Module 1 -

Introduction:

- **Cloud computing is important because data can be accessed anywhere and anytime.**
- **Can access a large pool of resources online.**

Computing:

- **Computing is the process of using computer technology to complete a goal-oriented task.**
- **In cloud computing most of the process is carried out online, that is, over the cloud.**
- **Computing consists of three things:**
    - **Managing,**
    - **Processing**
    - **And communicating information.**

Computing Evolution:

- **More computational capacity - we have more resources, GPU, high RAM, ROM now, so our computational capacity has increased over years.**
- **Less cost - pay per use model, just pay for what you use.**
- **Less size - previously when mainframe computing was in effect a computer would have taken the whole room and the computers barely take any space(because of transistors, integrated circuits.).**
- **More Efficient - computing has been made more efficient because of the increase in the availability of resources for a computer.**

Computers Generation:

First Generation	**Vacuum tubes**
Second Generation	**Transistors**
Third Generation	**Integrated circuits.**
Fourth Generation	**Microprocessors**
Five Generation	**Artificial intelligence**

Data processing methods:

- Single user processing
    - A single user can only use a single application or a single user can only run one process at a time.
    - Two different processes cannot be run at the same time.
- Time-sharing processing
    - Multiple processes can be run concurrently but they are given a time slot and the processes can only run in the given time slot.
    - Example: Round table scheduling algorithm.
    - For example, you want to execute three different processes p1, p2, p3 then you allow the process p1 to run for the first microsecond and then switch to process p2 for the next microsecond, and p3 in the third microsecond and p1 for the fourth microsecond again and so on.
    - Switching time is so less for the human eye to recognize.
    - Similar to time-division multiplexing.
    - Disadvantage: For example, in the previous example p1 just needs 2 microseconds for execution but because of time-sharing we have to wait for 4 microseconds to finish executing p1, that is, execution time increases, wait time increases.
- Multiple processing
    - Share your system resources for executing multiple processes at once.
    - Different processes run on different processors.
    - Threading.
    - But if the number of processes needed to run exceeds the number of processors then each process can get the divided processes that need to be run on it and apply time sharing on it.
    - Overall response time, waiting time would be reduced.
- Distributed processing
    - Parallelizing task.
    - Divide the task and ask various distributed computers to do their part of the work and merge the final result.
    - Distributed computers are connected through a network.

Mainframe Computing:

- First-generation computers, use vacuum tubes.
- Large sizes, some would even fit a whole room.
- Users would connect to the mainframe using terminals.
- Designed for scientific research, not for personal use.

Computing Environment:

When many computers are connected through a network and communicate with each other to handle multiple issues a computing environment is created.

Types of Computing Environment:

- Personal computing environment
    - Only a single user can access the whole computing environment.
    - Identical to our personal computers, laptops, etc.
    - Only meant for a single user, designed for personal use.
    - Size is reduced compared to mainframe computers.
    - Third-generation computers.
    - Every system has dedicated memory, computational power, and storage capacity.
    - Advantages:
        - Security - more secure, since you are only able to access.
    - Disadvantages:
        - Limited Storage.
        - Not utilizing the resources efficiently, not utilizing whole resources available.
- Time-sharing computing environment
    - To address the limitations of personal computing, in a time-sharing computing environment, sources are utilized to their fullest because multiple users share the environment periodically.
    - Advantages:
        - Effective resource utilization.
    - Disadvantages:
        - Waiting time is high (similar to time-sharing processing)
        - Response time is also high.
            - For example, there are 3 processes p1,p2,p3 and they take 1sec,2secs,3secs to finish execution respectively.
            - If each process has a timeout of 0.5secs then p1 needs 2secs to complete in a time-sharing environment whereas the optimal time for it to be finished is 1sec.
            - The order of how the processes are executed is:
                - P1 - 0.5 secs
                - P2 - 0.5secs
                - P3 - 0.5 secs
                - P1 - 0.5secs (p1 finishes)
- Networked computing environment
    - Internet.

