Introduction to Distributed Computing

Distributed computing is a system that involves multiple interconnected computers working together to complete a task. These systems are typically used for large-scale tasks that require significant computing power or data storage.

by Ranjeet Kaur

Advantages of Distributed Computing

1 Scalability

Distributed systems can easily be scaled to accommodate increased workloads by adding more computers to the system.

2 High Availability

If one computer fails, the system can continue to operate because the workload is distributed among multiple machines.

3 Cost-Effectiveness

Using multiple smaller computers is often more cost-effective than using a single large computer.

4 Flexibility

Distributed systems can be designed to be flexible, allowing for different types of computers and software to be used together.

Architectural Patterns in Distributed Systems

Client-Server

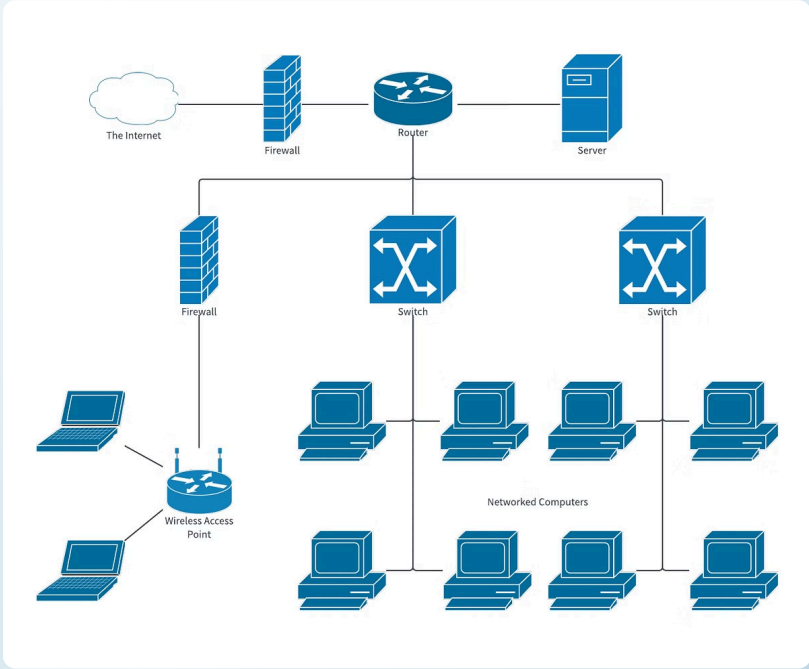
A central server provides services to multiple clients. This is the most common pattern used in web applications.

Peer-to-Peer

All computers in the system are equal and can communicate directly with each other. This pattern is often used in file sharing applications.

Cloud Computing

Resources are provided by a third-party provider over the internet. This allows businesses to access computing resources on demand.



Challenges in Distributed Computing

Data Consistency

Ensuring that data is consistent across all computers in the system.

Concurrency Control

Managing simultaneous access to shared resources.

Fault Tolerance

Designing systems that can continue to operate even if some computers fail.

Distributed Data Management

1

Data Partitioning

Splitting data into smaller parts that can be stored and managed by different computers.

2

Data Replication

Copying data to multiple computers to increase availability and performance.

3

Data Consistency

Ensuring that all copies of data are consistent across the system.

Distributed Algorithms and Coordination

1

Consensus

Achieving agreement among all computers in the system, even in the presence of failures.

2

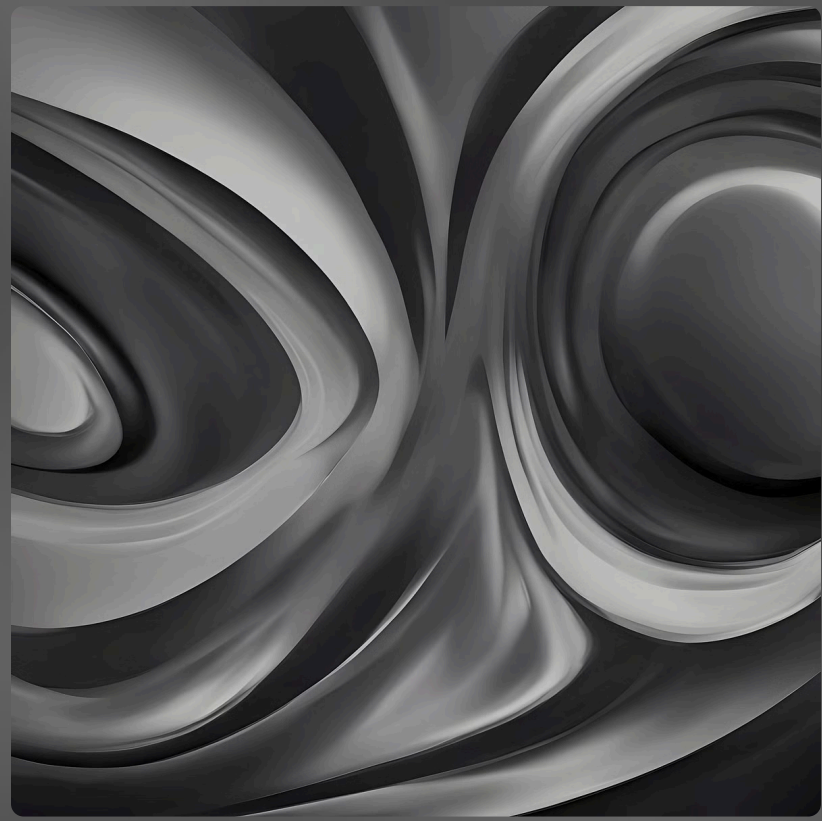
Leader Election

Choosing a single computer to act as the leader of the system.

3

Distributed Transactions

Coordinating multiple operations across multiple computers to ensure atomicity.



Fault Tolerance and Reliability

Redundancy

Having multiple copies of critical components, such as data or servers.

Timeouts

Setting time limits for operations to prevent the system from being blocked by slow or unresponsive computers.

Error Detection and Recovery

Monitoring the system for errors and implementing mechanisms to recover from failures.

Future Trends in Distributed Computing

Cloud-Native Applications	Applications designed specifically for the cloud, taking advantage of its scalability and flexibility.
Edge Computing	Processing data closer to the source, reducing latency and improving performance.
Serverless Computing	Running code without managing servers, simplifying development and deployment.
Blockchain Technology	Decentralized and secure systems for managing data and transactions.