ANSWER KEY – IAE 1 – CLOUD COMPUTING

1. Define cloud computing

Cloud computing is the delivery of computer system resources (e.g., servers, storage, databases, networking, software, analytics) over the internet

“The National Institute of Standards and Technology (NIST) defines cloud computing as a "payper-use model for enabling available, convenient and on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

2. List any two advantages of distributed computing.

Distributed computers offer two key advantages:

Easy scalability: Just add more computers to expand the system.

Redundancy: Since many different machines are providing the same service, that service can keep running even if one (or more) of the computers goes down.

3. Bring out the differences between private cloud and public cloud.

Public Cloud: Public clouds are owned and operated by third-party service providers who deliver computing resources like servers and storage over the Internet.

Microsoft Azure, Amazon Web Services (AWS), and Google Cloud Platform (GCP) are prime examples. They offer scalability and flexibility, making them ideal for many businesses.

Private Cloud: A private cloud refers to cloud computing resources used exclusively by one business or organization.

It can be hosted on-site or by a third-party provider, offering enhanced security and control, which is crucial for industries with strict data regulations.

S.No.	Public Cloud	Private Cloud
1.	When the computing infrastructure and resources are shared to the public via the internet, it is known as a public cloud.	When the computing infrastructure and the resources are shared to the private network via the internet, it is known as a private cloud.

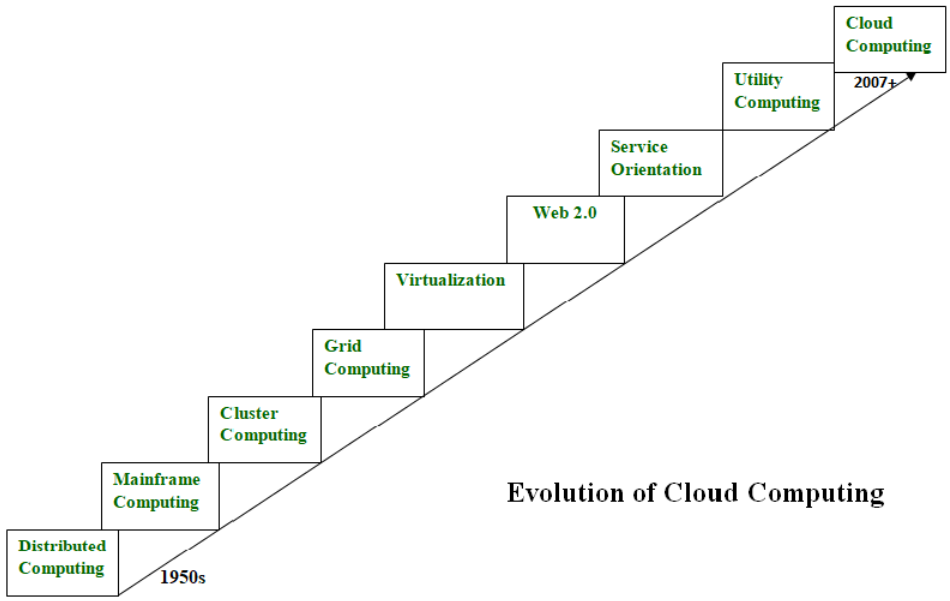
2.	A public cloud is like a multi-tenant in which the network is managed by your service provider.	A private cloud is like a single-tenant in which the network is handled by the in-house team.
3.	Here the data of several enterprises is stored.	Here the data of a single enterprise is stored.
4.	It supports the activity performed over the public network or internet.	It supports the activity performed over the private network or internet.
5.	The scalability is high in a public cloud.	The scalability is limited in a private cloud.
6.	Reliability is moderate here.	Reliability is high here.
7.	The security depends on the service provider.	It delivers a high class of security.
8.	It is affordable as compared to the private cloud.	It is expensive as compared to the public cloud.
9.	In the public cloud, the performance is low to medium.	The performance is high in a private cloud.
10.	It covers the shared servers.	It covers the devoted servers.

4. **Define SOA.**

SOA, or service-oriented architecture, defines a way to make software components reusable and interoperable through service interface. is a method of software development that uses software components called services to create business applications.

5. Write the name of Web services tools
   XML-RPC, UDDI, SOAP, and REST, Google Maps

6. a) Explain the evolution of cloud computing



Evolution of Cloud Computing

Evolution of Cloud Computing

Cloud Computing has evolved from the Distributed system to the current technology. Cloud computing has been used by all types of businesses, of different sizes and fields.

1. Distributed Systems

In the networks, different systems are connected. When they target to send the message from different independent systems which are physically located in various places but are connected through the network. Some examples of distributed systems are Ethernet which is a LAN technology, Telecommunication network, and parallel processing. The Basic functions of the distributed systems are –

- Resource Sharing – The Resources like data, hardware, and software can be shared between them.
- Open-to-all – The software is designed and can be shared.
- Fault Detection – The error or failure in the system is detected and can be corrected.

