

Evaluation and Measurements

1. Deploy your system on three machines, with each of the three components on a different machine. BE SURE NOT TO USE PORT 80 for your code since it may conflict with processes run by others (also port 80 is reserved and typically not allowed for user processes). Run a client on a separate 4th machine and show that your code works properly by making different types of requests and printing appropriate log messages at the client and the components.

You can refer to the logs generated on your specific EC2 instances which you have used for deploying the servers.

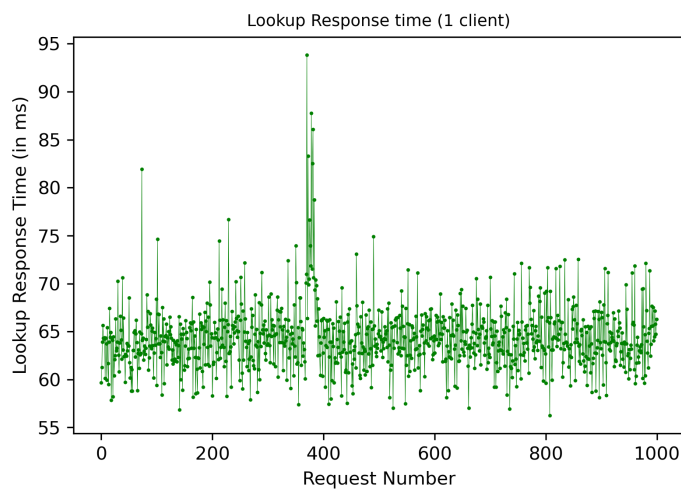
2. Compute the average response time per client search request by measuring the end-to-end response time seen by a client for , say, 1000 sequential requests. Also, measure the response times when multiple clients are concurrently making requests to the system, for instance, you can vary the number of clients and observe how the average response time changes.

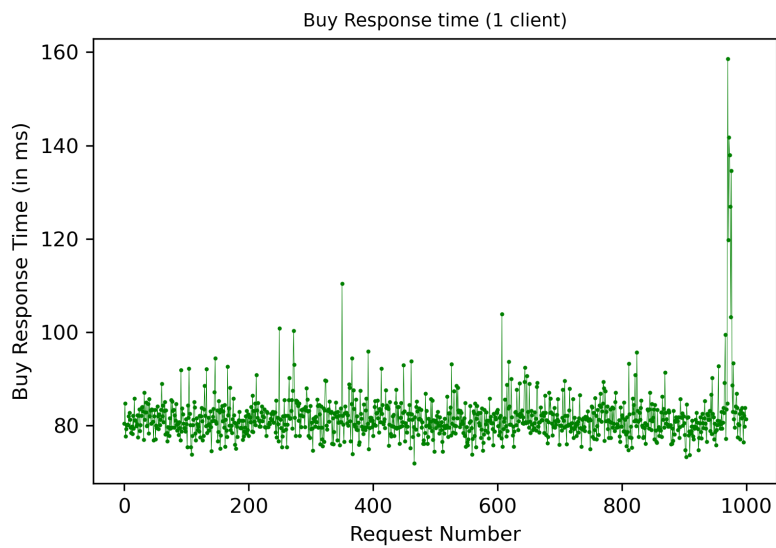
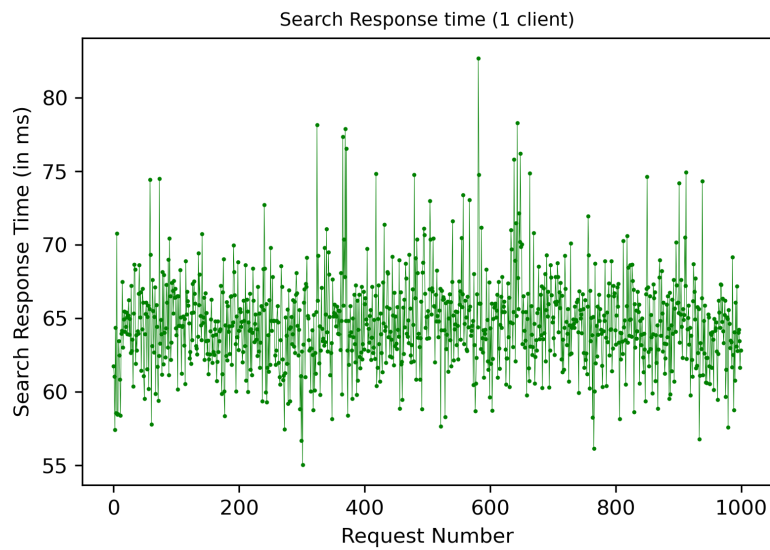
Single Client Metrics (Timings are as observed on client while calling the frontend server)

