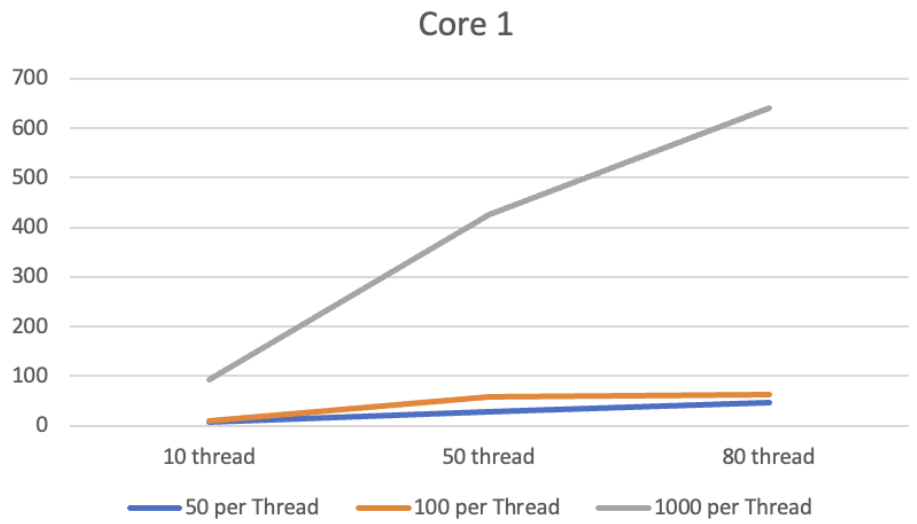
erss-HW 4

For testing our server's scalability, we have established a Client under app.client, with source code Client.java and ClientRunnable.java which will thread a number of threads sending several lines of queries to server at the same time.

We compose the docker environment by setting using core to be 1, 2, and 4. The followings are the table

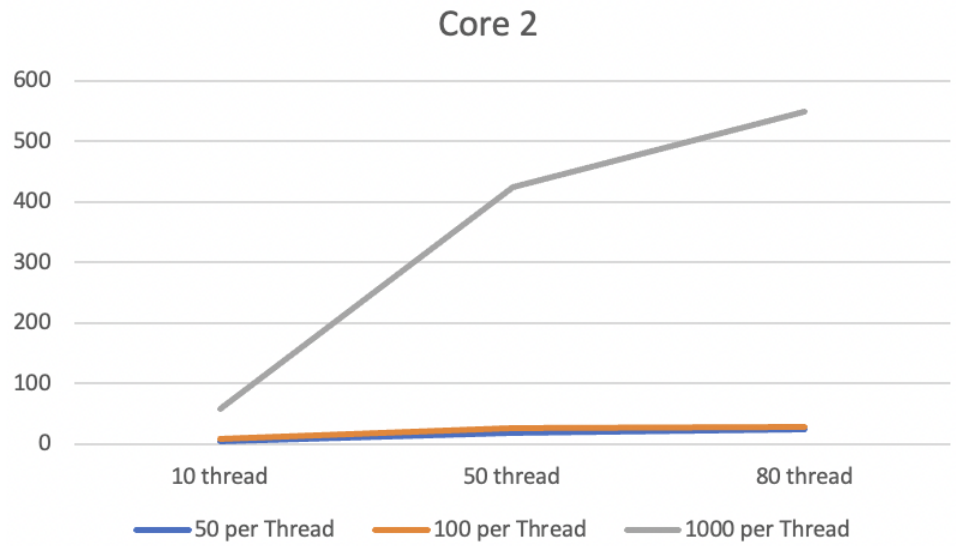
Docker CPU Core: 1

Request \Thread	10	50	80
50 per Thread	8 sec	28 sec	46 sec
100 per Thread	10 sec	58 sec	62 sec
1000 per Thread	92 sec	425 sec	640 sec



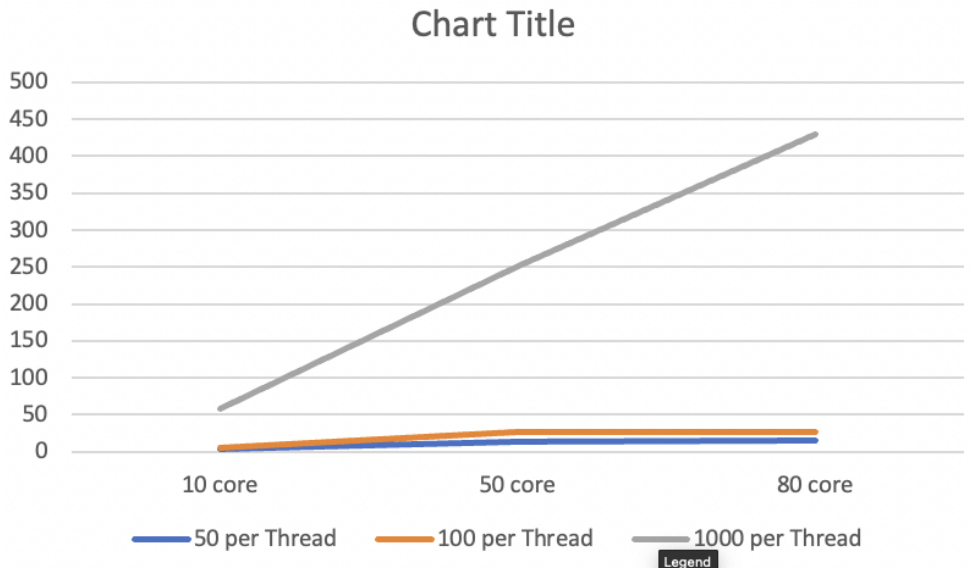
Docker CPU Core: 2

Request \Thread	10	50	80
50 per Thread	5 sec	18 sec	24 sec
100 per Thread	9 sec	27 sec	28 sec
1000 per Thread	58 sec	425 sec	550 sec



Docker CPU Core: 4

Request \Thread	10	50	80
50 per Thread	4 sec	14 sec	16 sec
100 per Thread	5 sec	26 sec	27 sec
1000 per Thread	58 sec	252 sec	430 sec



The corresponding Table and Diagram shows that server running on multicore CPU increase the efficiency and decreasing the running time. For lower queries, by increasing the thread number, the time consumption was increasing correspondingly. By increasing the queries of each thread, the differences of time consumption is increasing, and the time consumption rate is increasing when adding more thread as well. Our server running on 1 core CPU will takes 640 sec to solve 1000 queries per thread, which is only half of the time it takes while on 4 core CPU.