CMR
INSTITUTE OF
TECHNOLOGY

USN | | | | | | | | | |

Internal Assessment Test 1 – Nov. 2021

Sub:	**Cloud Computing**							**Sub Code:**	**20MCA342**
Date:	**12/11//2021**	**Duration:**	**90 min’s**	**Max Marks:**	**50**	**Sem:**	**III**	**Branch:**	**MCA**

Note : Answer FIVE FULL Questions, choosing ONE full question from each Module

		MARKS	OBE	
			CO	RBT
	PART I			
1	Explain with a diagram the evaluation of computing platform.	[10]	CO1	L1,L2
	OR			
2	Explain the concepts of IoT and CPS.	[6+4]	CO1	L2
	PART II			
3	Discuss the characteristics of a P2P system and overlay network.	[10]	CO1	L2
	OR			
4	What are the different deployment models of cloud computing?	[10]	CO1	L1

CMR
INSTITUTE OF
TECHNOLOGY

USN | | | | | | | | | |

Internal Assessment Test 1 – Nov. 2021

Sub:	**Cloud Computing**							**Sub Code:**	**20MCA342**
Date:	**12/11//2021**	**Duration:**	**90 min’s**	**Max Marks:**	**50**	**Sem:**	**III**	**Branch:**	**MCA**

Note : Answer FIVE FULL Questions, choosing ONE full question from each Module

		MARKS	OBE	
			CO	RBT
	PART I			
1	Explain with a diagram the evaluation of computing platform.	[10]	CO1	L1
	OR			
2	Explain the concepts of IoT and CPS.	[6+4]	CO1	L1
	PART II			
3	Discuss the characteristics of a P2P system and overlay network.	[10]	CO1	L2
	OR			
4	What are the different deployment models of cloud computing?	[10]	CO1	L1

PART III

5	Draw a diagram and discuss the cloud computing reference model.	[10]	CO1	L2
	OR			
6	Describe how cloud computing is cost effective compared to traditional IT computing. Discuss the other benefits of using cloud.	[5+5]	CO1	L2

PART IV

7	Discuss Levels of Parallelism and Laws of caution.	[6+4]	CO2	L1
	OR			
8	Compare the characteristics of parallel and distributed system. Draw and explain the layered view of distributed system.	[5+5]	CO2	L2

PART V

9	What are the various hardware architectures for parallel processing? Discuss each type and draw appropriate diagrams	[10]	CO2	L2
	OR			
10	Explain the software architectural styles.	[10]	CO2	L2

PART III

5	Draw a diagram and discuss the cloud computing reference model.	[10]	CO1	L2
	OR			
6	Describe how cloud computing is cost effective compared to traditional IT computing. Discuss the other benefits of using cloud.	[5+5]	CO1	L2

PART IV

7	Discuss Levels of Parallelism and Laws of caution.	[6+4]	CO2	L2
	OR			
8	Compare the characteristics of parallel and distributed system. Draw and explain the layered view of distributed system.	[5+5]	CO2	L3

PART V

9	What are the various hardware architectures for parallel processing? Discuss each type and draw appropriate diagrams	[10]	CO2	L3
	OR			
10	Explain the software architectural styles.	[10]	CO2	L3

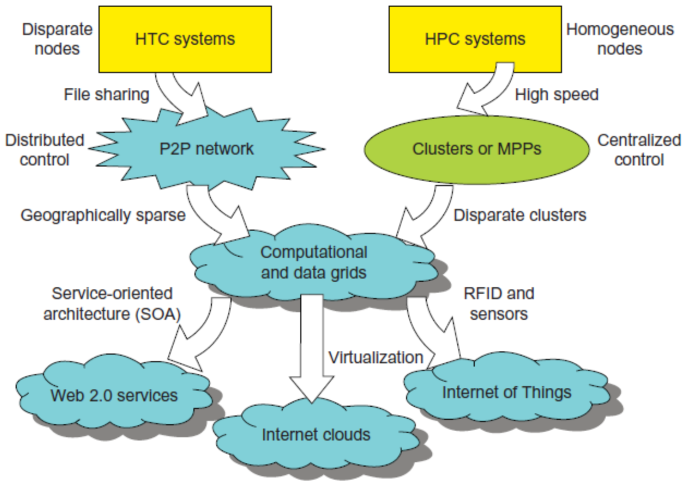
CMR
INSTITUTE OF
TECHNOLOGY

USN | | | | | | | | | | |

Internal Assessment Test 1– Nov 2021

Sub:	Cloud Computing					Sub Code:	**20MCA342**	Branch:	MCA
Date:	12/11/2021	Duration:	90 min's	Max Marks:	50	Sem	III		OBE

