Key Value Store - Performance Analysis

For doing performance analysis, we created a multithreaded client with a variable number of threads. We have taken the server thread pool size as 4.

Each thread is acting as a concurrent client which is running in batch mode, taking as input a file with 10000 randomised GET/PUT/DEL requests. Client spawns multiple threads which connect to the server, i.e. constantly increasing the load on the server, and we stored the timing data for each load to plot the below graphs. This was done for both LRU and LFU caches for a maximum of 15 concurrent clients, each with 10000 requests, i.e. in total 150000 requests at maximum load. We considered the average response time for each request in the graph.

PERFORMANCE GRAPHS

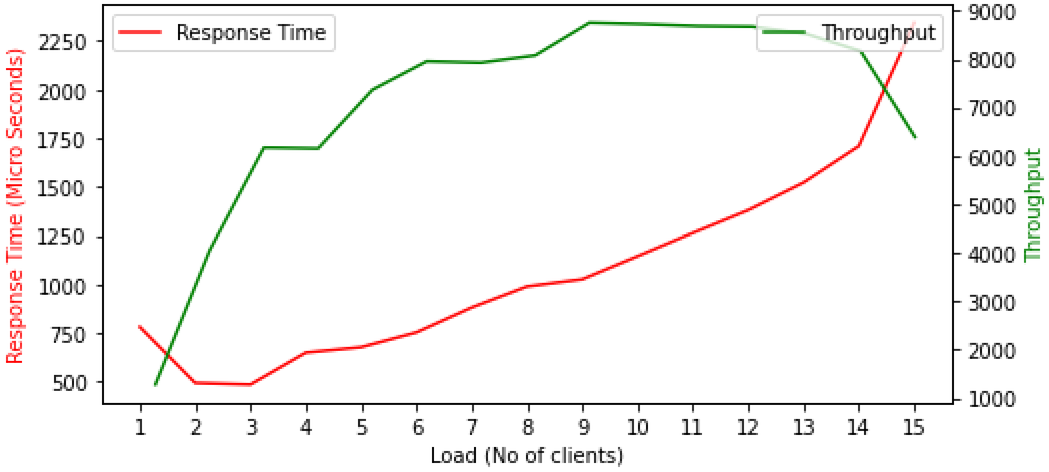

Fig. 1 (LRU Cache)

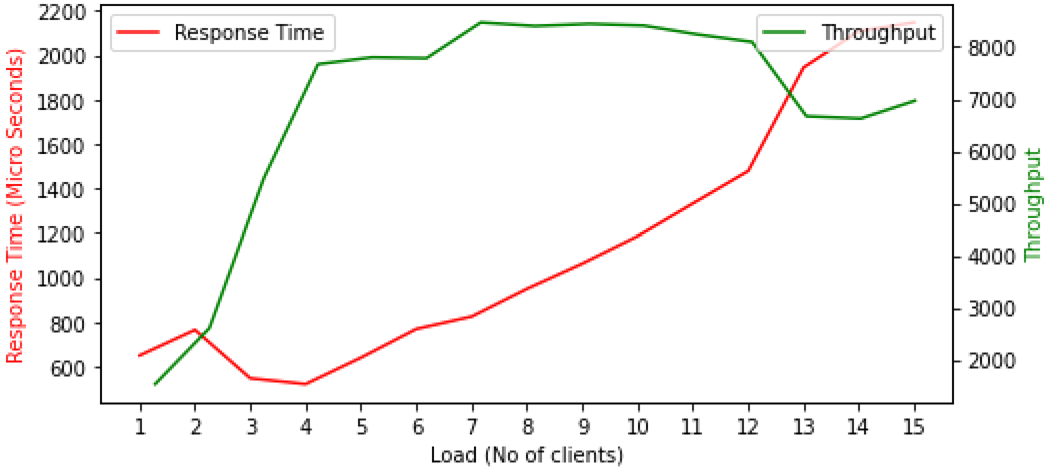

Fig. 2 (LFU Cache)

Fig. 1 shows the performance graph for LRU cache and Fig. 2 shows it for LFU cache.

Throughput vs Load :

Firstly, looking at the graphs for Throughput vs Load, for LRU and LFU, we observe that, in the start increasing the load dramatically increases the throughput, as can be seen from the slope of the graph at the beginning. This trend is seen upto a certain load, after which the slope of the graph reduces. For a small portion, the graph is horizontal or the slope is nearly zero meaning throughput remains constant, and the server has reached its peak capacity.Increasing the load even more now, causes the graph to go down or the slope to go negative, meaning the server has breached it's capacity to handle increasing loads and the throughput decreases drastically.

Response Time vs Load :

In both the figures we can observe that, in the start increasing the load increases the average response time of the requests. This thing continues only till the server is able to achieve its maximum capacity after which the burden increases on the server on further increasing the load, and because of this we observe that the average response time per request increases dramatically on further increasing the load.