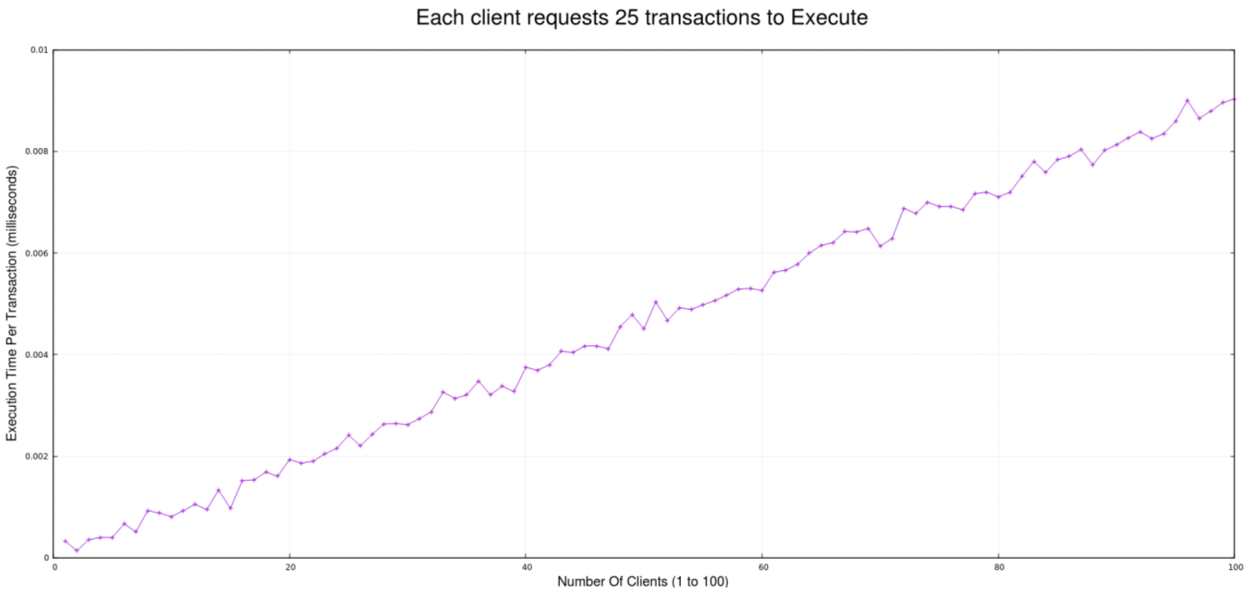
AOS Project 1

Name: Intekhab Naser ID: ZC11577

Performance Results

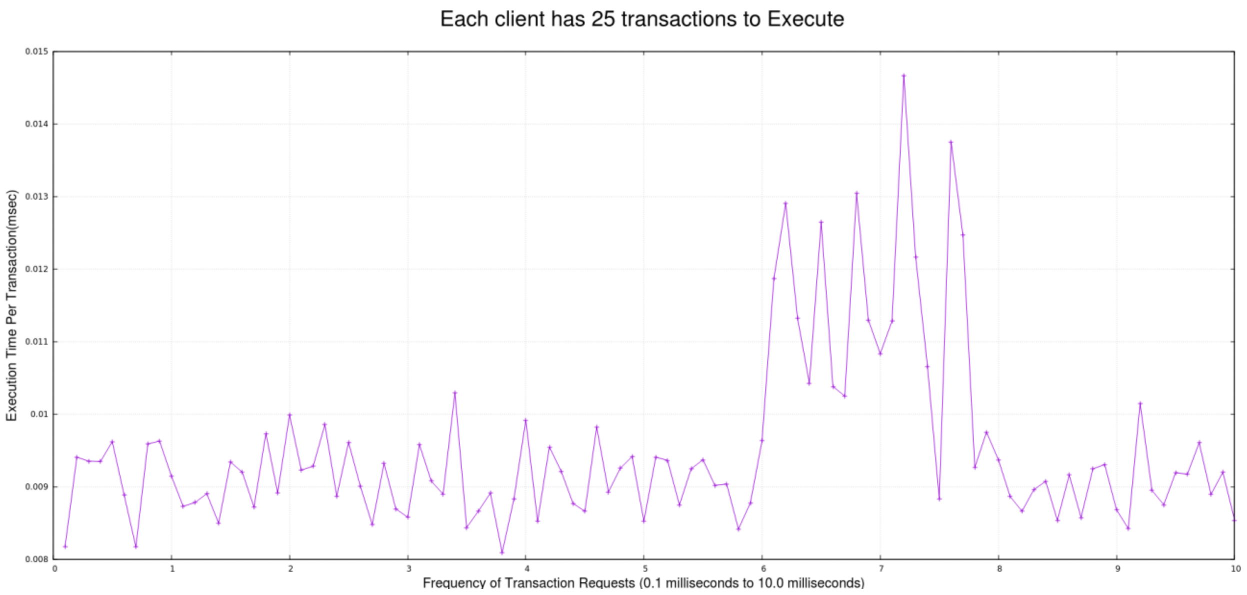
The program design is such that it uses sockets, processes, threads and synchronization. In order to analyze the efficiency of these implementations, I executed some performance evaluations. I divided the evaluation into two cases:

Case (i): In the first case of performance evaluation, I ran the tests by using a set of periodic requests at each client (for instance: each sending a request every 2 secs) and by varying the number of clients connected to the server. I measured the average time to complete each transaction for each client. The following is a plot for a graph to show average time to complete each transaction as number of clients are increased (from 1 to 100).



- In this case, I have iteratively increased the number of concurrent clients from 1 to 100 simultaneous clients.
- The number of points taken in this case on x-axis is 100.
- The total number of transaction in each of these concurrent clients is restricted to 25 transactions per client.
- The frequency of requesting each transaction to the server by all the servers is also restricted to 2 seconds.
- The execution time required per transaction gradually increases from less than 0.001 milliseconds to 0.010 milliseconds

Case (ii): In the second case of performance evaluation, we run tests by fixing the number of clients to 25 and then varying the request rate. i.e., one request every 0.1, 0.2, ...1 5 secs. We measured and plot the average time to complete each transaction as the request rate is varied.



- ➢ In this case, I have kept the number of concurrent clients invariable at 25 simultaneous/concurrent clients.
- ➢ The number of points taken in this case on x-axis is 100
- ➢ The total number of transaction in each of these concurrent clients is again restricted to 25 transactions per client.
- ➢ The frequency of requesting each transaction to the server by all the clients is gradually decreased from a request every 0.1 milliseconds to a request every 10 milliseconds.
- ➢ The execution time required per transaction remains in a small window of 0.08 to 0.01 milliseconds. While there is a small duration of extreme spikes, may be due to the interest service executing its add interest function and blocking the clients for that moment.