EEX5362

Performance Modelling

Deliverable 01

B. Pramuka Navodh
722514185

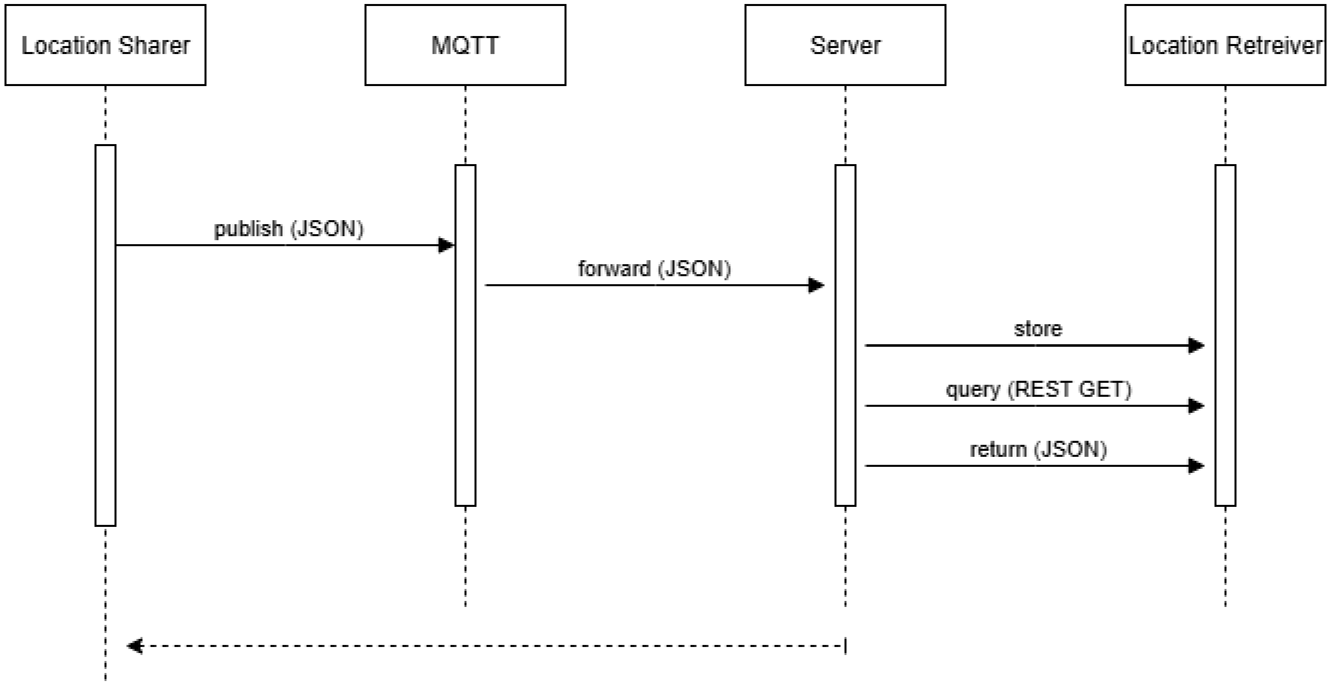
S22010189@ousl.lk

System Description

The selected scenario is the connection between a server and its clients of a ride tracking application. The application provides features for its users to share the live location of the public transport buses to a central server and also allows users to retrieve live bus locations from the server. The system is hosted on a shared VPS hosting as the server facility.

The server is implemented for constant use and handle large number of user parallelly. However, the resources of the server are limited (RAM, CPU, data transfer rates in its ports, and allowed traffic). In its objectives, retrieve location data from client devices, send location data to client devices, and manage the publisher-subscriber function in the mosquitto broker are listed. It's very important to efficiently manage this system to maintain it's best performances. For this server, a group of clients (location sharers) sends their live location as JSON packages every 10 seconds. A single package includes License Number, Route name, Latitude, longitude, and timespan. So, the size of a package is approximately 150 - 250 bytes. The other group is location retrievers. They receive the same data from the server, but in a REST API.

The server can be identified as the entity with controlled resources in this study. Given the server's limited resources and continuous operational demand, maintaining optimal performance is crucial. The server's specifications and client communications are crucial in this study for analysing response time, throughput, identifying bottlenecks, and optimizing resource allocation.



Performance Objectives

1. Response time

The objective of analysing the response time is to study the time taken by the server to process a client request and send back a response. In this scenario, CPU, RAM and data transfer rate of the server affect the response time. Long response times directly affect the user experience of the application. So, it's crucial to identify response time in various situations.

2. Concurrency

The objective of analysing the concurrency is to study how effectively the system handles multiple users or processes simultaneously. In this scenario, the data transfer rate of its ports of the server and CPU affect this aspect. Before the deployment, it's crucial to identify concurrency to avoid problems related to user or process handling.

3. Throughput

The objective of analysing the throughput is to study the total number of requests (messages and API calls) processed successfully by the server per unit time. With a limited bandwidth per day, it's important to serve functionalities to increase throughput.

4. Utilization

The objective of analysing the utilization is to study the proportion of server resources (CPU, memory, bandwidth) used during operation. Analysis of optimal utilization is very important to avoid performance degradation.

5. Bottleneck Identification

In this aspect, the objective is to detect performance-limiting components such as CPU saturation, message queue delays, or bandwidth congestion. This helps to identify weak points of the system.

6. Consistency

The objective of analysing consistency is to predict performance under various conditions. This can be used as a step-ahead solution for disaster recovery of the system.