2. Mainframe Computing

It was developed in the year 1951 and provides powerful features. Mainframe Computing is still in existence due to its ability to deal with a large amount of data. For a company that needs to access and share a vast amount of data then this computing is preferred. Among the four types of computers, mainframe computer performs very fast and lengthy computations easily.
The type of services handled by them is bulk processing of data and exchanging large-sized hardware. Apart from the performance, mainframe computing is very expensive.

3. Cluster Computing

In Cluster Computing, the computers are connected to make it a single computing. The tasks in Cluster computing are performed concurrently by each computer also known as the nodes which are connected to the network. So the activities performed by any single node are known to all the nodes of the computing which may increase the performance, transparency, and processing speed.

To eliminate the cost, cluster computing has come into existence. We can also resize the cluster computing by removing or adding the nodes.

4. Grid Computing

It was introduced in the year 1990. As the computing structure includes different computers or nodes, in this case, the different nodes are placed in different geographical places but are connected to the same network using the internet.

The other computing methods seen so far, it has homogeneous nodes that are located in the same place. But in this grid computing, the nodes are placed in different organizations. It minimized the problems of cluster computing but the distance between the nodes raised a new problem.

5. Web 2.0

This computing lets the users generate their content and collaborate with other people or share the information using social media, for example, Facebook, Twitter, and Orkut. Web 2.0 is a combination of the second-generation technology World Wide Web (WWW) along with the web services and it is the computing type that is used today.

6. Virtualization

It came into existence 40 years back and it is becoming the current technique used in IT firms. It employs a software layer over the hardware and using this it provides the customer with cloud-based services.

7. Utility Computing

Based on the need of the user, utility computing can be used. It provides the users, company, clients or based on the business need the data storage can be taken for rent and used.

6.b) Examine in detail about the multi core CPUs and multithreading technologies

Multicore CPUs:

Unlike multi-processor systems that make use of multiple processors to carry out concurrent operations, multicore CPUs make use of multiple cores within a single processor. The idea here is to divide the processor into multiple cores such as dual, quad etc to carry out operations in parallel. The main advantage of these systems is that it improves potential performance of the overall system. One of the major examples of such systems in Intel processors whose speed of processing increased from 10 MHZ to 4 GHZ. This value is considered as limit for most (or) all of the chips that are based on CMOS because of the power constraints. These constraints can be removed by employing ILP (Instruction level Parallelism) mechanisms which are based on super scalar architecture and speculative execution.

Some systems use many-core GPU (Graphics Processing Units) that make use of thousands of processor cores. These GPUs are capable of managing instructions with varying magnitudes similar to that of multi core CPU.

Some of the processors that are based on multi-core and multithreaded processing are Intel i7. AMD opteron, IBM power 6 and many more. Multithreading: Multithreading is a feature which enables multiple threads to execute on a single processor in an overlapping manner. A thread is an atomic unit of a process and many threads usually make up a process. In a multithreading environment, the resources of a processor are being shared by multiple threads, so each thread gets a separate copy of the functional unit or resource. Functional units generally include

a register file, a separate Program Counter (pc) or a separate page table to enable virtual memory access, which in turn enables multiple program to execute simultaneously by sharing the memory

Multithreading

To enable multithreading, the hardware must be able to perform threads switching which is more efficient than switching the processes, as each process consists of threads and usually takes many clock cycles for its execution. As threads are lightweight, they can execute and switch among themselves during the execution. Therefore, they are considered more efficient and fast than processes.

Multithreading can be implemented in two ways

1. Fine-grained multithreading

2. Coarse-grained multithreading.

1. Fine Grained Multithreading: In this approach, the threads are switched on each instruction. The delay caused because of the switch operation of threads is very little. The threads are switched only if the current running thread encounters a stall. The subsequent thread is chosen from a pool of waiting threads in a round-robin fashion. The approach becomes effective if threads are switched at every clock cycle.

Advantage: The advantage of fine-grained multithreading is that it can efficiently recover the losses of throughput which come from short and long stalls of the thread.

Disadvantage: The execution of the stalled thread is delayed that in turn decreases the execution speed of that individual thread since another thread is being executed in its place.

2. Coarse-Grained Multithreading: Coarse-grained multithreading is another approach for implementing multithreading. In a coarse grained multithreading approach, the threads switch only when a costly stall is encountered. A costly stall can be defined as a stall where a thread requires resources which usually consumes more CPU clock cycles than required.

A level 2 cache miss is an example of a costly stall. If such a case is encountered in a coarse-grained approach, then another thread replaces it and executes till the stalled thread has recovered.

In contrast to a fine-grained approach, in a coarse-grained approach, the threads without stalls can be executed completely without any interruption until a costly stall is encountered.

