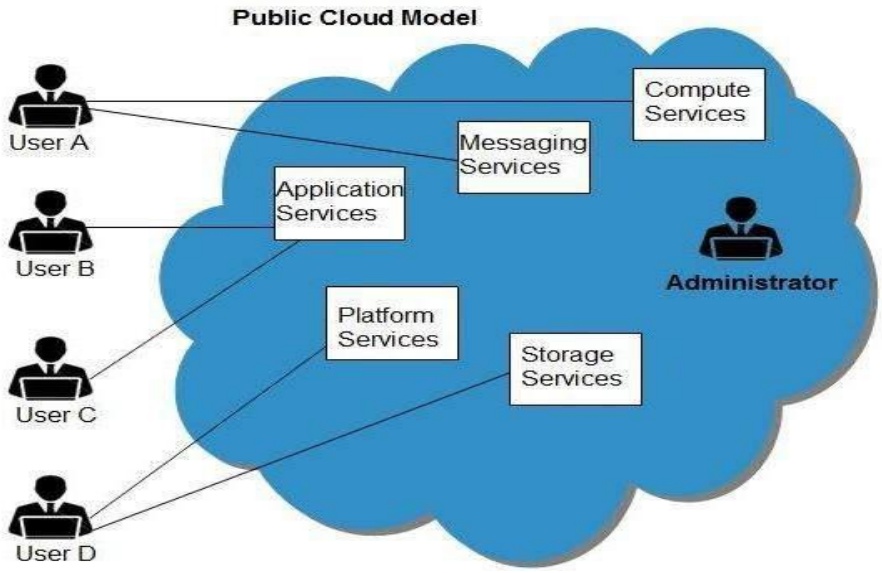
Types of Cloud

Public Cloud:

Public Cloud allows systems and services to be easily accessible to general public. The IT giants such as Google, Amazon and Microsoft offer cloud services via Internet.

The Public Cloud Model is shown in the diagram below.



Benefits:

There are many benefits of deploying cloud as public cloud model. The following diagram shows some of those benefits:

Cost Effective

Since public cloud shares same resources with large number of customers it turns out inexpensive.

Reliability

The public cloud employs large number of resources from different locations. If any of the resources fails, public cloud can employ another one.

Flexibility

The public cloud can smoothly integrate with private cloud, which gives customers a flexible approach.

Location Independence

Public cloud services are delivered through Internet, ensuring location independence.

Utility Style Costing

Public cloud is also based on pay-per-use model and resources are accessible whenever customer needs them.

High Scalability

Cloud resources are made available on demand from a pool of resources, i.e., they can be scaled up or down according the requirement.

Disadvantages:

Here are some disadvantages of public cloud model:

Low Security

In public cloud model, data is hosted off-site and resources are shared publicly, therefore does not ensure higher level of security.

Less Customizable

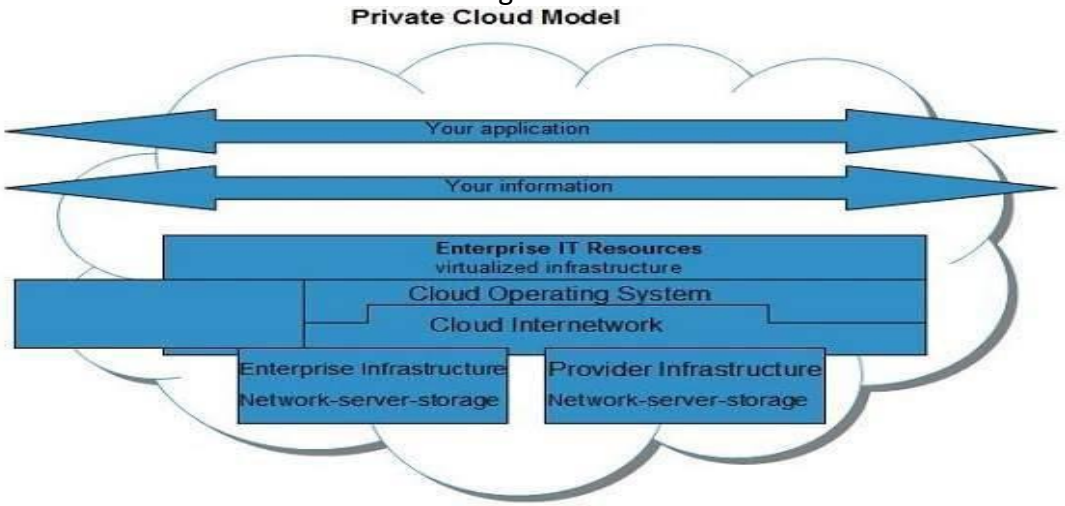
It is comparatively less customizable than private cloud.

Private Cloud:

Private Cloud allows systems and services to be accessible within an organization.

The Private Cloud is operated only within a single organization. However, it may be managed internally by the organization itself or by third-party.

The private cloud model is shown in the diagram below.



Benefits:

There are many benefits of deploying cloud as private cloud model. The following diagram shows some of those benefits:

High Security and Privacy

Private cloud operations are not available to general public and resources are shared from distinct pool of resources. Therefore, it ensures high security and privacy.

