

Distributed Systems (521290S)

Course Project Report

Team Tesla

Group Member
Md Mobusshar Islam
Muhammad Ahmed
Muhammad Talha Arshad

Student ID
2305578
2304796
2304797

Email
mislam23@student oulu.fi
mahmed23@student oulu.fi
Muhammad.arshad@student oulu.fi

Course project overview

The project aims to process the request for IoT devices based on certain factors like preference, urgency, etc, and respond to the requests accordingly.

The following are the components of the system.

❖ Smart IoT Devices:

Mocking the IoT devices with the code script that will serve as different nodes of the distributed system.

Each IoT device will be monitoring the environment, and it will be requesting the server based on different conditions like temperature, humidity etc

❖ Central Server:

Manages communication with all connected devices. Queues request from devices and prioritize them based on need. Respond to device requests with appropriate actions. Stores historical data for analysis and future optimizations.

❖ Request Queuing and Prioritization:

Implement a queuing system to manage incoming requests. Assign priority levels to requests based on factors like soil moisture levels, weather conditions, and plant types. High-priority requests get processed first.

Distributed System Topics Covered

❖ Message Queuing System:

➤ Round Robin Queuing:

Distributes incoming requests evenly among available server resources in a circular manner.

➤ Priority Queue:

Prioritizes requests based on urgency or importance, ensuring that high-priority requests are processed first.

❖ Load Balancing:

➤ Round Robin Load Balancing:

Distributes incoming network traffic or requests evenly across multiple servers.

➤ Weighted Round Robin:

Assigns different weights to servers based on their capacities, allowing for more efficient resource utilization.

❖ Priority Queue Management:

➤ Heap-based Priority Queue:

Utilizes a heap data structure to efficiently manage and extract the highest-priority requests.

➤ Weighted Priority Queue:

Assigns different weights to requests based on their criticality, influencing the order in which they are processed.

❖ Resource Management:

➤ Token Bucket Algorithm:

Controls the rate at which requests are processed to prevent resource exhaustion.

➤ Leaky Bucket Algorithm:

Smooth outbursts of incoming requests, preventing sudden spikes in resource usage.

Fault Tolerance and Redundancy:

In the above-mentioned basic topics covered, we will be trying different approaches to find out best possible solutions for a specific problem

Architecture Diagram

