UNIT-IV-CLOUD INFRASTRUCTURE

Hardware and Infrastructure – Thick and thin clients - Architectural Design – Layered Cloud Architecture Development – Design Challenges - Inter Cloud Resource Management – Resource Provisioning and Platform Deployment – Virtualization – VMware.

Hardware and infrastructure:

What is Cloud Computing Infrastructure?

Cloud computing infrastructure typically refers to the hardware and software that enables cloud computing. Hardware and software include storage devices, processors, networking equipment, operating systems, user interfaces, and applications. Infrastructure is generally divided into two parts:

1. Frontend cloud computing infrastructure consists of applications, machines, and user interfaces. Clients then use it to interact with the cloud.
2. Backend cloud computing infrastructure comprises all the hardware, storage, operating systems, networking logic, and security mechanisms that support the frontend infrastructure.

Cloud Infrastructure which comes under the backend part of cloud architecture represents the hardware and software component such as server, storage, networking, management software, deployment software and virtualization software etc. In backend, cloud infrastructure enables the complete cloud computing system.

Why Cloud Computing Infrastructure:

Cloud computing refers to providing on demand services to the customer anywhere and anytime irrespective of everything where the cloud infrastructure represents the one who activates the complete cloud computing system. Cloud infrastructure has more capabilities of providing the same services as the physical infrastructure to the customers. It is available for private cloud, public cloud, and hybrid cloud systems with low cost, greater flexibility and scalability.

Cloud infrastructure components:

Different components of cloud infrastructure support the computing requirements of a cloud computing model. Cloud infrastructure has number of key components but not limited to only server, software, network and storage devices.

First, frontend cloud computing infrastructure includes:

1) Graphical User Interface (GUI)

2) Client-Side Software

3) Client-Side Hardware

Graphical User Interface (GUI)

The GUI gives clients access to their cloud services and workloads. This may be a SaaS application such as Gmail or Microsoft 365. It could also be a web portal used to access IaaS resources or cloud software that the client developed themselves.

Client-Side Software

Client-side software refers to a local application installed on the user's device, which is used to access cloud services. One example would be using the Outlook desktop client to access email hosted in Microsoft 365. Another would be using a web browser (e.g., Chrome or Firefox) to access cloud-based EDA tools.

Client-Side Hardware

The client-side hardware includes the networking equipment (like routers and switches) used to connect users to the internet to access cloud services. Additionally, it includes the physical machines that run the client-side software (like laptops and smartphones).

Frontend cloud infrastructure is often referred to as "client-side infrastructure" because most of it (laptops, internet connection, and desktop software) is the client's responsibility to purchase and maintain. However, clients aren't always responsible for the GUI, especially in the case of SaaS cloud computing.

On the backend, cloud computing infrastructure includes:

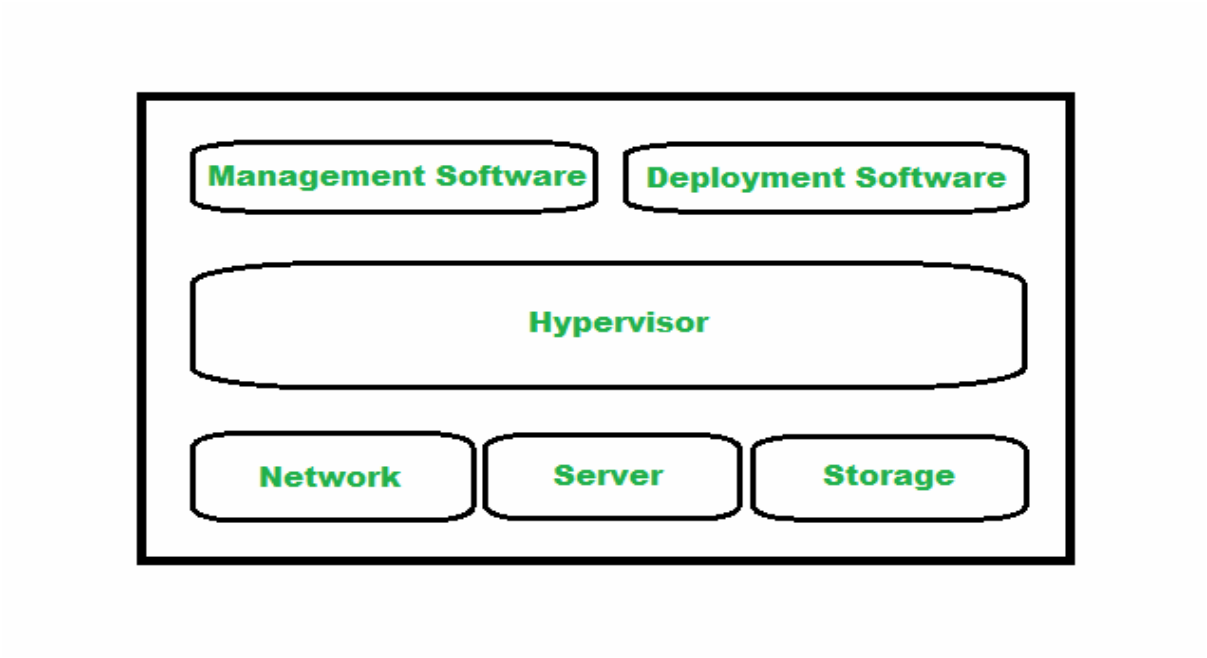
1) Cloud Hardware

2) Virtualization

3) Storage

4) Networking

5)Security



Components of Cloud Infrastructure

Cloud Hardware

When most people think about cloud infrastructure, they think about the underlying hardware on which everything runs. This infrastructure includes the physical servers, storage devices, processors, routers, switches, load balancers, and power distribution units (PDUs) that cloud providers maintain. This hardware resides in provider-controlled data centers. The best cloud vendors have many of these data centers distributed around the world for redundancy and global performance.

Hypervisor :

Hypervisor is a firmware or a low level program which is a key to enable virtualization. It is used to divide and allocate cloud resources between several customers. As it monitors and manages cloud services/resources that's why

hypervisor is called as VMM (Virtual Machine Monitor) or (Virtual Machine Manager).

2. Management Software :

Management software helps in maintaining and configuring the infrastructure. Cloud management software monitors and optimizes resources, data, applications and services.

3. Deployment Software :

Deployment software helps in deploying and integrating the application on the cloud. So, typically it helps in building a virtual computing environment.

4. Network :

It is one of the key component of cloud infrastructure which is responsible for connecting cloud services over the internet. For the transmission of data and resources externally and internally network is must required.

