

Report

XC187

CT300

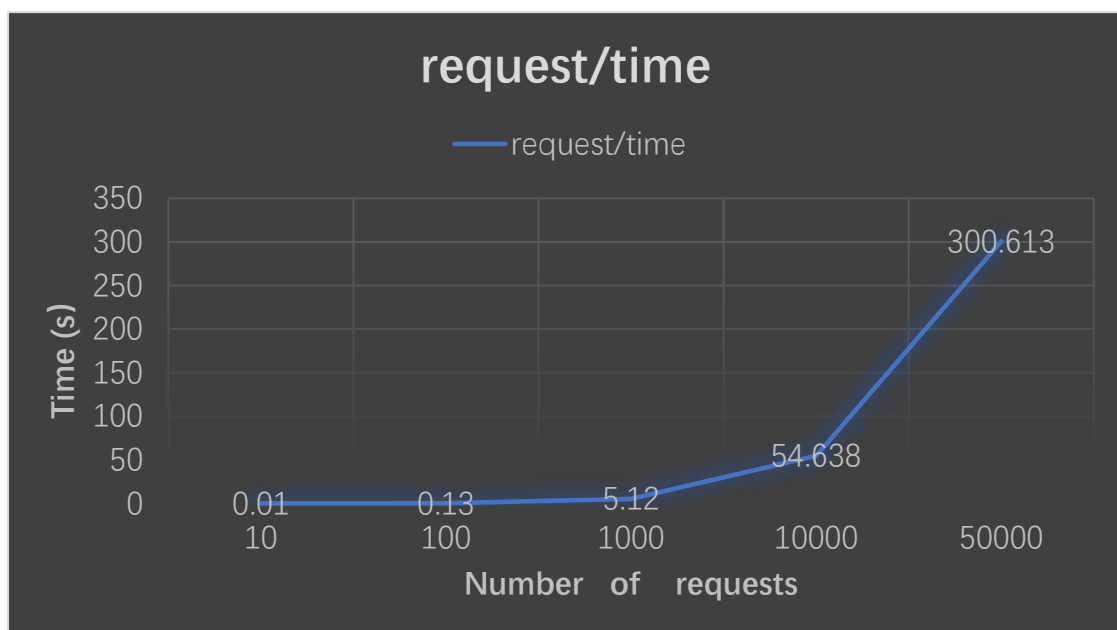
Introduction:

In this project, we implemented an exchange matching engine, which allows users to buy and sell orders, as well as match orders automatically. It is written using C++ with the multi-thread and socket programming, and we ran the test cases in multiple cores to have a better view of the scalability of the server. The report will give a demonstration of the experiment results and some in-depth analysis.

Number of threads and Execution time

In our testing process, we tested the number of requests, ranging from 10 to 50000, and the corresponding execution time. When the number of requests is low, the execution time is also very low. However, when the number of requests comes to a large scale, the execution time goes very high. The following table and graph shows the result.

Number of requests	Execution time(print)
10:	0.01
100:	0.013
1000:	5.120
10000:	54.638
50000:	300.613



Multi-core and the corresponding execution time

In this part of testing, we used multi-thread on the client-side code. There are totally 10 threads and each thread keeps sending requests.

We tested the runtime of server handling requests with different number of cores running the code, which are 1,2 and 4 cores. The running time under different cores is shown below in the table.

Multi-core	4	2	1
10	0.01	0.01	0.01
100:	0.013	0.02	0.023
1000:	5.120	5.07	8.648
10000:	44.638	54.0	56.036

The following table shows the throughput of different cores under 10000 requests. It shows that 4 cores has the highest throughput, while 1 core has the lowest. The scalability is based on hardware and load. When there are more cores, the server can therefore handle more requests as well as gain higher throughput.

Throughput	4	2	1
10000 requests	224.02	185.19	178.57

