Homework Report: Distributed Systems

By : Bekhouche Mohamed Amine

Date: 19 October 2025

Introduction

> *"A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable."*
> *— Leslie Lamport*

Distributed systems are collections of independent computers that appear to users as a single coherent system. They cooperate to achieve shared goals such as resource sharing, high performance, and fault tolerance. Among the different categories, three fundamental types are often discussed in the literature: **Distributed Computing Systems**, **Distributed Information Systems**, and **Distributed Pervasive Systems**

1 Distributed Computing Systems

Principle of Operation: These systems are designed to share computing power between several connected machines to execute complex tasks efficiently. A large computational job is divided into smaller subtasks distributed across multiple nodes. Each node processes a portion of the work, and results are later combined to produce the final output.

Architecture:

They typically follow a *Master–Slave* or *Client–Server* model, where a master node coordinates tasks and collects results. In some cases, a *Peer-to-Peer (P2P)* structure is used, allowing nodes to collaborate equally without centralized control.

Distribution:

Tasks are split into smaller independent units executed concurrently across nodes. The master node then aggregates the results to form the final solution.

Real Examples:

- Apache Hadoop and Apache Spark: distributed data processing frameworks.
- BOINC and SETI@home: volunteer computing platforms.
- Google MapReduce: parallel computation model for large datasets.

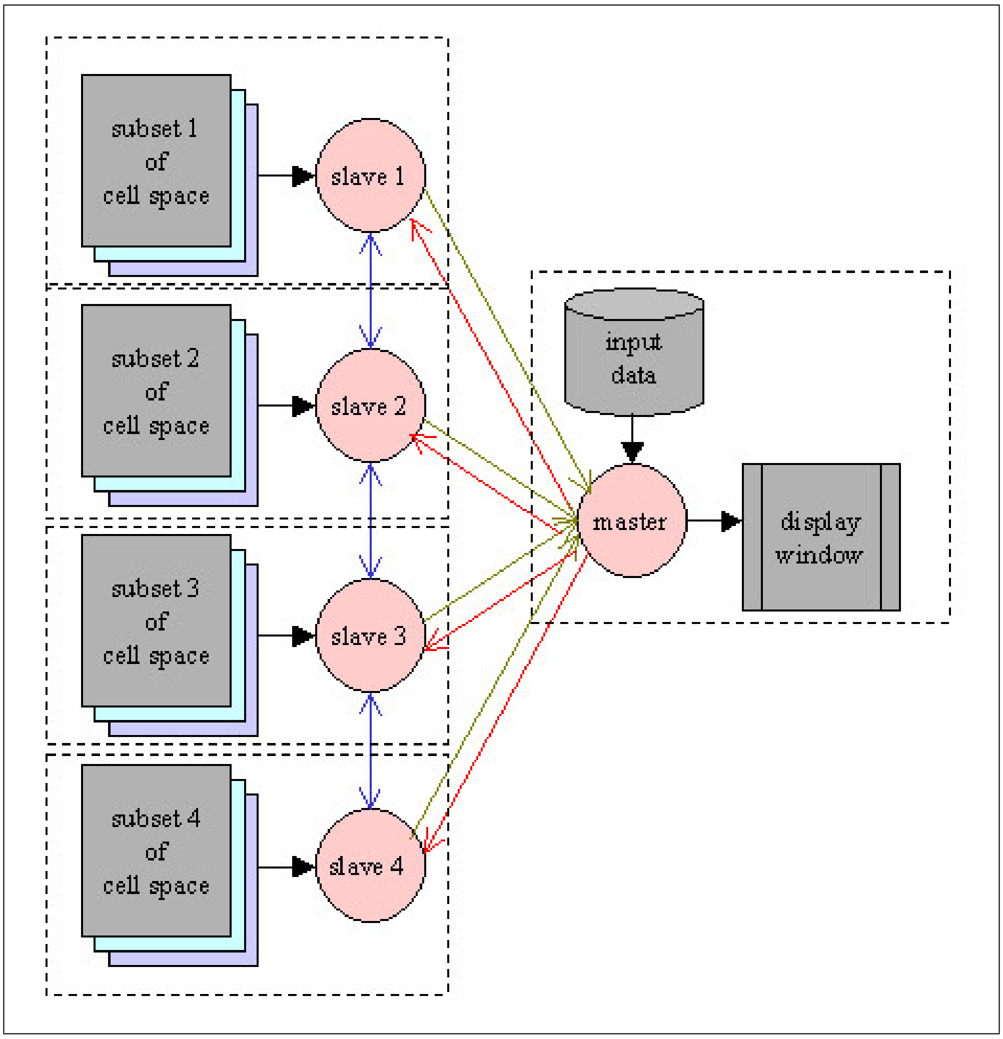

Figure 1: The master-slave structure of distributed computing with MPI programming.

2 Distributed Information Systems

Principle of Operation: These systems focus on managing and synchronizing data across multiple locations while maintaining consistency and reliability. They enable users and applications to access the same information seamlessly, regardless of where it is stored.

Architecture:

A middleware layer handles communication between distributed components and hides network complexity. Common models include *Client–Server* and *Publish–Subscribe* architectures.