In addition to the physical networking devices, cloud computing infrastructure relies on networking logic such as routing and load balancing. This logic may be tied to physical devices. It may also be virtualized using network function virtualization (NFV) or software-defined networking (SDN). Virtualization makes it easier for providers to manage and optimize the large and complex network architectures required to deliver cloud services.

5. Server :

Server which represents the computing portion of the cloud infrastructure is responsible for managing and delivering cloud services for various services and partners, maintaining security etc.

6. Storage :

Storage represents the storage facility which is provided to different organizations for storing and managing data. It provides a facility of extracting another resource if one of the resource fails as it keeps many copies of storage.

Cloud storage is another form of virtualization because the storage capacity available to clients is decoupled from the underlying storage hardware. That means an end-user can easily scale their storage capacity up and down on-

demand without worrying about buying and installing additional storage devices. On the provider's end, public cloud storage virtualization means they can distribute client data across whatever storage hardware is available, even in multiple data centers.

Along with this, virtualization is also considered as one of important component of cloud infrastructure. Because it abstracts the available data storage and computing power away from the actual hardware and the users interact with their cloud infrastructure through GUI (Graphical User Interface).

Virtualization

Virtualization decouples computing functions and services from the underlying hardware. This process allows providers to host platforms and software for multiple clients on shared hardware without anyone seeing or accessing each other's services. In the case of IaaS offerings, virtualization also gives users the ability to manage their cloud infrastructure through a user interface without accessing the physical hardware.

Security

Security infrastructure includes things like firewalls, access control, and malware prevention. Cloud security follows what's known as the shared responsibility model, meaning the duty of protecting cloud computing infrastructure is shared between the client and the provider. The provider must secure the physical infrastructure (using things like security cameras and door locks), the network infrastructure, the storage and computing systems, and their applications. Clients must secure the systems and infrastructure they use to access cloud services and the applications they develop and host on cloud infrastructure.

Thick and thin clients

Thin Clients

A thin client is a computer that uses resources from a central server rather than a local hard drive. Thin clients connect to a server-based computing environment via a remote connection, where most applications, sensitive data, and memory are stored.

- The majority of the work is done by the server, including starting software programmes, doing calculations, and storing data. Thin clients are part of a larger computing infrastructure in which multiple clients share computations with a single server or server farm.
- A thin client can be used in three ways; shared terminal services, desktop virtualisation, and browser-based. A graphical user interface (GUI), cloud access agents, a local web browser, terminal emulators, and a minimal set of local utilities are typical client software components.
- While the server must be capable of handling multiple client sessions simultaneously, thin client hardware requirements are minimal compared to a standard PC desktop. The majority of thin clients are equipped with low-power processors, flash storage, memory, and no moving parts.

Popular providers of thin clients include Wyse Technology, NComputing, Dell, HP and Samsung Electronics.

Pros of thin clients

Lower costs

Thin clients are less likely to break down because they have fewer internal parts than a regular computer. They have no hard drives and typically use less powerful processors, resulting in a lower cost per device. Connecting and setting up thin clients takes less effort and time, reducing IT costs.

Better security

Thin client devices are more resistant to malware because users cannot install programs or store files on their devices. Thin clients ensure that only trusted software is installed on the computer.

Minimal space requirements

Thin clients are small machines that don’t need powerful hardware as demanding processing happens at the server.

Protected data

Organizations can control who can access and use centrally stored files. In particular, thin client systems can be configured so that devices can access files without copying or deleting them.

Streamlined management

Installing new software, patching an application or operating system, or upgrading a network only requires work on the server instead of each client. Also, because user files are stored centrally, files can be found by searching in a single location. Thin clients make it easier to install software and ensure computers are up to date.

Effortless scaling

To scale a server-based system, all that needs to be done is to add a thin client and connect it to the server.

More efficient

Thin clients have a lower carbon footprint than regular computers because they have fewer moving parts. With simpler processors and no hard drives, thin clients also require less power and generate less waste heat.

Cons of thin clients

Single point of failure

Because everything a thin client does is delivered over a network connection, the network becomes a single point of failure and the most significant performance bottleneck in the system. If the network slows down, experiences delays, or fails completely, thin clients can do anything from delay to stop working altogether.

Requires powerful servers

Thin clients rely on powerful servers to do their jobs, and the entire system suffers when they don’t have enough power or performance. Even if a company already has servers, upgrading to handle thin client loads requires specialized hardware, which is a substantial initial investment.

Possible network problems

Thin clients cannot run without connecting to the server. If most employees’ work is done through thin clients, they can be affected by network latency, even on local computers.

Not ideal for power users

For users who only occasionally need to check email or access web content, thin clients will meet their needs without difficulty. But engineers, graphic designers, and others who frequently work with multimedia content or graphics-intensive software may be limited by thin clients.

Thick Clients

A thick client is a computing workstation that comprises most or all of the components required to operate and execute software applications independently.

- A thick client is a client-server computing component linked to the server via a network connection but does not use any of the server's computing resources to run applications. A thick client is also characterised as a fat, heavy, or wealthy client.
- A thick client is a client device with the bulk of hardware resources on board in a client-server architecture. It can execute computations, run applications, and perform other functions independently.
- Although a thick client is capable of performing most tasks, it still requires a connection to the primary server to download programmes and data and update the operating system. Therefore, they are generally implemented in computing environments when the primary servers have low network speeds, limited computing, or a need to work offline.

Some of its benefits include lower server requirements, working offline, more flexibility and better multimedia performance.

Pros of thick clients

Better offline performance

Thick clients can work offline because they don’t need to connect to a central server constantly. Thick clients typically have the hardware and software required to work on demand and sometimes do not need to connect to a central server at all.

More server capacity

Thick clients allow access to files and applications anytime, providing more server capacity for other tasks. Since servers receive a lower workload from each individual client, they can serve more clients.

Less costly server needs

The final performance of an application depends on the server it connects to. Since thick clients do most of the processing locally, they are cheaper and perform better because they don't have complex needs.

Better individual performance

Any resource or bandwidth-intensive application should perform well because resources are acquired from a single computer, not allocated from a central server, enabling work with multimedia content or graphics-intensive software.

No costly upgrades

Many organizations may already have local computers fast enough to implement an infrastructure that can easily run thick clients.

Cons of thick clients

Continuous maintenance

Every computer needs maintenance. Thick clients must be updated for security or any hardware and software fixes and necessary updates of each software program. Each computer's hardware must be maintained at a level acceptable to its software applications.