    - Connected through LANs.
    - Resource sharing and better communication.
    - Better than a time-sharing environment because waiting time is reduced and response time is also reduced.
    - Networks evolved to ARPANET and finally became the internet that we know today.
- Client-server computing environment
    - The client asks for a resource and the server responds with the resource.
    - Client and server communicate via the internet/network.
    - Client - which initiates the process, Server - which waits for the client to contact.
    - A server may have multiple clients at the same time while a client is in contact with only one server.
    - Advantages:
        - All the data is concentrated in a single place, the server, thus making it easier to protect.
        - The server is not located physically closer to the clients, yet they can access the server and request services.
        - Platform Independent - It does not matter which platform the client is using, yet they can periodically request data from the server.
    - Disadvantages:
        - Server Overloading/Congestion - If multiple clients request data simultaneously the server might overload and performance will be decreased.
        - Cost - Server maintenance and resetting the server in case it fails.
        - Server failure - If the server fails then none of the client's requests can be fulfilled and also resetting a failed server is costly.
- Distributed computing environment
    - Multiple Nodes are distributed geographically and are connected through a network and communicate with each other via the same.
    - Multiple nodes work together to perform a given task and communicate with each other in doing so while performing the given task.
    - Different systems can have different tasks to execute and they share their resources in doing so.
    - Advantages:
        - Nodes are distributed globally to reduce latency.
            - Latency - delay before a request can be served

        - Data is backed up - In case of some failure or an accident like a fire at one system, our data isn't lost because the nodes are geographically distributed and the data can be recovered.
        - Computation power - since so many systems share resources in executing a task, the computational power drastically increases.
    - Disadvantages:
        - Security - our nodes and connections need to be secure and it is difficult to provide all of them with adequate security.
        - Also, there might be some information loss when messages/data are transferred from one node to another.
- Grid Computing
    - Inherits all the properties from distributed computing but all the nodes/grids in the grid computing act as one single entity and all nodes in the grid execute/perform a single task.
    - In distributed computing different nodes might have different tasks to execute but in grid computing, all the nodes in the grid come together to perform a single common task.
    - Often the task is split and distributed among the system nodes.
    - Requires middleware - special software that moderates which part of a task to assign to which node, like a manager, if one node is not working properly it needs to reassign its tasks to other nodes in the grid to get the job done.
    - Advantages:
        - Improved resource utilization.
        - Parallel processing - can run complex tasks and provide results faster.
        - Easier collaboration
        - Increased robustness. (robustness - in case one node fails and cannot run its subtask assigned by the middleware, then the subtask can be executed by some other node in the grid without any problems, data won't be lost)
    - Disadvantages:
        - Integrity - A solution returned by a grid is not trustworthy, i.e, we don't know if it is right or wrong.
        - Dynamic nature of nodes - If nodes are added and removed dynamically from the grid system, it causes inconsistency and poor performance.
- Cloud computing environment
    - When the computing is made over a cloud, i.e, a group of systems managed by an organization or third party then we call such an environment a cloud computing environment.

    - The Cloud computing environment follows a structure similar to a client-server architecture.
    - The Cloud users only see the service being provided and not the internal details of how the service is provided.
    - Advantages:
        - On-demand self-service - Resources can be provisioned and released only when needed.
        - Scalability - ability to scale up or scale down the resources based on the usage.
        - Resource pooling - A large number of resources are pooled together.
        - Measured service - you only pay for what you have used unless it is not a private cloud.
        - Can be accessed anywhere and anytime using the internet.
    - Disadvantages:
        - Security - Security is the main concern since your data resides in a data center whose location is unknown to you and managed by a third party.
        - Cost - Even though cloud computing is a measured service, if you want a private cloud out of security concern, setting up and maintaining is costly.
        - When you don't have an internet connection you can't access the cloud.
- Cluster computing environment
    - Cluster computing refers that many computers are connected via a network and act as a single entity.
    - The clustered computing environment is similar to the parallel computing environment but with a major difference that in cluster computing we have multiple computer systems which give an impression of a single system while in parallel computing we have only a single system.
    - Advantages:
        - Faster computational speed
        - More computational power
        - Data integrity - Cluster computing environment provided correct solutions to even complicated problems.
        - Flexibility - Computational power can be increased just by adding a new system to our network.
    - Disadvantages:
        - Cost - Since the quality of each cluster in the network is high, the cost required to set up a cluster computing environment is quite high.

        - Maintenance - since the infrastructure is quite high, the cost and effort required for maintenance are also high.