Average Lookup Response Time : 64.36858248710632 ms

Average Search Response Time : 64.66713571548462 ms

Average Buy Response Time : 81.80399775505066 ms



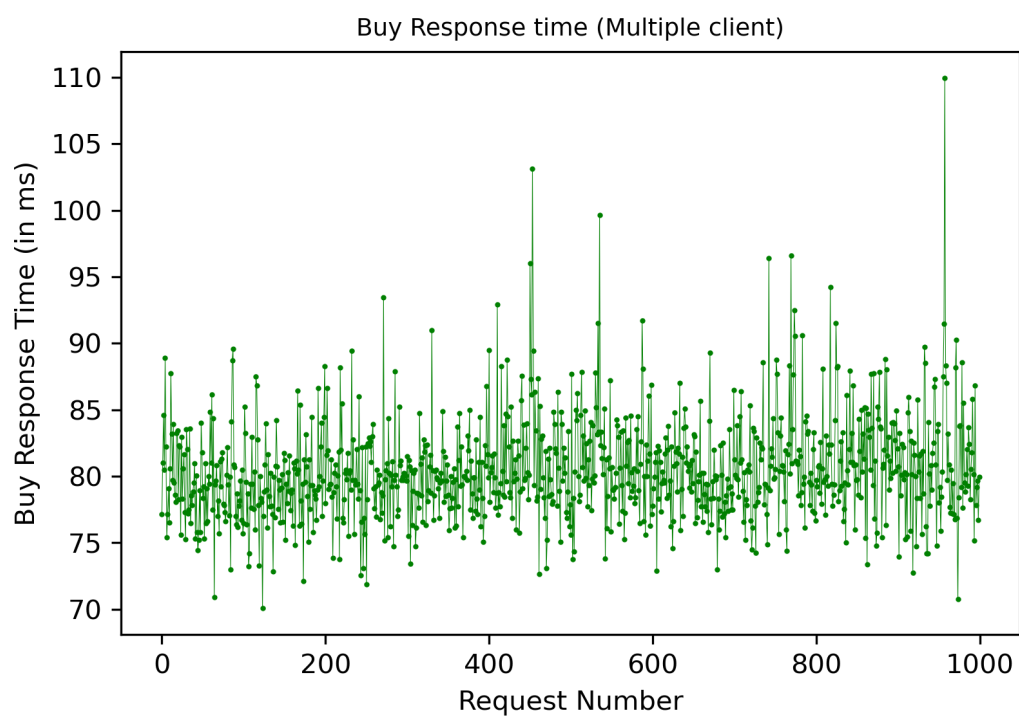
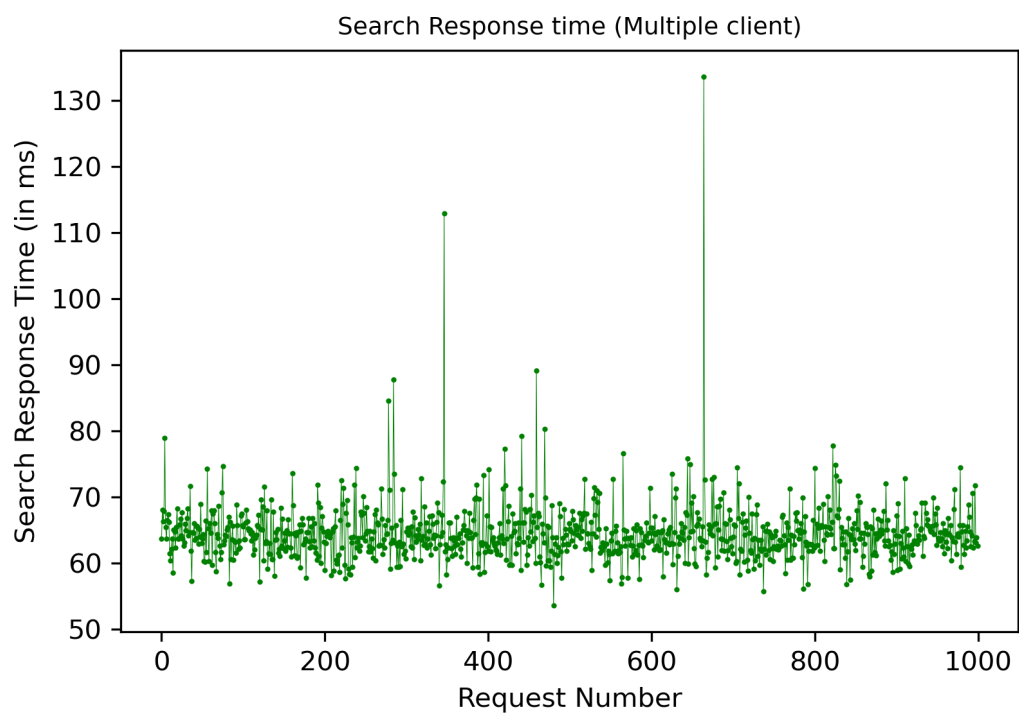