Demanding software changes

Because new software requires more resources, every computer that uses the application must be updated. New applications that employees may need may also need to be uploaded to other workstations.

Data security problems

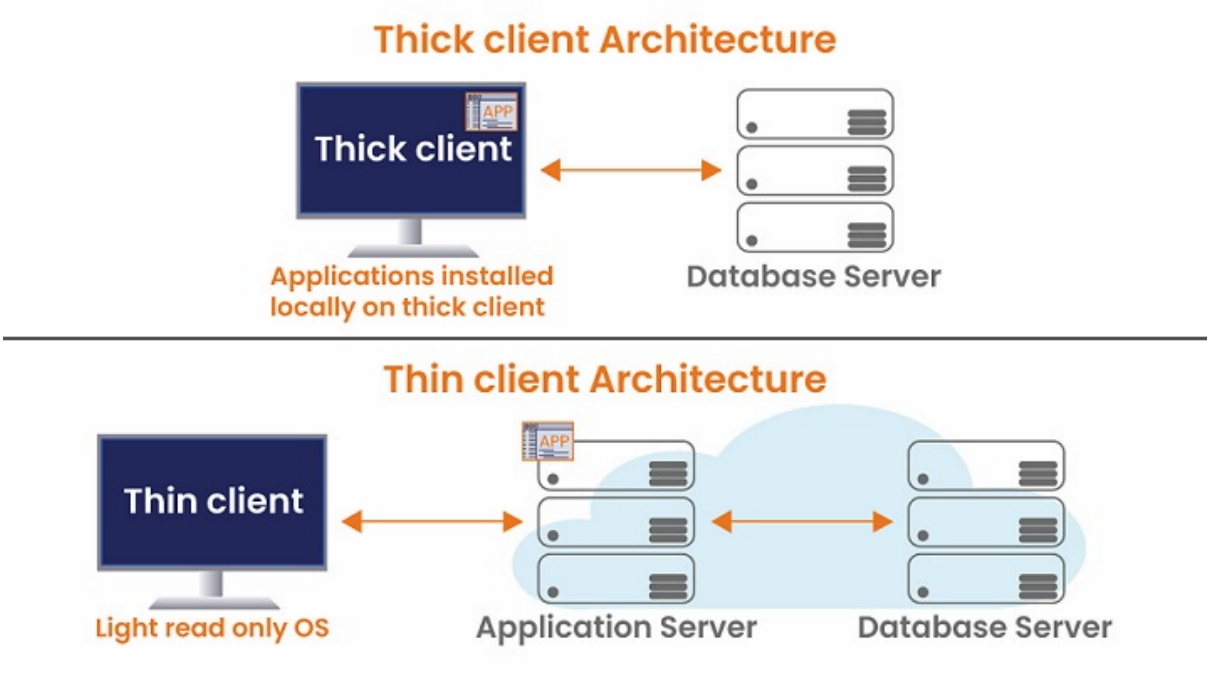
Data needs to be backed up from thick clients to ensure that if something goes wrong, information isn't lost forever. Thick clients also increase the responsibility of individuals for the security and protection of their computers.

Higher cost per unit

Organizations must continually invest in each workstation to ensure the hardware can run the latest version of the software.

High network traffic

Since each workstation handles all the data locally for each user over the network cable, there is usually a lot of network traffic. In modern networks, this may not be a problem. Still, when there is a considerable amount of data or multiple physical locations to communicate, the bandwidth capacity of the network may not be able to transfer all the necessary data quickly.



Difference between thin and thick clients

Thin Clients	**Thick Clients**
They are used by handheld devices	Customization systems use thick clients
They have browser-based installation	They are installed locally.
Easily deployable	More expensive to deploy
Data is typically stored on servers	More expensive to deploy
Designed to communicate with a server	It implements its own features

Architectural design

Layered cloud architecture development

The architecture of a cloud is developed at three layers:

Infrastructure

Platform

application.

These three development layers are implemented with virtualization and standardization of hardware and software resources provisioned in the cloud. The services to public, private, and hybrid clouds are conveyed to users through networking support over the Internet and intranets involved.

The infrastructure layer is deployed first to support IaaS services. The infrastructure layer is built with virtualized compute, storage, and network resources. The abstraction of these hardware resources is meant to provide the flexibility demanded by users. Internally, virtualization realizes automated provisioning of resources and optimizes the infrastructure management process. This infrastructure layer serves as the foundation for building the platform layer of the cloud for supporting PaaS services.

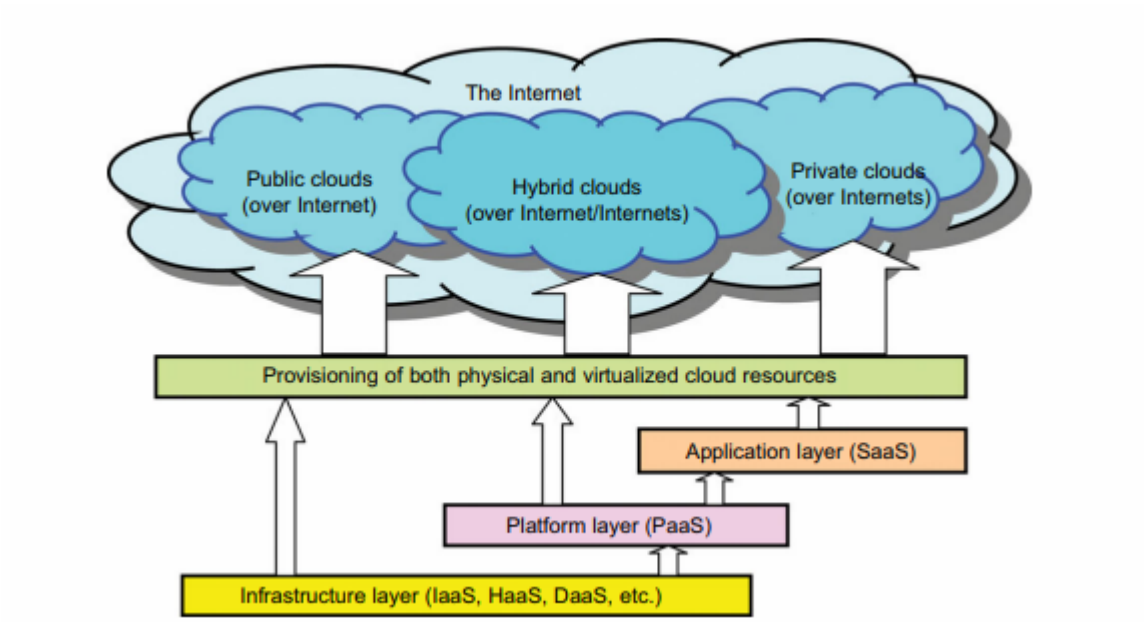
The platform layer is a foundation for implementing the application layer for SaaS applications. Different types of cloud services demand application of these resources separately. The platform layer is for general-purpose and repeated usage of the collection of software resources. This layer provides users with an environment to develop their applications, to test operation flows, and to monitor execution results and performance. The platform should be able to assure users that they have scalability, dependability, and security protection. In a way, the virtualized cloud platform serves as a “system middleware” between the infrastructure and application layers of the cloud.