Distribution:

Data is either replicated or partitioned across servers to ensure availability and fault tolerance. Distributed transactions maintain global data consistency.

Real Examples:

- Oracle RAC (Real Application Clusters): distributed relational database system.

- Cassandra and MongoDB: NoSQL distributed databases.
- CORBA and Java RMI: middleware enabling distributed communication.

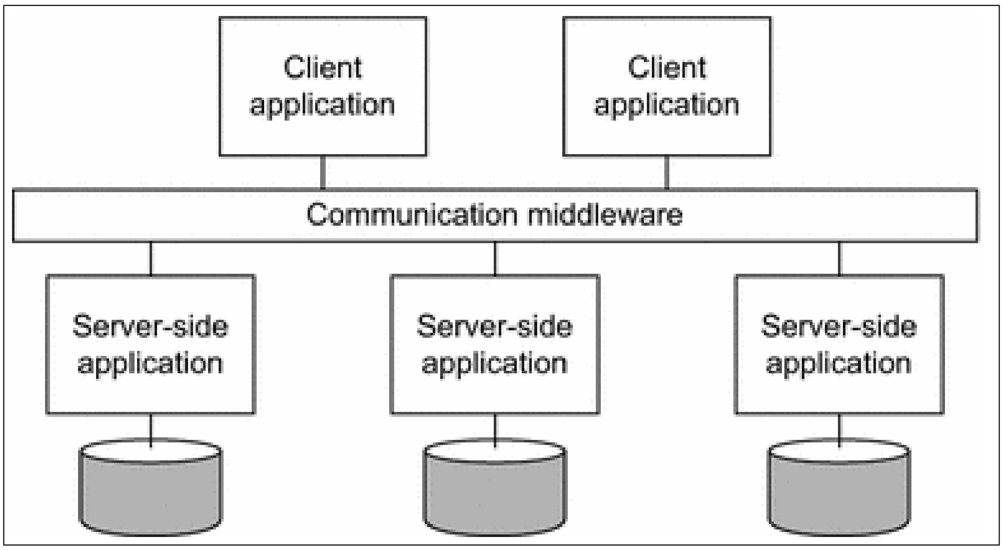

Figure 2: Middleware as a communication facilitator in enterprise application integration.

3 Distributed Pervasive Systems

Principle of Operation: These systems integrate computing capabilities into everyday environments through mobile, embedded, or sensor-based devices. They are designed to be context-aware and interact in real time with their surroundings.

Architecture:

Their architecture is typically decentralized, following *Edge Computing* or *Internet of Things (IoT)* paradigms. Computation is shared between local devices and cloud servers.

Distribution:

Devices and sensors communicate wirelessly (Wi-Fi, Bluetooth, Zigbee) and collaborate by sharing data or computation tasks across the network.

Real Examples:

- Smart Cities: intelligent management of traffic, lighting, and energy.
- Wireless Sensor Networks (WSN): environmental or agricultural monitoring.
- Connected Vehicles (V2X) and Smart Homes: IoT-based ecosystems.

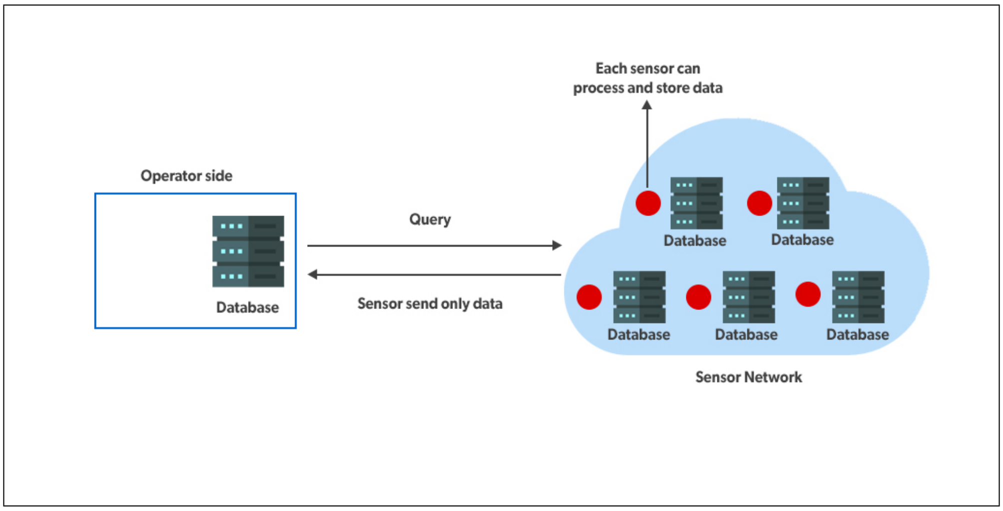

Figure 3: Distributed pervasive system with sensor network architecture.

Conclusion

Each category of distributed system reflects a different technological focus: computing systems aim for performance and scalability, information systems ensure consistency and reliability, while pervasive systems extend computing into our everyday surroundings. Together, these systems illustrate how distributed computing has evolved from centralized architectures into intelligent, interconnected ecosystems that drive modern innovation.

References

Tanenbaum, A. S. and Van Steen, M. (2007). *Distributed Systems: Principles and Paradigms*. Pearson Education, 2nd edition.