More Control

The private cloud has more control on its resources and hardware than public cloud because it is accessed only within an organization.

Cost and Energy Efficiency

The private cloud resources are not as cost effective as resources in public clouds but they offer more efficiency than public cloud resources.

Disadvantages:

Here are some disadvantages of using private cloud model:

Restricted Area of Operation

The private cloud is only accessible locally and is very difficult to deploy globally.

High Priced

Purchasing new hardware in order to fulfill the demand is a costly transaction.

Limited Scalability

The private cloud can be scaled only within capacity of internal hosted resources.

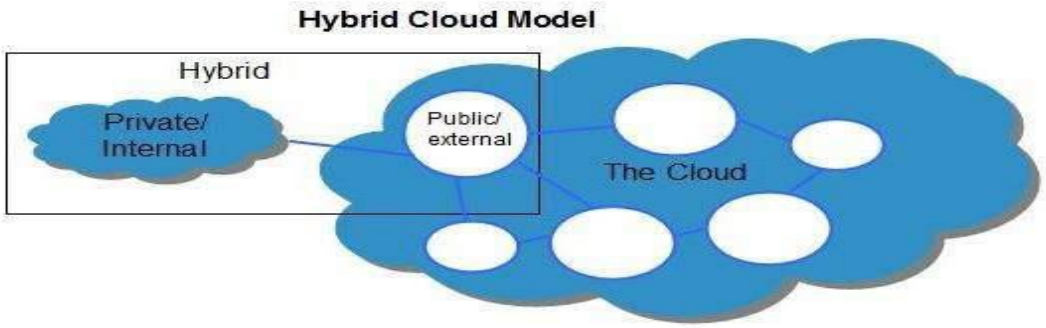
Additional Skills

In order to maintain cloud deployment, organization requires skilled expertise.

Hybrid Cloud:

Hybrid Cloud is a mixture of public and private cloud.

Non-critical activities are performed using public cloud while the critical activities are performed using private cloud. The Hybrid Cloud Model is shown in the diagram below.



Benefits:

There are many benefits of deploying cloud as hybrid cloud model. The following diagram shows some of those benefits:

Scalability

It offers features of both, the public cloud scalability and the private cloud scalability.

Flexibility

It offers secure resources and scalable public resources.

Cost Efficiency

Public clouds are more cost effective than private ones. Therefore, hybrid clouds can be cost saving.

Security

The private cloud in hybrid cloud ensures higher degree of security.

Disadvantages:

Networking Issues

Networking becomes complex due to presence of private and public cloud.

Security Compliance

It is necessary to ensure that cloud services are compliant with security policies of the organization.

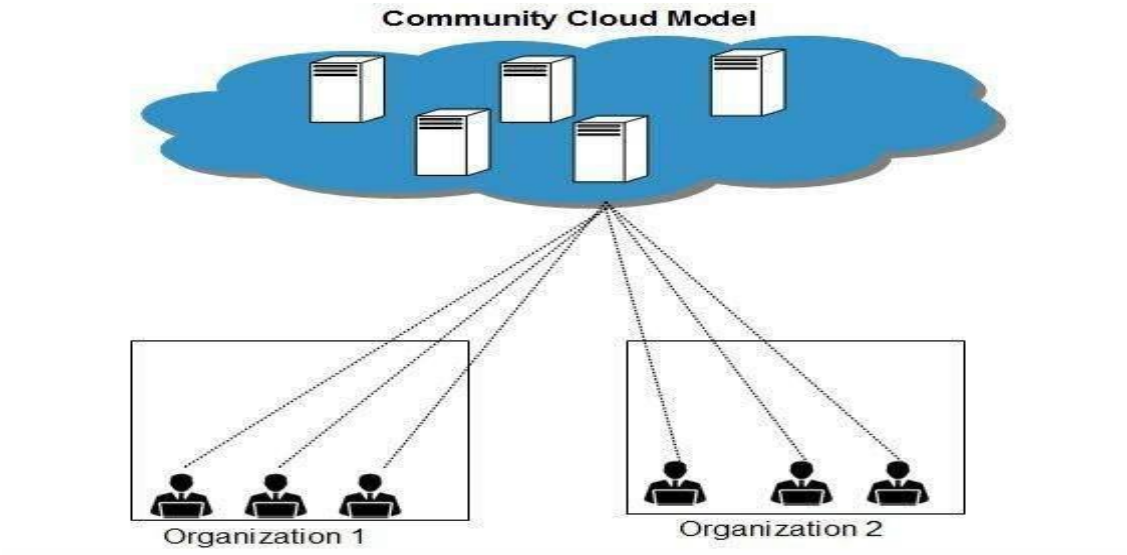
Infrastructure Dependency

The hybrid cloud model is dependent on internal IT infrastructure, therefore it is necessary to ensure redundancy across data centers.

Community Cloud:

Community Cloud allows system and services to be accessible by group of organizations. It shares the infrastructure between several organizations from a specific community.

It may be managed internally by organizations or by the third-party. The Community Cloud Model is shown in the diagram below.



Benefits:

There are many benefits of deploying cloud as community cloud model.