The application layer is formed with a collection of all needed software modules for SaaS applications. Service applications in this layer include daily office management work, such as information retrieval, document processing, and calendar and authentication services. The application layer is also heavily

used by enterprises in business marketing and sales, consumer relationship management (CRM), financial transactions, and supply chain management.

All cloud services are restricted to a single layer. Many applications may apply resources at mixed layers. After all, the three layers are built from the bottom up with a dependence relationship. From the provider's perspective, the services at various layers demand different amounts of functionality support and resource management by providers. In general, SaaS demands the most work from the provider, PaaS is in the middle, and IaaS demands the least.

For example, Amazon EC2 provides not only virtualized CPU resources to users, but also management of these provisioned resources. Services at the application layer demand more work from providers. The best example of this is the Salesforce.com CRM service, in which the provider supplies not only the hardware at the bottom layer and the software at the top layer, but also the platform and software tools for user application development and monitoring.

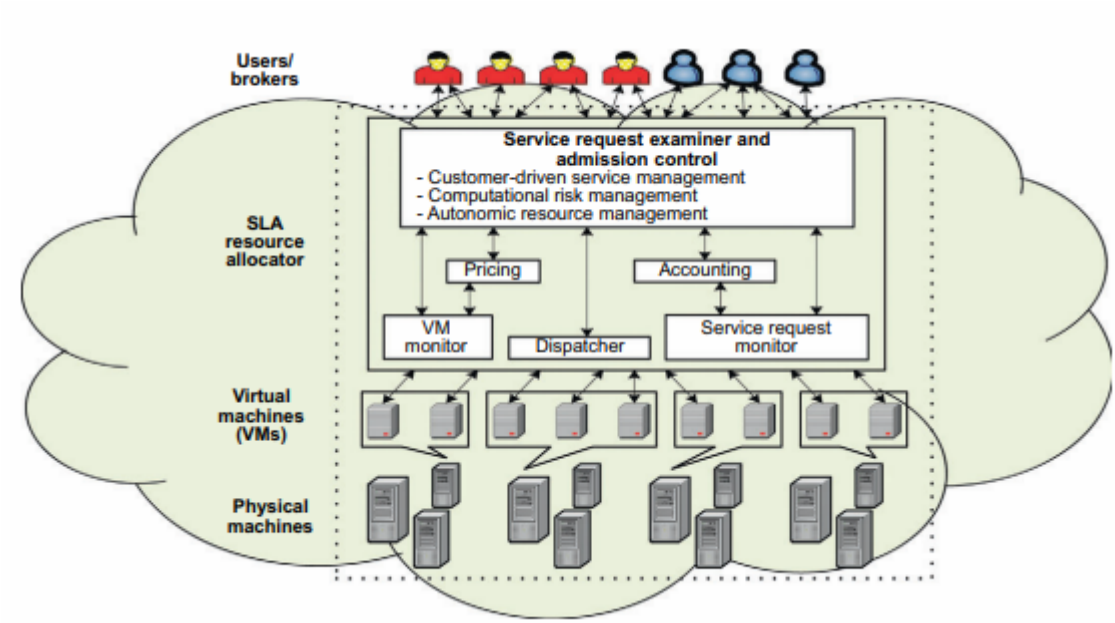


Layered architectural development of the cloud platform for IaaS, PaaS, and SaaS applications over the Internet.

Market-Oriented Cloud Architecture

As consumers rely on cloud providers to meet more of their computing needs, they will require a specific level of QoS to be maintained by their providers, in order to meet their objectives and sustain their operations. Cloud providers consider and meet the different QoS parameters of each individual consumer as negotiated in specific SLAs. To achieve this, the providers cannot deploy

traditional system-centric resource management architecture. Instead, market-oriented resource management is necessary to regulate the supply and demand of cloud resources to achieve market equilibrium between supply and demand. The designer needs to provide feedback on economic incentives for both consumers and providers. The purpose is to promote QoS-based resource allocation mechanisms. In addition, clients can benefit from the potential cost reduction of providers, which could lead to a more competitive market, and thus lower prices. Figure below shows the high-level architecture for supporting market-oriented resource allocation in a cloud computing environment.



Market-oriented cloud architecture to expand/shrink leasing of resources with variation in QoS/demand from users

This cloud is basically built with the following entities:

Users or brokers acting on user's behalf submit service requests from anywhere in the world to the data center and cloud to be processed.

The SLA resource allocator acts as the interface between the data center/cloud service provider and external users/brokers. It requires the interaction of the following mechanisms to support SLA-oriented resource management. When a service request is first submitted the service request examiner interprets the submitted request for QoS requirements before determining whether to accept or reject the request.

The request examiner ensures that there is no overloading of resources whereby many service requests cannot be fulfilled successfully due to limited resources. It also needs the latest status information regarding resource

availability (from the VM Monitor mechanism) and workload processing (from the Service Request Monitor mechanism) in order to make resource allocation decisions effectively. Then it assigns requests to VMs and determines resource entitlements for allocated VMs.

The Pricing mechanism decides how service requests are charged. For instance, requests can be charged based on submission time (peak/off-peak), pricing rates (fixed/changing), or availability of resources (supply/demand). Pricing serves as a basis for managing the supply and demand of computing resources within the data center and facilitates in prioritizing resource allocations effectively.

The Accounting mechanism maintains the actual usage of resources by requests so that the final cost can be computed and charged to users. In addition, the maintained historical usage information can be utilized by the Service Request Examiner and Admission Control mechanism to improve resource allocation decisions.

The VM Monitor mechanism keeps track of the availability of VMs and their resource entitlements. The Dispatcher mechanism starts the execution of accepted service requests on allocated VMs.

The Service Request Monitor mechanism keeps track of the execution progress of service requests. Multiple VMs can be started and stopped on demand on a single physical machine to meet accepted service requests, hence providing maximum flexibility to configure various partitions of resources on the same physical machine to different specific requirements of service requests.

