

Unit – I

1.1 Introduction to Cloud Computing

- Over the last three decades, businesses that use computing resources have learned to face a vast array of buzzwords like grid computing, utility computing, autonomic computing, on-demand computing and so on.
- A new buzzword named cloud computing is presently in state-of-the-art and it is generating all sorts of confusion about what it actually means.
- In history, the term cloud has been used as a metaphor for the Internet.

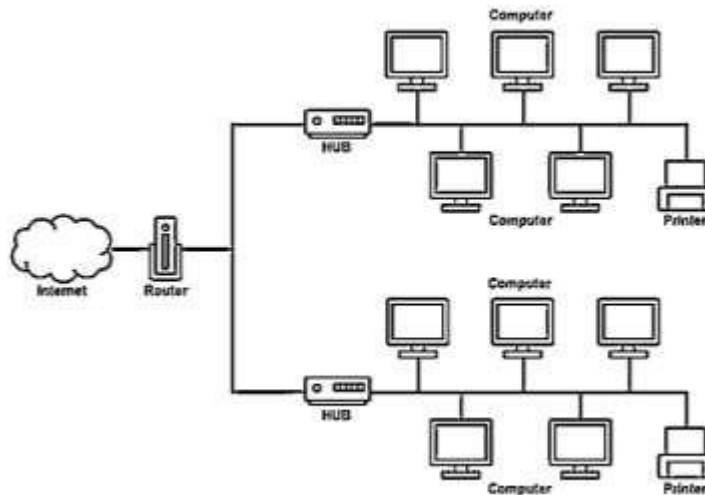


Figure 1.1 illustration of network diagram

- This usage of the term was originally derived from its common illustration in network diagrams as an outline of a cloud and the symbolic representation used to represent the transport of data across the network to an endpoint location on the other side of the network.
- Figure 1.1 illustrates the network diagram which includes the symbolic representation of cloud
- The cloud computing concepts were initiated in 1961, when Professor John McCarthy suggested that computer time-sharing technology might lead to a future where computing power and specific applications might be sold through a utility-type business model.
- This idea became very popular in the late 1960s, but in mid 1970s the idea vanished away when it became clear that the IT Industries of the day were unable to sustain such a innovative computing model. However, since the turn of the millennium, the concept has been restored.
- Utility computing is the provision of computational resources and storage resources as a metered service, similar to those provided by a traditional public utility company. This is not a new idea. This form of computing is growing in popularity, however, as companies have begun to extend the model to a cloud computing paradigm providing virtual servers that IT departments and users can access on demand.
- In early days, enterprises used the utility computing model primarily for non-mission-critical requirements, but that is quickly changing as trust and reliability issues are resolved.
- Research analysts and technology vendors are inclined to define cloud computing very closely, as a new type of utility computing that basically uses virtual servers that have been made available to third parties via the Internet.

- Others aimed to describe the term cloud computing using a very broad, all-inclusive application of the virtual computing platform. They confront that anything beyond the network firewall limit is in the cloud.
- A more softened view of cloud computing considers it the delivery of computational resources from a location other than the one from which the end users are computing.
- The cloud sees no borders and thus has made the world a much smaller place. Similar to that the Internet is also global in scope but respects only established communication paths.
- People from everywhere now have access to other people from anywhere else.
- Globalization of computing assets may be the major contribution the cloud has made to date. For this reason, the cloud is the subject of many complex geopolitical issues.
- Cloud computing is viewed as a resource available as a service for virtual data centers. Cloud computing and virtual data centers are different one.
- For example, Amazon's S3 is Simple Storage Service. This is a data storage service designed for use across the Internet. It is designed to create web scalable computing easier for developers.
- Another example is Google Apps. This provides online access via a web browser to the most common office and business applications used today. The Google server stores all the software and user data.
- Managed service providers (MSPs) offers one of the oldest form of cloud computing.
- A managed service is an application that is accessible to an organization's IT infrastructure rather than to end users which include virus scanning for email, anti spamservices such as Postini, desktop management services offered by CenterBeam or Everdream, and application performance monitoring.

- Grid computing is often confused with cloud computing. Grid computing is a form of distributed computing model that implements a virtual supercomputer made up of a cluster of networked or Inter networked computers involved to perform very large tasks.
- Most of the cloud computing deployments in market today are powered by grid computing implementations and are billed like utilities, but cloud computing paradigm is evolved next step away from the grid utility model.
- The majority of cloud computing infrastructure consists of time tested and highly reliable services built on servers with varying levels of virtualized technologies, which are delivered via large scale data centers operating under various service level agreements that require 99.9999% uptime.

1.2 Definition of cloud

- Cloud computing is a model for delivering IT services in which resources are retrieved from the internet through web based tools and applications rather than a direct connection to the server.

