**Chapter 1 Characterization of Distributed Systems**

**Distributed system** refers to the system that its **components** locate at **networked computers**, communicate and coordinate their actions by passing only **messages**

**The characteristics of distributed systems:**

* Concurrency of components
* Lack of a global clock: The **accuracy** that computers in a network can synchronize their **clocks** is very **limited** – there is **no** exact global notion of the **correct time**.
* Independent failures of components

**Distributed systems** mainly rely on **resource sharing**. Resource refers to things that can be shared in a networked computer system, EX: hardware, file, database, object, etc.

**Resources** are managed by **servers** and accessed by **clients**, or they can be encapsulated as **objects** and accessed by other **client objects**.

**The challenges of distributed systems include:**

* Heterogeneity of components (ability of the system to operate on a variety of different hardware and software components)
* Openness (components to be added or replaced)
* Security
* Scalability (the ability to work well when the load increases)
* Failure handling
* Concurrency of components
* Transparency
* Quality of service (QoS)

**Name: SOR SOPHEAK**

**Group: GIC(I4-C)**

TP 1: Introduction to Distributed Systems

1. What’s distributed system?

* Distributed system is a network that distributes resources among several servers without having a single central server. Each of these servers can act as an independent central server.

1. What are the characteristics of distributed systems? Explain.

* Characteristics of distributed systems
* Concurrency of components: In distributed systems, multiple components or processes run concurrently, performing independent tasks simultaneously. This concurrency allows for efficient utilization of resources and improved system performance. However, managing concurrent access to shared resources and ensuring synchronization between components becomes a critical challenge. Techniques like locking, synchronization primitives, and distributed algorithms are employed to coordinate the concurrent actions of distributed components.
* Lack of a global clock: Distributed systems typically lack a global clock that can provide a synchronized notion of time across all components. Each component may have its own local clock, but due to variations in clock speeds and network delays, it is challenging to establish a consistent global time across the entire system. This lack of a global clock can lead to difficulties in ordering events and coordinating actions. Distributed algorithms and protocols often employ logical clocks or timestamping mechanisms to order events causally or globally, enabling coordination and consistency across the distributed system.
* Independent failures of components: Distributed systems are composed of multiple interconnected components, and each component operates independently. As a result, individual components can fail independently due to hardware faults, software errors, or network issues. Independent failures are a significant concern in distributed systems, and fault tolerance mechanisms are employed to handle such failures and ensure system resilience. Techniques such as replication, redundancy, and error detection and recovery mechanisms help mitigate the impact of individual component failures and ensure the overall system's availability and reliability.

1. List down the 8 challenges of distributed systems? Explain.

* Heterogeneity = refers to the differences that arise in networks, programming languages, hardware, operating systems, and differences in software implementation.
* Openness = refers to the system's extensibility and ability to be reimplemented that is determined by the degree to which new resource-sharing services can be added and be available to different client programs.
* Security = is especially important in distributive systems due to their association with sensitive and private data.

For information resources has 3 components:

* + Confidentiality: protection against disclosure of data to unauthorized individuals
  + Integrity: protection against alteration or corruption of data
  + Availability: protection against interference which about to access the resources
* Scalability = A system is described as scalable if it will remain effective when there is a significant increase in the number of resources and the number of users.
* Failure handling = Failures in a distributed system are partial – some components may fail while others continue to function. In any program, this is a major problem. However, in distributive systems, with so many processes and users, the consequences of failures are exacerbated.
* Concurrency of components = In distributed systems, several clients will attempt to access a shared resource at the same time.
* Transparency = is the concealment from the user and the application programmer of the separated components in a distributed system so that the system is perceived as a whole rather than as a collection of independent components.
* Quality of service (QoS) = The main nonfunctional properties of systems that affect the QoS experienced by clients and users are reliability, security and performance. Performance in here refers to the ability to meet timeliness guarantees.

1. What are the disadvantages of distributed systems? Explain.

* The advantages of distributed systems
* It is difficult to provide adequate security in distributed systems because the nodes as well as the connections need to be secured.
* Some messages and data can be lost in the network while moving from one node to another.
* The database connected to the distributed systems is quite complicated and difficult to handle as compared to a single user system.
* Overloading may occur in the network if all the nodes of the distributed system try to send data at once.