In addition, multiple VMs can concurrently run applications based on different operating system environments on a single physical machine since the VMs are isolated from one another on the same physical machine.

Quality of Service Factors

The data center comprises multiple computing servers that provide resources to meet service demands. In the case of a cloud as a commercial offering to enable crucial business operations of companies, there are critical QoS parameters to consider in a service request, such as time, cost, reliability, and trust/security. QoS requirements cannot be static and may change over time due to continuing changes in business operations and operating environments. There should be greater importance on customers since they pay to access

services in clouds. In addition, the state of the art in cloud computing has no or limited support for dynamic negotiation of SLAs between participants and mechanisms for automatic allocation of resources to multiple competing requests. Negotiation mechanisms are needed to respond to alternate offers protocol for establishing SLAs. Commercial cloud offerings must be able to support customer-driven service management based on customer profiles and requested service requirements. Commercial clouds define computational risk management tactics to identify, assess, and manage risks involved in the execution of applications with regard to service requirements and customer needs. The cloud also derives appropriate market-based resource management strategies that encompass both customer-driven service management and computational risk management to sustain SLA-oriented resource allocation. The system incorporates autonomic resource management models that effectively self-manage changes in service requirements to satisfy both new service demands and existing service obligations, and leverage VM technology to dynamically assign resource shares according to service requirements.

Design Challenges

Architectural Design Challenges

In this section, we will identify six open challenges in cloud architecture development.

Challenge 1—Service Availability and Data Lock-in Problem

The management of a cloud service by a single company is often the source of single points of failure. To achieve HA, one can consider using multiple cloud providers. Even if a company has multiple data centers located in different geographic regions, it may have common software infrastructure and accounting systems. Therefore, using multiple cloud providers may provide more protection from failures. Another availability obstacle is distributed denial of service (DDoS) attacks. Criminals threaten to cut off the incomes of SaaS providers by making their services unavailable. Some utility computing services offer SaaS providers the opportunity to defend against DDoS attacks by using quick scale-ups. Software stacks have improved interoperability among different cloud platforms, but the APIs itself are still proprietary. Thus, customers cannot easily extract their data and programs from one site to run on another. The obvious solution is to standardize the APIs so that a SaaS developer can deploy services and data across multiple cloud providers. This

will rescue the loss of all data due to the failure of a single company. In addition to mitigating data lock-in concerns, standardization of APIs enables a new usage model in which the same software infrastructure can be used in both public and private clouds. Such an option could enable "surge computing," in which the public cloud is used to capture the extra tasks that cannot be easily run in the data center of a private cloud

Challenge 2—Data Privacy and Security Concerns

Current cloud offerings are essentially public (rather than private) networks, exposing the system to more attacks. Many obstacles can be overcome immediately with well-understood technologies such as encrypted storage, virtual LANs, and network middleboxes (e.g., firewalls, packet filters). For example, you could encrypt your data before placing it in a cloud. Many nations have laws requiring SaaS providers to keep customer data and copyrighted material within national boundaries. Traditional network attacks include buffer overflows, DoS attacks, spyware, malware, rootkits, Trojan horses, and worms. In a cloud environment, newer attacks may result from hypervisor malware, guest hopping and hijacking, or VM rootkits. Another type of attack is the man-in-the-middle attack for VM migrations. In general, passive attacks steal sensitive data or passwords. Active attacks may manipulate kernel data structures which will cause major damage to cloud servers. We will study all of these security and privacy problems on clouds.

Challenge 3—Unpredictable Performance and Bottlenecks

Multiple VMs can share CPUs and main memory in cloud computing, but I/O sharing is problematic. For example, to run 75 EC2 instances with the STREAM benchmark requires a mean bandwidth of 1,355 MB/second. However, for each of the 75 EC2 instances to write 1 GB files to the local disk requires a mean disk write bandwidth of only 55 MB/second. This demonstrates the problem of I/O interference between VMs. One solution is to improve I/O architectures and operating systems to efficiently virtualize interrupts and I/O channels. Internet applications continue to become more data-intensive. If we assume applications to be "pulled apart" across the boundaries of clouds, this may complicate data placement and transport. Cloud users and providers have to think about the implications of placement and traffic at every level of the system, if they want to minimize costs. This kind of reasoning can be seen in Amazon's development of its new CloudFront service. Therefore, data transfer

bottlenecks must be removed, bottleneck links must be widened, and weak servers should be removed.

Challenge 4—Distributed Storage and Widespread Software Bugs

The database is always growing in cloud applications. The opportunity is to create a storage system that will not only meet this growth, but also combine it with the cloud advantage of scaling arbitrarily up and down on demand. This demands the design of efficient distributed SANs. Data centers must meet programmers' expectations in terms of scalability, data durability, and HA. Data consistence checking in SAN-connected data centers is a major challenge in cloud computing. Large-scale distributed bugs cannot be reproduced, so the debugging must occur at a scale in the production data centers. No data center will provide such a convenience. One solution may be a reliance on using VMs in cloud computing. The level of virtualization may make it possible to capture valuable information in ways that are impossible without using VMs. Debugging over simulators is another approach to attacking the problem, if the simulator is well designed. Challenge 5—Cloud Scalability, Interoperability, and Standardization

The pay-as-you-go model applies to storage and network bandwidth; both are counted in terms of the number of bytes used. Computation is different depending on virtualization level. GAE automatically scales in response to load increases and decreases; users are charged by the cycles used. AWS charges by the hour for the number of VM instances used, even if the machine is idle. The opportunity here is to scale quickly up and down in response to load variation, in order to save money, but without violating SLAs. Open Virtualization Format (OVF) describes an open, secure, portable, efficient, and extensible format for the packaging and distribution of VMs. It also defines a format for distributing software to be deployed in VMs. This VM format does not rely on the use of a specific host platform, virtualization platform, or guest operating system. The approach is to address virtual platform-agnostic packaging with certification and integrity of packaged software. The package supports virtual appliances to span more than one VM.

OVF also defines a transport mechanism for VM templates, and can apply to different virtualization platforms with different levels of virtualization. In terms of cloud standardization, we suggest the ability for virtual appliances to run on any virtual platform. We also need to enable VMs to run on heterogeneous

hardware platform hypervisors. This requires hypervisor-agnostic VMs. We also need to realize cross-platform live migration between x86 Intel and AMD technologies and support legacy hardware for load balancing. All these issue are wide open for further research.

