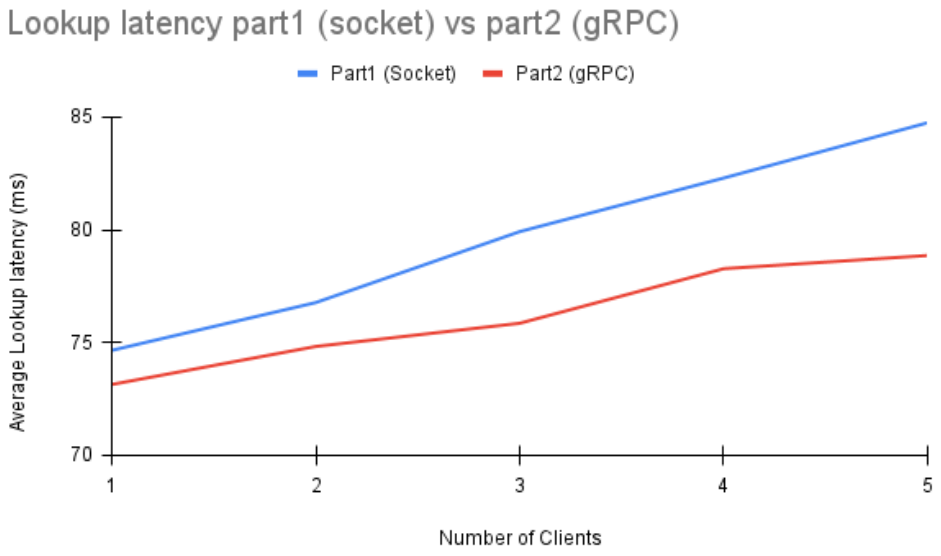
Evaluation and Performance Measurement

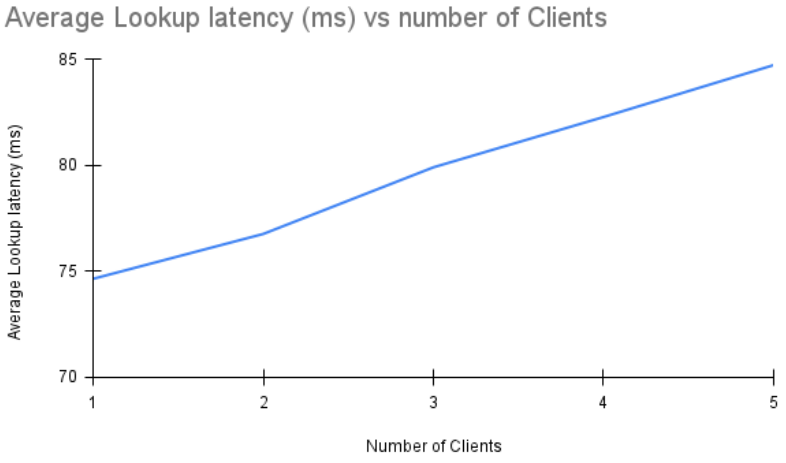
1. **Lookup latency part1 vs part2:**



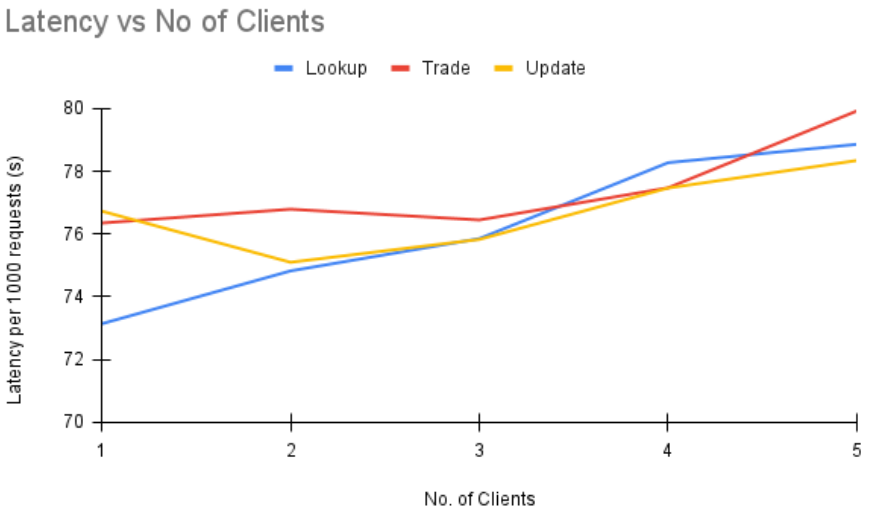
We observe that the Lookup latency for the socket communication (part1) is more than the lookup latency for gRPC communication (part2). Socket uses TCP protocol while gRPC uses HTTP/2 protocol. Most HTTP transfers are short and bursty, whereas TCP is optimized for long-lived, bulk data transfers. By reusing the same connection, HTTP/2 is able to both make more efficient use of each TCP connection and also significantly reduce the overall protocol overhead.

2. **Average latency vs number of clients (load):**

Part1 - socket communication:

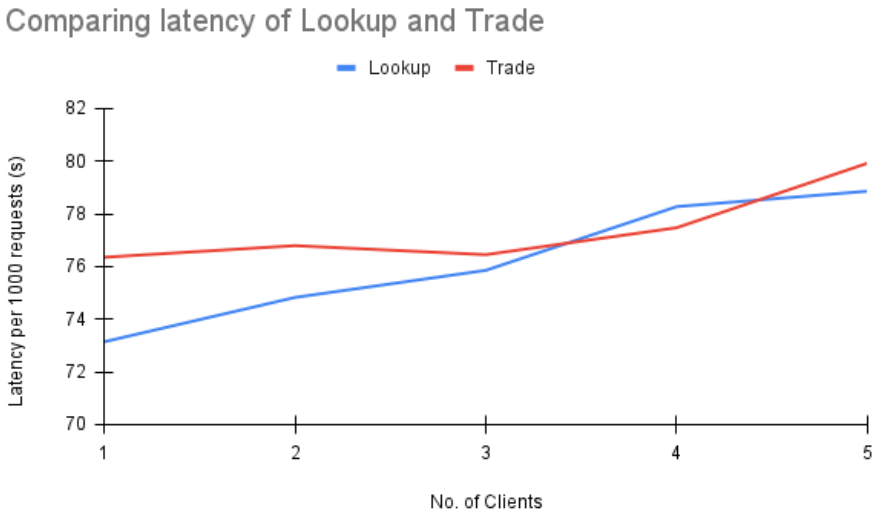


Part2 - gRPC communication:



From the above graphs, we observe that the latency increases with an increase in the number of clients (load). This is expected because, with the increase in clients, the overall number of requests to the server increases. Since the server processes these requests concurrently, threads handling requests from different clients will wait on the mutex lock more often increasing the average response time for each request.

3. **Lookup latency vs Trade latency in gRPC:**



From the above graph, we see that trade call latency is in general more than lookup latency. In the Lookup function, since threads just read the date from the database, we do not need to use any mutex locks. But in Trade functions, we write to the database. So in order to avoid race conditions (which lead to corrupt data), we use mutex locks. Since threads wait on the lock, the execution time of the trade function can be more than the execution time of the Lookup function.

4. **Latency thread pool size vs number of clients:**

   For thread pool size = 2

Number of Clients	Average Latency (ms)
1	93.8367
2	113.5667
3	158.7692

   The average latency increases when the number of clients is larger than the static thread pool size due to the request waiting. When the number of clients is the same as the thread pool size, both the threads can handle requests from the 2 clients. But when the third client is added to make requests, the request from one of the clients has to wait while the other 2 client requests are being handled. This will increase the overall average latency.