CS677 LAB3 Testing and performance document –

Testing

We performed manual testing for each test scenario possible and took screenshots of the output. These are included in the output directory. The tests performed were –

1. Checking for cache gets on subsequent reads.
2. Checking if cache is erased when a write is performed.
3. Checking round-robin routing works properly.
4. Checking if heartbeat and fault tolerance work properly.
5. Increasing heart-beat duration to 20 seconds to make sure fallback request works properly.
6. Checking if fault recovery works properly.

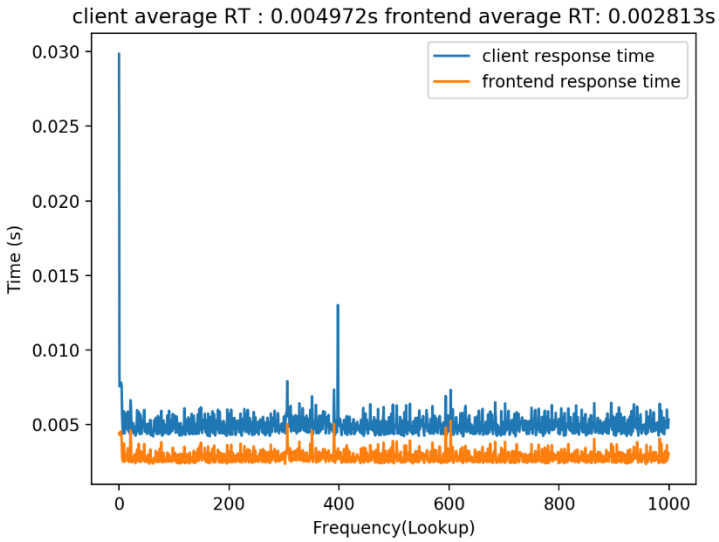
The outputs of all these tests are included in documents in the output folder.

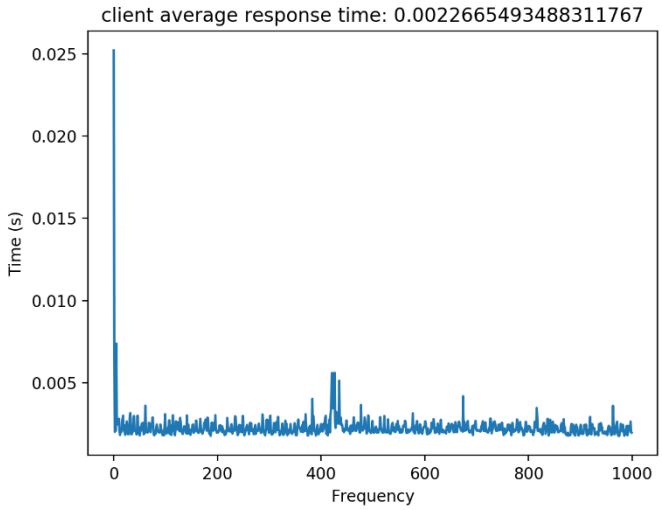
Docker

We created docker containers for each service and ran the images on our local machine. We were able to get them working and the outputs are added to the output folder. The images are uploaded to the UMass box for access. Instructions to run these containers can be found in the README page inside the Docker folder on the repository.