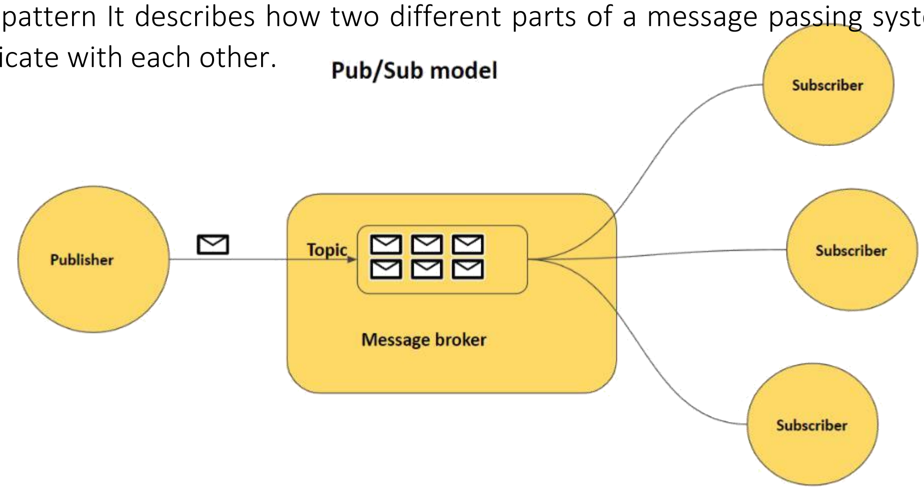
The main disadvantage of coarse-grained multithreading is that, when a thread encounters a costly stall, its instruction pipeline which is carrying out the execution gets frozen. The new thread which replaces this frozen thread has to wait until the emptied pipeline is filled, prior to completion of instruction execution. The time delay is significant and appears to be an overhead. Coarse-grained multithreaded approach doesn't have the ability.

The key advantage of coarse grained multithreading is that it stops the execution of threads which encounter costly stall and replaces with a new thread. A costly stall consumes more clock cycles when compared to the time taken to remove a frozen thread and replace a new thread into the pipeline.

7. a) Discuss the purpose of Publish-Subscribe Model.

Publish/Subscribe systems are nowadays considered a key technology for information diffusion. Each participant in a publish/subscribe communication system can play the role of a publisher or a subscriber of information. Publishers produce information in form of events, which are then consumed by subscribers. Subscribers can declare their interest on a subset of the whole information issuing subscriptions. There are two major roles: Publisher and Subscriber

Push strategy: It is the responsibility of the publisher to notify all the subscribers. Eg: Method invocation.
Pull strategy : The publisher simply makes available the message for a specific event. It is the responsibility of the subscribers to check whether there are messages on the events that are registered. Subscriptions are used to filter out part of the events produced by publishers. In Software Architecture, Publish/Subscribe pattern is a message pattern and a network oriented architectural pattern It describes how two different parts of a message passing system connect and communicate with each other.



There are three main components to the Publish Subscribe Model:

Publishers: Broadcast messages, with no knowledge of the subscribers.
Subscribers: They _listen' out for messages regarding topic/categories that they are interested in without any knowledge of who the publishers are.
Event Bus: Transfers the messages from the publishers to the subscribe

Pub-Sub messaging is an asynchronous communication method used in microservice architecture.
The Pub-Sub model consists of three components.

- A publisher who publishes message.
- A message broker or topic where the messages are pushed.
- A subscriber who receives the message via a message broker

A pub/sub model allows messages to be broadcasted asynchronously across multiple sections of the applications. In publish-subscribe, the sender of the message doesn't know anything about receivers. The message is being sent to the topic. After that, it's distributed among all endpoints subscribed to that topic. It can be useful e.g. for implementing notifications mechanism or distributing independent tasks.

Pub/Sub messaging allows to create decoupled applications easily with a reliable communication method and enables users to create Event-driven architectures. Event-driven architecture (EDA) is a software design pattern that enables a system to detect events (such as a transaction or site visit) and act on them in real time or near real time. This pattern replaces the traditional Request/Response architecture where services would have to wait for a reply before they could move onto the next task.

7. b) Discuss classification or taxonomy of virtualization at different levels

In computing, virtualization refers to the act of creating a virtual (rather than actual) version of something, like computer hardware platforms, operating systems, storage devices, and computer network resources

Work of Virtualization in Cloud Computing

Virtualization has a prominent impact on Cloud Computing. In the case of cloud computing, users store data in the cloud, but with the help of Virtualization, users have the extra benefit of sharing the infrastructure. Cloud Vendors take care of the required physical resources, but these cloud providers charge a huge amount for these services which impacts every user or organization. Virtualization helps Users or Organizations in maintaining those services which are required by a company through external (third-party) people, which helps in reducing costs to the company. This is the way through which Virtualization works in Cloud Computing.