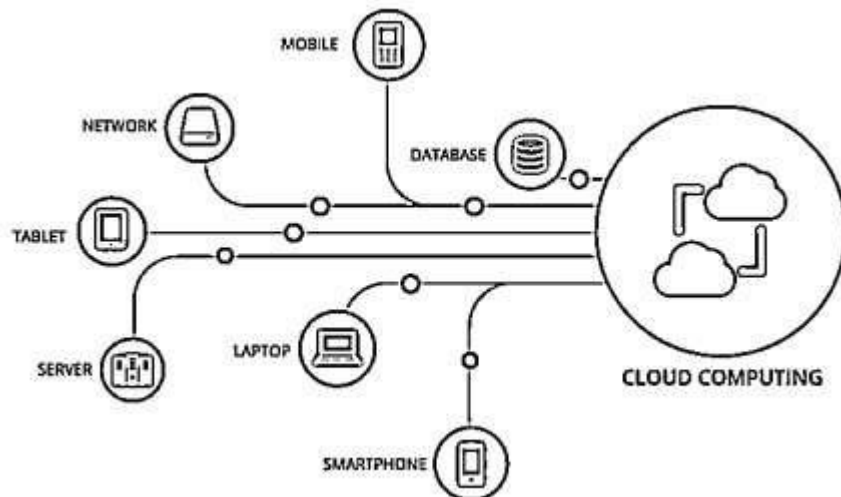


Figure 1.2 Cloud Computing Paradigm

- In other words, cloud computing is a distributed computing model over a network and means the ability to run a program on many connected components at a same time

- In the cloud computing environment, real server machines are replaced by virtual machines. Such virtual machines do not physically exist and can therefore be moved around and scaled up or down on the fly without affecting the cloud user as like a natural cloud.
- Cloud refers to software, platform, and Infrastructure that are sold as a service. The services accessed remotely through the Internet
- The cloud users can simply log on to the network without installing anything. They do not pay for hardware and maintenance. But the service providers pay for physical equipment and maintenance.
- The concept of cloud computing becomes much more understandable when one begins to think about what modern IT environments always require scalable capacity or additional capabilities to their infrastructure dynamically, without investing money in the purchase of new infrastructure, all the while without needing to conduct training for new personnel and without the need for licensing new software.
- The cloud model is composed of three components.

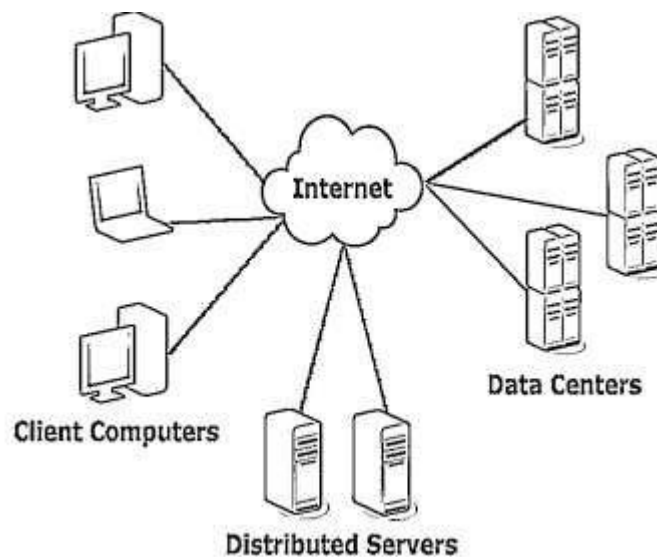


Figure 1.3 Cloud Components

- **Clients** are simple computers might be laptop, tablet, mobile phone.
- Categories of clients are Mobile clients, Thin clients and Thick clients.
- Mobile clients which includes smartphones and PDAs
- Thin clients which include servers without internal hardware. Usage of this type of clients leads to Low hardware cost, Low IT Cost, Less power consumption and less noise.
- Thick clients which includes regular computers.
- **Data Center** is a collection of servers and it contains clients requested applications.
- **Distributed Server** in which server is distributed in different geographical locations

1.3 Evolution of Cloud Computing

- It is important to understand the evolution of computing in order to get an appreciation of how IT based environments got into the cloud environment. Looking at the evolution of the computing hardware itself, from the first generation to the fourth generation of computers, shows how the IT industry's got from there to here.
- The hardware is a part of the evolutionary process. As hardware evolved, so did the software. As networking evolved, so did the rules for how computers communicate. The development of such rules or protocols, helped to drive the evolution of Internet software.
- Establishing a common protocol for the Internet led directly to rapid growth in the number of users online.
- Today, enterprises discuss about the uses of IPv6 (Internet Protocol version 6) to ease addressing concerns and for improving the methods used to communicate over the Internet.
- Usage of web browsers led to a stable migration away from the traditional data center model to a cloud computing based model. And also, impact of technologies such as

server virtualization, parallel processing, vector processing, symmetric multiprocessing, and massively parallel processing fueled radical change in IT era.

1.3.1 Hardware Evolution

- The first step along with the evolutionary path of computers was occurred in 1930, when the first binary arithmetic was developed and became the foundation of computer processing technology, terminology, and programming languages.
- Calculating devices date back to at least as early as 1642, when a device that could mechanically add numbers was invented.
- Adding devices were evolved from the abacus. This evolution was one of the most significant milestones in the history of computers.
- In 1939, the Berry brothers were invented an electronic computer that capable of operating digital aspects. The computations were performed using vacuum tube technology.
- In 1941, the introduction of Z3 at the German Laboratory for Aviation purpose in Berlin was one of the most significant events in the evolution of computers because Z3 machine supported both binary arithmetic and floating point computation. Because it was a "Turing complete" device, it is considered to be the very first computer that was fully operational.

1.3.1.1 First Generation Computers

- The first generation of modern computers traced to 1943, when the Mark I and Colossus computers were developed for fairly different purposes.
- With financial support from IBM, the Mark I was designed and developed at Harvard University. It was a general purpose electro, mechanical, programmable computer.

- Colossus is an electronic computer built in Britain at the end 1943. Colossus was the world's first programmable, digital, electronic, computing device.