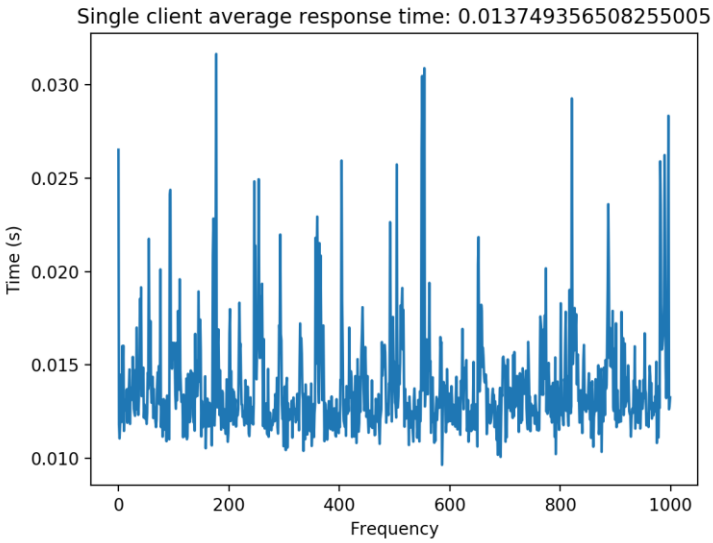
Performance analysis –

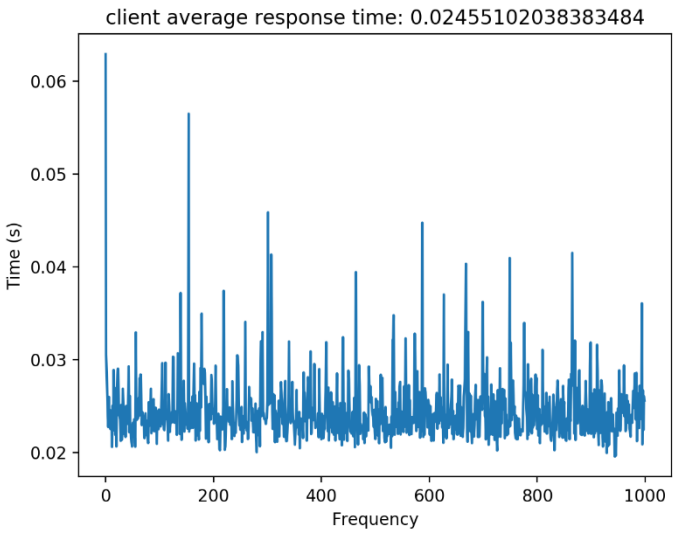
1. We performed a comparison of lookups for implementations with and without cache. There is a significant performance bump for subsequent requests with a cache as can be seen in the graph below. The first graph indicates client response time without a cache and the second one shows the client response time with a cache. Average response time without a cache was 0.05s vs with a cache was 0.02s.

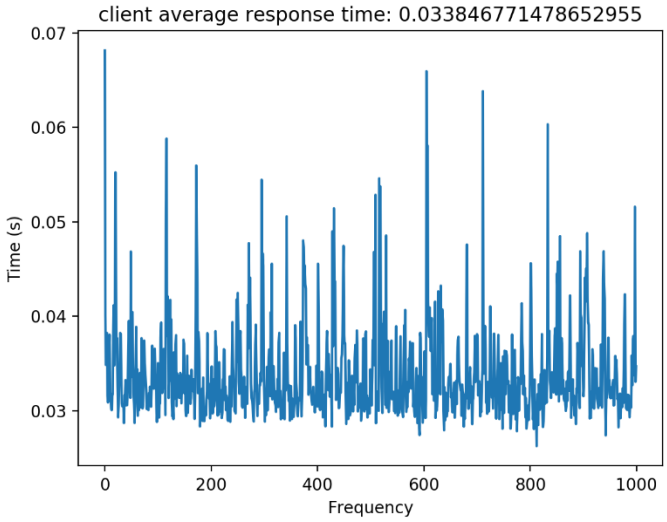


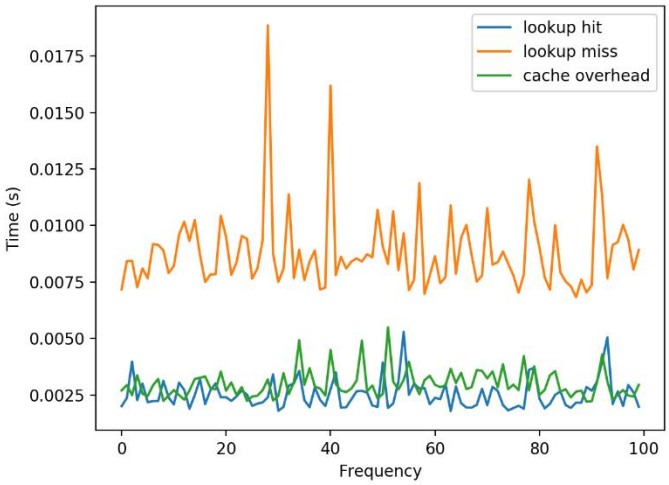


2. We also performed a sequential experiments on buy operations as we added several middle layers and consistency protocols we hypothesized that the time for a buy operation should increase. This is visible in the graphs below. Previously the average response time for a buy request was 0.014s and now it is 0.024s. This can be attributed to overhead for maintaining consistency and for updating the distributed cache.









3. This experiment covers the buy requests for 2 order and 2 catalog servers which has added time as the buy request locks the load balancer and then an update is performed to each catalog server. The average response time in this case is 0.033s.

4. Cache performance experiments – Finally we performed cache performance experiments to prove to show the differene in time taken in a cache hit and a miss. The cache hit takes multiple times over more time than a hit.

Cache hits reduce the time taken by atleast 2 times than cache misses. Here a cache miss needs to go through the load balancer to the server and then return back. The graph can be seen above.