Project 4 Part 1 Report

1.Team Members

Name: Zixun Wang UFID: 3725-9823

Name: Yixin Wei UFID: 5114-6181

2.Implementation

We implement the engine actor and a specified number of user actors in separate processes and use remote actors. We use one engine process and one user process to simulate the twitter.

1. We implement a Twitter like engine with the following functionality:

a. Register account:

When creating an account, we create a user actor with a unique username and a password.

b. Subscribe to user's tweets:

After registration, we assign some subscribers to existing users. The number of subscribers for each user is a random number which is less than 1/3 of the number of users.

c. Send tweet:

Tweets have hashtags (e.g. #COP5615isgreat) and mentions (e.g. @user1). And each user can send tweets to their followers.

d. Re-tweets:

Users can re-tweet tweets which they have received or sent before. And each user can re-tweet to their followers.

e. Querying tweets subscribed to:

Stored tweets in a hashmap which has username as key and the content of tweets they sent as value. We can query tweets in that hash map by the specified username.

f. Querying tweets with specific hashtags (e.g. #COP5615isgreat):

Stored tweets in a hashmap which has hashtag as key and the content of tweets they sent as value. We can query tweets in that hash map by the specified hashtags.

g. Querying tweets in which the user is mentioned (e.g. @user1):

Stored tweets in a hashmap which has mentioned username as key and the content of tweets they sent as value. We can query tweets in that hash map by the specified mentioned username.

h. If the user is connected, deliver the above types of tweets live (without querying):

One user has two status, active and inactive (connected and disconnected). One user can log in and log out anytime when the program is running. If the user logs out, we set a hashmap to store tweets from its followings. And once the user logs in again, it can receive all these tweets during its logging out.

2. We implement a tester/simulator to test the above:

a. Simulate as many users as you can:

We can input the number of users to simulate the twitter program. As the number of users increases, we need more time to initialize the information (mainly refer to the number of subscribers) randomly. And we select 4000 as the biggest number of users in order to reduce running time of program.

At first, we register 4000 users with the username “user0” to “user3999” and with the password “123”. Then we assign a random amount of subscribers to these users which is less than 1/3 of the amount of the users.

b. Simulate periods of live connection and disconnection for users:

Firstly, we choose 1/10 of users to be connected at the beginning of the program. Then we run all types of commands randomly, including send tweets, re-tweets, query by username subscribed to, query by hashtags, query by mentioned username, log in and log out. The probability of sending tweets command is twice that of other commands. And when the command of logging in and logging out be selected, the users can be connected and disconnected, and the periods of live connection and disconnection can be simulated.

c. Simulate a Zipf distribution on the number of subscribers:

For users with a lot of subscribers, we increase the number of tweets and re-tweets to simulate a Zipf distribution. We increase the probability of sending tweets of a user if it has more subscribers. And the probability of sending tweets is in direct proportion to the number of subscribers of one user.

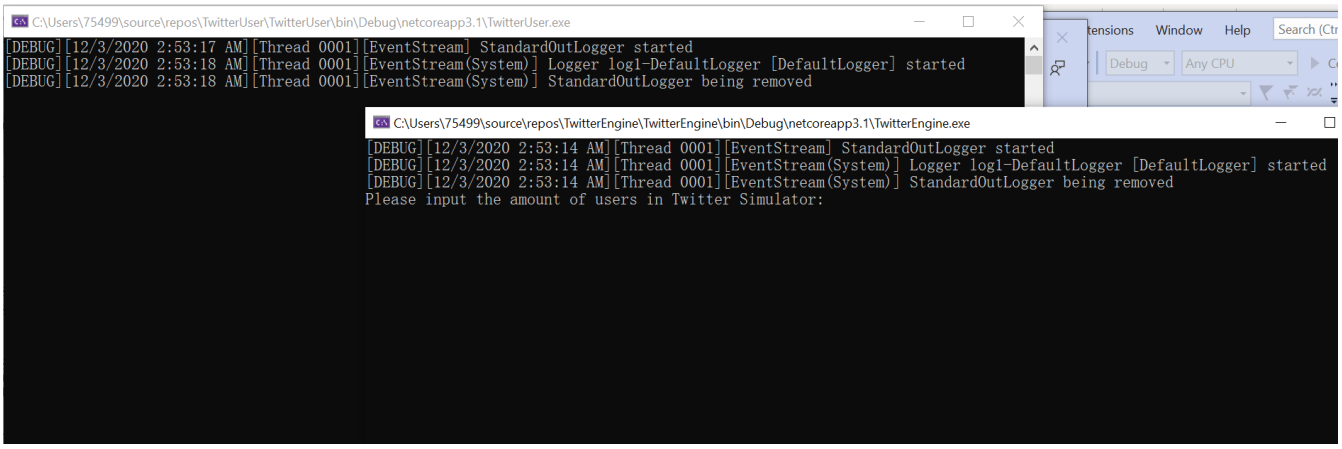
We let the user0 has the highest number of subscribers and the client user1 has the second highest number of subscribers and so on.

