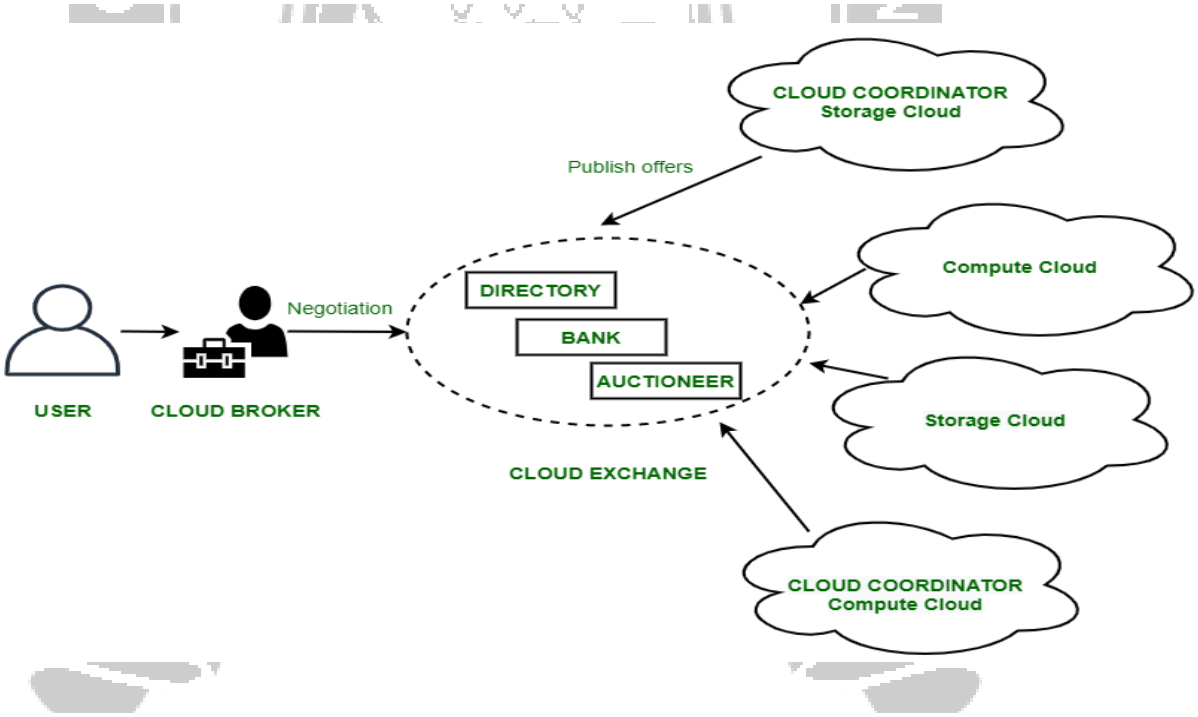
The architecture of Federated Cloud:

The architecture of Federated Cloud consists of three basic components:

1. Cloud Exchange
The Cloud Exchange acts as a mediator between cloud coordinator and cloud broker. The demands of the cloud broker are mapped by the cloud exchange to the available services provided by the cloud coordinator. The cloud exchange has a track record of what is the present cost, demand patterns, and available cloud providers, and this information is periodically reformed by the cloud coordinator.

2. Cloud Coordinator
The cloud coordinator assigns the resources of the cloud to the remote users based on the quality of service they demand and the credits they have in the cloud bank. The cloud enterprises and their membership are managed by the cloud controller.

3. Cloud Broker
The cloud broker interacts with the cloud coordinator, analyzes the Service-level agreement and the resources offered by several cloud providers in cloud exchange. Cloud broker finalizes the most suitable deal for their client.



Properties of Federated Cloud:

1. In the federated cloud, the users can interact with the architecture either centrally or in a decentralized manner. In centralized interaction, the user interacts with a broker to mediate between them and the organization. Decentralized interaction permits the user to interact directly with the clouds in the federation.
2. Federated cloud can be practiced with various niches like commercial and non-commercial.

3. The visibility of a federated cloud assists the user to interpret the organization of several clouds in the federated environment.
4. Federated cloud can be monitored in two ways. MaaS (Monitoring as a Service) provides information that aids in tracking contracted services to the user. Global monitoring aids in maintaining the federated cloud.
5. The providers who participate in the federation publish their offers to a central entity. The user interacts with this central entity to verify the prices and propose an offer.
6. The marketing objects like infrastructure, software, and platform have to pass through federation when consumed in the federated cloud.

Benefits of Federated Cloud:

1. It minimizes the consumption of energy.
2. It increases reliability.
3. It minimizes the time and cost of providers due to dynamic scalability.
4. It connects various cloud service providers globally. The providers may buy and sell services on demand.
5. It provides easy scaling up of resources.

Challenges in Federated Cloud:

1. In cloud federation, it is common to have more than one provider for processing the incoming demands. In such cases, there must be a scheme needed to distribute the incoming demands equally among the cloud service providers.
2. The increasing requests in cloud federation have resulted in more heterogeneous infrastructure, making interoperability an area of concern. It becomes a challenge for cloud users to select relevant cloud service providers and therefore, it ties them to a particular cloud service provider.
3. A federated cloud means constructing a seamless cloud environment that can interact with people, different devices, several application interfaces, and other entities.

Federated Cloud technologies:

The technologies that aid the cloud federation and cloud services are:

1. OpenNebula

It is a cloud computing platform for managing heterogeneous distributed data center infrastructures. It can use the resources of its interoperability, leveraging existing information technology assets, protecting the deals, and adding the application programming interface (API).

2. Aneka coordinator

The Aneka coordinator is a proposition of the Aneka services and Aneka peer components (network architectures) which give the cloud ability and performance to interact with other cloud services.

3. Eucalyptus

Eucalyptus defines the pooling computational, storage, and network resources that can be measured scaled up or down as application workloads change in the utilization of the software. It is an open-source framework that performs the storage, network, and many other computational resources to access the cloud environment.