Q1. Explain with a diagram the evolution of computing platform



Computer technology has gone through five generations of development, with each generation lasting from 10 to 20 years. Successive generations are overlapped in about 10 years. For instance, from 1950 to 1970, a handful of mainframes, including the IBM 360 and CDC 6400, were built to satisfy the demands of large businesses and government organizations. From 1960 to 1980, lower-cost mini-computers such as the DEC PDP 11 and VAX Series became popular among small businesses and on college campuses.

From 1970 to 1990, we saw widespread use of personal computers built with VLSI microprocessors. From 1980 to 2000, massive numbers of portable computers and pervasive devices appeared in both wired and wireless applications. Since 1990, the use of both HPC and HTC systems hidden in clusters, grids, or Internet clouds has proliferated. These systems are employed by both consumers and high-end web-scale computing and information services.

The general computing trend is to leverage shared web resources and massive amounts of data over the Internet. The figure above illustrates the evolution of HPC and HTC systems. On the HPC side, supercomputers (massively parallel processors or MPPs) are gradually replaced by clusters of cooperative computers out of a desire to share computing resources. The cluster is often a collection of homogeneous compute nodes that are physically connected in close range to one another.

On the HTC side, peer-to-peer (P2P) networks are formed for distributed file sharing and content delivery applications. A P2P system is built over many client machines. Peer machines are globally distributed in nature. P2P, cloud computing, and web service platforms are more focused on HTC applications than on HPC applications. Clustering and P2P technologies lead to the development of computational grids or data grids.

Q2. Explain the concepts of IoT and CPS

Internet of Things

The traditional Internet connects machines to machines or web pages to web pages. The concept of the IoT was introduced in 1999 at MIT. The IoT refers to the networked interconnection of everyday objects, tools, devices, or computers. One can view the IoT as a wireless network of sensors that interconnect all things in our daily life. These things can be large or small and they vary with respect to time and place. The idea is to tag every object using RFID or a related sensor or electronic technology such as GPS.

With the introduction of the IPv6 protocol, 2^{128} IP addresses are available to distinguish all the objects on Earth, including all computers and pervasive devices. The IoT researchers have estimated that every human being will be surrounded by 1,000 to 5,000 objects. The IoT needs to be designed to track 100 trillion static or moving objects simultaneously. The IoT demands universal addressability of all of the objects or things. To reduce the complexity of identification, search, and storage, one can set the threshold to filter out fine-grain objects. The IoT obviously extends the Internet and is more heavily developed in Asia and European countries.

In the IoT era, all objects and devices are instrumented, interconnected, and interacted with each other intelligently. This communication can be made between people and things or among the things themselves. Three communication patterns co-exist: namely H2H (human-to-human), H2T (human-to-thing), and T2T (thing-to-thing). Here things include machines such as PCs and mobile phones. The idea here is to connect things (including human and machine objects) at any time and any place intelligently with low cost. Any place connections include at the PC, indoor (away from PC), outdoors, and on the move. Any time connections include daytime, night, outdoors and indoors, and on the move as well.

The dynamic connections will grow exponentially into a new dynamic network of networks, called the Internet of Things (IoT). The IoT is still in its infancy stage of development. Many proto-type IoTs with restricted areas of coverage are under experimentation at the time of this writing. Cloud computing researchers expect to use the cloud and future Internet technologies to support fast, efficient, and intelligent interactions among humans, machines, and any objects on Earth. A smart Earth should have intelligent cities, clean water, efficient power, convenient transportation, good food supplies, responsible banks, fast telecommunications, green IT, better schools, good health care, abundant resources, and so on. This dream living environment may take some time to reach fruition at different parts of the world.