Grid computing Architecture:

- Two entities, one is the control server(manager), and the other is the nodes of the grid.
- The control server has a special software middleware that does the task of dividing and merging the task and its output provided by nodes of the grid.
- The grid nodes also have the middleware installed to communicate with each other.
- This middleware acts as the manager of all nodes.
- Two grid nodes can be connected unidirectional or bi-directional, depending on the architecture.
- If a grid node specializes in a certain application(for example .net client or CPP client) then the control server assigns the task related to that application to the respective grid node and if all grid nodes can execute that process/application then the tasks are divided based on the load, response time and waiting time to the multiple nodes and later merge their outputs.
- Grid architecture can also be constructed locally using LANs and doesn't necessarily have to be geographically separated.
- Users usually do not pay for their services.

In-Memory data grid:

- Generally, grid nodes do not share resources but share their outputs of the tasks they were assigned, they have their data stored locally and use the data stored locally.
- If you want all the grid nodes to share a common database you can use the in-memory data grid architecture where all nodes are tightly connected(sharing their resource) and share a common database.
- All nodes are bidirectionally connected so each grid node can communicate with every other node.
- Offer high throughput.
    - Throughput - number of requests served per unit time.
    - All nodes share the same database and don't need to share their data.
    - And since we reduce latency automatically throughput increases.
- Offer low latency.

    - In memory, a data grid offers low latency because it is a type of local data grid architecture and is nearby to your place and so the delay gets reduced.

Types of grids:

- Computational grids
    - Uses high-performance servers to perform tasks
    - Used for high computational tasks.
- Data grids
    - Data grids are the grids that we have explored until now.
    - Access, transfer and modify huge amounts of data.
- Scavenging grids
    - Use idle CPU cycles of nodes to execute tasks.
    - For example, if one of the nodes is highly loaded and one other node has some CPU cycles available and low load then the highly loaded node can utilize the CPU cycle of a low loaded node to execute its tasks.

Design issues in grid computing:

- Network failure - if nodes cannot periodically connect then latency would increase.
- Security - Trustworthy solution? Each node will perform its task and return its output but what is the guarantee that the solution provided is correct? There might be attacks on the nodes and might provide incorrect solutions.
- Heterogeneity - Each node may run a different OS, might have different specs.
- Dynamic nature - If nodes are added or nodes are removed because they are not working properly or something, then how can you reassign the tasks and manage the load?

Utility Computing:

- Pay per use model.
- Computing is considered as a utility similar to electricity.
- Every connection is metered.
- Users can utilize the computing power according to their needs.

Difference between utility computing and grid computing: Link

Notes:

- Utility computing is similar to cloud computing.

- Cloud computing follows client-server architecture while grid computing follows distributed systems architecture.
- Cloud computing and grid computing are based on different concepts.

Issue/Challenges for cloud computing:

Shortly when there will be many smart cities and have many IoT devices in them, the data that is generated is high and it needs to be processed for the devices to run periodically, and to process such high volume data we need a cloud service which processes the data and returns the output, this gives rise to several problems that we might face with the cloud service that we ought to provide.

- The distance between the cloud and the data.
    - The distance between the host and the cloud/server might be high.
- Network bandwidth latency and cost.
    - When a high amount of data is being sent then latency and cost would be high.
- Response time, Processing time, Operational cost.
    - More response time and high processing time and operational cost when a high amount of data is being sent and since the distance between server and client are high.
- Security.
    - Ensure security of the data that is being transmitted.

To address these issues we have layers of computing, and by introducing these layers we are decreasing the distance between the data generation and data processing, which automatically solves most of the issues that we discussed earlier.

Layers of computing:

- Cloud Layer
    - Industrial big data, databases, and data “warehousing”.
- Fog Layer
    - Local network assets and micro-data centers.
    - Computing at gateway/LAN’s.
- Edge Layer
    - Computing is done locally, for example, a smartwatch has some sensors like pulse rate sensor, motion sensor, etc, if you want to send the data to the cloud it might take a lot of bandwidth and high processing time, so the computing is done locally, this is called edge computing.

Cloud Computing:

- Cloud computing = grid computing(some features) + utility computing(pay per use model).
- Managed by service providers.
- Pay as you go model.
- Accessing through a network.
- The illusion of unlimited resources.