Challenge 6—Software Licensing and Reputation Sharing

Many cloud computing providers originally relied on open source software because the licensing model for commercial software is not ideal for utility computing. The primary opportunity is either for open source to remain popular or simply for commercial software companies to change their licensing structure to better fit cloud computing. One can consider using both pay-for-use and bulk-use licensing schemes to widen the business coverage. One customer's bad behavior can affect the reputation of the entire cloud. For instance, blacklisting of EC2 IP addresses by spam-prevention services may limit smooth VM installation. An opportunity would be to create reputation-guarding services similar to the "trusted e-mail" services currently offered (for a fee) to services hosted on smaller ISPs. Another legal issue concerns the transfer of legal liability. Cloud providers want legal liability to remain with the customer, and vice versa.

Inter Cloud Resource Management

Resource Provisioning and Platform Deployment

The emergence of computing clouds suggests fundamental changes in software and hardware architecture. Cloud architecture puts more emphasis on the number of processor cores or VM instances. Parallelism is exploited at the cluster node level.

Provisioning of Compute Resources (VMs):

Providers supply cloud services by signing SLAs with end users. The SLAs must commit sufficient resources such as CPU, memory, and bandwidth that the user can use for a preset period. Under provisioning of resources will lead to broken SLAs and penalties. Overprovisioning of resources will lead to resource underutilization, and consequently, a decrease in revenue for the provider. Deploying an autonomous system to efficiently provision resources to users is a challenging problem. The difficulty comes from the unpredictability of consumer demand, software and hardware failures, heterogeneity of services,

power management, and conflicts in signed SLAs between consumers and service providers.

To deploy VMs, users treat them as physical hosts with customized operating systems for specific applications. For example, Amazon's EC2 uses Xen as the virtual machine monitor (VMM). The same VMM is used in IBM's Blue Cloud.

In the EC2 platform, some predefined VM templates are also provided. Users can choose different kinds of VMs from the templates. IBM's Blue Cloud does not provide any VM templates. Power-efficient schemes for caching, query processing, and thermal management are mandatory due to increasing energy waste by heat dissipation from data centers. Public or private clouds promise to streamline the on-demand provisioning of software, hardware, and data as a service, achieving economies of scale in IT deployment and operation.

Resource Provisioning Methods

Figure 4.24 shows three cases of static cloud resource provisioning policies. In case (a), overprovisioning with the peak load causes heavy resource waste (shaded area). In case (b), underprovisioning (along the capacity line) of resources results in losses by both user and provider in that paid demand by the users (the shaded area above the capacity) is not served and wasted resources still exist for those demanded areas below the provisioned capacity. In case (c), the constant provisioning of resources with fixed capacity to a declining user demand could result in even worse resource waste. The user may give up the service by cancelling the demand, resulting in reduced revenue for the provider. Both the user and provider may be losers in resource provisioning without elasticity. Three resource-provisioning methods are presented in the following sections. The demand-driven method provides static resources and has been used in grid computing for many years. The event driven method is based on predicted workload by time. The popularity-driven method is based on Internet traffic monitored.

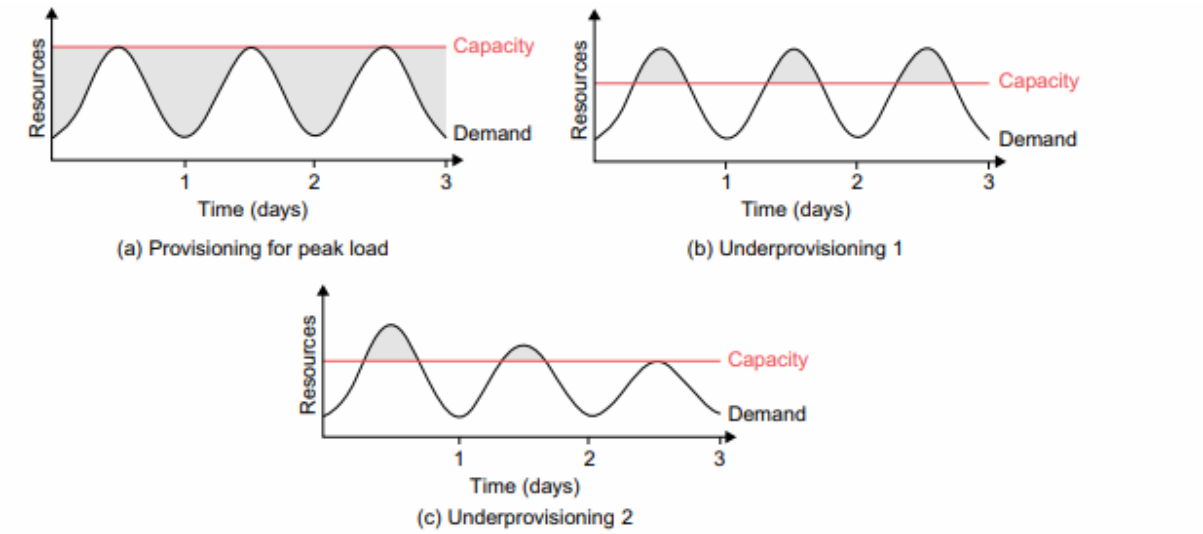


FIGURE 4.24

Three cases of cloud resource provisioning without elasticity: (a) heavy waste due to overprovisioning, (b) underprovisioning and (c) under- and then overprovisioning.

4.5.2.3 Demand-Driven Resource Provisioning

This method adds or removes computing instances based on the current utilization level of the allocated resources. The demand-driven method automatically allocates two Xeon processors for the user application, when the user was using one Xeon processor more than 60 percent of the time for an extended period. In general, when a resource has surpassed a threshold for a certain amount of time, the scheme increases that resource based on demand. When a resource is below a threshold for a certain amount of time, that resource could be decreased accordingly. Amazon implements such an auto-scale feature in its EC2 platform. This method is easy to implement. The scheme does not work out right if the workload changes abruptly.

The x-axis in Figure 4.25 is the time scale in milliseconds. In the beginning, heavy fluctuations of CPU load are encountered. All three methods have demanded a few VM instances initially. Gradually, the utilization rate becomes more stabilized with a maximum of 20 VMs (100 percent utilization) provided for demand-driven provisioning in Figure 4.25(a). However, the event-driven method reaches a stable peak of 17 VMs toward the end of the event and drops quickly in Figure 4.25(b). The popularity provisioning shown in Figure 4.25(c) leads to a similar fluctuation with peak VM utilization in the middle of the plot. **Event-Driven Resource Provisioning**

This scheme adds or removes machine instances based on a specific time event. The scheme works better for seasonal or predicted events such as

Christmastime in the West and the Lunar New Year in the East. During these events, the number of users grows before the event period and then decreases during the event period. This scheme anticipates peak traffic before it happens. The method results in a minimal loss of QoS, if the event is predicted correctly. Otherwise, wasted resources are even greater due to events that do not follow a fixed pattern.