Cost Effective

Community cloud offers same advantages as that of private cloud at low cost.

Sharing Among Organizations

Community cloud provides an infrastructure to share cloud resources and capabilities among several organizations.

Security

The community cloud is comparatively more secure than the public cloud but less secured than the private cloud.

Issues:

- Since all data is located at one place, one must be careful in storing data in community cloud because it might be accessible to others.
- It is also challenging to allocate responsibilities of governance, security and cost among organizations.

Cloud computing architecture

Cloud computing architecture refers to the components and subcomponents required for cloud computing. These components typically consist of a front end platform (fat client, thin client, mobile device), back end platforms (servers, storage), a cloud based delivery, and a network (Internet, Intranet, Intercloud).

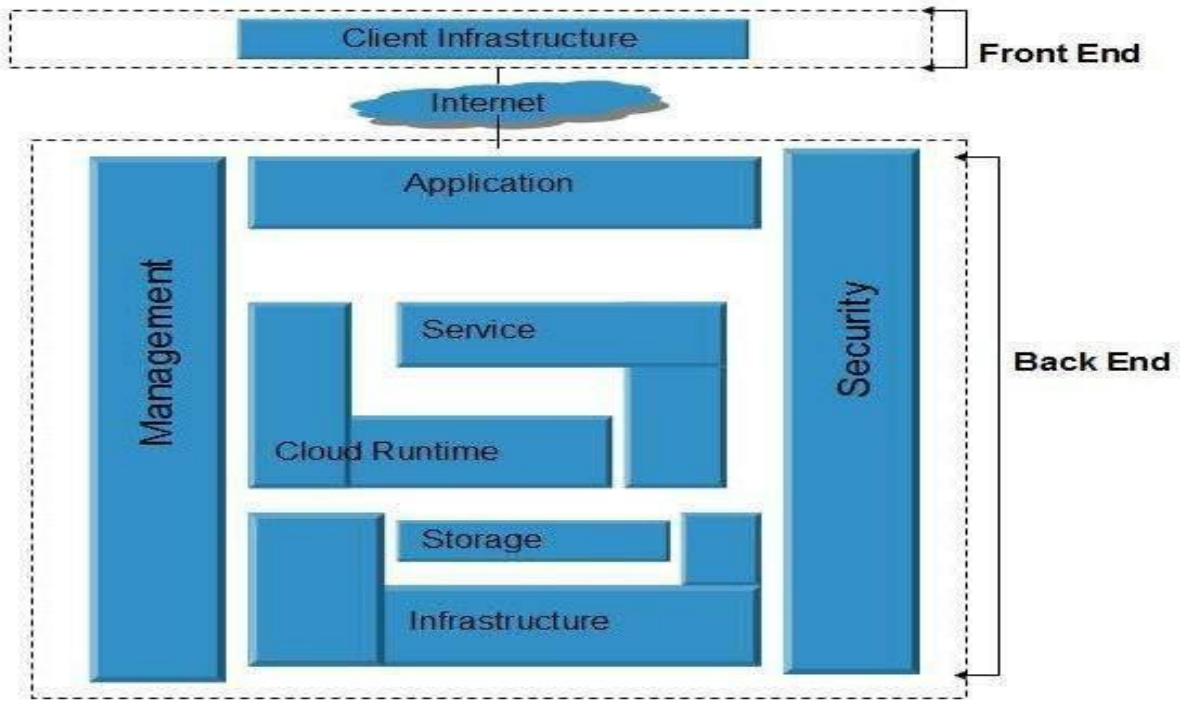
[A **fat client** (also called heavy, **rich** or **thick client**) is a computer (**clients**), in **client**–server architecture or networks, that typically provides **rich** functionality independent of the central server. It is Originally known as just a "client" or "thick client," the name is contrasted to thin client, which describes a computer heavily dependent on a server's applications.]

Cloud Computing architecture comprises of many cloud components, which are loosely coupled. We can broadly divide the cloud architecture into two parts:

- Front End
- Back End

Each of the ends is connected through a network, usually Internet.

The following diagram shows the graphical view of cloud computing architecture:



Front End:

The front end refers to the client part of cloud computing system. It consists of interfaces and applications that are required to access the cloud computing platforms, Example - Web Browser.

Back End:

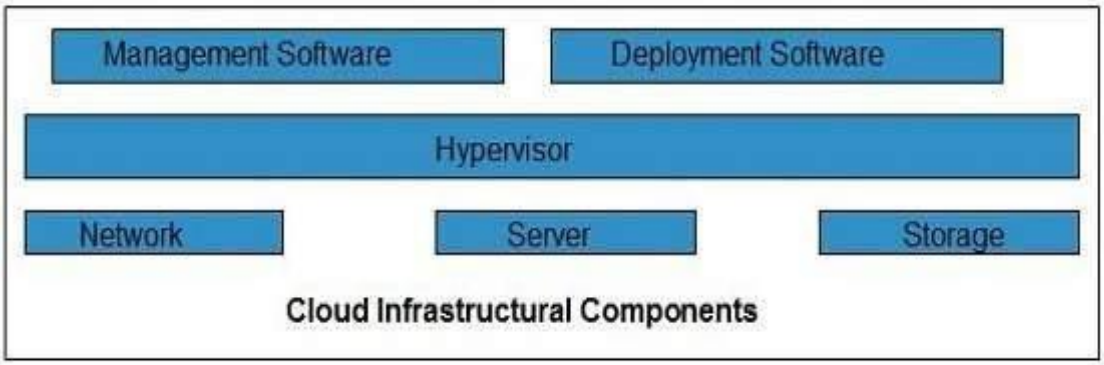
The back End refers to the cloud itself. It consists of all the resources required to provide cloud computing services. It comprises of huge data storage, virtual machines, security mechanism, services, deployment models, servers, etc.