Taxonomy of Virtualization

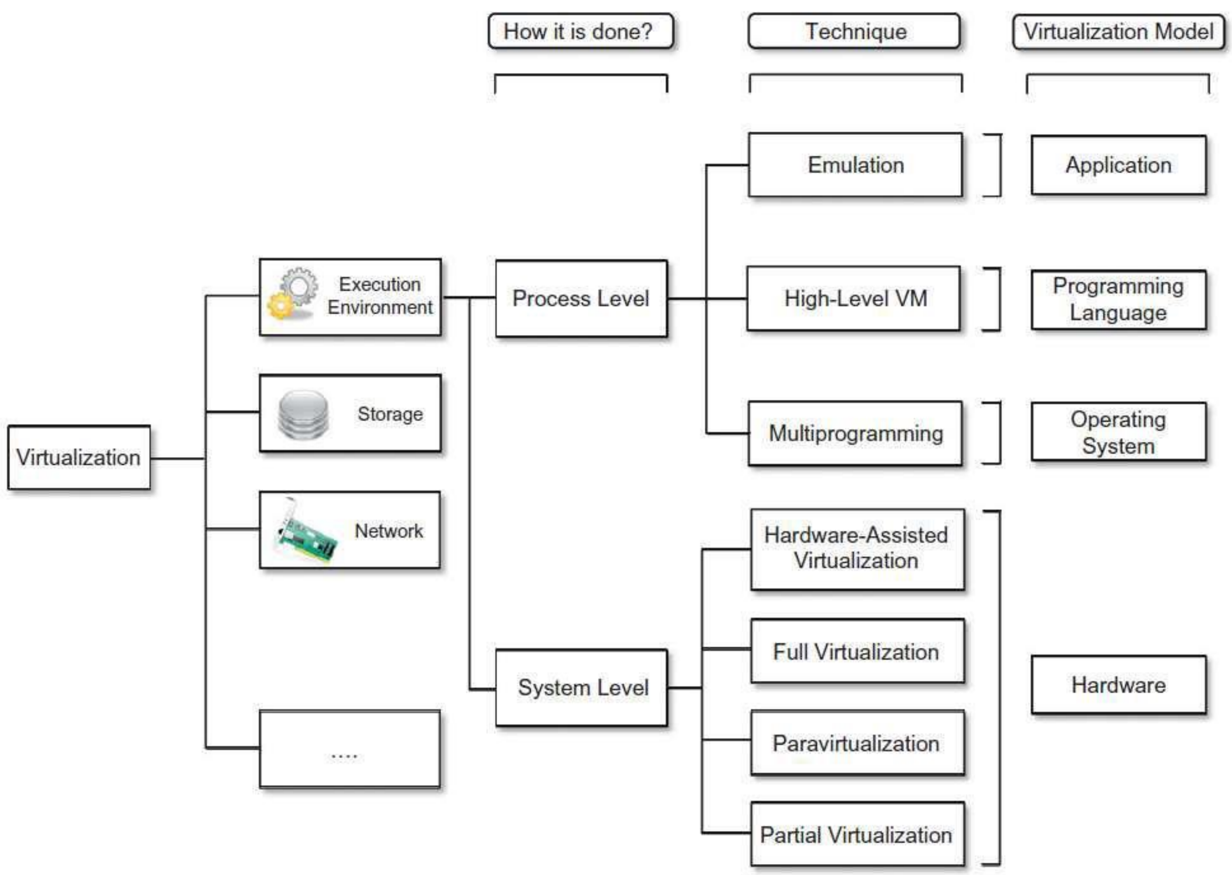
Virtualization is technology that you can use to create virtual representations of servers, storage, networks, and other physical machines. Virtual software mimics the functions of physical hardware to run multiple virtual machines simultaneously on a single physical machine.

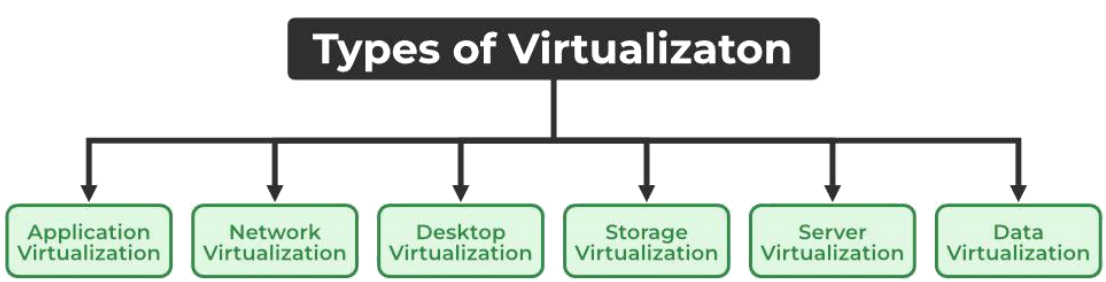
Virtualization is mainly used to emulate the execution environment, storage, and networks.