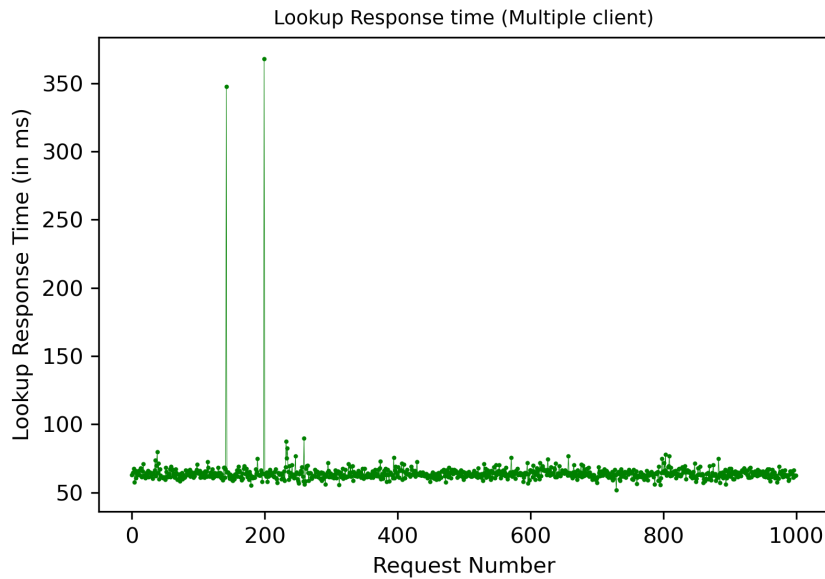
Multiple Client Metrics (Timings are as observed on client while calling the frontend server)

Average Lookup Response Time : 64.37189745903015 ms

Average Search Response Time : 64.53242754936218 ms

Average Buy Response Time : 80.49859762191772





Conclusion-

As it can be seen, the metrics for single client and multiple client systems are quite similar in nature, which represents the multi-threading which is automatically applied at the server end to deal with concurrent multiple client requests.

3. Following the idea shown above, break down the end-to-end response time into component-specific response times by computing the per-tier response time for query and buy requests.

Breakdown of Lookup Request

FrontEnd Server - 3.392816781997684 ms

Catalog Server - 60.29456806182861 ms

Conclusion

These values align with our expectations of these values. The front-end server merely transfers the request to the catalog server whereas the catalog server reads from the database and returns the data, thus taking up the majority of the time in the total response time breakup.

Breakdown of Buy Request

FrontEnd Server - 3.56614854861261 ms

Order Server - 19.273147583007814 ms

Catalog Server - 58.735509604921326 ms

Conclusion

These values align with our expectations of these values.

The catalog server again takes the majority of time since it reads and accordingly updates the final catalog database.

The order server takes a significant amount of time as well since it interacts with its own db and accordingly forwards the request.

The frontend server again takes the least amount since all it effectively does is request forwarding.