Report:

Introduction:

In this Homework, we are testing the Scalability of a Stock-exchange program. This program is implemented by C/C++, and the DataBase is Postgresql. By using this program, user can create account, add symbol and make transactions, queries and cancel their opened transactions. The system will match the opened transactions and find the suitable price for the user.

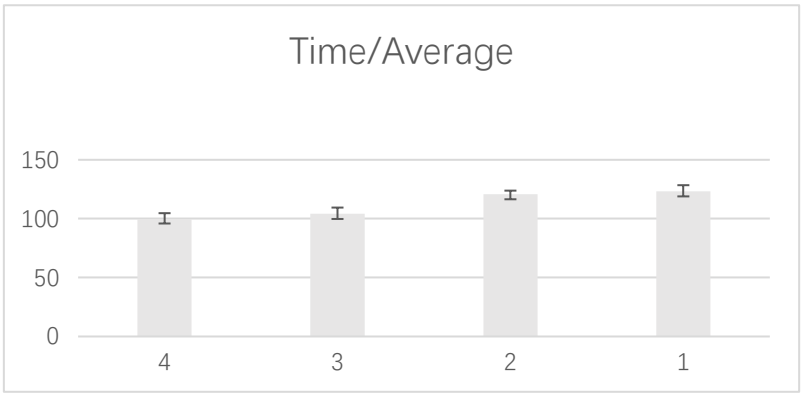
In conclusion, our program can helper user exchange their shares and match the buyer and seller ‘s request.

Testing strategy:

For testing the scalability of our program, we came up with 2 strategies. Firstly, we can test the only one program that running. That means only one client send the request and the server will deal with the only client’s request. Secondly, we can test multiple client running parallel and the server will deal with multiple client’s request and the Data will be processed by the program and the database.

Single Client Testing:

For our scalability test, we tested the create account process and flood our server with 10000 requests. We have put in 10000 create request to test our program.
The data is shown below:



From the difference shown above, we found that running on more cores does reduce the total time consumed, but the reduced amount is not as significant as we predicted. After reviewing our code and test, we still thought that our programme should have run faster on more cores, so we guessed it might be the socket that constrainted our programme's performance but unfortunately we didn't have time to prove that.
We also tested different Core’s effiency.

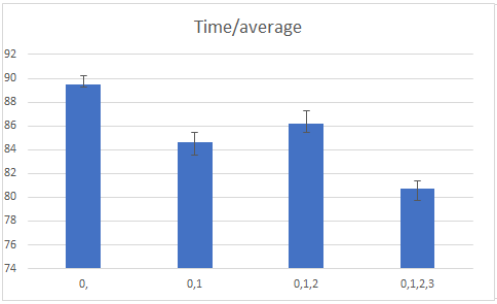
core Number	Time/average	#	Time/average
0,1,2,3	99.7096	4	99.7096
0,1,2	104.245	3	104.245
0,1	120.815	2	120.815
0	123.251	1	123.251

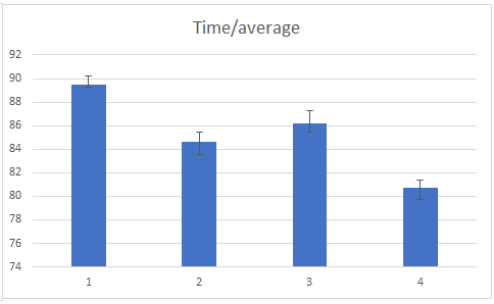
2	108.976	1	108.976
3	98.9149	1	98.9149
1	102.47	1	102.47
1,2,3	107.767	3	107.767

In this table, we can conclude that the node will influence the time. Because 2 cores and 3 cores have larger difference than 3 and 4. Which may result from the node. As the data transform between Nodes will cost more time.

Testing Multiple Client:

In Multiple Client, we test 29600 query and use it to test our program. And then we tested our program running time and the throughput of the program. The data is as the following:





#	Time/average	Throughput(request/second)
1	89.5106	330.68
2	84.6241	349.78
3	86.1972	343.4
4	80.7209	366.69

From the data figure, we can conclude that the trend is descending but it has some buldge. We thought that the node may effect the test result. As the data transform between Nodes will cost more time.