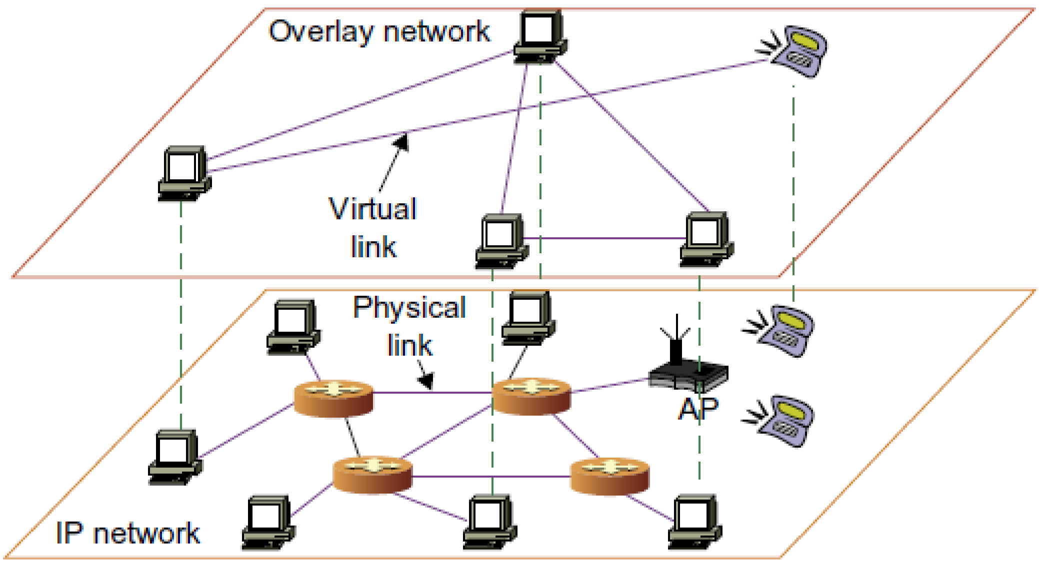
Cyber Physical Systems

A cyber-physical system (CPS) is the result of interaction between computational processes and the physical world. A CPS integrates "cyber" (heterogeneous, asynchronous) with "physical" (concur-rent and information-dense) objects. A CPS merges the "3C" technologies of computation, communication, and control into an intelligent closed feedback system between the physical world and the information world, a concept which is actively explored in the United States. The IoT emphasizes various networking connections among physical objects, while the CPS emphasizes exploration of virtual reality (VR) applications in the physical world. We may transform how we interact with the physical world just like the Internet transformed how we interact with the virtual world.

Q3. Discuss the characteristics of a P2P system and overlay network

In a P2P system, every node acts as both a client and a server, providing part of the system resources. Peer machines are simply client computers connected to the Internet. All client machines act autonomously to join or leave the system freely. This implies that no master-slave relationship exists among the peers. No central coordination or central database is needed. In other words, no peer machine has a global view of the entire P2P system. The system is self-organizing with distributed control.

The figure below shows the architecture of a P2P network at two abstraction levels. Initially, the peers are totally unrelated. Each peer machine joins or leaves the P2P network voluntarily. Only the participating peers form the physical network at any time. Unlike the cluster or grid, a P2P network does not use a dedicated interconnection network. The physical network is simply an ad hoc network formed at various Internet domains randomly using the TCP/IP and NAI protocols. Thus, the physical network varies in size and topology dynamically due to the free membership in the P2P network.

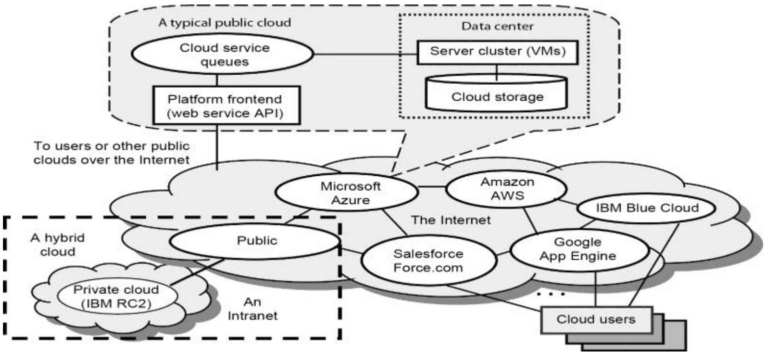


Overlay Networks

Data items or files are distributed in the participating peers. Based on communication or file-sharing needs, the peer IDs form an overlay network at the logical level. This overlay is a virtual network formed by mapping each physical machine with its ID, logically, through a virtual mapping as shown in the figure. When a new peer joins the system, its peer ID is added as a node in the overlay network. When an existing peer leaves the system, its peer ID is removed from the overlay network automatically. Therefore, it is the P2P overlay network that characterizes the logical connectivity among the peers.