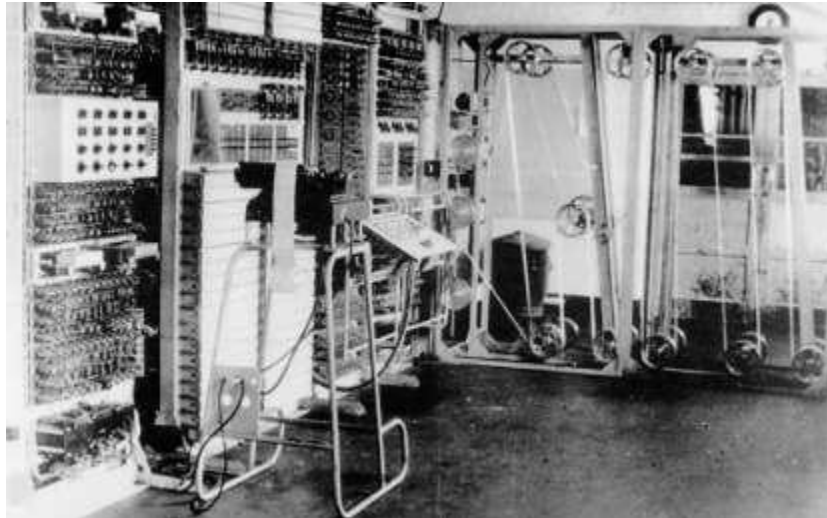


Figure 1.4 Colossus

- In general, First generation computers were built using hard-wired circuits and vacuum tubes.
- Data were stored using paper punch cards.

1.3.1.2 Second Generation Computers

- Another general-purpose computer of this era was ENIAC (Electronic Numerical Integrator and Computer), which was built in 1946. This was the first Turing complete, digital computer that capable of reprogramming to solve a full range of computing problems.
- ENIAC composed of 18,000 thermionic valves, weighed over 60,000 pounds, and consumed 25 kilowatts of electrical power per hour. ENIAC was capable of performing one lakh calculations a second.



Figure 1.5 ENIAC

- Transistorized computers marked the initiation of second generation computers, which dominated in the late 1950s and early 1960s. The computers were used mainly by universities and government agencies.
- The integrated circuit or microchip was developed by Jack St. Claire Kilby, an achievement for which he received the Nobel Prize in Physics in 2000.

1.3.1.3 Third Generation Computers

- Claire Kilby's invention initiated an explosion in third generation computers. Even though the first integrated circuit was produced in 1958, microchips were not used in programmable computers until 1963.
- In 1971, Intel released the world's first commercial microprocessor called Intel 4004.



Figure 1.6 Intel 4004

- Intel 4004 was the first complete CPU on one chip and became the first commercially available microprocessor. It was possible because of the development of new silicon gate technology that enabled engineers to integrate a much greater number of transistors on a chip that would perform at a much faster speed.

1.3.1.4 Fourth Generation Computers

- The fourth generation computers that were being developed at this time utilized a microprocessor that put the computer's processing capabilities on a single integrated circuit chip.
- By combining random access memory, developed by Intel, fourth generation computers were faster than ever before and had much smaller footprints.
- The first commercially available personal computer was the MITS Altair 8800, released at the end of 1974. What followed was a flurry of other personal computers to market, such as the Apple I and II, the Commodore PET, the VIC-20, the Commodore 64, and eventually the original IBM PC in 1981. The PC era had begun in earnest by the mid-1980s.
- Even though microprocessing power, memory and data storage capacities have increased by many orders of magnitude since the invention of the 4004 processor, the technology for Large Scale Integration (LSI) or Very Large Scale Integration (VLSI) microchips has not changed all that much.
- For this reason, most of today's computers still fall into the category of fourth generation computers.

1.3.2 Internet Software Evolution

- The Internet is named after the evolution of Internet Protocol which is the standard communications protocol used by every computer on the Internet.

- Vannevar Bush was written a visionary description of the potential uses for information technology with his description of an automated library system called MEMEX.
- Bush introduced the concept of the MEMEX in late 1930s as a microfilm based device in which an individual can store all his books and records.

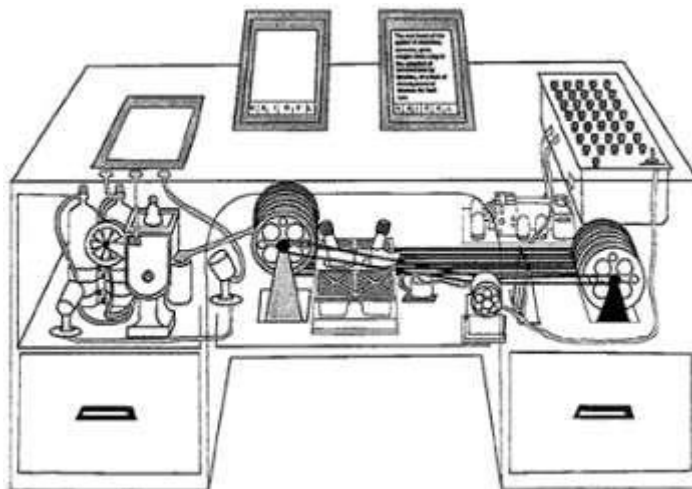


Figure 1.7 MEMEX Systems

- The second individual who has shaped the Internet was Norbert Wiener.
- Wiener was an early pioneer in the study of stochastic and noise processes. Norbert Wiener work in stochastic and noise processes was relevant to electronic engineering, communication, and control systems.
- SAGE refers Semi Automatic Ground Environment. SAGE was the most ambitious computer project and started in the mid 1950s and became operational by 1963. It remained in continuous operation for over 20 years, until 1983.
- A minicomputer was invented specifically to realize the design of the Interface Message Processor (IMP). This approach provided a system independent interface to the ARPANET.
- The IMP would handle the interface to the ARPANET network. The physical layer, the data link layer, and the network layer protocols used internally on the ARPANET were implemented using IMP.

