CS 677 Lab 2 Performance Evaluation Report

Dan Cline, Kyle Stevens, Tian Xia *UMass Amherst, Amherst College*

This document provides a performance evaluation of our implementation.

Performance of Sequential Requests

In this test, we perform 100 requests of each for all the items and topics, and measure the average latency for each combination of item/topic and requests. The test can be found in `tests/sequential`. The output can be found in `tests/output`.

item/topic	Response Time of `buy`(s)	Response Time of `lookup`(s)	Response Time of `search`(s)
item1	0.035602	0.014066	N/A
item2	0.036503	0.014149	N/A
item3	0.035496	0.014100	N/A
item4	0.034645	0.014235	N/A
gradschool	N/A	N/A	0.014113
systems	N/A	N/A	0.014110

From these numerical results, we can see that the response time for `buy` request is much higher than the response time for `lookup` and `search` requests. This can be attributed to the fact that `buy` requests need to go through another tier (frontend-order-catalog) compared to other requests (frontend-catalog). We also observer that the response time for `lookup` and `search` requests are surprisingly similar. This is mostly likely because these two requests both go through the same number of tiers.

Performance of Per-tier Sequential Requests

The test can be found in `tests/sequential`. The output can be found in `tests/output`. We only tested the per-tier requests on item 1 and the topic systems since the results for the other items or topics should be more or less uniform. The frontend performance logs can be found on the frontend server. The client-frontend time can be found by subtracting the end-to-end run time with the frontend-catalog time or frontend-order time. The results is as below.

Tier	`buy` item 1(s)	`lookup` item 1(s)	`search` topic systems (s)
Frontend - Order/buy	0.01029	N/A	N/A
Frontend - Catalog/query	N/A	0.00814	0.00714
Order - Catalog/update	0.01715	N/A	N/A
Client - Frontend/lookup	N/A	0.00678	N/A
Client - Frontend/search	N/A	N/A	0.00537
Client - Frontend/buy	0.00611	N/A	N/A

From these numerical results, we can see that the end-to-end response time is roughly split into

half per tier for both `lookup` and `search`. This makes intuitive sense because the client needs to go through two layers with similar time cost. The frontend-order seems to be the most time intensive. This is probably because it needs to both query the catalog server and update with a json payload.

Performance of Concurrent Requests

In this test, we ran varying numbers of clients to buy the same item and measured the average end-to-end response time for 100 sequential requests for each spawned client. All test can be found in `tests/concurrent`. The output can be found in `tests/output`.

Number of Concurrent Clients	Average Response Time(s)
2	0.03687
5	0.08694
10	0.15386
20	0.26983
40	0.43128

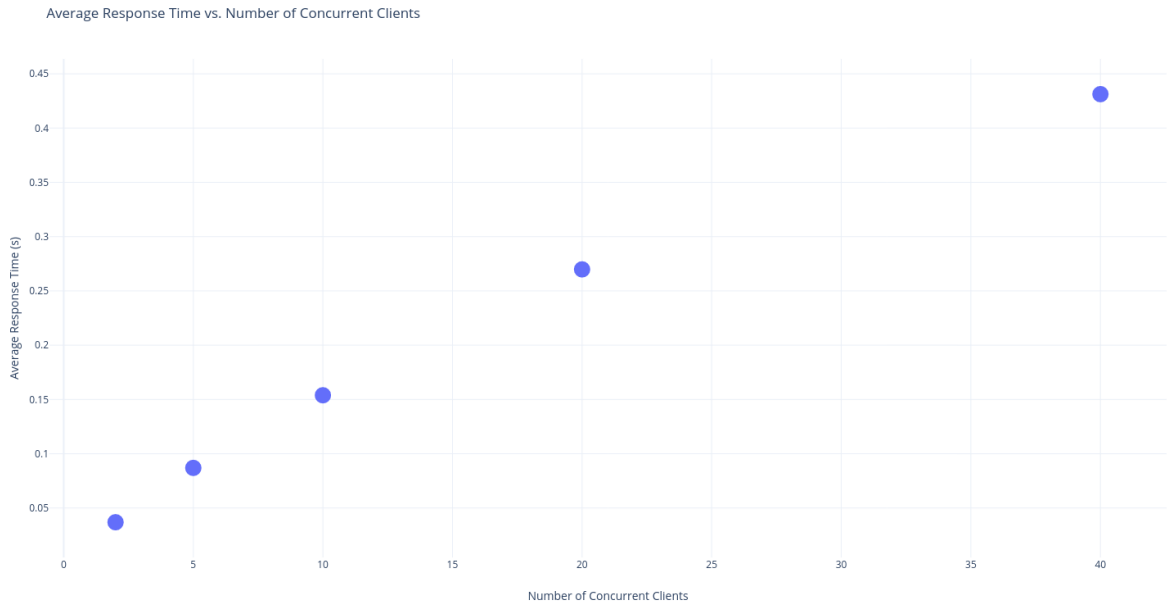

Figure 1: Latency for 140 Sequential Buy Calls

From the graph, we can say that the response time increases when the seller receives increasing amount of concurrent requests from buyers.