There are two types of overlay networks: unstructured and structured. An unstructured overlay network is characterized by a random graph. There is no fixed route to send messages or files among the nodes. Often, flooding is applied to send a query to all nodes in an unstructured overlay, thus resulting in heavy network traffic and nondeterministic search results. Structured overlay net-works follow certain connectivity topology and rules for inserting and removing nodes (peer IDs) from the overlay graph. Routing mechanisms are developed to take advantage of the structured overlays.

Q4. What are the different deployment models of cloud computing?



What is a public cloud?

A public cloud is built over the Internet and can be accessed by any user who has paid for the service. Public clouds are owned by service providers and are accessible through a subscription. The callout box in top of Figure shows the architecture of a typical public cloud. Many public clouds are available, including Google App Engine (GAE), Amazon Web Services (AWS), Microsoft Azure, IBM Blue Cloud, and Salesforce.com's Force.com. The providers of the aforementioned clouds are commercial providers that offer a publicly accessible remote interface for creating and managing VM instances within their proprietary infrastructure. A public cloud depublicrs a selected set of business processes. The application and infrastructure services are offered on a flexible price-per-use basis.

Advantages of public clouds:

- Lower costs—no need to purchase hardware or software and you pay only for the service you use.
- No maintenance—your service provider provides the maintenance.
- Near-unlimited scalability—on-demand resources are available to meet your business needs.
- High reliability—a vast network of servers ensures against failure.

What is a private cloud?

A private cloud is built within the domain of an intranet owned by a single organization. Therefore, it is client owned and managed, and its access is limited to the owning clients and their partners. Its deployment was not meant to sell capacity over the Internet through publicly accessible interfaces. Private clouds give local users a flexible and agile private infrastructure to run service workloads within their administrative domains. A private cloud is supposed to deliver more efficient and convenient cloud services. It may impact the cloud standardization, while retaining greater customization and organizational control.

Advantages of private clouds:

- More flexibility—your organisation can customise its cloud environment to meet specific business needs.
- Improved security—resources are not shared with others, so higher levels of control and security are possible.
- High scalability—private clouds still afford the scalability and efficiency of a public cloud.

What is a hybrid cloud?

A hybrid cloud is built with both public and private clouds, as shown at the lower-left corner of Figure. Private clouds can also support a hybrid cloud model by supplementing local infrastructure with computing capacity from an external public cloud. For example, the Research Compute Cloud (RC2) is a private cloud, built by IBM, that interconnects the computing and IT resources at eight IBM Research Centers scattered throughout the United States, Europe, and Asia. A hybrid cloud provides access to clients, the partner network, and third parties.

Advantages of hybrid clouds:

- Control—your organisation can maintain a private infrastructure for sensitive assets.
- Flexibility—you can take advantage of additional resources in the public cloud when you need them.
- Cost-effectiveness—with the ability to scale to the public cloud, you pay for extra computing power only when needed.
- Ease—transitioning to the cloud does not have to be overwhelming because you can migrate gradually—phasing in workloads over time.

Q5. Draw a diagram and discuss the cloud computing reference model.