- Using this approach, each site would only have to write one interface to the commonly deployed IMP.
- The first networking protocol that was used on the ARPANET was the Network Control Program (NCP). The NCP provided the middle layers of a protocol stack running on an ARPANET connected host computer.
- The lower-level protocol layers were provided by the IMP host interface, the NCP essentially provided a transport layer consisting of the ARPANET Host-to-Host Protocol (AHHP) and the Initial Connection Protocol (ICP).
- The AHHP defines how to transmit a unidirectional and flow controlled stream of data between two hosts.
- The ICP specifies how to establish a bidirectional pair of data streams between a pair of connected host processes.
- Robert Kahn and Vinton Cerf who built on what was learned with NCP to develop the TCP/IP networking protocol commonly used nowadays. TCP/IP quickly became the most widely used network protocol in the world.
- Over time, there evolved four increasingly better versions of TCP/IP (TCP v1, TCP v2, a split into TCP v3 and IP v3, and TCP v4 and IPv4). Now, IPv4 is the standard protocol, but it is in the process of being replaced by IPv6.
- The amazing growth of the Internet throughout the 1990s caused a huge reduction in the number of free IP addresses available under IPv4. IPv4 was never designed to scale to global levels. To increase available address space, it had to process data packets that were larger.
- After examining a number of proposals, the Internet Engineering Task Force (IETF) settled on IPv6, which was released in early 1995 as RFC 1752. IPv6 is sometimes called the Next Generation Internet Protocol (IPNG) or TCP/IP v6.

1.3.3 Server Virtualization

- Virtualization is a method of running multiple independent virtual operating systems on a single physical computer. This approach maximizes the return on investment for the computer.
- The creation and management of virtual machines has often been called platform virtualization.
- Platform virtualization is performed on a given computer (hardware platform) by software called a control program.
- Parallel processing is performed by the simultaneous execution of multiple program instructions that have been allocated across multiple processors with the objective of running a program in less time.
- The next advancement in parallel processing was multiprogramming.
- In a multiprogramming system, multiple programs submitted by users are allowed to use the processor for a short time, each taking turns and having exclusive time with the processor in order to execute instructions.
- This approach is called as round robin scheduling (RR scheduling). It is one of the oldest, simplest, fairest, and most widely used scheduling algorithms, designed especially for time-sharing systems.
- Vector processing was developed to increase processing performance by operating in a multitasking manner.
- Matrix operations were added to computers to allow a single instruction to manipulate two arrays of numbers performing arithmetic operations. This was valuable in certain types of applications in which data occurred in the form of vectors or matrices.

- The next advancement was the development of symmetric multiprocessing systems (SMP) to address the problem of resource management in master or slave models. In SMP systems, each processor is equally capable and responsible for managing the workflow as it passes through the system.
- Massive parallel processing (MPP) is used in computer architecture circles to refer to a computer system with many independent arithmetic units or entire microprocessors, which run in parallel.

1.4 Principles of Parallel and Distributed Computing

- The two fundamental and dominant models of computing environment are sequential and parallel. The sequential computing era was begun in the 1940s. The parallel and distributed computing era was followed it within a decade.
- The four key elements of computing developed during these eras are architectures, compilers, applications, and problem solving environments.
- Every aspect of this era will undergo a three phase process.
 - Research and Development (R&D)
 - Commercialization
 - Commoditization

1.4.1 Parallel vs distributed computing

- The terms parallel computing and distributed computing are often used interchangeably, even though which meant somewhat different things.
- The term parallel implies a tightly coupled system, whereas distributed refers to a wider class of system which includes tightly coupled systems.
- More specifically, the term parallel computing refers to a model in which the computation is divided among several processors which sharing the same memory.

- The architecture of a parallel computing system is often characterized by the homogeneity of components.
- In parallel computing paradigm, each processor is of the same type and it has the same capability. The shared memory has a single address space, which is accessible to all the processors.
- Processing of multiple tasks simultaneously on multiple processors is called as parallel processing.
- The parallel program consists of multiple active processes or tasks simultaneously solving a given problem.
- A given task is divided into multiple subtasks using a divide and conquer technique, and each subtask is processed on a different Central Processing Unit (CPU).
- Programming on a multiprocessor system using the divide and conquer technique is called parallel programming.
- The term distributed computing encompasses any architecture or system that allows the computation to be broken down into units and executed concurrently on different computing elements, whether these are processors on different nodes, processors on the same computer, or cores within the same processor.
- Therefore, distributed computing includes a wider range of systems and applications than parallel computing and is often considered a most common term.

1.4.2 Elements of parallel computing

- The core elements of parallel processing are CPUs. Based on the number of instruction streams and data streams that can be processed simultaneously, computing systems are classified into four categories proposed by Michael J. Flynn in 1966.

- Single Instruction Single Data systems (SISD)
 - Single Instruction Multiple Data systems (SIMD)
 - Multiple Instruction Single Data systems (MISD)
 - Multiple Instruction, Multiple Data systems (MIMD)
- An SISD computing system is a uniprocessor system capable of executing a single instruction, which operates on a single data stream.

