Load Balancing

Load balancing is the process of distributing network or application traffic across multiple servers. It ensures that no single server bears too much load, which helps:

- Prevent server overload
- Improve response time
- Increase fault tolerance
- Enhance scalability

Load balancers can be hardware-based or software-based and are commonly used in web applications, cloud services, and enterprise networks.

Load Balancing Strategies in Detail

1. Round Robin

How It Works:

- Requests are distributed sequentially to each server in the pool.
- After the last server is reached, it starts again from the first.

Example:

Imagine 3 servers: S1, S2, S3
Requests come in this order:
→ S1 → S2 → S3 → S1 → S2 → S3 ...

Advantages:

- Simple and easy to implement.
- Works well when all servers have equal capacity and performance.

Disadvantages:

- Doesn't consider current server load.
- Can lead to uneven distribution if some servers are slower or busier.

---

2. Least Connections

How It Works:

- The load balancer tracks the number of active connections on each server.

- It sends the new request to the server with the fewest active connections.

Example:

If S1 has 2 connections, S2 has 5, and S3 has 1 → the next request goes to S3.

Advantages:

- Dynamically balances based on real-time server load.
- Ideal for long-lived connections (e.g., streaming, chat apps).

Disadvantages:

- Requires tracking and updating connection counts.
- Slightly more complex to implement.

---

3. Random

How It Works:

- Each incoming request is assigned to a randomly selected server.

Example:

Request 1 → S2
Request 2 → S1
Request 3 → S3
Request 4 → S2 (random again)

Advantages:

- Very simple and fast.
- Can be surprisingly effective in evenly distributing load over time.

Disadvantages:

- Doesn't consider server capacity or current load.
- May lead to uneven distribution in short bursts.