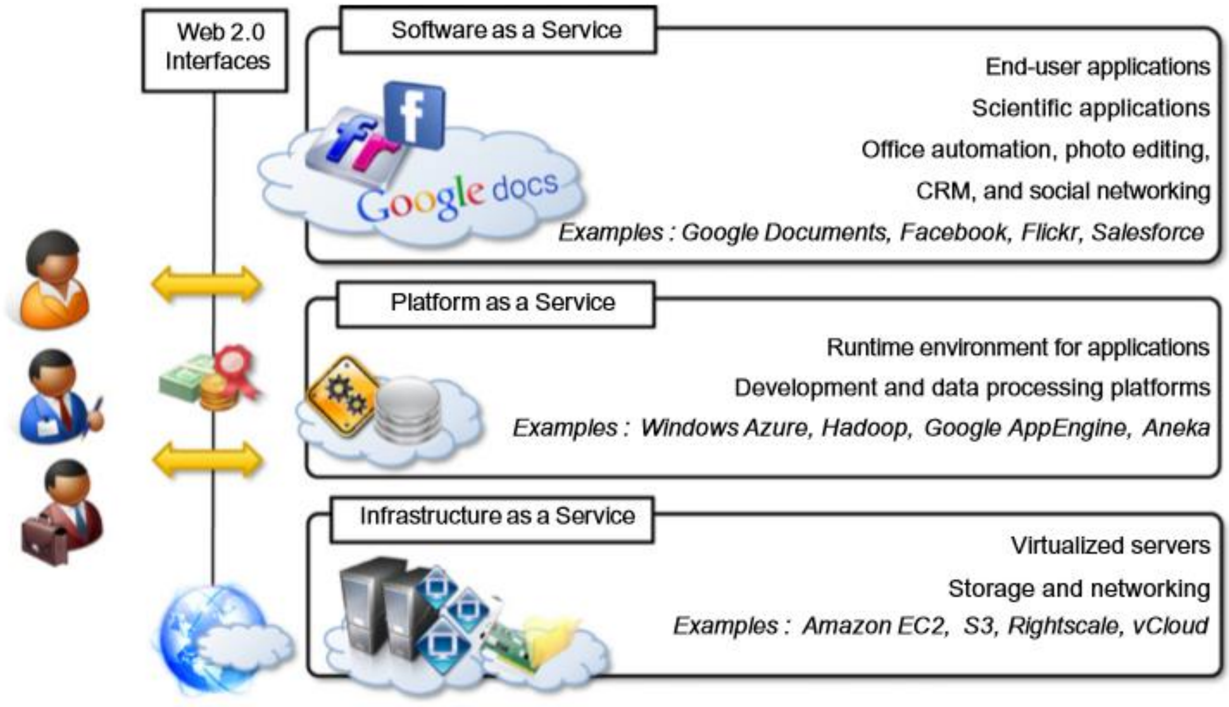


FIGURE 1.5

The Cloud Computing Reference Model.

A fundamental characteristic of cloud computing is the capability to deliver, on demand, a variety of IT services that are quite diverse from each other. This variety creates different perceptions of what cloud computing is among users. Despite this lack of uniformity, it is possible to classify cloud computing services offerings into three major categories: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). These categories are related to each other as described in Figure 1.5, which provides an organic view of cloud computing. We refer to this diagram as the Cloud Computing Reference Model, and we will use it throughout the book to explain the technologies and introduce the relevant research on this phenomenon. The model organizes the wide range of cloud computing services into a layered view that walks the computing stack from bottom to top. At the base of the stack, Infrastructure-as-a-Service solutions deliver infrastructure on demand in the form of virtual hardware, storage, and networking. Virtual hardware is utilized to provide compute on demand in the form of virtual machine instances. These are created at users' request on the provider's infrastructure, and users are given tools and interfaces to configure the software stack installed in the virtual machine. The pricing model is usually defined in terms of dollars per hour, where the hourly cost is influenced by the characteristics of the virtual hardware. Virtual storage is delivered in the form of raw disk space or object store. The former complements a virtual hardware offering that requires persistent storage. The latter is a more high-level abstraction for storing enti ties rather than files. Virtual networking identifies the collection of services that manage the net working among virtual instances and their connectivity to the Internet or private networks. Platform-as-a-Service solutions are the next step in the stack. They deliver scalable and elastic runtime environments on demand and host the execution of applications. These services are backed by a core middleware platform that is responsible for creating the abstract environment where applications are deployed and executed. It is the responsibility of the service provider to provide scalability and to manage fault tolerance, while users are requested to focus on the logic of the application developed by leveraging the provider's APIs and libraries. This approach increases the level of abstraction at which cloud computing is leveraged but also constrains the user in a more controlled environment. At the top of the stack, Software-as-a-Service solutions provide applications and services on demand. Most of the common functionalities of desktop applications—such as office automation, document management, photo editing, and customer relationship management (CRM) software—are replicated on the provider's infrastructure and made more scalable and accessible through a browser on demand. These applications are shared across multiple users whose interaction is isolated from the other users. The SaaS layer is also the area of social networking Websites, which leverage cloud-based infrastructures to sustain the load generated by their popularity. Each layer provides a different service to users. IaaS solutions are sought by users who want to leverage cloud computing from building dynamically scalable computing systems requiring a specific software stack. IaaS services are therefore used to develop scalable Websites or for background processing. PaaS solutions provide scalable programming platforms for developing applications and are more appropriate when new systems have to be developed. SaaS solutions target mostly end users who want to benefit from the elastic scalability of the cloud without doing any software development, installation, configuration, and maintenance. This solution is appropriate when there are existing SaaS services that fit users needs (such as email, document management, CRM, etc.) and a minimum level of customization is needed.