The execution environment is classified into two:

1. **Process-level** — implemented on top of an existing operating system.

2. System-level — implemented directly on hardware and does not or minimum requirement of the existing operating system.





Benefits of Virtualization

- More flexible and efficient allocation of resources.
- Enhance development productivity.
- It lowers the cost of IT infrastructure.
- Remote access and rapid scalability.
- High availability and disaster recovery.
- Pay peruse of the IT infrastructure on demand.
- Enables running multiple operating systems.

Drawback of Virtualization

- **High Initial Investment:** Clouds have a very high initial investment, but it is also true that it will help in reducing the cost of companies.
- **Learning New Infrastructure:** As the companies shifted from Servers to Cloud, it requires highly skilled staff who have skills to work with the cloud easily, and for this, you have to hire new staff or provide training to current staff.
- **Risk of Data:** Hosting data on third-party resources can lead to putting the data at risk, it has the chance of getting attacked by any hacker or cracker very easily.

Characteristics of Virtualization

- **Increased Security:** The ability to control the execution of a guest program in a completely transparent manner opens new possibilities for delivering a secure, controlled execution environment. All the operations of the guest programs are generally performed against the virtual machine, which then translates and applies them to the host programs.
- **Managed Execution:** In particular, sharing, aggregation, emulation, and isolation are the most relevant features.
- **Sharing:** Virtualization allows the creation of a separate computing environment within the same host.
- **Aggregation:** It is possible to share physical resources among several guests, but virtualization also allows aggregation, which is the opposite process.

1. Application Virtualization: Application virtualization helps a user to have remote access to an application from a server. The server stores all personal information and other characteristics of the application but can still run on a local workstation through the internet. An 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: The ability to run multiple virtual networks with each having a separate control and data plan. It co-exists together on top of one physical network. It can be managed by individual parties that are potentially confidential to each other. Network virtualization provides a facility to create and provision virtual networks, logical switches, routers, firewalls, load balancers, Virtual Private Networks (VPN), and workload security within days or even weeks.

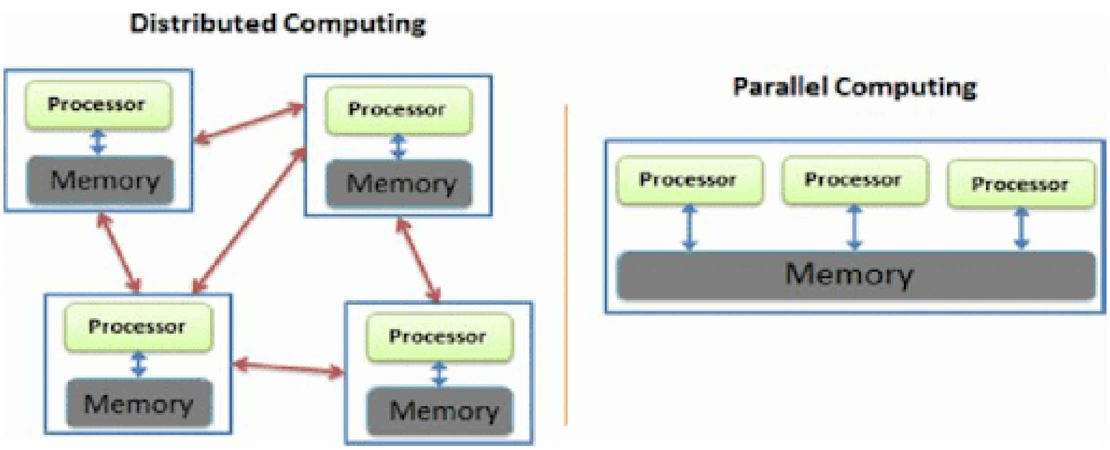
3. Desktop Virtualization: Desktop virtualization allows the users' OS to be remotely stored on a server in the data center. It allows the user to access their desktop virtually, from any location by a different machine. Users who want specific operating systems other than Windows Server will need to have a virtual desktop. The main benefits of desktop virtualization are user mobility, portability, and 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 and instead function more like worker bees in a hive. It makes managing storage from multiple sources 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, breaks down, and differences in the underlying equipment.

5. Server Virtualization: This is a kind of virtualization in which the masking of server resources takes place. Here, the central server (physical server) is divided into multiple different virtual servers by changing the identity number, and processors. So, each system can operate its operating systems in an isolated manner. Where each sub-server knows the identity of the central server. It causes an increase in performance and reduces the operating cost by the deployment of main server resources into a sub-server resource. It's beneficial in virtual migration, reducing energy consumption, reducing infrastructural costs, etc.