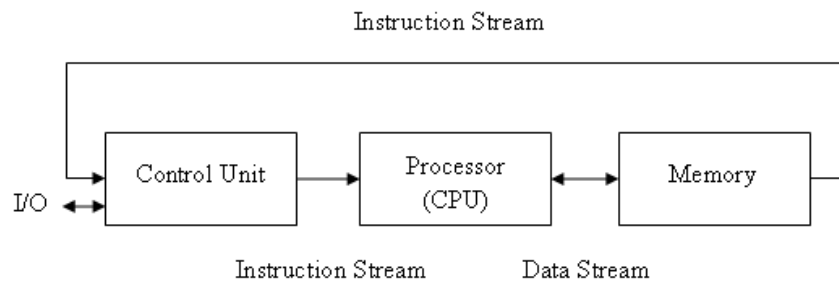


Figure 1.8 SISD

- An SIMD computing system is a multiprocessor system capable of executing the single instruction on all the CPUs but operating on different data streams.

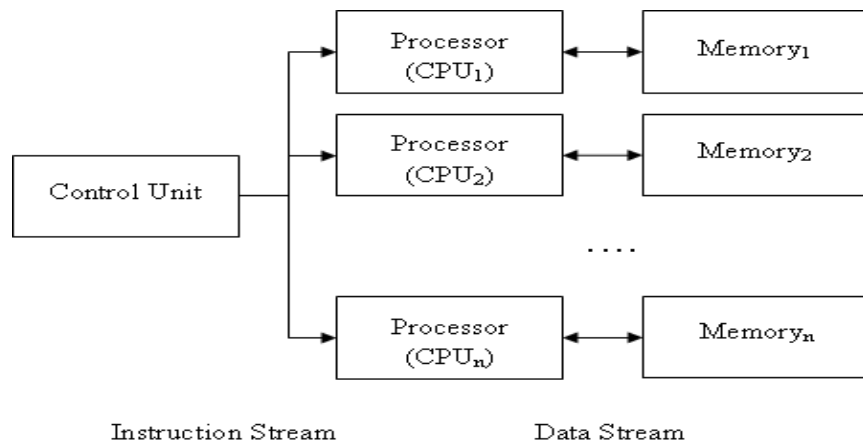


Figure 1.9 SIMD

- An MISD computing system is a multiprocessor system capable of executing different instructions on different processing elements but all of them operating on the same data streams.

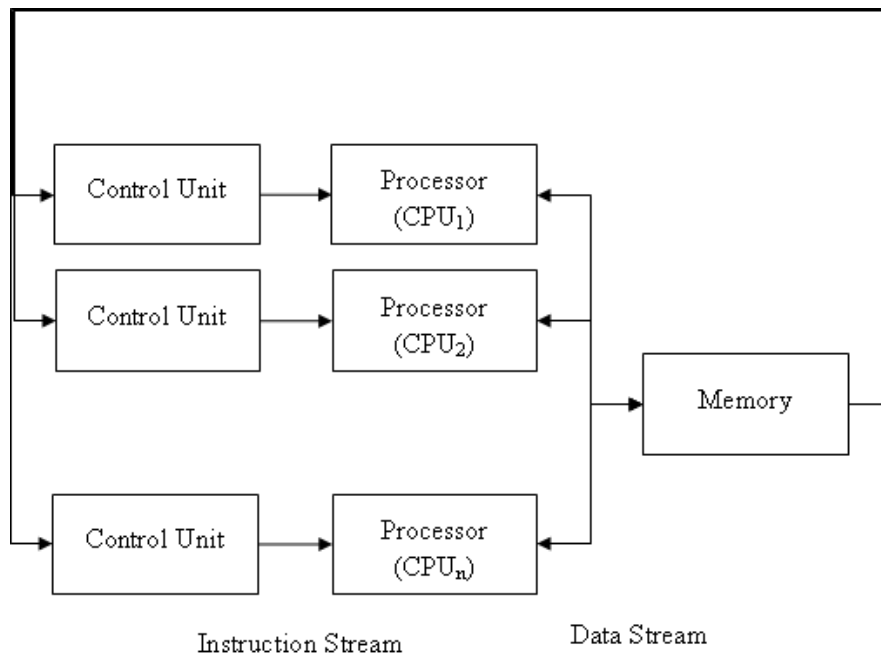


Figure 1.10 MISD

- An MIMD computing system is a multiprocessor system capable of executing multiple instructions on multiple data streams.

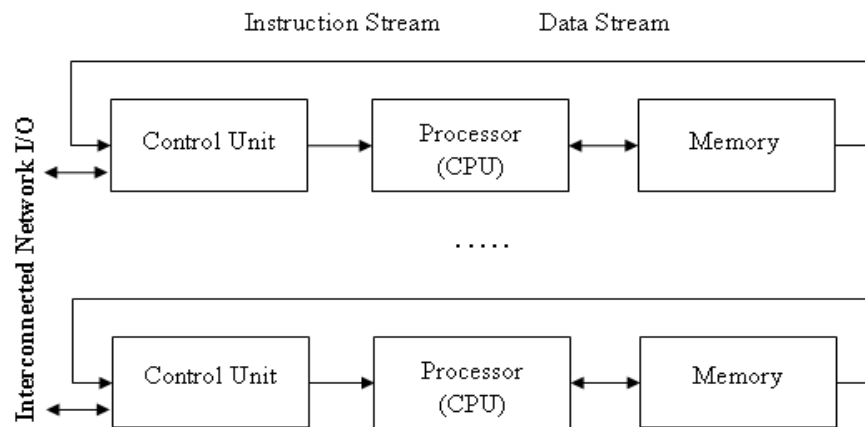


Figure 1.11 MIMD

- MIMD systems are broadly categorized into shared memory MIMD and distributed memory MIMD based on the way processing elements are coupled to the main memory.