Q6. Describe how cloud computing is cost effective compared to traditional IT computing.

In terms of their running costs, cloud computing often comes out on top. And it's definitely a cost-effective IT solution.

Firstly, a cloud-based service can be hosted and have core aspects maintained by the third-party service provider. This means you don't need to hire a large in-house IT team to manage its daily back office operations.

The host will take care of running regular security upgrades and health checks and have built in failover and continuity systems. As such, this reduces the chances of the server going down. Which would result in unnecessary downtime for your company.

Secondly, cloud-based software operates in such a way that you only pay for the storage space and processing power that you use. For instance, think of it like how you would pay for your monthly utility bills.

For new and small businesses, this can be particularly appealing. As it makes those first few years of growth as cost-effective as possible.

With traditional IT storage, you need to purchase additional hardware and server space upfront to allow your business to grow. Hardware like this can be pretty costly. Especially if you end up investing in resources that you never end up using.

In addition, despite your IT hardware being an expensive asset to purchase in the first place, it's also the one with a value that significantly decreases year on year.

Any hardware you do purchase will have a low return on investment and will eventually need replacing. So do bear this mind when considering the cost of implementing a traditional IT infrastructure.

Q7. Discuss Levels of Parallelism and Laws of caution.

Levels of parallelism are decided based on the lumps of code (grain size) that can be a potential candidate for parallelism. Table 2.1 lists categories of code granularity for parallelism. All these approaches have a common goal: to boost processor efficiency by hiding latency. To conceal latency, there must be another thread ready to run whenever a lengthy operation occurs. The idea is to execute concurrently two or more single-threaded applications, such as compiling, text format ting, database searching, and device simulation.

As shown in the table and depicted in Figure 2.7, parallelism within an application can be detected at several levels:

- Large grain (or task level)
- Medium grain (or control level)
- Fine grain (data level)
- Very fine grain (multiple-instruction issue)

In this book, we consider parallelism and distribution at the top two levels, which involve the distribution of the computation among multiple threads or processes

Laws of Caution: Now that we have introduced some general aspects of parallel computing in terms of architectures and models, we can make some considerations that have been drawn from experience designing and implementing such systems. These considerations are guidelines that can help us understand how much benefit an application or a software system can gain from parallelism. In particular, what we need to keep in mind is that parallelism is used to perform multiple activities together so that the system can increase its throughput or its speed. But the relations that control the increment of speed are not linear. For example, for a given n processors, the user expects speed to be increased by n times. This is an ideal situation, but it rarely happens because of the communication overhead.

Here are two important guidelines to take into account:

- Speed of computation is proportional to the square root of system cost; they never increase linearly. Therefore, the faster a system becomes, the more expensive it is to increase its speed (Figure 2.8).
- Speed by a parallel computer increases as the logarithm of the number of processors (i.e., y = k*log(N)).

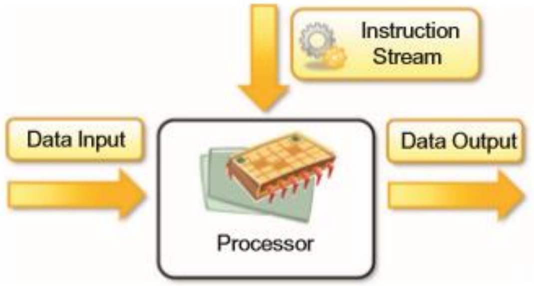
Q8) Compare the characteristics of parallel and distributed system. Draw and explain the layered view of distributed system.

