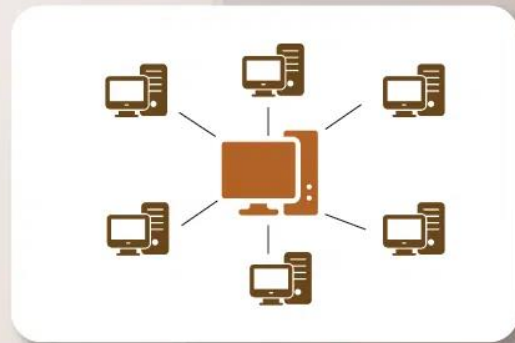


What is a Distributed System?

A distributed system is a collection of independent computers that appear to the users of the system as a single coherent system. These computers or nodes work together, communicate over a network, and coordinate their activities to achieve a common goal by sharing resources, data, and tasks.

What is a Distributed System?



Difference between centralized system and distributed system

All data and computational resources are kept and controlled in a single central place, such as a server, in a centralized system. Applications and users connect to this hub in order to access and handle data. Although this configuration is easy to maintain and secure, if too many users access it simultaneously or if the central server malfunctions, it could become a bottleneck.

A distributed system, on the other hand, disperses data and resources over several servers or locations, frequently across various physical places. Better scalability and reliability are made possible by this configuration since the system can function even in the event of a component failure. However, because of their numerous points of interaction, distributed systems can be more difficult to secure and administer.

Architectures of Distributed systems

Below are some of the common distributed system architectures:

- [Client-Server Architecture:](#)

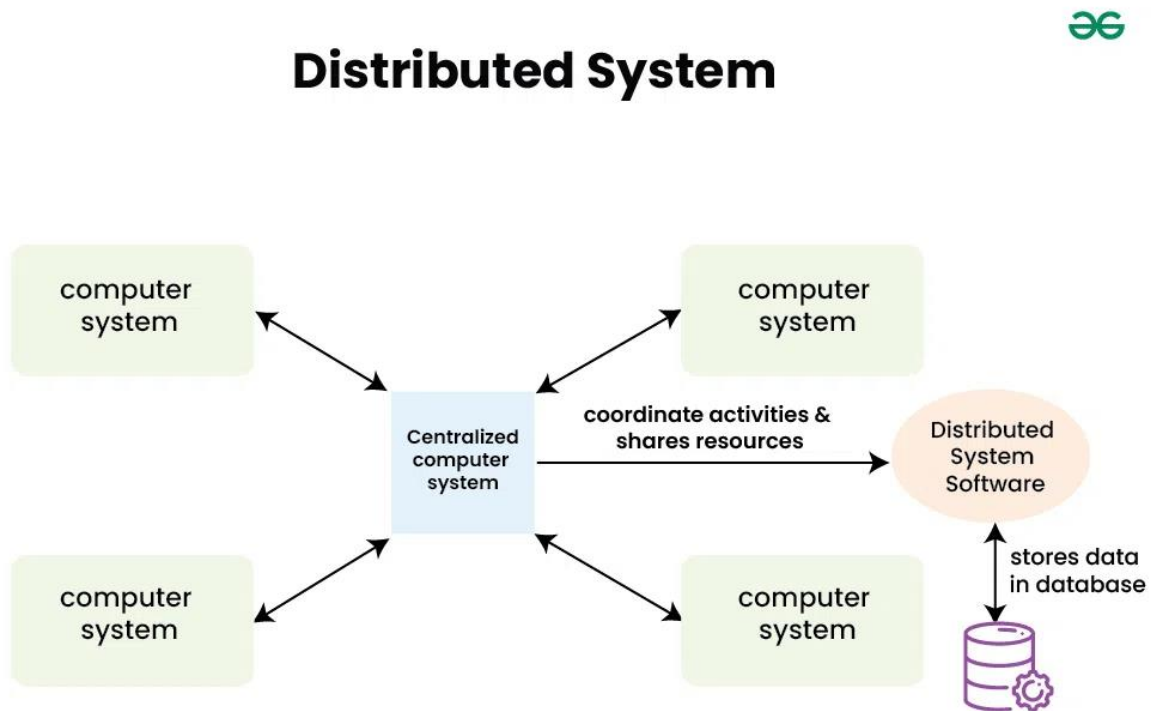
- In this setup, servers provide resources or services, and clients request them. Clients and servers communicate over a network.
 - **Examples:** Web applications, where browsers (clients) request pages from web servers.
- **Peer-to-Peer (P2P) Architecture:**
 - Each node, or “peer,” in the network acts as both a client and a server, sharing resources directly with each other.
 - **Examples:** File-sharing networks like BitTorrent, where files are shared between users without a central server.
- **Three-Tier Architecture:**
 - This model has three layers: presentation (user interface), application (business logic), and data (database). Each layer is separated to allow easier scaling and maintenance.
 - **Examples:** Many web applications use this to separate user interfaces, logic processing, and data storage.
- **Microservices Architecture:**
 - The application is split into small, independent services, each handling specific functions. These services communicate over a network, often using REST APIs or messaging.
 - **Examples:** Modern web applications like Netflix or Amazon, where different services handle user accounts, orders, and recommendations independently.
- **Service-Oriented Architecture (SOA):**
 - Similar to microservices, SOA organizes functions as services. However, SOA typically uses an enterprise service bus (ESB) to manage communication between services.
 - **Examples:** Large enterprise applications in finance or government, where different services handle various aspects of business processes.
- **Event-Driven Architecture:**
 - Components interact by sending and responding to events rather than direct requests. An event triggers specific actions or processes in various parts of the system.