- In the shared memory MIMD model, all the processing elements are connected to a single global memory and they all have access to it.
- In the distributed memory MIMD model, all processing elements have a local memory. Systems based on this model are also called loosely coupled multiprocessor systems.
- In general, Failures in a shared memory MIMD affects the entire system, where as this is not the case of the distributed model, in which each of the processing elements can be easily isolated.
- A wide variety of parallel programming approaches are available in computing environment. The most prominent among them are the following:
 - Data parallelism
 - Process parallelism
 - Farmer-and-worker model
- In data parallelism, the divide and conquer methodology is used to split data into multiple sets, and each data set is processed on different processing elements using the same instruction.
- In process parallelism, a given operation has multiple distinct tasks that can be processed on multiple processors.
- In farmer and worker model, a job distribution approach is used in which one processor is configured as master and all other remaining processing elements are designated as slaves. The master assigns jobs to slave processing elements and, on completion, they inform the master, which in turn collects results.
- Parallelism within an application can be detected at several levels such as Large grain (or task level), Medium grain (or control level), Fine grain (data level), Very fine grain (multiple-instruction issue)

- Speed of computation is never increase linearly. It is proportional to the square root of system cost. Therefore, the faster a system becomes, the more expensive it is to increase its speed.

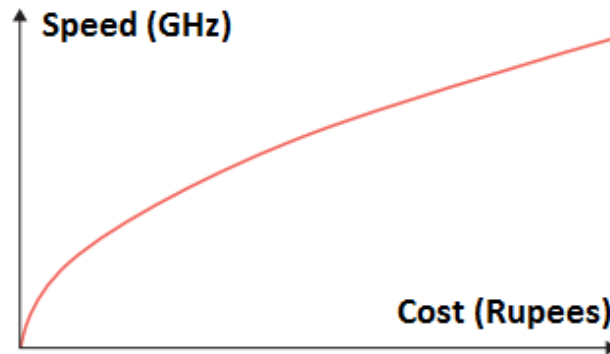


Figure 1.12 Cost versus Speed

- Speed by a parallel computer increases as the logarithm of the number of processors (i.e., $y = k \log(N)$).

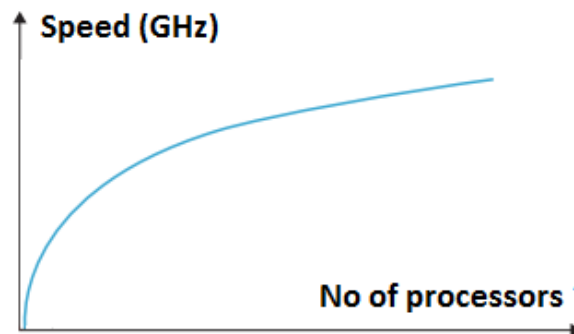


Figure 1.13 No of processors versus Speed

1.4.3 Elements of distributed computing

- A distributed system is the collection of independent computers that appears to its users as a single coherent system.
- A distributed system is the result of the interaction of several components that pass through the entire computing stack from hardware to software.

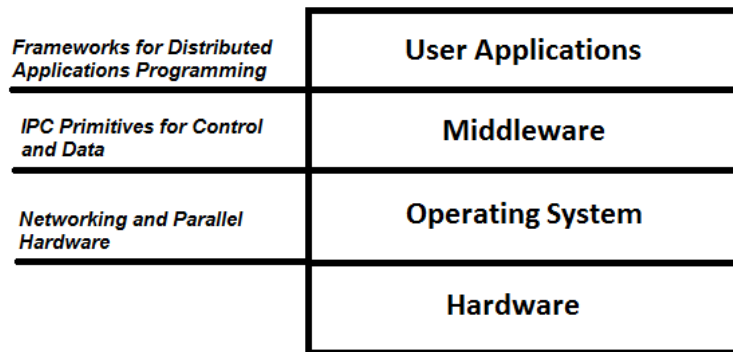


Figure 1.14 A layered view of a distributed system

- At the very bottom layer, computer and network hardware constitute the physical infrastructure.
- The hardware components are directly managed by the operating system, which provides the basic services for inter process communication (IPC), process scheduling and management, and resource management in terms of file system and local devices.
- The use of well-known standards at the operating system level and even more at the hardware and network levels allows easy harnessing of heterogeneous components and their organization into a coherent and uniform system.
- The middleware layer leverages such services to build a uniform environment for the development and deployment of distributed applications.
- The top of the distributed system stack is represented by the applications and services designed and developed to use the middleware.
- In distributed computing, Architectural styles are mainly used to determine the vocabulary of components and connectors that are used as instances of the style together with a set of constraints on how they can be combined.
- Architectural styles are classified into two major classes.

- Software architectural styles
 - System architectural styles
- The first class relates to the logical organization of the software.
 - The second class includes all those styles that describe the physical organization of distributed software systems in terms of their major components.
 - A component represents a unit of software that encapsulates a function or a feature of the system. Examples of components can be programs, objects, processes, pipes, and filters.
 - A connector is a communication mechanism that allows cooperation and coordination among components. Differently from components, connectors are not encapsulated in a single entity, but they are implemented in a distributed manner over many system components.
 - Software architectural styles are based on the logical arrangement of software components.
 - According to Garlan and Shaw, architectural styles are classified as shown in Table 1.1

Category	Most Common Architectural Styles
Data-centered	<ul style="list-style-type: none"> ● Repository ● Blackboard
Data flow	<ul style="list-style-type: none"> ● Pipe and filter ● Batch sequential
Virtual machine	<ul style="list-style-type: none"> ● Rule-based system ● Interpreter

