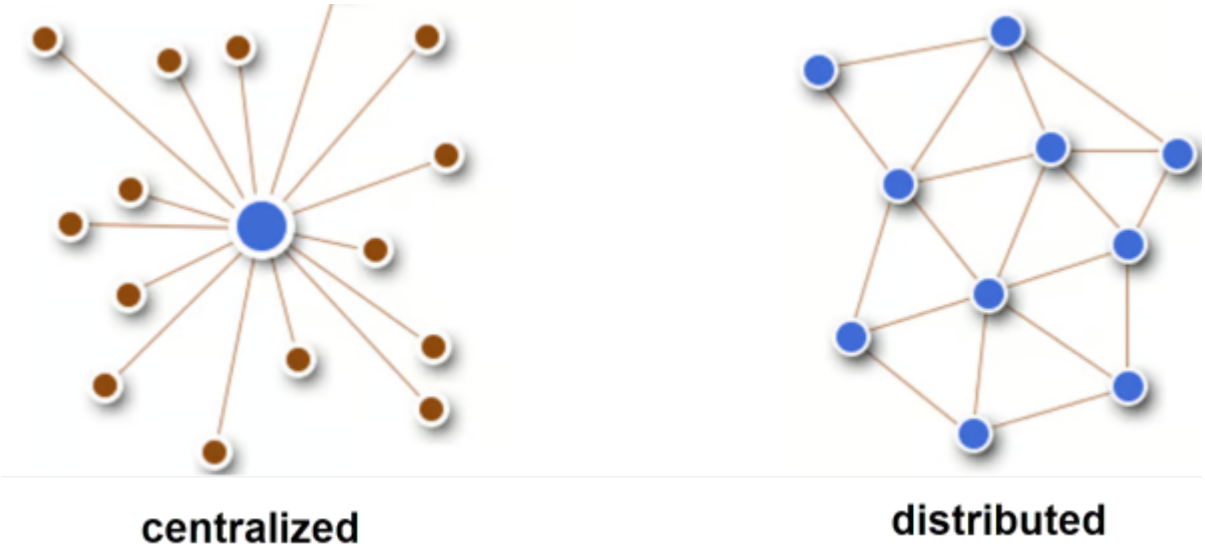
Centralized vs Distributed System Architecture



Centralized System Architecture

A centralized system architecture refers to a computing architecture where all or most of the processing, data storage, and management are performed by a single central server or a mainframe. This central entity is responsible for handling all requests and serving all client devices connected to it. Here are the main characteristics and components of a centralized system architecture:

Characteristics

- **Single Point of Control**: All processing, data storage, and management are handled by a central server or mainframe.
- **Tight Integration**: Components are closely integrated, typically within a single physical location.
- **Simplified Management**: Easier to administer, update, maintain, and back up since all resources are centralized.
- **Scalability Constraints**: Limited by the capacity of the central server; scaling often requires upgrading the central server's hardware.
- **Reliability Issues**: System depends heavily on the central server; a failure in the central server can make the entire system unavailable.
- **Performance**: Can be fast for small-scale operations due to reduced network latency and overhead within the local environment.

Components

1. **Central Server**: Handles major processing and data storage tasks.
2. **Client Devices**: Connect to the central server for processing and data access.
3. **Network Infrastructure**: Typically involves local area networks (LAN) connecting clients to the central server.

4. **Data Storage**: Centralized storage managed by the central server.

Examples

- Mainframe systems
- Client-server architecture
- Single data center cloud computing

Distributed System Architecture

A distributed system architecture refers to multiple independent components (or nodes) work together to appear as a single coherent system to the end-users. These components are spread across different physical or virtual machines.

Characteristics

- **Multiple Points of Control**: Processing and data storage are spread across multiple independent nodes or servers.
- **Loose Integration**: Components are often geographically dispersed and communicate over a network.
- **Complex Management**: Requires more sophisticated coordination, synchronization, and administration.
- **Scalability**: Can scale horizontally by adding more nodes to the system, making it suitable for large-scale operations.
- **Reliability**: More fault-tolerant; failure of one node does not necessarily bring down the entire system.
- **Performance**: Can handle large-scale operations more effectively by distributing the load; network latency and communication overhead can be a challenge.

Components

1. **Nodes/Servers**: Multiple independent servers or nodes handle processing and data storage.
2. **Network Infrastructure**: Involves wide area networks (WAN) and internet connectivity for communication between nodes.
3. **Distributed Data Storage**: Data is distributed across multiple nodes, often with redundancy to ensure availability and fault tolerance.
4. **Middleware**: Software layer that manages communication, data consistency, and coordination between distributed components.

Examples

- Internet services (e.g., Google, Facebook)
- Peer-to-peer networks

- Distributed databases (e.g., Apache Cassandra, Amazon DynamoDB)
- Cloud computing (multi-data center)

Comparison

Feature	Centralized System	Distributed System
————————	————————————————	————————————————
Control	Single central server	Multiple independent nodes
Integration	Tight integration, often local	Loose integration, often geographically dispersed
Management	Simplified management	Complex management
Scalability	Vertical scaling (hardware upgrades)	Horizontal scaling (adding nodes)
Reliability	Single point of failure	Fault-tolerant, no single point of failure
Performance	Low latency in small-scale operations	High performance for large-scale operations; may face network latency
Examples	Mainframes, client-server systems	Internet services, peer-to-peer networks, cloud computing

In summary, centralized systems are simpler to manage and more straightforward in design but face challenges in scalability and reliability. Distributed systems offer better scalability and fault tolerance but are more complex to design, implement, and manage.