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

8. a) Explain in detail underlying principles of Parallel and Distributed Computing. Tabulate the differences between these computing.



Parallel computing, also known as parallel processing, speeds up a computational task by dividing it into smaller jobs across multiple processors inside one computer. Distributed computing,
on the other hand, Distributed computing uses a distributed system, such as the internet, to increase the available computing power and enable larger, more complex tasks to be executed across multiple machines.

Parallel computing is the process of performing computational tasks across multiple processors at once to improve computing speed and efficiency. It divides tasks into sub-tasks and executes them simultaneously through different processors.

There are three main types, or "levels," of parallel computing: bit, instruction, and task.

- Bit-level parallelism: Uses larger "words," which is a fixed-sized piece of data handled as a unit by the instruction set or the hardware of the processor, to reduce the number of instructions the processor needs to perform an operation.
- Instruction-level parallelism: Employs a stream of instructions to allow processors to execute more than one instruction per clock cycle (the oscillation between high and low states within a digital circuit).
- Task-level parallelism: Runs computer code across multiple processors to run multiple tasks at the same time on the same data.

Distributed computing is the process of connecting multiple computers via a local network or wide area network so that they can act together as a single ultra-powerful computer capable of performing computations that no single computer within the network would be able to perform on its own.

Distributed computers offer two key advantages:

- Easy scalability: Just add more computers to expand the system.
- Redundancy: Since many different machines are providing the same service, that service can keep running even if one (or more) of the computers goes down.

S.No	Parallel Computing	Distributed Computing
1	Parallel computing involves the use of multiple processors within a single computer to work on a problem.	Distributed computing involves the use of multiple computers across a network to work together on a problem.
2	If one processor fails, others can continue working.	If one computer fails, others can continue working.
3	In parallel computing, communication between processors is usually fast because the processors are housed within the same computer	In distributed computing, communication between computers is slower because the computers are connected over a network.
4	Parallel computing often requires a high degree of synchronization between processors, as they must coordinate their efforts to solve a problem.	In distributed computing, synchronization is more complex because the computers work independently and must coordinate their efforts over a network.
5	In parallel computing, the computer can have a shared memory.	In distributed computing, each computer has its own memory.

8. b) Give the importance of Virtualization Support and its implementation.

Virtualization is a computer architecture technology by which multiple virtual machines (VMs) are multiplexed in the same hardware machine. The idea of VMs can be dated back to the 1960s. The purpose of a VM is to enhance resource sharing by many users and improve computer performance in terms of resource utilization and application flexibility.

Hardware resources (CPU, memory, I/O devices, etc.) or software resources (operating system and software libraries) can be virtualized in various functional layers. This virtualization technology has been revitalized as the demand for distributed and cloud computing increased sharply in recent years.

Virtualization uses software to create an abstraction layer over computer hardware, enabling the division of a single computer's hardware components—such as processors, memory and storage—into multiple virtual machines (VMs). Each VM runs its own operating system (OS) and behaves like an independent computer, even though it is running on just a portion of the actual underlying computer hardware.

It follows that virtualization enables more efficient use of physical computer hardware and allows a greater return on an organization's hardware investment. Today, virtualization is a standard practice in enterprise IT architecture. It is also the technology that drives cloud computing economics. Virtualization enables cloud providers to serve users with their existing physical computer hardware. It enables cloud users to purchase only the computing resources they need when they need it, and to scale those resources cost-effectively as their workloads grow.

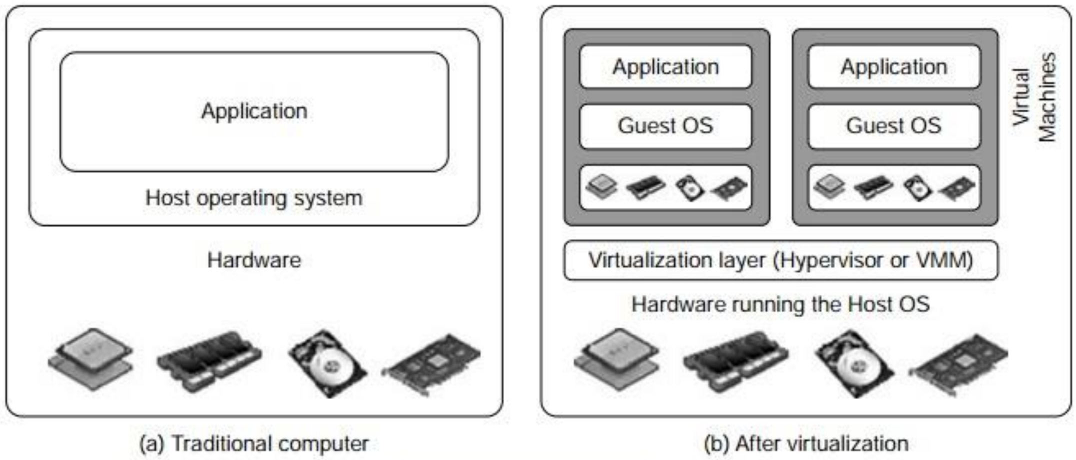


FIGURE 3.1

The architecture of a computer system before and after virtualization, where VMM stands for virtual machine monitor.