1. What are the 8 forms of transparency in distributed systems? Explain.

* The 8 forms of transparency in distributed system
* Access Transparency: Access Transparency allows the same operations to be used to access local and remote resources. The file distribution must be hidden from the clients. The storing of data on separate servers that are physically separated, and a common set of actions should be available to access both remote and local files. Applications for local files are to be designed such that they should be able to run on remote files as well.
* Location Transparency: Location Transparency permits access to resources regardless of their physical or network location. There should be a view of a consistent file namespace for the clients. It must possess the feature of moving files such that their pathnames are not to be affected. There is no information regarding the physical location of the object in case of a location transparent name. It is a quite vital and critical feature for facilitating resource movement and service availability. Location and Access Transparency together makes Network transparency.
* Concurrency Transparency: Concurrency Transparency permits many processes to run in parallel using shared resources without interfering with one another. As we know distributed systems exhibit concurrent environments so the shareable items are all accessed at the same time. It is hard to control Concurrency and implementation.
* Concurrency Transparency: Concurrency Transparency permits many processes to run in parallel using shared resources without interfering with one another. As we know distributed systems exhibit concurrent environments so the shareable items are all accessed at the same time. It is hard to control Concurrency and implementation.
* Failure Transparency: Failure Transparency permits fault abstraction in the background, allowing users and application programs to execute tasks even when hardware and software components fail. The fault tolerance property is exhibited by the procedures that deal with access transparency. The main concern in the distributed system is that they are more prone to failure since any of the components could fail, resulting in a degraded or non-existent/unavailable service. It is quite difficult to tell the difference between a failed and a slow-running operation since the complexities are hidden.
* Mobility (Migration) Transparency: Mobility Transparency lets a system or resources move around without disrupting user or software processes. It also bolsters the load balancing of any client that may be overburdened.
* Performance Transparency: Performance Transparency enables system reconfiguration to increase or enhance performance.
* Scaling (Size) Transparency: Scaling Transparency enables systems and applications to scale up without requiring changes to the system architecture or application techniques. The resources that are not of any relevance to the user are hidden from the user and application programmers. The smooth development and evolution with this type of transparency are critical for most businesses. A system should be able to scale down to small surroundings, when necessary, as well as be space and/or time-efficient when required.

1. Give 5 types of hardware/software resources that can be shared. Give examples of their sharing as it occurs in practice in distributed systems.

* Five types of hardware resource and five types of data or software resources that can usually be shared are printer, plotter, storage space, cd drive,dvd drives, and processing power.

For example printer which takes graphics and texts from the computer and later it gets transferred into a paper which is of standard size.

1. What is single-point-of-failure and how can distribution help here?

* Single-point-of-failure is a system component which, upon failure, renders an entire system unavailable or unreliable. When you design a highly available deployment, you identify potential SPOFs and investigate how these SPOFs can be mitigated.
* Distribution can help
  + Backup and redundant systems and software components ensure against the loss of a primary system.
  + A second channel or conduit for redundant network cabling protects against loss of connections to local carriers and internet service providers.
  + Load balancers send requests for service only to servers that are online and in use. As a result, load balancing reduces the threat of SPOFs where multiple servers are in use.
  + Backup power and other electrical systems protect against the loss of power and intermittent power fluctuations that can disrupt business operations. For instance, lightning arrestors and electrical grounding reduce the threat of power surges.
  + An up-to-date data security infrastructure mitigates the threat from cybersecurity attacks. This includes firewalls that have current database rules and security tools set and patched for the level of software in use.
  + People can also be SPOFs. For example, an organization can be vulnerable if one person has all knowledge of a critical system. Cross-training employees is a wise approach.

1. When will we need distributed systems? Giving your ideas.

* Distributed systems are used when a workload is too great for a single computer or device to handle. They're also helpful in situations when the workload is subject to change, such as e-commerce traffic on Cyber Monday.