COP5615: Distributed Operating Systems
Project -4.1

Submitted by:
Aarthi Kashikar, UFID: 0366-8968
Sri Kanth Juluru, UFID: 9279-1918

Specifications of the machine where the experiments were performed on are as follows.
Model: Dell Inspiron 5593
OS: Windows 10 Home
Processor: Intel(R) Core(TM) i7-1065G7 CPU @ 1.30GHz
Memory: 16 GB RAM, 2667 MHz

Contents of zip folder:

1. Main project folder (project4_1)
2. Report.pdf

We can run the project by going inside the project2 folder, and run this command
dotnet fsi --langversion:preview Simulator.fsx numUsers

Implementation details:

The implementation follows as below:

1) Number of users are taken as input parameter and those many clients will be spawned and stored.
2) Each user needs to be registered with the engine before performing any operations.
3) Client will make a register request to the twitter engine, which in turn registers the user.
4) Then all the users gets registered by calling the Register case of Twitter engine.
5) The client makes a call to the engine where the user is registered with the name provided.
6) After a user is registered, they can perform operations like login, tweet, query for hashtags, or his mentions and logout by calling engine.
7) Based on the request from the client, respective cases in twitter engine will be triggered and corresponding action will be performed.
8) To test these operations, a simulator is present which sends the request from the client to twitter engine and also gets back acknowledgements in some scenarios.
9) Here all the given number of users are registered and randomly users are picked to login, logout, tweet etc.,

Operations Implemented:

Below is the description of implementation of all the operations.

Register:

One the request to register is received by the engine, the unique username is added to a list called usersList. This acts as the bank of users. Then, entries into two maps are created. One is followers where it maintains a list of followers. Another is the timeline where the activities in the timeline are stored.

Login:

Here the given user is logged in. In the engine a list of activeUsers is present where this username is added as he is logged in. Once the user is logged in he can follow, tweet, retweet tweets on his timeline.

Tweet:

Here the user sends the tweet in the form of a string of words with any hashtags if he wants and users he likes to mention in the tweet. This tweet is given a unique tweetId and is added to his timeline. The list of followers is taken and the tweetId is added to their timeline as well. From the tweet, the hashtags are picked and added to hashtag map with hashtagged word as the key and the list of tweetIds as the value. Similarly mentions are picked and added to mention map where mentioned username is the key and the list of tweetIds as value.

In the simulator tweet is generated randomly from a library of words from words.txt and hashtags are added from Hashtag.txt and random users are picked and mentioned.

Follow:

Here the engine receives two parameters, one is the user and the second is the user to be followed. The user(first parameter) is added as a value to the map with the key user to be followed(second parameter).

Retweet:

The user can pick any tweet from his timeline and send a request to retweet. Then the engine follows the same procedure as tweet.

Logout:

The user is removed from activeUser list since he is logged out. Now he cannot perform some functions like tweeting, retweeting or following. He shall be prompted to login first

Simulator:

We have added a file called Simulator.fsx file that simulates scenarios of user registration, login, tweet etc., The simulator is provided with number of users that we want to simulate. The file first spawns and actor TwitterEngine. It then spawns client actors as that of number of users provided. Then the clients are then made to register at engine and hence they all send register request to TwitterEngine. Once all the users are registered, all the other operations like following, tweeting, and querying are performed. We randomly made users login and follow other users. For this users will send a login request to twitter engine which in turns perform the requested action and updates the necessary timelines. If a user is logged in, engine will keep track of active users and updates it whenever a user logs out. Similarly user will randomly follow other users. Number of followers/subscribers for a user are generated by performing a zipf distribution on number of users. To achieve popularity, users with more subscribers will be making more number of tweets. Each tweet is generated randomly with words collected from words file and also having some hashtags and mentions taken from respective files. By doing this, each tweet will be more unique with different mentions and hashtags.

After users follows other users, we will start making tweets. As said earlier, users with more popularity will be making more tweets. We are also making some retweets here. So whenever a retweet happens, that tweet will be added to their followers timeline and if a follower is active/logged in, he can see that tweet on his screen. We will directing the outputs to a output.txt file for better logging. Whenever a user logs in or follows, or for any activity by engine, it will be logged into the output.txt file. Users will start loggin off randomly during this process and active user list will be updated accordingly. When all user logs out, program will be terminated and time taken to process the tweets and user count will be printed in the console.

Program took 1265.058200 milliseconds to make 1610 tweets by 50 users
Program took 5238.247300 milliseconds to make 2950 tweets by 100 users
Program took 46.720100 milliseconds to register 100 users
Program took 21681.541200 milliseconds to make 5640 tweets by 200 users
Program took 249.922000 milliseconds to perform operations by 200 users
Program took 115.899000 milliseconds to register 500 users
Program took 125601.722100 milliseconds to make 14020 tweets by 500 users
Program took 117667.977600 milliseconds to perform operations by 500 users

Findings:

Time taken to perform all the operations from registering the user to program termination:

Num of users	Number of tweets	Total time taken
50	1570	1320 ms
100	3270	3137 ms
200	5960	5876 ms
500	14570	117743 ms
1000	30730	75609 ms
2000	62300	234728 ms

Time taken to perform login, follow, tweet, query operations:

Num of users	Number of tweets	Time taken for ops
50	1570	1310 ms
100	3270	3115 ms
200	5960	5861 ms
500	14570	117667 ms
1000	30730	75609 ms
2000	62300	234649 ms

Maximum number of users on which simulation was tested was 1000 and maximum number of tweets made were 30,730.