Evolution of Applications: Link

- Stand Alone
    - These applications reside on local systems.
    - Use local resources.
    - Update frequently.
    - In non-shareable mode.
    - Example - Adobe reader, etc
    - Disadvantages -
        - Low Not shareable.
        - Low resource utilization - not using all the applications completely.
- Web applications
    - Reside on remote systems.
    - Need internet to connect with web apps.
    - Client-server architecture.
    - Quality of service depends upon the number of users, if users are high then latency might be high and users might suffer some delays because of limited resources on remote systems.
    - Inflexible usage model.
- Cloud applications
    - Similar to web applications - reside on remote systems, client-server model, need internet.
    - On-demand - only provides resources on demand of the user, pay per use model.
    - Multitenancy - more than one user can access different virtual machines at the same time.
    - Elasticity -
    - Heterogeneity - Cloud apps can be accessed by various machines with different specifications, machines(mobiles).

A simple Architecture - Link

- Multiple edge devices communicate with the cloud through a gateway.

- Gateways are generally the LANs.
- If computing happens at the edge devices then it is called edge computing.
- If computing happens at the gateway it is called fog computing.
- If computing happens in the cloud then it is called cloud computing.
- Cloud computing is not always preferred, it is not the best choice always.
- Consider a case where an automatic car observes a car ahead of it, since the distance between the car/client and cloud/server is high, by the time it processes that brake needs to be hit we might have an accident. Since the distance causes the high latency cloud computing is not always preferred. And in this case edge computing is preferred since there is no traffic and latency is low the car hits the brake immediately.
- In real-time applications edge computing is preferred because not a large amount of data is sent to be processed and can be processed locally.

Features of edge computing:

- Advantages
    - Low latency
    - Faster decisions.
    - Privacy
    - Security.
- Disadvantages
    - Low computational power.
    - Lack of global(network) view.
        - Consider the case of automated museum tour guides,
        - Suppose they use edge computing, if the order of the artifacts changes or some exhibits is closed then they need to be programmed again and let them know that some exhibits are closed or the order has changed, this is a problem with edge computing.
        - Instead, we can use fog computing here where the edge device that is working has a local network view(here the museum) and can work accordingly.

Fog computing:

- Fog and edge computing are extensions of cloud computing, we find the disadvantages of cloud computing and then solve using fog and edge computing.
- Fog computing is done at the gateways - a location in between the cloud and the client.

- Can periodically connect with cloud servers and cache some useful information.

Features of Fog computing:

- Contextual location awareness(museum case) + low latency(compared to cloud)
- Geographical distribution
- Heterogeneity - can be accessed with different devices with different specifications.
- Interoperability - Can exchange with other fog devices if they are connected.
- Real-time interactions - Because of local view and awareness it offers real-time interactions, in the case of a museum tour guide if an exhibit is closed suddenly then the tour guide will not take you to that particular exhibit now that it is closed.
- Scalability - if demand is increased then we can scale up or else scale down the resources used.
- Advantages -
    - Low latency - distance reduced between the data generation and data processing.
    - Better data control - data processed at local area network, so we can control the data as we like.
    - Flexible storage system - can use when we have internet or when we don't have internet as well.
    - Connecting centralized and decentralized storage systems - the cloud is centralized while edge systems are decentralized, fog is in between these two and hence connects them.

Notes:

1. Edge computing is preferred when not much data is being generated and a low response time is required.
2. Fog computing when a local network level view is required and still the low response is also needed,
3. Cloud computing when more computational power is required.

Role of IoT and cloud computing:

These days we are using these technologies together.

- Increased Scalability -
    - IoT end devices are resource-constrained devices, that is they have the less computational power and less storage capacity.

    - We use IoT along with cloud computing to overcome the problem when an IoT device needs to process a large amount of data.
    - We can scale up and scale down the resources based on requirements.
- Improved safety -
    - IoT devices are less secured and more prone to attackers.
    - When merged with cloud computing, security increases because there is server-levela agreement with cloud computing and the security is provided by the cloud service provider.
- Enable IoT
    - Decentralized data storage corresponds with the main IoT needs, such as accessibility, safety, mobility, and scalability. Cloud services process massive amounts of data and quickly distribute the information among multiple servers, where it can be later accessed by a connected device anytime.

