Homework 4: Scalability: Exchange Matching

Suchuan Xing(sx80) Xuhui Gong(xg73)

Dr. Brain Rogers

ECE - 568

Contents

1 Scalability Test

The project is built in java using synchronized to handle multi request. This is not as high performance as Optimistic concurrency control but can handle multi request without making too many threads which will make program crushed.

We design a java script(a client) which can generate many create and transaction requests at the same time, each of those will return error message or successfully change or get information with database, and we will show our result following.

With the client designed, we send our server with 10, 50, 100, 150, 200, 500, 1000, 1500requests at the same time respectively to test our server's scalability. And our client will return the executed time and latency of each request. The relationship between average time and number of request, the Frequency distribution curve of executed time and the number of cores and the overall executed time are listed as followed.

2 Results and Analysis

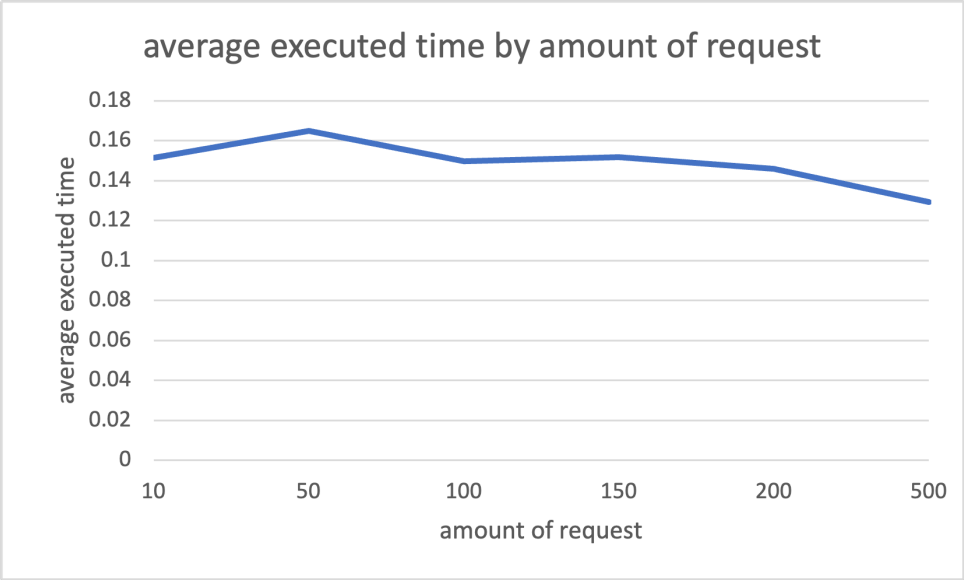

Figure 1: the relation between average executed time and request amount

From this picture, we can tell that our program can handle more request within a set time when the request's number is larger. This is because when number of requests is getting larger and larger, the time used to handle thread switching between each other can be omitted regarding to the latency.

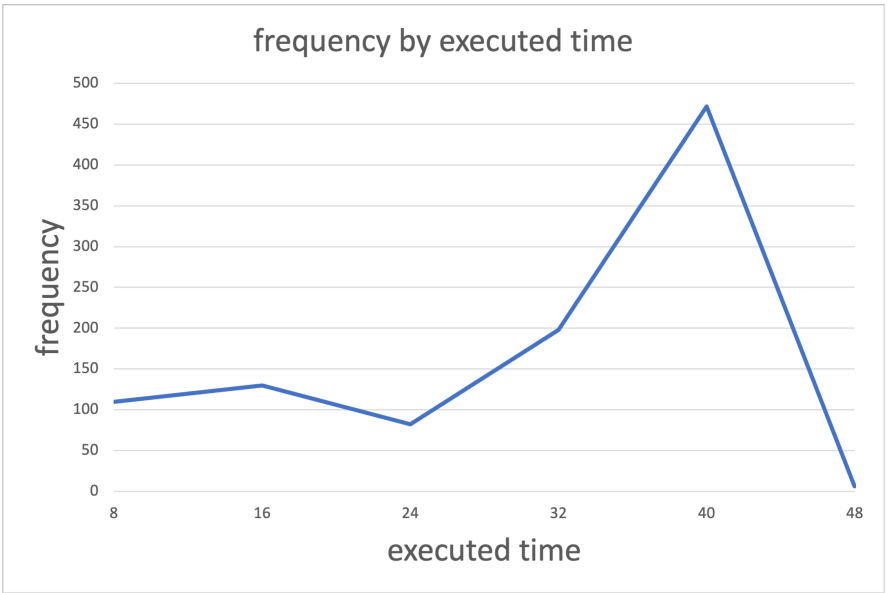

Figure 2: the relation between frequency and executed time

As the picture shows, most of our request can be done within time 40s and the overall amount of request is 1000. This shows our server's average **latency** is around 40s and **throughput** is around $1000/40 = 25$request/s.

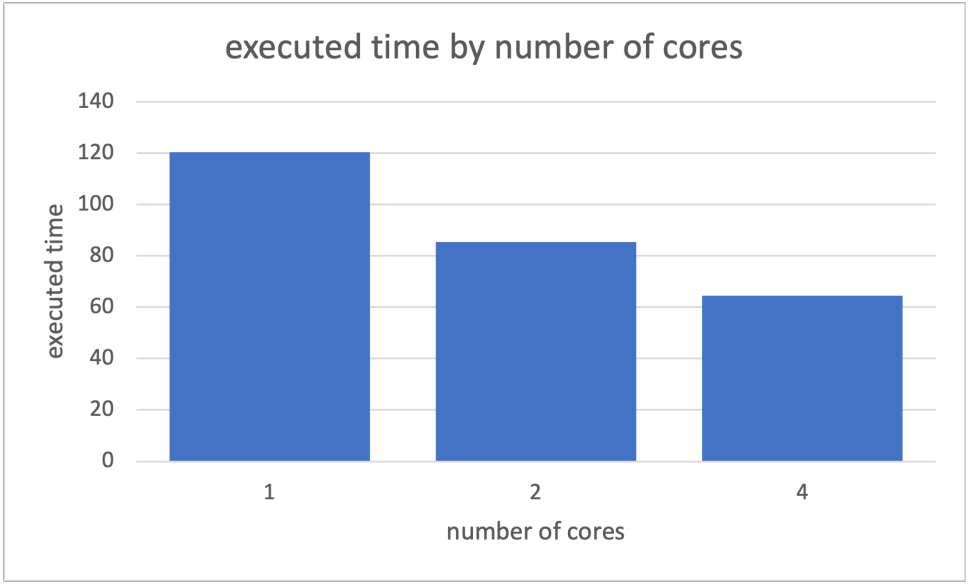

Figure 3: the relation between core number and executed time

This picture shows the runtime of handling 1500 requests with 1 core, 2 cores, 4 cores. The difference is not increasing completely consistent with the increasing of number of cores. But we can tell when dealing with large amount of requests, with more number of cores, our handling speed will increase at a higher standard.