Note:

- It is the responsibility of the back end to provide built-in security mechanism, traffic control and protocols.
- The server employs certain protocols known as middleware, which help the connected devices to communicate with each other.

Cloud Computing Infrastructure

Cloud infrastructure consists of servers, storage devices, network, cloud management software, deployment software, and platform virtualization.



Hypervisor:

Hypervisor is a firmware or low-level program that acts as a Virtual Machine Manager. It allows to share the single physical instance of cloud resources between several occupants.

Management Software:

It helps to maintain and configure the infrastructure.

Deployment Software:

It helps to deploy and integrate the application on the cloud.

Network:

It is the key component of cloud infrastructure. It allows to connect cloud services over the Internet. It is also possible to deliver network as a utility over the Internet, which means, the customer can customize the network route and protocol.

Server:

The **server** helps to compute the resource sharing and offers other services such as resource allocation and de-allocation, monitoring the resources, providing security etc.

Storage:

Cloud keeps multiple replicas of storage. If one of the storage resources fails, then it can be extracted from another one, which makes cloud computing more reliable.

Infrastructural Constraints

Fundamental constraints that cloud infrastructure should implement are shown in the following diagram:

Transparency:

Virtualization is the key to share resources in cloud environment. But it is not possible to satisfy the demand with single resource or server. Therefore, there must be transparency in resources, load balancing and application, so that we can scale them on demand.

Scalability:

Scaling up an application delivery solution is not that easy as scaling up an application because it involves configuration overhead or even re-architecting the network. So, application delivery solution is need to be scalable which will require the virtual infrastructure such that resource can be provisioned and de-provisioned easily.

Intelligent Monitoring:

To achieve transparency and scalability, application solution delivery will need to be capable of intelligent monitoring.

Security:

The mega data center in the cloud should be securely architected. Also the control node, an entry point in mega data center, also needs to be secure.

Virtualization

Virtualization in Cloud Computing is a process in which the user of cloud shares the data present in the cloud which can be application software etc. It provides a virtual environment in the cloud which can be software hardware or any other thing.

Virtualization is a technique how to separate a service from the underlying physical delivery of that service. It is the process of creating a virtual version of something like computer hardware.

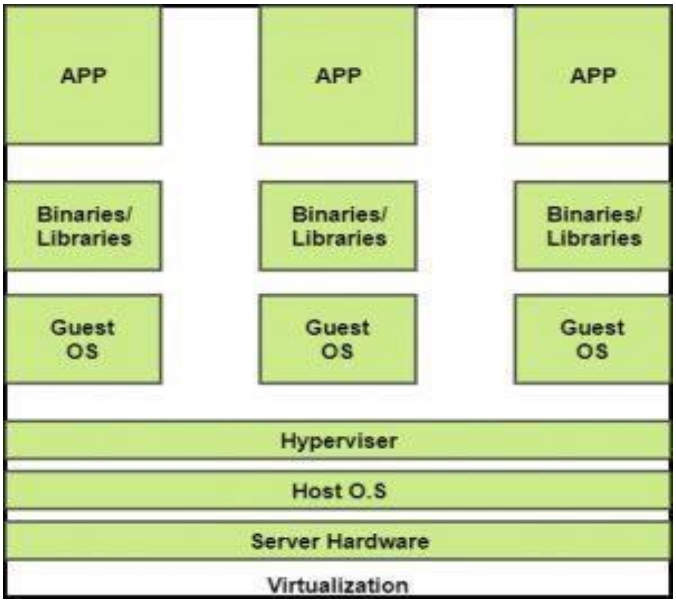
It involves using specialized software to create a virtual or software-created version of a computing resource rather than the actual version of the same resource.

With the help of Virtualization multiple operating systems and applications can run on same Machine and its same hardware at the same time increasing the utilization and flexibility of hardware.

In other words, One of the main cost effective, hardware reducing, energy saving techniques used by cloud providers is virtualization.

Virtualization is a technique, which allows to share a single physical instance of a resource or an application among multiple customers and organizations at one time. It does by assigning a logical name to a physical storage and providing a pointer to that physical resource on demand.

The term virtualization is often synonymous with hardware virtualization, which plays a fundamental role in efficiently delivering Infrastructure-as-a-Service (IaaS) solutions for cloud computing. Moreover, virtualization technologies provide a virtual environment for not only executing applications but also for storage, memory, and networking.