Popularity-Driven Resource Provisioning

In this method, the Internet searches for popularity of certain applications and creates the instances by popularity demand. The scheme anticipates increased traffic with popularity. Again, the scheme has a minimal loss of QoS, if the predicted popularity is correct. Resources may be wasted if traffic does not occur as expected. In Figure 4.25(c), EC2 performance by CPU utilization rate (the dark curve with the percentage scale shown on the left) is plotted against the number of VMs provisioned (the light curves with scale shown on the right, with a maximum of 20 VMs provisioned).

Dynamic Resource Deployment

The cloud uses VMs as building blocks to create an execution environment across multiple resource sites. Dynamic resource deployment can be implemented to achieve scalability in performance. The InterGrid is a Java-implemented software system that lets users create execution cloud environments on top of all participating grid resources. Peering arrangements established between gateways enable the allocation of resources from multiple grids to establish the execution environment. '

In Figure 4.26, a scenario is illustrated by which an intergrid gateway (IGG) allocates resources from a local cluster to deploy applications in three steps: (1) requesting the VMs, (2) enacting the leases, and (3) deploying the VMs as requested. Under peak demand, this IGG interacts with another IGG that can allocate resources from a cloud computing provider. A grid has predefined peering arrangements with other grids, which the IGG manages. Through multiple IGGs, the system coordinates the use of InterGrid resources. An IGG is aware of the peering terms with other grids, selects suitable grids that can provide the required resources, and replies to requests from other IGGs. Request redirection policies determine which peering grid InterGrid selects to process a request and a price for which that grid will perform the task. An IGG can also allocate resources from a cloud provider. The cloud system creates a

virtual environment to help users deploy their applications. These applications use the distributed grid resources. The InterGrid allocates and provides a distributed virtual environment (DVE). This is a virtual cluster of VMs that runs isolated from other virtual clusters. A component called the DVE manager performs resource allocation and management on behalf of specific user applications. The core component of the IGG is a scheduler for implementing provisioning policies and peering with other gateways. The communication component provides an asynchronous message-passing mechanism. Received messages are handled in parallel by a thread pool

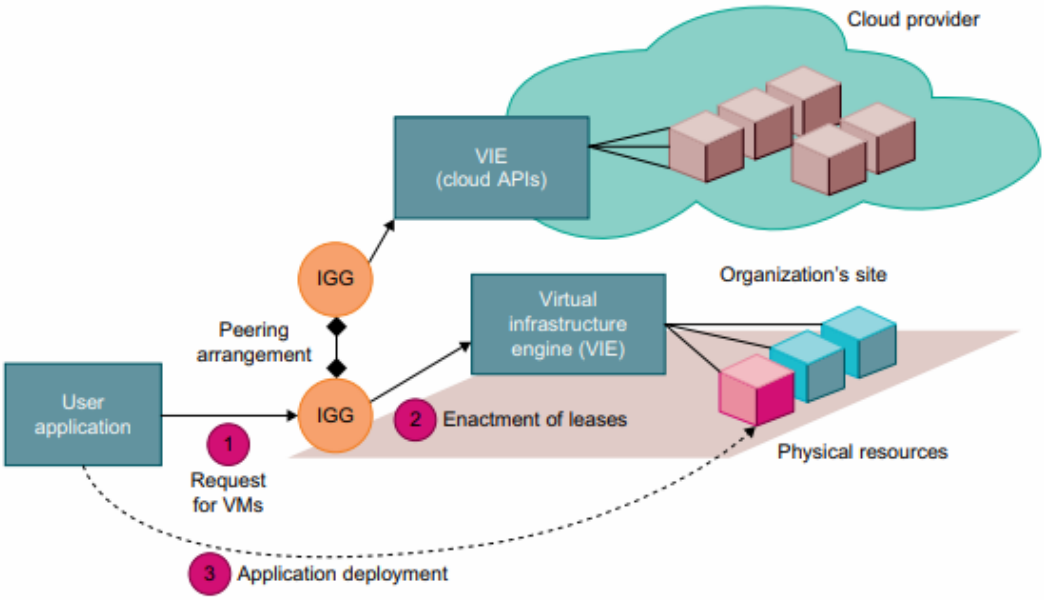


FIGURE 4.26

Cloud resource deployment using an IGG (intergrid gateway) to allocate the VMs from a Local cluster to interact with the IGG of a public cloud provider.

