Distributed systems and distributed databases

Distributed systems and distributed databases are closely related concepts that play a crucial role in modern computing. A distributed database is a type of distributed system designed specifically for managing and storing data across multiple nodes. Below is an overview of distributed systems in the context of distributed databases, including their architecture, advantages, challenges, and examples.

Overview of Distributed Databases

A **distributed database** is a database that is not stored in a single location but rather distributed across multiple nodes or servers, which may be located in different physical locations. These nodes can work together to provide a unified database system to users.

Key Characteristics of Distributed Databases

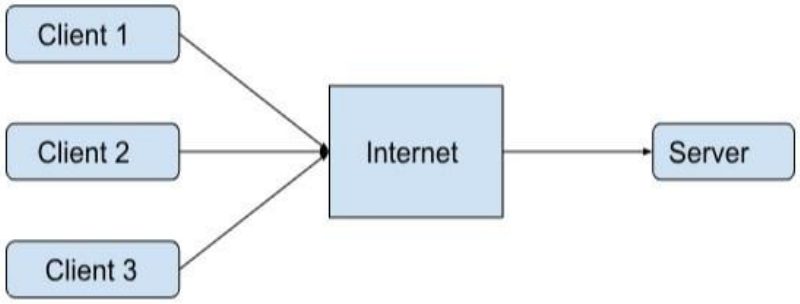
1. **Data Distribution**:
   - Data is distributed across various nodes to improve performance, reliability, and scalability. This can be done through techniques like sharding (horizontal partitioning) or replication.
2. **Transparency**:
   - Users and applications should perceive the distributed database as a single, cohesive system, even though the data is stored across multiple locations. This includes location transparency, access transparency, and failure transparency.
3. **Concurrency Control**:
   - Distributed databases must manage concurrent access to data from multiple nodes and users, ensuring data consistency and integrity.
4. **Fault Tolerance**:
   - Distributed databases are designed to withstand failures. If one node goes down, the system should still function correctly by using replication or other techniques.
5. **Scalability**:

- Distributed databases can scale horizontally by adding more nodes to handle increased load, improving performance without significant redesign.

Architectures of Distributed Databases

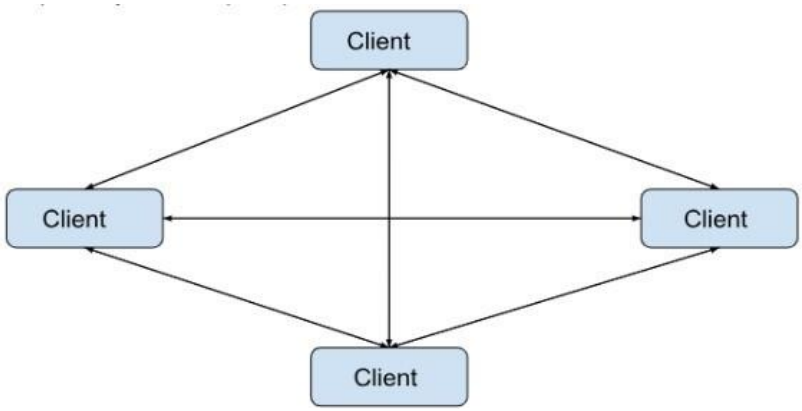
Client−Server Architecture

In this model, clients send requests to a centralized server or a group of servers that manage the database. The servers process the requests and return the results to the **clients.**



Peer−to−Peer Architecture

Each node in the network acts as both a client and a server. Nodes can share data directly with each other, allowing for decentralized control and resource sharing



Shared–Nothing Architecture

Data is divided up and spread among several nodes in a shared–nothing architecture, with each node in charge of a particular portion of the data. Resources are not shared across nodes, and each node runs independently. Due to the system's capacity to add additional nodes as needed without affecting the current nodes, this design offers great scalability and fault tolerance. Large–scale distributed systems, such as data warehouses or big data analytics platforms, frequently employ shared–nothing designs