- **Examples:** Real-time applications like IoT systems, where sensors trigger actions based on detected events.

The most common forms of distributed systems today operate over the internet, handing off workloads to dozens of cloud-based virtual server instances that are created as needed, and then terminated when the task is complete.

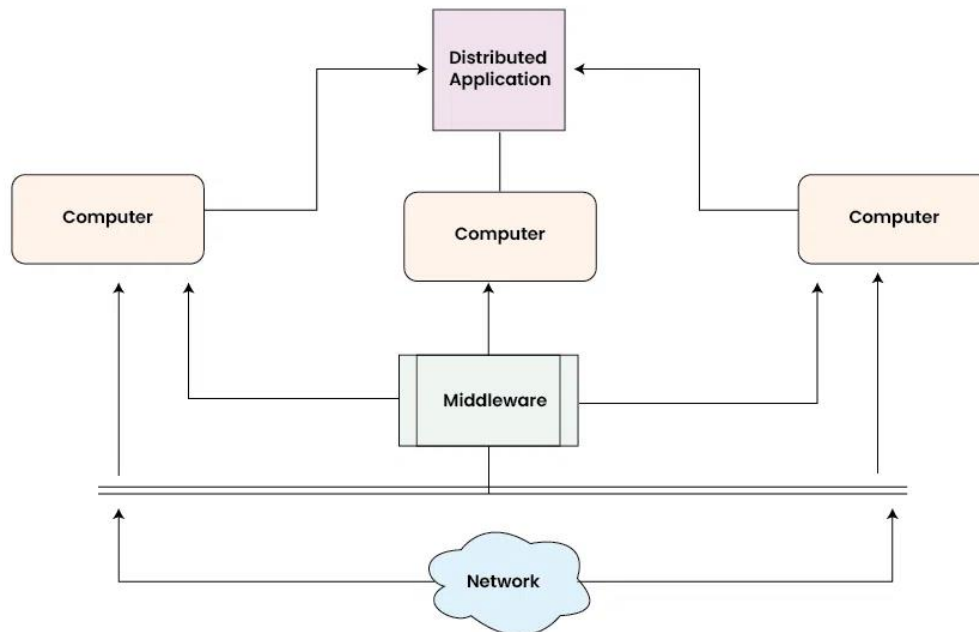
Example of a Distributed System

Any Social Media can have its Centralized Computer Network as its Headquarters and computer systems that can be accessed by any user and using their services will be the Autonomous Systems in the Distributed System Architecture.



- **Distributed System Software:** This Software enables computers to coordinate their activities and to share the resources such as Hardware, Software, Data, etc.
- **Database:** It is used to store the processed data that are processed by each Node/System of the Distributed systems that are connected to the Centralized network.