Functionality, Applications	Computer Clusters	Peer-to-Peer Networks	Data/ Computational Grids	Cloud Platforms
Architecture, Network Connectivity, and Size	Network of compute nodes interconnected by SAN, LAN, or WAN hierarchically	Flexible network of client machines logically connected by an overlay network	Heterogeneous clusters interconnected by high-speed network links over selected resource sites	Virtualized cluster of servers over data centers via SLA
Control and Resources Management	Homogeneous nodes with distributed control, running UNIX or Linux	Autonomous client nodes, free in and out, with self-organization	Centralized control, server-oriented with authenticated security	Dynamic resource provisioning of servers, storage, and networks
Applications and Network-centric Services	High-performance computing, search engines, and web services, etc.	Most appealing to business file sharing, content delivery, and social networking	Distributed supercomputing, global problem solving, and data center services	Upgraded web search, utility computing, and outsourced computing services
Representative Operational Systems	Google search engine, SunBlade, IBM Road Runner, Cray XT4, etc.	Gnutella, eMule, BitTorrent, Napster, KaZaA, Skype, JXTA	TeraGrid, GriPhyN, UK EGEE, D-Grid, ChinaGrid, etc.	Google App Engine, IBM Bluecloud, AWS, and Microsoft Azure

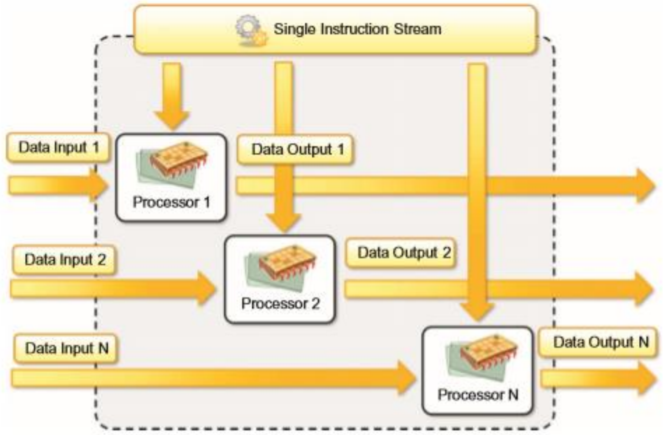
Q9) What are the various hardware architectures for parallel processing? Discuss each type and draw appropriate diagrams

The core elements of parallel processing are CPUs. Based on the number of instruction and data streams that can be processed simultaneously, computing systems are classified into the following four categories:

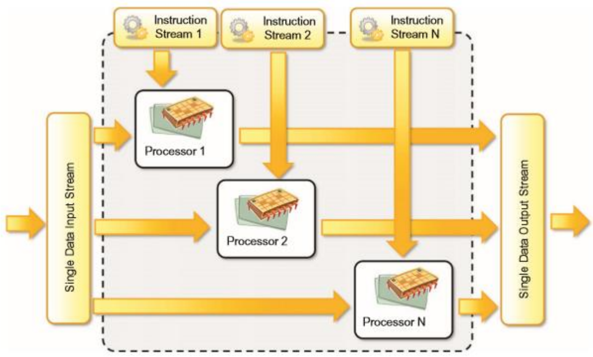
• Single-instruction, single-data (SISD) systems - An SISD computing system is a uniprocessor machine capable of executing a single instruction,
which operates on a single data stream (see Figure 2.2). In SISD, machine instructions are processed sequentially; hence computers adopting this model are popularly called sequential computers. Most conventional computers are built using the SISD model. All the instructions and data to be processed have to be stored in primary memory. The speed of the processing element in the SISD model is limited by the rate at which the computer can transfer information internally. Dominant representative SISD systems are IBM PC, Macintosh, and workstations



• Single-instruction, multiple-data (SIMD) systems - An SIMD computing system is a multiprocessor machine capable of executing the same instruction on all the CPUs but operating on different data streams (see Figure 2.3). Machines based on an SIMD model are well suited to scientific computing since they involve lots of vector and matrix operations. For instance, statements such as
$C_i = A_i * B_i$
can be passed to all the processing elements (PEs); organized data elements of vectors A and B can be divided into multiple sets (N-sets for N PE systems); and each PE can process one data set. Dominant representative SIMD systems are Cray's vector processing machine and Thinking Machines' cm.