Call and return	<ul style="list-style-type: none"> • Top down systems • Object oriented systems • Layered systems
Independent components	<ul style="list-style-type: none"> • Communicating processes • Event system

Table 1.1 Software Architectural Styles

- The repository architectural style is the most relevant reference model in this category. It is characterized by two main components: the central data structure, which represents the current state of the system, and a collection of independent components, which operate on the central data.
- The batch sequential style is characterized by an ordered sequence of separate programs executing one after the other. These programs are chained together by providing as input for the next program the output generated by the last program after its completion, which is most likely in the form of a file.
- The pipe and filter style is a variation of the previous style for expressing the activity of a software system as a sequence of data transformations. Each component of the processing chain is called a filter, and the connection between one filter and the next is represented by a data stream.
- Rule-Based Style architecture is characterized by representing the abstract execution environment as an inference engine. Programs are expressed in the form of rules or predicates that hold true.
- The core feature of the interpreter style is the presence of an engine that is used to interpret a pseudo code expressed in a format acceptable for the interpreter. The interpretation of the pseudo-program constitutes the execution of the program itself.
- Top Down Style is quite representative of systems developed with imperative programming, which leads to a divide and conquer approach to problem resolution.

- Object Oriented Style encompasses a wide range of systems that have been designed and implemented by leveraging the abstractions of object oriented programming
- The layered system style allows the design and implementation of software systems in terms of layers, which provide a different level of abstraction of the system.
- Each layer generally operates with at most two layers: the one that provides a lower abstraction level and the one that provides a higher abstraction layer.
- In Communicating Processes architectural style, components are represented by independent processes that leverage IPC facilities for coordination management.
- On the other hand, Event Systems architectural style where the components of the system are loosely coupled and connected.
- System architectural styles cover the physical organization of components and processes over a distributed infrastructure. They provide two fundamental reference styles: client/server and peer-to-peer.
- The client/server model features two major components: a server and a client. These two components interact with each other through a network connection using a given protocol. The communication is unidirectional. The client issues a request to the server, and after processing the request the server returns a response.
- The important operations in the client-server paradigm are request, accept (client side), and listen and response (server side).
- The client/server model is suitable in many-to-one scenarios.
- In general, multiple clients are interested in such services and the server must be appropriately designed to efficiently serve requests coming from different clients. This consideration has implications on both client design and server design.

- For the client design, there are two models: Thin client model and Fat client model.
- Thin client model, the load of data processing and transformation is put on the server side, and the client has a light implementation that is mostly concerned with retrieving and returning the data it is being asked for, with no considerable further processing.
- Fat client model, the client component is also responsible for processing and transforming the data before returning it to the user, whereas the server features a fairly light implementation that is mostly concerned with the management of access to the data.
- The three major components in the client-server model are presentation, application logic, and data storage.
- Presentation, application logic, and data maintenance can be seen as conceptual layers, which are more appropriately called tiers.
- The mapping between the conceptual layers and their physical implementation in modules and components allows differentiating among several types of architectures, which go under the name of multi-tiered architectures.
- Two major classes are Two-tier architecture and Three-tier architecture.
- Two-tier architecture partitions the systems into two tiers, which are located one in the client component and the other on the server. The client is responsible for the presentation tier by providing a user interface. The server concentrates the application logic and the data store into a single tier.
- Three-tier architecture separates the presentation of data, the application logic, and the data storage into three tiers. This architecture is generalized into an N-tier model in case it is necessary to further divide the stages composing the application logic and storage tiers.

- The peer-to-peer model introduces a symmetric architecture in which all the components are called as peers, play the same role and incorporate both client and server capabilities of the client/server model.
- The most relevant example of peer-to-peer systems is constituted by file sharing applications such as Gnutella, BitTorrent, and Kazaa.

1.4.4 Models for inter process communication

- There are several different models in which processes can interact with each other; these map to different abstractions for IPC. Among the most relevant models are shared memory, remote procedure call (RPC), and message passing.
- Message passing introduces the concept of a message as the main abstraction of the model. The entities exchanging information explicitly encode in the form of a message the data to be exchanged. The structure and the content of a message vary according to the model. Examples of this model are the Message-Passing Interface (MPI) and OpenMP.
- Remote procedure call paradigm extends the concept of procedure call beyond the boundaries of a single process, thus triggering the execution of code in remote processes. In this case, underlying client/server architecture is implied. A remote process hosts a server component, thus allowing client processes to request the invocation of methods, and returns the result of the execution.

1.4.5 Models for message-based communication

Point-to-point message model

- This model organizes the communication among single components. Each message is sent from one component to another, and there is a direct addressing to identify the message receiver. In a point-to-point communication model it is necessary to know the location of or how to address another component in the system.

Publish-and-subscribe message model

- This model introduces a different strategy, one that is based on notification among components.
- There are two major roles: the publisher and the subscriber.
- There are two major strategies for dispatching the event to the subscribers:
 - Push strategy. In this case it is the responsibility of the publisher to notify all the subscribers. For example, with a method invocation.
 - Pull strategy. In this case the publisher simply makes available the message for a specific event, and it is the responsibility of the subscribers to check whether there are messages on the events that are registered.

Request-reply message model

- The request-reply message model identifies all communication models in which, for each message sent by a process, there is a reply.
- This model is quite popular and provides a different classification that does not focus on the number of the components involved in the communication but rather on how the dynamic of the interaction evolves.

1.5 Technologies for distributed computing

Remote procedure call

- RPC is the fundamental abstraction enabling the execution of procedures on client's request.
- RPC allows extending the concept of a procedure call beyond the boundaries of a process and a single memory address space.
- The called procedure and calling procedure may be on the same system or they may be on different systems in a network.