Virtualization example

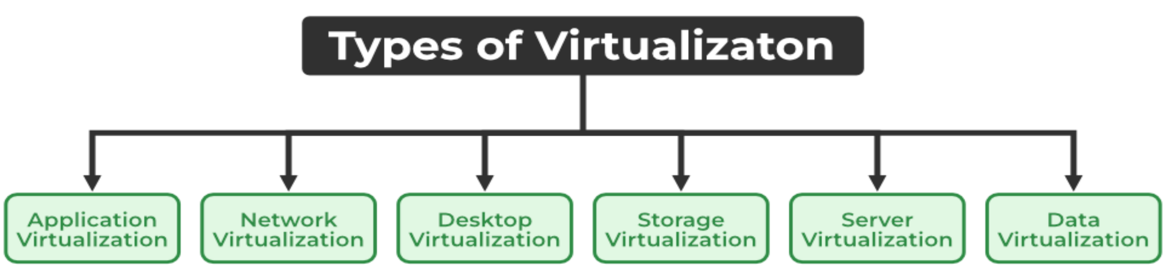
Consider a company that needs servers for three functions:

1. Store business email securely
2. Run a customer-facing application
3. Run internal business applications

Each of these functions has different configuration requirements:

- The email application requires more storage capacity and a Windows operating system.
- The customer-facing application requires a Linux operating system and high processing power to handle large volumes of website traffic.
- The internal business application requires iOS and more internal memory (RAM).

To meet these requirements, the company sets up three different dedicated physical servers for each application. The company must make a high initial investment and perform ongoing maintenance and upgrades for one machine at a time. The company also cannot optimize its computing capacity. It pays 100% of the servers' maintenance costs but uses only a fraction of their storage and processing capacities.



How does virtualization work?

Virtualization uses specialized software, called a hypervisor, to create several cloud instances or virtual machines on one physical computer.

Cloud instances or Virtual Machines

After you install virtualization software on your computer, you can create one or more virtual machines. You can access the virtual machines in the same way that you access other applications on your computer. Your computer is called the host, and the virtual machine is called the guest. Several guests can run on the host. Each guest has its own operating system, which can be the same or different from the host operating system.

From the user's perspective, the virtual machine operates like a typical server. It has settings, configurations, and installed applications. Computing resources, such as central processing units (CPUs), Random Access Memory (RAM), and storage appear the same as on a physical server. You can also configure and update the guest operating systems and their applications as necessary without affecting the host operating system.

Hypervisors

The hypervisor is the virtualization software that you install on your physical machine. It is a software layer that acts as an intermediary between the virtual machines and the underlying hardware or host operating system. The hypervisor coordinates access to the physical environment so that several virtual machines have access to their own share of physical resources.

For example, if the virtual machine requires computing resources, such as computer processing power, the request first goes to the hypervisor. The hypervisor then passes the request to the underlying hardware, which performs the task.

The following are the two main types of hypervisors.

Type 1 hypervisors

A type 1 hypervisor—also called a bare-metal hypervisor—runs directly on the computer hardware. It has some operating system capabilities and is highly efficient because it interacts directly with the physical resources.

Type 2 hypervisors

A type 2 hypervisor runs as an application on computer hardware with an existing operating system. Use this type of hypervisor when running multiple operating systems on a single machine.

Levels of Virtualization Implementation

The virtualization software creates the abstraction of VMs by interposing a virtualization layer at various levels of a computer system. The main function of the software layer for virtualization is to virtualize the physical hardware of a host machine into virtual resources to be used by the VMs, exclusively. This can be implemented at various operational levels as mentioned below