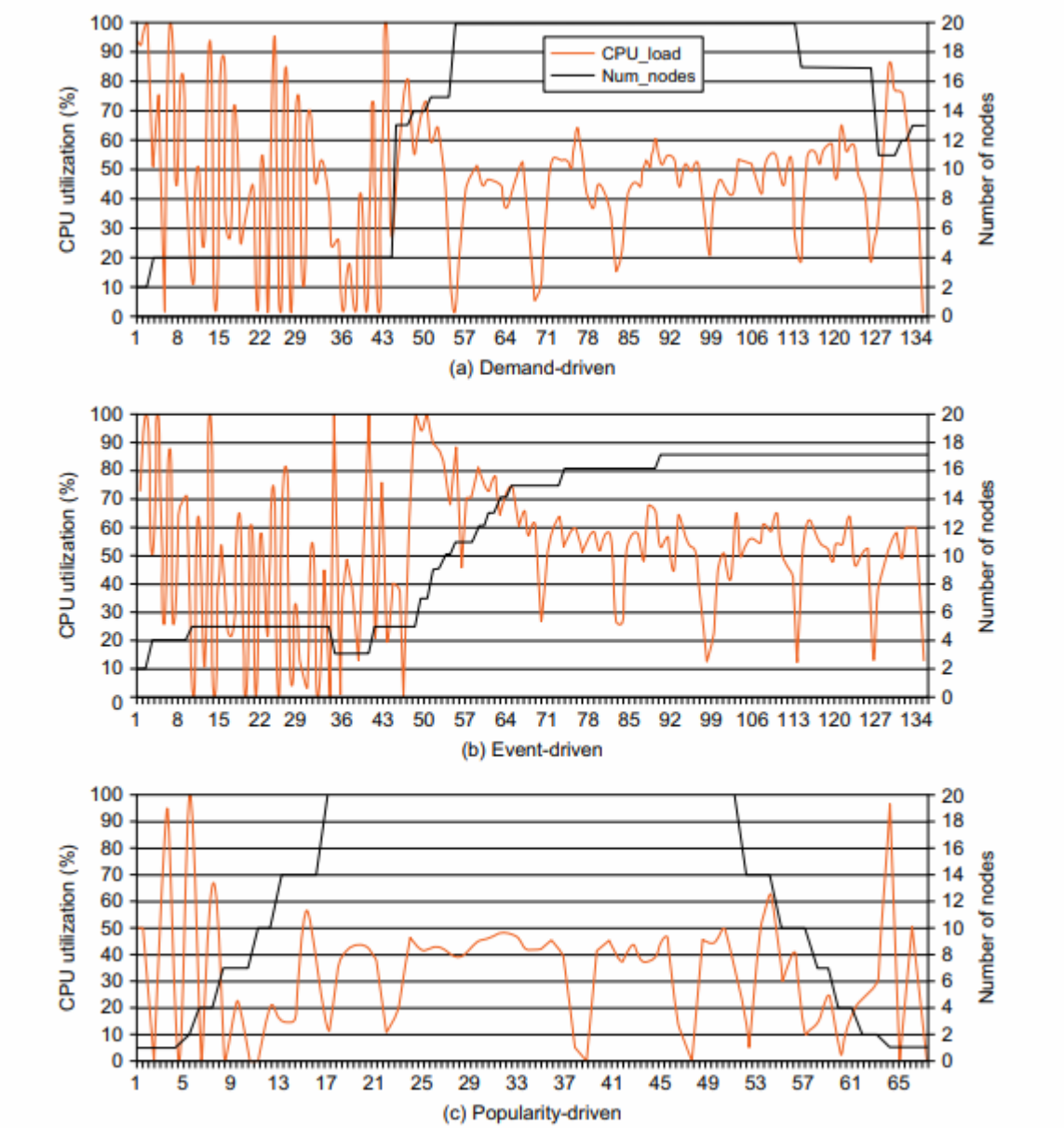


FIGURE 4.25

EC2 performance results on the AWS EC2 platform, collected from experiments at the University of Southern California using three resource provisioning methods.

Provisioning of Storage Resources

The data storage layer is built on top of the physical or virtual servers. One example is e-mail systems. A typical large e-mail system might have millions of users and each user can have thousands of e-mails and consume multiple gigabytes of disk space. Another example is a web searching application. A distributed file system is very important for storing large-scale data. However, other forms of data storage also exist. Some data does not need the namespace of a tree structure file system, and instead, databases are built with stored data files. In cloud computing, another form of data storage is (Key, Value) pairs.

Amazon S3 service uses SOAP to access the objects stored in the cloud. Table 4.8 outlines three cloud storage services provided by Google, Hadoop, and Amazon. Typical cloud databases include BigTable from Google, SimpleDB from Amazon, and the SQL service from Microsoft Azure

Table 4.8 Storage Services in Three Cloud Computing Systems

Storage System	Features
GFS: Google File System	Very large sustainable reading and writing bandwidth, mostly continuous accessing instead of random accessing. The programming interface is similar to that of the POSIX file system accessing interface.
HDFS: Hadoop Distributed File System	The open source clone of GFS. Written in Java. The programming interfaces are similar to POSIX but not identical.
Amazon S3 and EBS	S3 is used for retrieving and storing data from/to remote servers. EBS is built on top of S3 for using virtual disks in running EC2 instances.

Virtualization – VMware.

Virtualization is a technology that enables the creation of virtual environments from a single physical machine, allowing for more efficient use of resources by distributing them across computing environments.

Using software, virtualization creates an abstraction layer over computer hardware, dividing a single system's components such as processors, memory, networks and storage into multiple virtual machines (VMs). Each VM runs its own operating system (OS) and behaves like a separate physical computer, despite sharing the same underlying hardware.

What is VMware?

VMware develops virtualization software products that are crucial to many enterprises' IT infrastructures. VMware is a cloud computing and virtualization startup formed in 1998 that has played a significant role in transforming the way hardware configurations power workloads and support designs.

VMware virtualization essentially replaces certain pieces of hardware with VMware workstation that performs functions of conventional physical servers and PCs performed previously in the virtualization era.

The VMware cloud takes advantage of this transition from one virtualization era to the other with its products and services.

These VMware resources may be split over several virtual servers that act much like a single physical machine in the appropriate configurations – for example, storing data, developing and distributing programs, maintaining a workspace, and much more.

Each VMware workstation may run its own operating system and behave in the way that it has been commanded to. VMware for Desktops is available for Windows, macOS, and Linux whereas the enterprise option, vSphere requires no underlying operating system and runs directly on the hardware and is maintained remotely.

Virtual machines (VMs): The basic units of VMware

A virtual machine (VM) is a virtual representation or emulation of a physical computer that uses software instead of hardware to run programs and deploy applications.

A virtual machine (VM) is the base unit of VMware virtualization. It is a software-based representation of a physical computer. An operating system (OS) running in a VM is called a guest OS.

Each VM includes the following:

- A configuration file that stores the VM's settings.
- A virtual disk file that is a software version of a hard disk drive.
- A log file that keeps track of the VM's activities. This includes system failures, hardware changes, migrations of virtual machines from one host to another and the VM's status.

VMware offers various tools for managing these files. You can configure virtual machine settings using the vSphere Client, a command-line interface for VM management. You can also use the vSphere Web Services software development kit to configure VMs using other programs. For example, you could enable your software development environment to create a virtual machine to test a software program.

Benefits of VMware

Improved return on investment (ROI)

VMware enables you to use more of a physical computer's resources. Administrators don't like running multiple mission-critical applications on a single server OS because if one application crashes, it can make the OS unstable and crash other applications. One way to eliminate this risk is to run each application in its own OS on its own dedicated physical server. However, this is inefficient because each OS might only use 30% of a server's CPU power. With

VMware, you can run each application in its own OS on the same physical server and make better use of the physical server's available CPU power.

More efficient use of energy and space

VMware lets you run more applications using fewer physical servers. Fewer physical servers require less space in your data center and less energy to power and cool.

Optimize IT operations

VMware can help organizations better provision applications and resources and optimize their IT operations by balancing workloads across virtualized infrastructure.

- security based on a zero-trust model, along with better security than container systems like Kubernetes;
- better provisioning of applications and resources;
- simplified data center management; and
- increased efficiency and agility of data center systems.