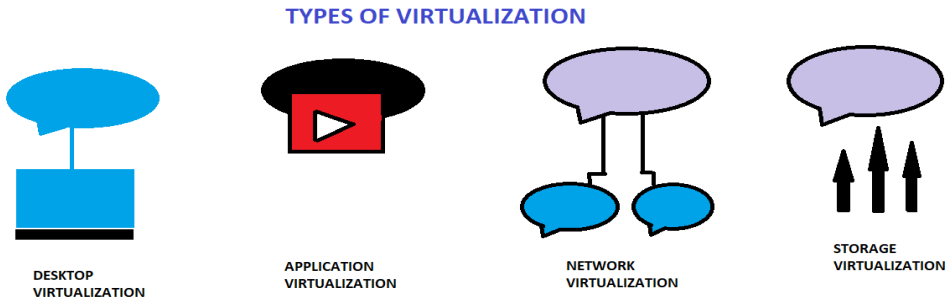
The machine on which the virtual machine is going to be build is known as Host Machine and that virtual machine is referred as a Guest Machine.

BENEFITS OF VIRTUALIZATION:

1. More flexible and efficient allocation of resources.
2. Enhance development productivity.
3. It lowers the cost of IT infrastructure.
4. Remote access and rapid scalibility.
5. High availability and disaster recovery.
6. Pay per use of the IT infrastructure on demand.
7. Enables running multiple operating system.

Types of Virtualization:

1. Application Virtualization.
2. Network Virtualization.
3. Desktop Virtualization.
4. Storage Virtualization.



1. Application Virtualization:

Application virtualization helps user to have a 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 internet. Example of this would be a user who needs to run two different versions of the same software. Technologies that use application virtualization are hosted applications and packaged applications.

2. Network Virtualization:

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

3. Desktop Virtualization:

Desktop virtualization allows the users' OS to be remotely stored on a server in the data center. It allows the user to access their desktop virtually, from any location by different machine. Users who wants specific operating systems other than Windows Server will need to have a virtual desktop. Main benefits of desktop virtualization are user mobility, portability, easy management of software installation, updates and patches.

4. 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. 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.

Hypervisor

Hypervisor is a form of virtualization software used in Cloud hosting to divide and allocate the resources on various pieces of hardware.

The program which provide partitioning, isolation or abstraction is called virtualization hypervisor.

Hypervisor is a hardware virtualization technique that allows multiple guest operating systems (OS) to run on a single host system at the same time.

A hypervisor is sometimes also called a virtual machine manager(VMM).

Types of Hypervisor:-

TYPE-1 Hypervisor:

Hypervisor runs directly on underlying host system. It is also known as "Native Hypervisor" or "Bare metal hypervisor". It dose not require any base server operating system. It has direct access to hardware resources.

TYPE-2 Hypervisor:

A Host operating system runs on undrlying host system. It is also known as 'Hosted Hypervisor".

Basically a software installed on an operating system. Hypervisor asks operating system to make hardware calls. Hosted hypervisors are often found on endpoints like PCs.

Choosing the right hypervisor:

Type 1 hypervisors offer much better performance than Type 2 because there's no middle layer, making them the logical choice for mission-critical applications and workloads.

But that's not to say that hosted hypervisors don't have their place – they're much simpler to set up, so they're a good.

One of the best ways to determine which hypervisor meets your needs is to compare their performance metrics. These include CPU overhead, amount of maximum host and guest memory, and support for virtual processors.

The following factors should be examined before choosing a suitable hypervisor:

1. Understand your needs: The company and its applications are the reason for the data centre (and your job). Besides your company's needs, you (and your co-workers in IT) also have your own needs.

Needs for a virtualization hypervisor are:

a. Flexibility
b. Scalability
c. Usability
d. Availability
e. Reliability
f. Efficiency
g. Reliable support

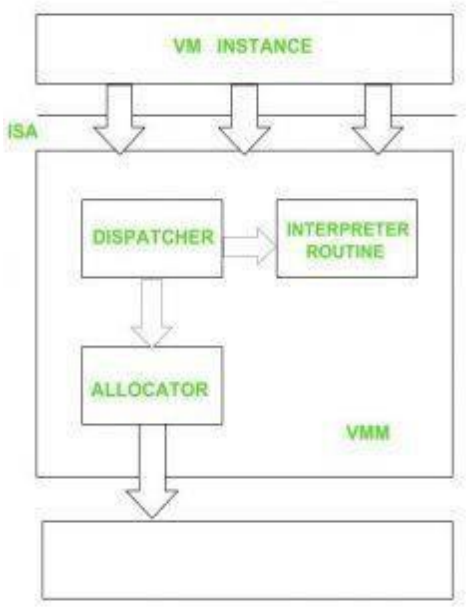
2. The cost of a hypervisor: For many buyers, the toughest part of choosing a hypervisor is striking the right balance between cost and functionality. While a number of entry-level solutions are free, or practically free, the prices at the opposite end of the market can be overwhelming. Licensing frameworks also vary, so it's important to be aware of exactly what you're getting for your money.

3. Virtual machine performance: Virtual systems should meet or exceed the performance of their physical counterparts, at least in relation to the applications within each server. Everything beyond meeting this benchmark is profit.

