CS490-Project 1-Report
Zhiyuan Zheng
Mingsheng Xu

Performance

We test the performance of different server implementations by simultaneously running 4 instances of dummy client which will sends out 100k registration requests.

We use 4 terminals in which we ssh to four different machines. We type the command needed to start a dummy client in each terminal in advance. Then we select each terminal and press enter quickly after we spin up the server.

Server Type	Avg. Latency (in ms)	Std dev. Latency	Avg. Throughput (responses/sec)	Std dev. Throughput
Single-threaded	12.27905	4.73913	12.39734	0.00747
4-thread	26.32031	14.63499	18.71521	0.01441
8-thread	24.72410	13.14913	22.34985	1.04275
16-thread	22.87186	16.79049	25.91714	1.44545
32-thread	4.0584	0.71574	27.00851	2.35117
RMI	2.10702	1.11662	241.411	54.1847

From the table above, we could see that as the number of threads server has increases, the throughput tends to increase non-linearly, and the latency tends to decrease.

Regarding to RMI server, it is much faster than server implementations that using TCP. because it objects which are in fact remote can be treated as though they were local.

Implementation

1. Single-threaded Server: It is relatively easy to implement. We don't need to worry about concurrent access to a UserGroup data structure. The only concurrent issue is a thread wants to update an user liveness while another is going to check the liveness of same user.

2. Multi-threaded Server: It is toughest to implement. We should solve concurrent access problems by modifying underlying data structure that stores user information.
3. RMI Server:Both Client and Server have three component:Main function,interface,its classes.Server contain a list of client