Benefits of Merging IoT and cloud computing:

- Smart analytics - Internet of Things systems produce a lot of data. Developers and businesses can use it to know their users better. Cloud services provide a safe environment where this data could be analyzed, managed, and stored.
- Better security - security for the IoT devices is increased because of cloud computing as cloud service providers provide more security to the IoT devices storage in the cloud.
- Inter device interactions - we can use cloud, fog, edge computing as per our needs to fulfill our needs. The cloud improves communication between devices and applications.

Mist computing:

- We form clusters of devices that have similar tasks to do and they share their resources in doing so
- Lightweight version of fog computing as not all devices are connected but only similar devices(clusters) are connected.

Dew computing:

- When cloud computing is duplicated in local computing then it is dew computing.
- Useful when the internet connection is not available.

A simple Scenario:

- ABC is an e-commerce company and they have hosted their website on an on-premise server.
- Recently they have been getting a lot of business and their IT head bob observes that the network traffic increases and their server operates very near its capacity.
- Now bob has two options -
    - Horizontal scaling - Bob can increase the number of systems and use distributed systems architecture to solve the issue, these are less expensive in the long run, have no downtime, potentially unlimited scaling.
    - Vertical scaling - Bob can increase the specifications of the server, that is, improve the hardware, this increases the cost, in the long run, possible downtime, limited scaling.
- But what if the network traffic is high only on every tuesday, if bob chooses horizontal scaling then his systems goes unused on other days, but if chooses not to improve they lose revenue.
- This is when cloud computing is useful, Bob chooses cloud to proceed.
- Using cloud is preferred here because of the pay per use model and the scalability that it provides, we can scale up the resources on tuesday to meet the needs and scale down on other days when the load is low.

Features of Cloud computing:

- On-demand self service
    - Whenever needed we can use, that is, on demand.
    - And this is self service so whenever we need cloud services we can use them by ourselves and no human intervention is required.
- Broad network access
    - Can be accessed using any device(Heterogeneity).
- Resource pooling
    - We have a large amount of resources and can be allocated to users on requirements.
- Rapid elasticity
    - Scalability - when high resources are required we can use high resources and when we don’t need high amounts of resources we can scale down.
- Measured service
    - Pay per use.

Service level agreement - The warranty of the cloud services:

- SLA is a bond for performance negotiated between cloud service providers and the clients.
- Availability and Performance
- Security / privacy of the data
- Disaster Recovery expectations
- Location of the data
- Access and portability to the data
- Process to identify problems and resolution expectations
- Dispute mediation process (e.g. escalation process, consequences
- Exit Strategy with expectations on the provider to ensure smooth transition.

Service models:

- Software as service
    - Google photos, docs, and facebook, etc
- Platform as service
    - Provide runtime environment for applications,
    - Development and data processing platforms
    - Examples - microsoft azure, hadoop, google appengine, etc
- Infrastructure as service
    - Virtualized servers
    - Storage and networking
    - Examples - S3, Amazon EC2, vcloud etc.

Enabling technologies for cloud computing:

- Browser as a platform
    - We are using browsers to communicate with cloud servers.
- Parallel Processing
    - Multiple users accessing the same resources at the same time - multi tenancy.
- Virtualization
    - Virtualization is a core technology for cloud computing.
    - Virtualization is essentially a technology that allows creation of different computing environments.
    - VMware, Virtual box are examples of how we use virtualization.
- Web 2.0
    - The Web is the primary interface through which cloud computing delivers its services.
    - Web 2.0 enables developers to architect applications and deliver services through the Internet.
    - Web 2.0 brings interactivity and flexibility into Webpages.

    - These capabilities are obtained by integrating a collection of standards and technologies such as XML, Asynchronous JavaScript and XML (AJAX), Web Services, and others.
    - Examples of Web 2.0 applications are Google Documents, Google Maps, Flickr, Facebook, Twitter, YouTube, delicious, Blogger, and Wikipedia.
- Service oriented architecture
    - Service orientation is the core reference model for cloud computing systems.(Check service models 👆)
    - This approach adopts the concept of services as the main building blocks of application and system development.
    - A service is supposed to be loosely coupled, reusable, programming language independent, and location transparent.
    - Service-oriented computing introduces and diffuses two fundamental concepts to cloud computing: Quality of Service (QoS) and Software-as-a-Service (SaaS).
    - Quality of service (QoS) identifies a set of functional and non-functional attributes that can be used to evaluate the behavior of a service from different perspectives