3.Results and Observations

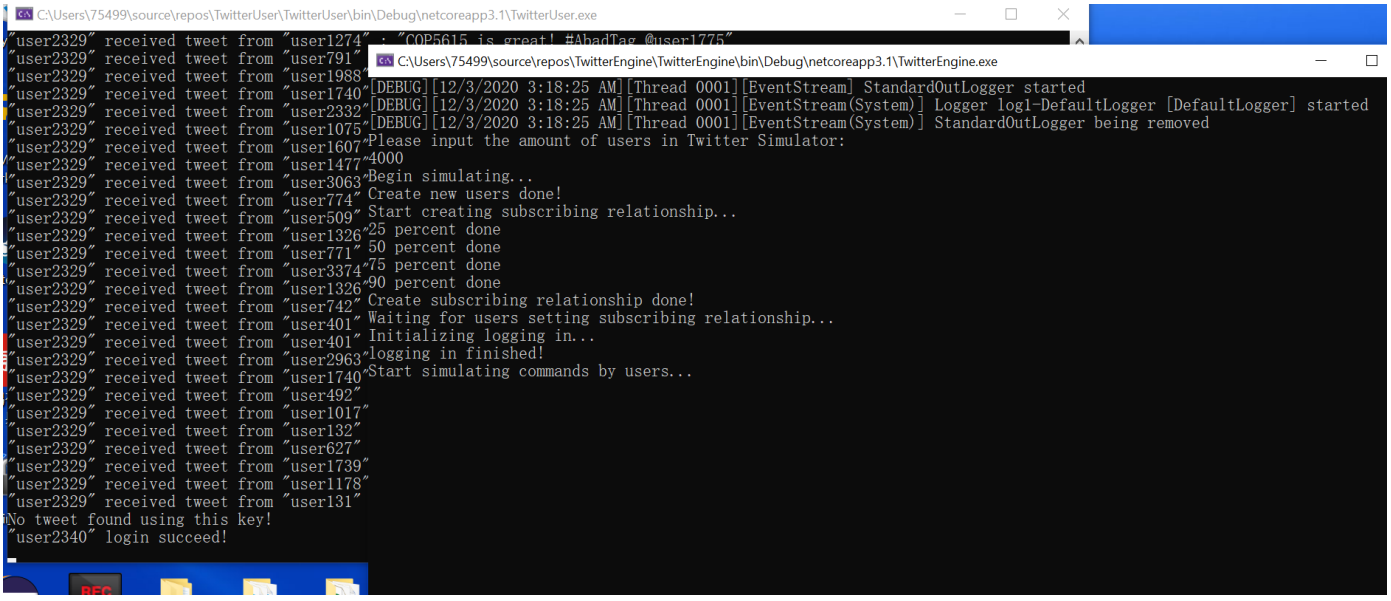
In this project, we simulate 4000 users in our twitter engine. Once the twitter engine has initialized and begin tweeting, we let the program run for 3 minutes. Then we record the total amount of tweets when the program is stopped. Here is the result of our simulation:

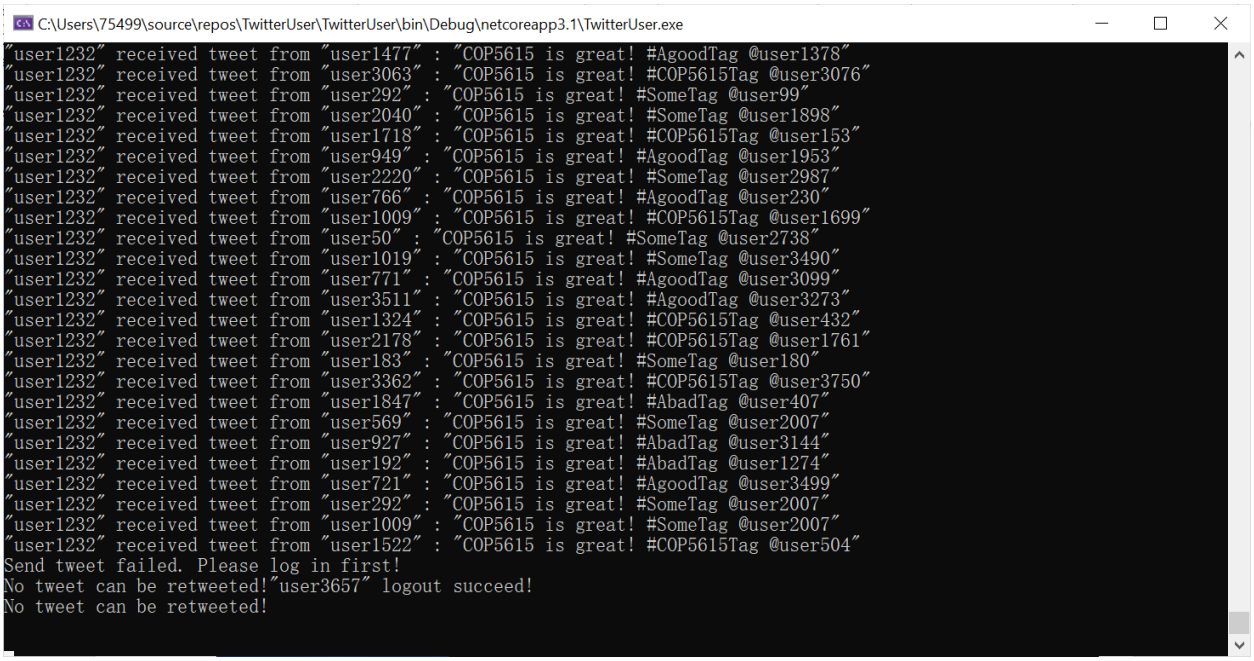
Firstly, we open our two processes to simulate the engine and client part.

To run this program, first run twitterEngine, then run twitterUser. Input the amount of users of Twitter Simulater in twitterEngine, and the program will run.



We display the content of tweets in the process of TwitterUser. And we display the output of amount of tweets in the process of TwitterEngine.





Here is a chart about the number of users and tweets:

Users	Max Subscribers	Total Tweets	Total Commands	Send tweet	Re-tweets	Query(tag)	Query(mentioned)	Query(subscribers)	Log in	Log out
500	166	169091	11571	2922	1419	1433	1441	1456	1467	1433
1000	333	272837	11566	2923	1472	1427	1445	1401	1434	1464
2000	666	440384	10922	2721	1319	1377	1375	1438	1416	1276
3000	1000	463708	10627	2570	1366	1268	1364	1335	1366	1358
4000	1333	430329	8938	2237	1072	1147	1107	1117	1124	1134

We can see that as the number of users increases, the total amount of tweets increases, and the total commands decreases a bit, because the program takes longer time to run each loop.

And when the amount of users is 4000, the number total tweets is lower than that of users of 3000. Because it takes longer time to run and some tweets are not processed well in the program.