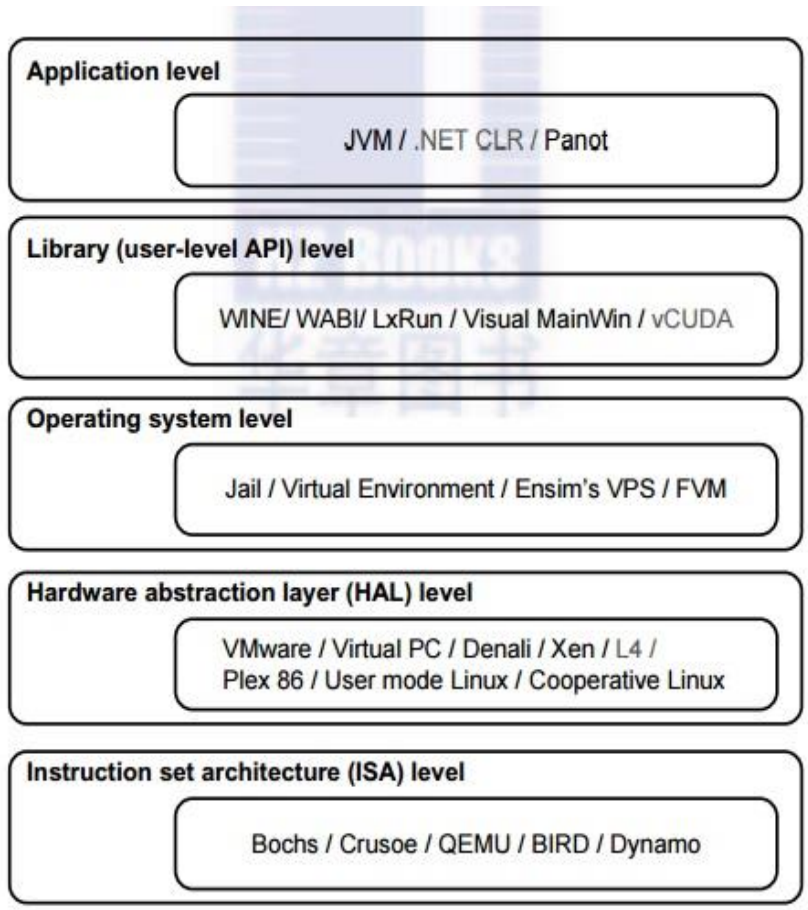


FIGURE 3.2

Virtualization ranging from hardware to applications in five abstraction levels.

a) Instruction Set Architecture Level

At the ISA level, virtualization is performed by emulating a given ISA by the ISA of the host machine. For example, MIPS binary code can run on an x86-based host machine with the help of ISA emulation. With this approach, it is possible to run a large amount of legacy binary code written for various processors on any given new hardware host machine. Instruction set emulation leads to virtual ISAs created on any hardware machine.

The basic emulation method is through code interpretation. An interpreter program interprets the source instructions to target instructions one by one. One source instruction may require tens or hundreds of native target instructions to perform its function. Obviously, this process is relatively slow. For better performance, dynamic binary translation is desired. This approach translates basic blocks of dynamic source instructions to target instructions. The basic blocks can also be extended to program traces or super blocks to increase translation efficiency. Instruction set emulation requires binary translation and optimization. A virtual instruction set architecture (V-ISA) thus requires adding a processor-specific software translation layer to the compiler.

b) Hardware Abstraction Level

Hardware-level virtualization is performed right on top of the bare hardware. On the one hand, this approach generates a virtual hardware environment for a VM. On the other hand, the process manages the underlying hardware through virtualization. The idea is to virtualize a computer's resources, such as its processors, memory, and I/O devices. The intention is to upgrade the hardware utilization rate by multiple users concurrently. The idea was implemented in the IBM VM/370 in the 1960s. More recently, the Xen hypervisor has been applied to virtualize x86-based machines to run Linux or other guest OS applications. We will discuss hardware virtualization approaches in more detail in Section 3.3.

c) Operating System Level

This refers to an abstraction layer between traditional OS and user applications. OS-level virtualization creates isolated containers on a single physical server and the OS instances to utilize the hard-ware and software in data centers. The containers behave like real servers. OS-level virtualization is commonly used in creating virtual hosting environments to allocate hardware resources among a large number of mutually distrusting users. It is also used, to a lesser extent, in consolidating server hardware by moving services on separate hosts into containers or VMs on one server. OS-level virtualization is depicted in Section 3.1.3.

d) Library Support Level

Most applications use APIs exported by user-level libraries rather than using lengthy system calls by the OS. Since most systems provide well-documented APIs, such an interface becomes another candidate for virtualization. Virtualization with library interfaces is possible by controlling the communication link between applications and the rest of a system through API hooks. The software tool WINE has implemented this approach to support Windows applications on top of UNIX hosts. Another example is the vCUDA which allows applications executing within VMs to leverage GPU hardware acceleration. This approach is detailed in Section 3.1.4.

e) User-Application Level

Virtualization at the application level virtualizes an application as a VM. On a traditional OS, an application often runs as a process. Therefore, application-level virtualization is also known as process-level virtualization. The most popular approach is to deploy high level language (HLL) VMs. In this scenario, the virtualization layer sits as an application program on top of the operating system, and the layer exports an abstraction of a VM that can run programs written and compiled to a particular abstract machine definition. Any program written in the HLL and compiled for this VM will be able to run on it. The Microsoft .NET CLR and Java Virtual Machine (JVM) are two good examples of this class of VM. Other forms of application-level virtualization are known as application isolation, application sandboxing, or application streaming. The process involves wrapping the application in a layer that is isolated from the host OS and other applications.