Simulation

Problem:

Record the time needed for each message delivered from client A to client B, and draw a chart showing the average delivery time of messages between 10, 100, 1000, 10000 clients. For this requirement, you should automate the process.

Brief explanation on how this practices above was solved:

The idea was to create an array of client objects and start creating a thread for each client object. Timestamp class was used to measure the timing, when a particular message (A by default text message "ping" has been sent to each client) is delivered. For each client, the timing of message sending and receiving is tracked by writing those values into two different text files. Later on, by reading the message sending and receiving time of each pair of clients from those two different text files, the message delivery time was figured out (By measuring the difference). Then for a number of connections, the average has been calculated and plotted on a graph where the X-axis represents the NUMBER OF CONNCECTIONS and the Y-axis represents the AVERAGE DELAY TIME PER NUMBER OF CONNECTIONS.

User Manual:

1. There are primarily 4 files in each directory, which are: **AverageTimeDisplay.java, AverageTimeGraph. java, Clientx.java, Serverx.java**

2. Run the **Serverx.java** file. An IP address will pop up. Update the IP address in **Clientx.java** file. The port number used for the purpose is **6667**.

3. Run the **Clientx.java** file.

4. Running **Clientx.java** file will create two text files called **test1.txt** and **test2.txt** in the same directory**.**

5. There is a file named **AverageTimeDisplay.java,** which will give the output value of average delivery time of all the connections. The screenshot is given below:

PTO

```
83
84
85
86
        List<String> listing2 = new ArrayList<String>();
        List<String> hourpart2 = new ArrayList<String>();
        List<String> minutepart2 = new ArrayList<String>();
```