4. Ecosystem: It's tempting to overlook the role of a hypervisor's ecosystem – that is, the availability of documentation, support, training, third-party developers and consultancies, and so on – in determining whether or not a solution is cost-effective in the long term.

5. Test for yourself: You can gain basic experience from your existing desktop or laptop. You can run both Hypervisor to create a nice virtual learning and testing environment.

HYPERVISOR REFERENCE MODEL



There are 3 main modules coordinate in order to follow the fundamental hardware:

1. Dispatcher
2. Allocator
3. Interpreter

DISPATCHER:
The dispatcher behaves like the entry point of the monitor and reroutes the instructions of the virtual machine instance to one of the other two modules.

ALLOCATOR:
The allocator is responsible for deciding the system resources to be provided to the virtual machine instance. It means whenever virtual machine tries to execute an instruction that results in changing the machine resources associated with the virtual machine, the allocator is invoked by the dispatcher.

INTERPRETER:
The interpreter module consists of interpreter routines. These are executed, whenever virtual machine executes a privileged instruction.

CPU Virtualization

Virtualization of the hardware is known as CPU Virtualization. This is where any hardware platform that can be controlled by the user or a guest software over a virtual machine on a platform, virtually. This is not limited to guest software but also several operating systems.

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 attentive 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.

Memory Virtualization

Virtual memory virtualization is similar to the virtual memory support provided by modern operating 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 look aside 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.

Since each page table of the guest OS 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 performance overhead and cost of memory will be very high.

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.

I/O Virtualization

I/O virtualization involves managing the routing of I/O requests between virtual devices and the shared physical hardware.

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.

All the functions of a device or bus infrastructure, such as device enumeration, identification, interrupts, 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.

A single hardware device can be shared by multiple VMs that run concurrently. However, software emulation runs much slower than the hardware.

The para-virtualization method of I/O virtualization is typically used in Xen. [Xen Project is a type-1 hypervisor, providing services that allow multiple computer operating systems to execute on the same computer hardware concurrently.]

It is also known as the split driver model consisting of a front end driver and a backend driver. The frontend driver is running in Domain U and the backend driver 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 OS 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.

Direct I/O virtualization lets the VM access devices directly. It can achieve close-to-native performance without high CPU costs. However, current direct I/O virtualization implementations focus on networking for mainframes.

There are a lot of challenges for commodity hardware devices. For example, when a physical device is reclaimed (required by workload migration) for later reassignment, it may have been set to an arbitrary state that can function incorrectly or even crash the whole system.

Since software-based I/O virtualization requires a very high overhead of device emulation, hardware-assisted I/O virtualization is critical.

VIRTUAL CLUSTERS AND RESOURCE MANAGEMENT

A physical cluster is a collection of servers (physical machines) interconnected by a physical network such as a LAN.

When a traditional VM is initialized, the administrator needs to manually write configuration information or specify the configuration sources. When more VMs join a network, an inefficient configuration always causes problems with overloading or underutilization.

Amazon's Elastic Compute Cloud (EC2) is a good example of a web service that provides elastic computing power in a cloud. EC2 permits customers to create VMs and to manage user accounts over the time of their use.

Most virtualization platforms, including XenServer and VMware ESX Server, support a bridging mode which allows all domains to appear on the network as individual hosts. By using this mode, VMs can communicate with one another freely through the virtual network interface card and configure the network automatically.

Physical versus Virtual Clusters:-

Virtual clusters are built with VMs installed at distributed servers from one or more physical clusters.

The VMs in a virtual cluster are interconnected logically by a virtual network across several physical networks.

Each virtual cluster is formed with physical machines or a VM hosted by multiple physical clusters. The virtual cluster boundaries are shown as distinct boundaries.

The provisioning of VMs to a virtual cluster is done dynamically to have the following interesting properties:

- The virtual cluster nodes can be either physical or virtual machines. Multiple VMs running with different OS can be deployed on the same physical node.
- A VM runs with a guest OS, which is often different from the host OS, that manages the resources in the physical machine, where the VM is implemented.
- The purpose of using VMs is to consolidate multiple functionalities on the same server. This will greatly enhance server utilization and application flexibility.