- An important aspect of RPC is marshaling, which identifies the process of converting parameters and return values into a form that is more suitable to be transported over a network through a sequence of bytes. The term unmarshaling refers to the opposite procedure.

Distributed object frameworks

- Distributed object frameworks extend object-oriented programming systems by allowing objects to be distributed across a heterogeneous network and provide facilities so that they can coherently act as though they were in the same address space.

Service-oriented computing

- Service-oriented computing organizes distributed systems in terms of services, which represent the major abstraction for building systems.
- Service orientation expresses applications and software systems as aggregations of services that are coordinated within a service-oriented architecture (SOA).
- SOA is an architectural style supporting service orientation. It organizes a software system into a collection of interacting services.
- SOA encompasses a set of design principles that structure system development and provide means for integrating components into a coherent and decentralized system.
- SOA-based computing packages functionalities into a set of interoperable services, which can be integrated into different software systems belonging to separate business domains.
- There are two major roles within SOA: the service provider and the service consumer.

1.5 Cloud Characteristics

From the cloud computing's various definitions; a certain set of key characteristics emerges. Figure 1.15 illustrates various key characteristics related to cloud computing paradigm.

1.5.1 On-demand Provisioning

- On-demand provisioning is the single most important characteristic of cloud computing, it allows the users to request or release resources whenever they want.
- These demands are thereafter automatically granted by a cloud provider's service and the users are only charged for their usage, i.e., the time they were in possession of the resources.
- The reactivity of a cloud solution, with regard to resource provisioning is indeed of prime importance as it is closely related to the cloud's pay-as-you-go business model.
- It is one of the important and valuable features of Cloud Computing as the user can continuously monitor the server uptime, capabilities, and allotted network storage. With this feature, the user can also monitor the computing capabilities.

1.5.2 Universal Access

- Resources in the cloud need not only be provisioned rapidly but also accessed and managed universally, using standard Internet protocols, typically via RESTful web services.
- This enables the users to access their cloud resources using any type of devices, provided they have an Internet connection.
- Universal access is a key feature behind the cloud's widespread adoption, not only by professional actors but also by the general public that is nowadays familiar with cloud based solutions such as cloud storage or media streaming.

- Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms such as mobile phones, tablets, laptops, and workstations.

1.5.3 Enhanced Reliability

- Cloud computing enables the users to enhance the reliability of their applications.
- Reliability is already built in many cloud solutions via storage redundancy.
- Cloud providers usually have more than one data center and further reliability can be achieved by backing data up in different locations.
- This can also be used to ensure service availability, in the case of routine maintenance operations or the rarer case of a natural disaster.
- The user can achieve further reliability using the services of different cloud providers.

1.5.4 Measured Services

- Cloud computing refers generally to paid services.
- The customers are entitled to a certain quality of service, guaranteed by the Service Level Agreement that they should be able to supervise.
- Therefore, cloud providers offer monitoring tools, either using a graphical interface or via an API.
- These tools also help the providers themselves for billing and management purposes.

1.5.5 Multitenancy

- As the grid before, the cloud's resources are shared by different simultaneous users. These users had to reserve in advance a fixed number of physical machines for a fixed amount of time.
- In virtualized data centers, a user's provisioned resources no longer correspond to the physical infrastructure and can be dispatched over multiple physical machines.
- They can also run alongside other users' provisioned resources thus requiring a lesser amount of physical resources. Consequently, important energy savings can be made by shutting down the unused resources or putting them in energy saving mode.

1.5.6 Resource pooling

- The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.

- There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter).
- Examples of resources include storage, processing, memory, and network bandwidth.

1.5.7 Rapid elasticity and Scalability

- Elasticity is the ability of a system to include and exclude resources like CPU cores, memory, Virtual Machine and container instances to adapt to the load variation in real time.
- Elasticity is a dynamic property for cloud computing. There are two types of elasticity. Horizontal and Vertical.
- Horizontal elasticity consists in adding or removing instances of computing resources associated with an application.
- Vertical elasticity consists in increasing or decreasing characteristics of computing resources, such as CPU time, cores, memory, and network bandwidth.
- There are other terms such as scalability and efficiency, which are associated with elasticity but their meaning is different from elasticity while they are used interchangeably in some cases.
- Scalability is the ability of the system to sustain increasing workloads by making use of additional resources, it is time independent and it is similar to the provisioning state in elasticity but the time has no effect on the system (static property).
- The following equation that summarizes the elasticity concept in cloud computing.

Auto scaling = Scalability +Automation

Elasticity = Auto scaling + Optimization

- It means that the elasticity is built on top of scalability. It can be considered as an automation of the concept of scalability, however, it aims to optimize at best and as quickly as possible the resources at a given time.
- Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand.
- To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

1.5.8 Easy Maintenance

- The servers are easily maintained and the downtime is very low and even in some cases, there is no downtime.
- Cloud Computing comes up with an update every time by gradually making it better. The updates are more compatible with the devices and perform faster than older ones along with the bugs which are fixed.

1.5.9 High Availability

- The capabilities of the Cloud can be modified as per the use and can be extended a lot. It analyzes the storage usage and allows the user to buy extra Cloud storage if needed for a very small amount.

1.5.10 Security

- Cloud Security is one of the best features of cloud computing. It creates a snapshot of the data stored so that the data may not get lost even if one of the servers gets damaged.
- The data is stored within the storage devices, which cannot be hacked and utilized by any other person. The storage service is quick and reliable

