Performance Report

Scenario 1 - 1 instance of seller and buyer client

Assignment 1 Response Time - 0.002 secs
Assignment 2 Response Time - 0.004 secs

Assignment 1 Throughput - 25 operations/sec
Assignment 2 Throughput -33 operations/sec

Scenario 2 - 10 instance of seller and buyer client

Assignment 1 Response Time - 0.03 secs
Assignment 2 Response Time - 0.009 secs

Assignment 1 Throughput - 180 operations/sec
Assignment 2 Throughput - 191 operations/sec

Scenario 3 - 100 instance of seller and buyer client

Assignment 1 Response Time - 2.3 secs
Assignment 2 Response Time - 0.009 secs

Assignment 1 Throughput - 405 operations/sec
Assignment 2 Throughput - 427 operations/sec

As we increase the number of instances of both buyer and seller, the response time starts increasing and the throughput starts to decrease. It could happen due to multiple factors like network congestion, server processing limits, and database access bottlenecks. Each client instance consumes resources (CPU, memory, network bandwidth) on the server, leading to longer processing times for each request as the system becomes saturated. Additionally, simultaneous database access requests can lead to contention, further slowing down response times. Optimizing system architecture, increasing server capacity, and implementing efficient data handling techniques can help mitigate these issues.

The observed performance increase in our distributed system, after shifting client-server communication to REST API and server-database communication to gRPC is due to many reasons:

1. gRPC uses Protocol Buffers for serializing structured data. Protobufs are more efficient (both in terms of CPU usage and bandwidth) than the JSON typically used in socket communications.
2. gRPC is built on top of HTTP/2, which introduces a binary framing layer which reduces overhead and improves the efficiency of data transmission compared to the more verbose text-based formats like HTTP/1.x used in socket communications.
3. By using REST API for client-server communication and gRPC for server-database communication, which often requires more complex, high-performance interactions, each part of the system is leveraging a protocol optimized for its specific use case.