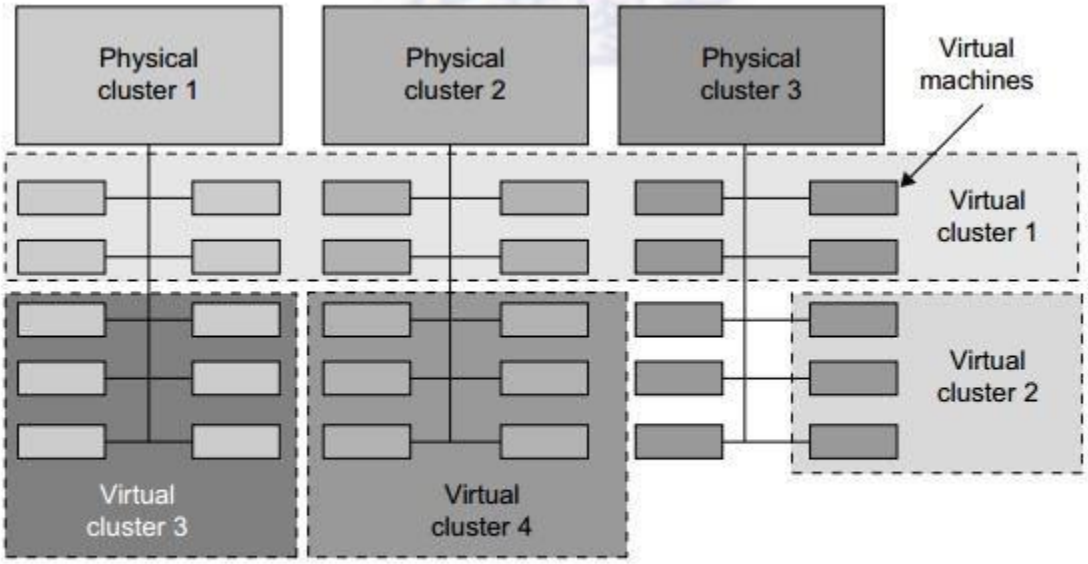


FIGURE 3.18

A cloud platform with four virtual clusters over three physical clusters shaded differently.

- VMs can be colonized (replicated) in multiple servers for the purpose of promoting distributed parallelism, fault tolerance, and disaster recovery.

- The size (number of nodes) of a virtual cluster can grow or shrink dynamically, similar to the way an overlay network varies in size in a peer-to-peer (P2P) network.

- The failure of any physical nodes may disable some VMs installed on the failing nodes. But the failure of VMs will not pull down the host system.

Fast Deployment and Effective Scheduling:

The system should have the capability of fast deployment. Here, deployment means two things: to construct and distribute software stacks (OS, libraries, applications) to a physical node inside clusters as fast as possible, and to quickly switch runtime environments from one user's virtual cluster to another user's virtual cluster.

If one user finishes using his system, the corresponding virtual cluster should shut down or suspend quickly to save the resources to run other VMs for other users.

The concept of "green computing" has attracted much attention recently. However, previous approaches have focused on saving the energy cost of components in a single workstation without a global vision. Consequently, they do not necessarily reduce the power consumption of the whole cluster.

The live migration of VMs allows workloads of one node to transfer to another node. However, it does not guarantee that VMs can randomly migrate among themselves. In fact, the probable overhead caused by live migrations of VMs cannot be ignored.

The overhead may have serious negative effects on cluster utilization and throughput issues. Therefore, the challenge is to determine how to design migration strategies to implement green computing without influencing the performance of clusters.

Another advantage of virtualization is load balancing of applications in a virtual cluster. Load balancing can be achieved using the load index and frequency of user logins. The automatic scale-up and scale-down mechanism of a virtual cluster can be implemented based on this model. Consequently, we can increase the resource utilization of nodes and shorten the response time of systems.

Mapping VMs onto the most appropriate physical node should promote performance. Dynamically adjusting loads among nodes by live migration of VMs is desired, when the loads on cluster nodes become quite unbalanced.

High-Performance Virtual Storage:

The template VM can be distributed to several physical hosts in the cluster to customize the VMs. In addition, existing software packages reduce the time for customization as well as switching virtual environments.

It is important to efficiently manage the disk spaces occupied by template software packages. Some storage architecture design can be applied to reduce duplicated blocks in a distributed file system of virtual clusters.

Basically, there are four steps to deploy a group of VMs onto a target cluster: preparing the disk image, configuring the VMs, choosing the destination nodes, and executing the VM deployment command on every host.

Many systems use templates to simplify the disk image preparation process. A template is a disk image that includes a preinstalled operating system with or without certain application software.

Users choose a proper template according to their requirements and make a duplicate of it as their own disk image. Templates could implement the COW (Copy on Write) format. A new COW backup file is very small and easy to create and transfer. Therefore, it definitely

reduces disk space consumption. In addition, VM deployment time is much shorter than that of copying the whole raw image file.
Every VM is configured with a name, disk image, network setting, and allocated CPU and memory. One needs to record each VM configuration into a file. However, this method is inefficient when managing a large group of VMs.

Live VM Migration Steps and Performance Effects:-

In a cluster built with mixed nodes of host and guest systems, the normal method of operation is to run everything on the physical machine. When a VM fails, its role could be replaced by another VM on a different node, as long as they both run with the same guest OS.

In other words, a physical node can fail over to a VM on another host. This is different from physical-to-physical failover in a traditional physical cluster. The advantage is enhanced failover flexibility.

The potential drawback is that a VM must stop playing its role if its residing host node fails. However, this problem can be mitigated with VM life migration. The migration copies the VM state file from the storage area to the host machine.

There are four ways to manage a virtual cluster:

First, you can use a guest-based manager, by which the cluster manager resides on a guest system. In this case, multiple VMs form a virtual cluster. For example, openMosix is an open source Linux cluster running different guest systems on top of the Xen hypervisor. Another example is Sun's cluster Oasis, an experimental Solaris cluster of VMs supported by a VMware VMM.

Second, you can build a cluster manager on the host systems. The host-based manager supervises the guest systems and can restart the guest system on another physical machine. A good example is the VMware HA system that can restart a guest system after failure. These two cluster management systems are either guest-only or host-only, but they do not mix.