Working of Distributed System



- As we can see that each Autonomous System has a common Application that can have its own data that is shared by the Centralized Database System.
- To Transfer the Data to Autonomous Systems, Centralized System should be having a Middleware Service and should be connected to a Network.
- Middleware Services enable some services which are not present in the local systems or centralized system default by acting as an interface between the Centralized System and the local systems. By using components of Middleware Services systems communicate and manage data.
- The Data which is been transferred through the database will be divided into segments or modules and shared with Autonomous systems for processing.
- The Data will be processed and then will be transferred to the Centralized system through the network and will be stored in the database.

Characteristics of Distributed System

- **Resource Sharing:** It is the ability to use any Hardware, Software, or Data anywhere in the System.

- **Openness:** It is concerned with Extensions and improvements in the system (i.e., How openly the software is developed and shared with others)
- **Concurrency:** It is naturally present in Distributed Systems, that deal with the same activity or functionality that can be performed by separate users who are in remote locations. Every local system has its independent Operating Systems and Resources.
- **Scalability:** It increases the scale of the system as a number of processors communicate with more users by accommodating to improve the responsiveness of the system.
- **Fault tolerance:** It cares about the reliability of the system if there is a failure in Hardware or Software, the system continues to operate properly without degrading the performance the system.
- **Transparency:** It hides the complexity of the Distributed Systems to the Users and Application programs as there should be privacy in every system.

Advantages of Distributed System

Below are some of the advantages of Distributed System:

- **Scalability:** Distributed systems can easily grow by adding more computers (nodes), allowing them to handle increased demand without significant reconfiguration.
- **Reliability and Fault Tolerance:** If one part of the system fails, others can take over, making distributed systems more resilient and ensuring services remain available.
- **Performance:** Workloads can be split across multiple nodes, allowing tasks to be completed faster and improving overall system performance.
- **Resource Sharing:** Distributed systems allow resources like data, storage, and computing power to be shared across nodes, increasing efficiency and reducing costs.
- **Geographical Distribution:** Since nodes can be in different locations, distributed systems can serve users globally, providing faster access to resources based on location.

Disadvantages of Distributed System

Below are some of the disadvantages of Distributed System:

- Relevant Software for Distributed systems does not exist currently.
- Security possess a problem due to easy access to data as the resources are shared to multiple systems.

- Networking Saturation may cause a hurdle in data transfer i.e., if there is a lag in the network then the user will face a problem accessing data.
- In comparison to a single user system, the database associated with distributed systems is much more complex and challenging to manage.
- If every node in a distributed system tries to send data at once, the network may become overloaded.

Use cases of Distributed System

- **Finance and Commerce:** Amazon, eBay, Online Banking, E-Commerce websites.
- **Information Society:** Search Engines, Wikipedia, Social Networking, Cloud Computing.
- **Cloud Technologies:** AWS, Salesforce, Microsoft Azure, SAP.
- **Entertainment:** Online Gaming, Music, youtube.
- **Healthcare:** Online patient records, Health Informatics.
- **Transport and logistics:** GPS, Google Maps.

Are Distributed Systems and Microservices the Same?

Distributed systems and [microservices](#) are related concepts but not the same. Let's break down the differences:

1. Distributed Systems:

- A distributed system is a collection of independent computers that appear to its users as a single coherent system.
- In a distributed system, components located on networked computers communicate and coordinate their actions by passing messages.
- Distributed systems can encompass various architectures, including client-server, peer-to-peer, and more.

2. [Microservices](#):

- Microservices is an architectural style that structures an application as a collection of small, autonomous services, modeled around a business domain.
- Each microservice is a self-contained unit that can be developed, deployed, and scaled independently.

- Microservices communicate with each other over a network, typically using lightweight protocols like HTTP or messaging queues.

While microservices can be implemented in a distributed system, they are not same.

Microservices focus on architectural design principles, emphasizing modularity, scalability, and flexibility, whereas distributed systems encompass a broader range of concepts, including communication protocols, fault tolerance, and concurrency control, among others.