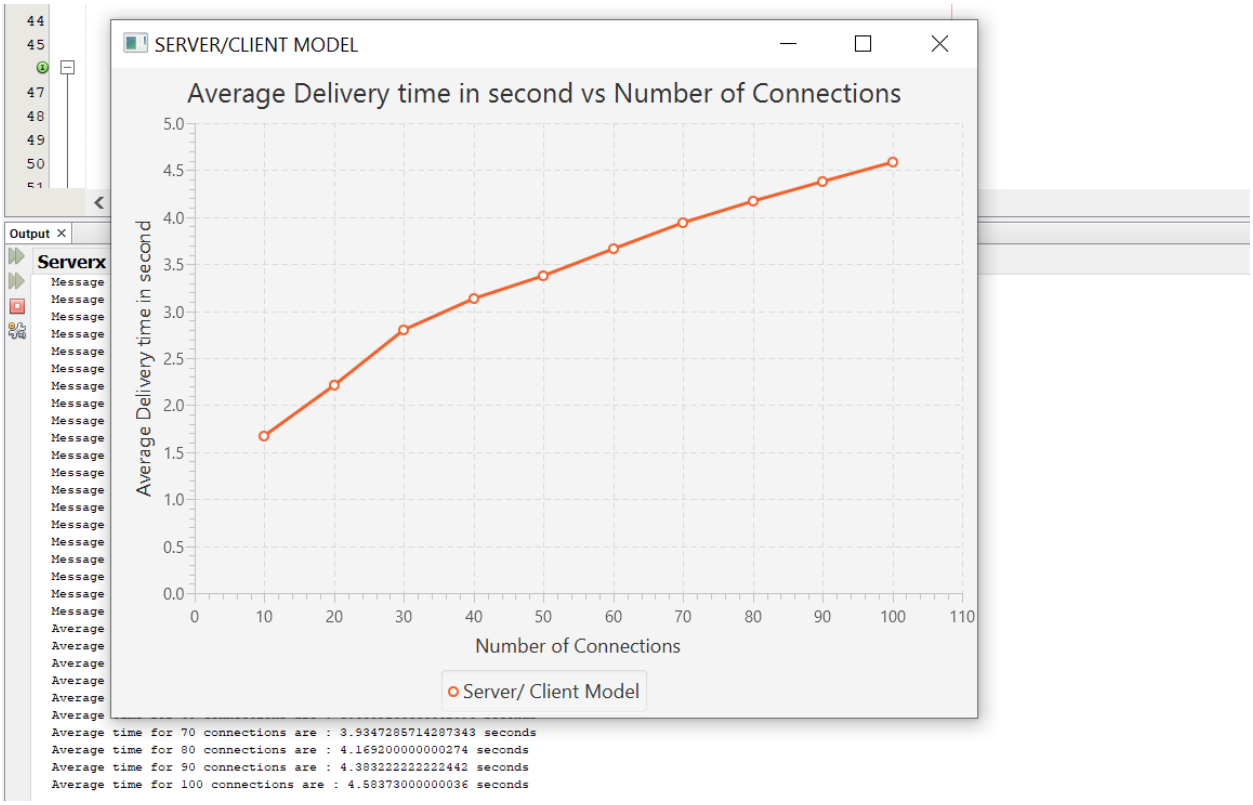
Output

Serverx (run) | Clientx (run) | AverageTimeDisplay (run)

```
Message delay time for number 81 pair of clients:6.009999999994761
Message delay time for number 82 pair of clients:6.009999999994761
Message delay time for number 83 pair of clients:6.073999999993248
Message delay time for number 84 pair of clients:6.073000000003958
Message delay time for number 85 pair of clients:6.127999999996973
Message delay time for number 86 pair of clients:6.128000000011525
Message delay time for number 87 pair of clients:6.167000000001281
Message delay time for number 88 pair of clients:6.168000000005122
Message delay time for number 89 pair of clients:6.237999999997555
Message delay time for number 90 pair of clients:6.239000000001397
Message delay time for number 91 pair of clients:6.283999999999651
Message delay time for number 92 pair of clients:6.283999999999651
Message delay time for number 93 pair of clients:6.334000000002561
Message delay time for number 94 pair of clients:6.334000000002561
Message delay time for number 95 pair of clients:6.422000000005937
Message delay time for number 96 pair of clients:6.420999999987544
Message delay time for number 97 pair of clients:6.505000000004657
Message delay time for number 98 pair of clients:6.505000000004657
Message delay time for number 99 pair of clients:6.555000000007567
Average time for 10 connections are : 1.6654999999984283 seconds
Average time for 20 connections are : 2.215349999999307 seconds
Average time for 30 connections are : 2.796266666665906 seconds
Average time for 40 connections are : 3.1345500000003086 seconds
Average time for 50 connections are : 3.3723200000001814 seconds
Average time for 60 connections are : 3.6589166666662396 seconds
Average time for 70 connections are : 3.9347285714287343 seconds
Average time for 80 connections are : 4.169200000000274 seconds
Average time for 90 connections are : 4.383222222222442 seconds
Average time for 100 connections are : 4.58373000000036 seconds
BUILD SUCCESSFUL (total time: 0 seconds)
```

6. For Graphical representation, there is **AverageTimeGraph.Java** file. "**jfxrt.jar"** file should be kept in the same directory to run the **AverageTimeGraph.Java** file. Download the jarfile from the link given below.

Run, Javac –classpath "……\Simulation_upto_100_clients\jfxrt.jar" **AverageTimeGraph.Java**




```
Average time for 70 connections are : 3.9347285714287343 seconds
Average time for 80 connections are : 4.169200000000274 seconds
Average time for 90 connections are : 4.383222222222442 seconds
Average time for 100 connections are : 4.58373000000036 seconds
```

7**. AverageTimeDisplay.java** and **AverageTimeGraph.Java** will read value from **test1.txt (Message Sending time)** and **test2.txt (Message Receiving time)** files and calculate the final average delivery time for a number of clients.

8. There are two different folders: **Simulation_upto_100_clients** and **Simulation_upto_10000_clients.**

9**. Simulation_upto_100_clients** directory will run the server where 100 clients are connected. And it will show average message delivery time of 10,20,30,40,50,60,70,80,90 clients.

8. **Simulation_upto_10000_clients** directory will run the server where 10000 clients are connected. And it will show average message delivery time of 10,100,1000,10000 clients.

Jarfile link:

https://unmm-my.sharepoint.com/:f:/g/personal/hmdadal_unm_edu/Ejnf27-hIHxGjzzOheBZUkoBLm4BsO68T2qN5CwsYiVWDQ?e=MiAba4