A **third way** to manage a virtual cluster is to use an independent cluster manager on both the host and guest systems. This will make infrastructure management more complex, however.

Finally, you can use an integrated cluster on the guest and host systems. This means the manager must be designed to distinguish between virtualized resources and physical resources. Various cluster management schemes can be greatly enhanced when VM life migration is enabled with minimal overhead.

VMs can be live-migrated from one physical machine to another; in case of failure, one VM can be replaced by another VM. Virtual clusters can be applied in computational grids, cloud platforms, and high-performance computing (HPC) systems.

The major attraction of this scenario is that virtual clustering provides dynamic resources that can be quickly put together upon user demand or after a node failure. In particular, virtual clustering plays a key role in cloud computing.

When a VM runs a live service, it is necessary to make an exchange to ensure that the migration occurs in a manner that minimizes all three metrics.

Furthermore, we should ensure that the migration will not disrupt other active services residing in the same host through resource contention (e.g., CPU, network bandwidth).

A VM can be in one of the following four states:

An **inactive state** is defined by the virtualization platform, under which the VM is not enabled.

An **active state** refers to a VM that has been instantiated at the virtualization platform to perform a real task.

A **paused state** corresponds to a VM that has been instantiated but disabled to process a task or paused in a waiting state.

A VM enters the **suspended state** if its machine file and virtual resources are stored back to the disk.

Llive migration of a VM consists of the following six steps:

Steps 0 and 1: Start migration.

This step makes preparations for the migration, including determining the migrating VM and the destination host. Although users could manually make a VM migrate to an appointed host, in most circumstances, the migration is automatically started by strategies such as load balancing and server consolidation.

Steps 2: Transfer memory.

Since the whole execution state of the VM is stored in memory, sending the VM's memory to the destination node ensures continuity of the service provided by the VM. All of the memory data is transferred in the first round, and then the migration controller recopies the memory data which is changed in the last round. These steps keep iterating until the dirty portion of the memory is small enough to handle the final copy.

Step 3: Suspend the VM and copy the last portion of the data.

The migrating VM's execution is suspended when the last round's memory data is transferred. Other non memory data such as CPU and network states should be sent as well. During this step, the VM is stopped and its applications will no longer run. This "service unavailable" time is called the "downtime" of migration, which should be as short as possible so that it can be negligible to users.

Steps 4 and 5: Commit and activate the new host.

After all the needed data is copied, on the destination host, the VM reloads the states and recovers the execution of programs in it, and the service provided by this VM continues. Then the network connection is redirected to the new VM and the dependency to the source host is cleared. The whole migration process finishes by removing the original VM from the source host.

Migration of Memory, Files, and Network Resources:-

Since clusters have a high initial cost of ownership, including space, power conditioning, and cooling equipment, leasing or sharing access to a common cluster is an attractive solution when demands vary over time.

Shared clusters offer economies of scale and more effective utilization of resources by multiplexing. When one system migrates to another physical node, we should consider the following issues.

Memory Migration:

This is one of the most important aspects of VM migration. Moving the memory instance of a VM from one physical host to another can be approached in any number of ways. But traditionally, the concepts behind the techniques tend to share common implementation paradigms. The techniques employed for this purpose depend upon the characteristics of application/workloads supported by the guest OS.

Memory migration can be in a range of hundreds of megabytes to a few gigabytes in a typical system today, and it needs to be done in an efficient manner.

File System Migration:

To support VM migration, a system must provide each VM with a consistent, location-independent view of the file system that is available on all hosts. A simple way to achieve this is to provide each VM with its own virtual disk which the file system is mapped to and transport the contents of this virtual disk along with the other states of the VM. However, due to the current trend of high-capacity disks, migration of the contents of an entire disk over a network is not a viable solution.

Another way is to have a global file system across all machines where a VM could be located. This way removes the need to copy files from one machine to another because all files are network-accessible.

The relevant VM files are explicitly copied into the local file system for a resume operation and taken out of the local file system for a suspend operation. This approach relieves developers from the complexities of implementing several different file system calls for different distributed file systems.

Network Migration:

A migrating VM should maintain all open network connections without relying on forwarding mechanisms on the original host or on support from mobility or redirection mechanisms.

To enable remote systems to locate and communicate with a VM, each VM must be assigned a virtual IP address known to other entities. This address can be distinct from the IP address of the host machine where the VM is currently located.

Each VM can also have its own distinct virtual MAC address. The VMM maintains a mapping of the virtual IP and MAC addresses to their corresponding VMs.

In general, a migrating VM includes all the protocol states and carries its IP address with it.

Live migration means moving a VM from one physical node to another while keeping its OS environment and applications unbroken. This capability is being increasingly utilized in today's enter-prise environments to provide efficient online system maintenance, reconfiguration, load balancing, and proactive fault tolerance.

It provides desirable features to satisfy requirements for computing resources in modern computing systems, including server consolidation, performance isolation, and ease of management.

Live migration is a key feature of system virtualization technologies. Only memory and CPU status needs to be transferred from the source node to the target node.