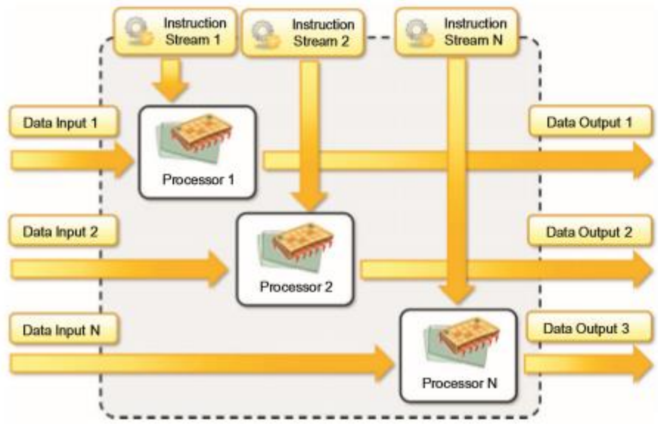


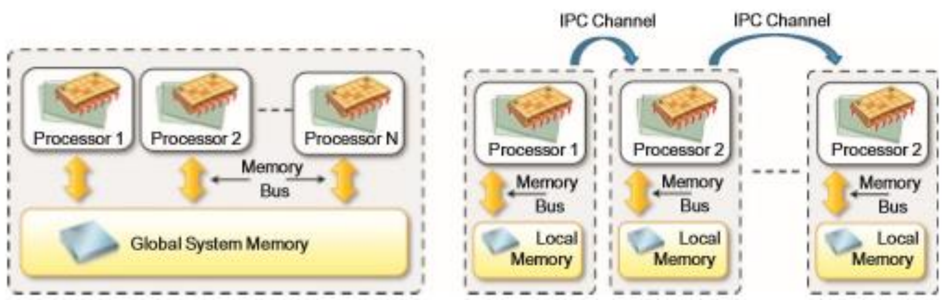
• Multiple-instruction, single-data (MISD) systems - An MISD computing system is a multiprocessor machine capable of executing different instructions on different PEs but all of them operating on the same data set (see Figure 2.4). For instance, statements such as perform different operations on the same data set. Machines built using the MISD model are not useful in most of the applications; a few machines are built, but none of them are available commercially. They became more of an intellectual exercise than a practical configuration.



• Multiple-instruction, multiple-data (MIMD) systems - An MIMD computing system is a multiprocessor machine capable of executing multiple instructions on multiple data sets (see Figure 2.5). Each PE in the MIMD model has separate instruction and data streams; hence machines built using this model are well suited to any kind of application. Unlike SIMD and MISD machines, PEs in MIMD machines work asynchronously. MIMD machines are broadly categorized into shared-memory MIMD and distributed-memory MIMD based on the way PEs are coupled to the main memory.

Shared memory MIMD machines





In the shared memory MIMD model, all the PEs are connected to a single global memory and they all have access to it (see Figure 2.6). Systems based on this model are also called tightly coupled multiprocessor systems. The communication between PEs in this model takes place through the shared memory; modification of the data stored in the global memory by one PE is visible to all other PEs. Dominant representative shared memory MIMD systems are Silicon Graphics machines and Sun/IBM's SMP (Symmetric Multi-Processing).

Distributed memory MIMD machines

In the distributed memory MIMD model, all PEs have a local memory. Systems based on this model are also called loosely coupled multiprocessor systems. The communication between PEs in this model takes place through the interconnection network (the interprocess communication channel, or IPC). The network connecting PEs can be configured to tree, mesh, cube, and so on. Each PE operates asynchronously, and if communication/synchronization among tasks is necessary, they can do so by exchanging messages between them.

The shared-memory MIMD architecture is easier to program but is less tolerant to failures and harder to extend with respect to the distributed memory MIMD model. Failures in a shared-memory MIMD affect the entire system, whereas this is not the case of the distributed model, in which each of the PEs can be easily isolated. Moreover, shared memory MIMD architectures are less likely to scale because the addition of more PEs leads to memory contention. This is a situation that does not happen in the case of distributed memory, in which each PE has its own memory. As a result, distributed memory MIMD architectures are most popular today

Q10) Explain the software architectural styles

Category	Most Common Architectural Styles
Data-centered	Repository Blackboard
Data flow	Pipe and filter Batch sequential
Virtual machine	Rule-based system Interpreter
Call and return	Main program and subroutine call/top-down systems Object-oriented systems Layered systems
Independent components	Communicating processes Event systems