

1. We Deployed the system on three machines, with each of the three component on a different edlab machine (elinux2,3,7) with the replicas being on different machines. Then ran a client on a separate 4th machine (elinux1) and tested the client code working properly by making different types of requests. The configuration in detail is mentioned in the config files. We then computed the average response time per client search request by measuring the end-to-end response time seen by a client for 1000 concurrent requests with caching. The 2 variants of this experiment were that we used just one replica and sequential requests for both replicas(Catalog) Table 1 depicts the average response time for different functionalities in seconds with caching.

Operations	get_quantity	update cost	update_quantity	buy	lookup	list	Total
Concurrent requests- 2 catalog replicas	0.004267	0.09589	0.155693	0.1817	0.00408	9.97E-07	0.3
Concurrent requests-1 catalog replicas	0.01936	0.0744	0.0809	2.1104	0.0352	0.0	2.37
Sequential requests	1.7795	2.6064	3.1005	1.83	3.351	2.931	15.78

Table 1: With Caching

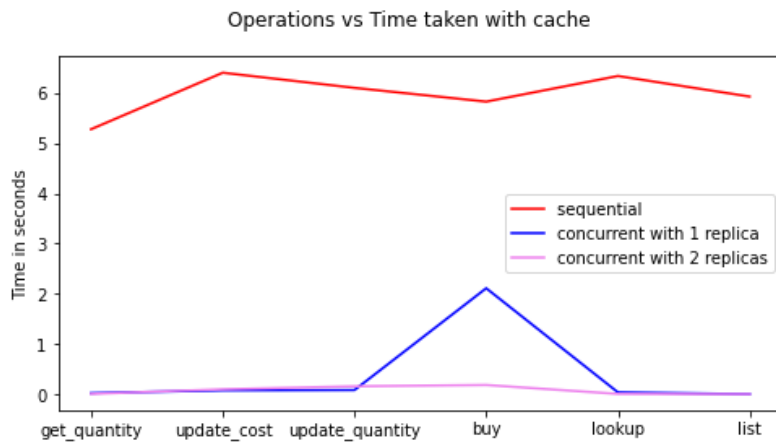


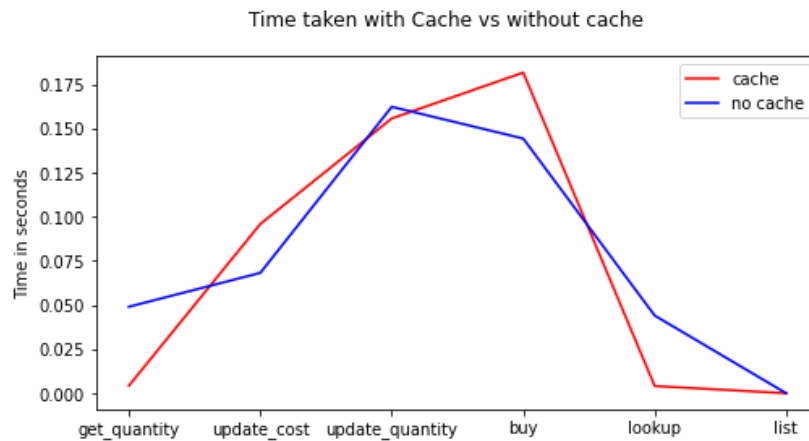
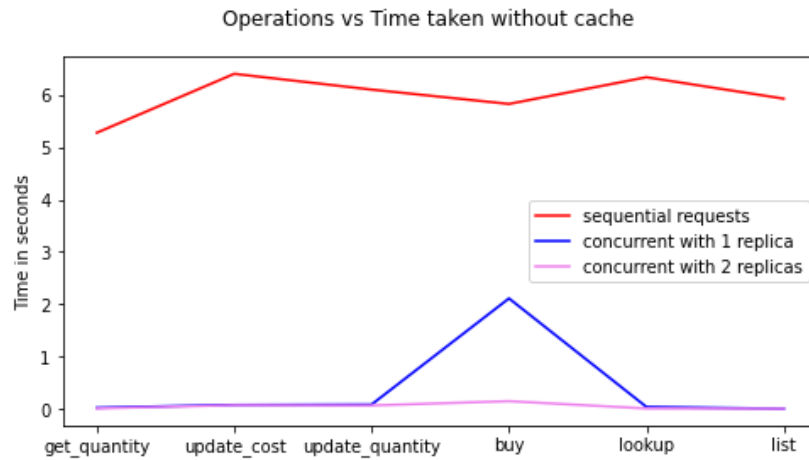
Figure 1

We also computed the average response time per client search request by measuring the end-to-end response time seen by a client for 1000 concurrent requests without caching. Table 2 depicts the average response time for different functionalities in seconds.

Operations	get_quantity	update cost	update_quantity	buy	lookup	list	Total
Concurrent requests- 2 catalog replicas	0.049	0.0682	0.1623	0.1443	0.044	0.0010	0.663
Concurrent requests-1 catalog replicas	0.04576	0.0515	0.1730	2.079	0.044	0.01	3.5264

Sequential requests	5.2795	6.4064	6.1045	5.83	6.341	5.931	23.94
---------------------	--------	--------	--------	------	-------	-------	-------

Table 2: Without caching



Analysis:

Thus we can see significant improvements with caching in case of read requests (about 20-100 times greater efficiency). Cache significantly decreases time of read transactions. We can also see that cache highly increases efficiency when there are sequential requests as well as just one catalog replica.

2. In this experiment we executed a write request(update_cost) before a read request(lookup). We measured the end-to-end average response time seen by a client for 1000 concurrent requests of update item followed by lookup of same item with cache. We used the same configuration as above.

update_cost	lookup
0.08089	0.01033

The overhead of the cache is not significantly high because the update cost transaction took almost the same time on a average. However the latency due to the subsequent cache miss is significant because the transaction took more than double the time it takes with a cache hit.

3. We have performed experiment 3 in the starter script