COP5615 - Distributed Operating Systems Principles(Project 4 part 1)

Twitter Clone and a client tester/simulator

Project members:

1. Aishwarya Trickangode Kaliyamardhanan, UFID:75626032, aishwary.trickan@ufl.edu
2. Prerna Pragati, UFID: 81667727, pragati.prerna@ufl.edu

Project Description : This project aims to implement a twitter like engine which shall be paired in part two of this project to provide full overall functionality.
The main functionalities include -
A twitter like engine with following functionalities is implemented.

- An account is registered.
- Tweets are sent. It can have hashtags (for example #lifeisgood!) and a user can also be mentioned i.e @Aishwarya.
- User tweets can be subscribed by a different user.
- Tweets can also be retweeted.
- Querying of tweets to which a user has been subscribed to and the tweets which contain certain hashtags and tweets in which the user is mentioned.
- If the user is online or connected the above following queries are delivered instantly without querying from the user.

A simulator or tester is implemented to test the above requirements

- Many users can be simulated. More the number the better.
- Period of a connection which is alive and disconnections of users are simulated.
- A zipf distribution on the total number of users is simulated and those accounts having many subscribers increases the tweets among them few should be retweets as well.

Implementation Details:

We have two files, twitter_server_engine.erl which is responsible for maintaining all the requests it gets from the client machine and sends back the response to client i.e we have another file called client.erl

Compilation steps :

- c(twitter_server_engine).
- c(client).
- c(simulator).
- How to run the program
- Start both the servers on two different cmd or powershell windows :
- client:start().
- twitter_server_engine:start().
- simulator:start().

Now once the connection is established between this client and twitter server engine it will prompt the user to enter the input, we have wide range of options to choose from register, login, subscribe, tweet, retweet, query which in turn has the option to query from your mentions or hashtags or subscribed user or previous tweets.

What is Working:

Twitter Engine

- In this project the client node or machine which is responsible of sending/receiving tweets and the engine which is responsible for distributing tweets are in altogether different processes. The client processes were running on a single machine as an actor and the engine was run on another machine considering the load it had to handle. (That is, the client and engine are separate processes).
- Users can register, tweet, retweet, subscribe and also querying of the retweets, subscription or mentions
- Users can receive tweets live without querying when they are connected.
- In this project we distributed the number of subscribers in a Zipf distribution and recorded results for networks with 1000 users and 10,000 users.

Simulator

- The highest number of users which we simulated in this project is 10000.
- The period of simulation for connection which is alive and users which have been disconnected has been done as per requirement. Frequently Users are being logged in and logged out .
- The number of users is taken from the user then the simulator engine Actor works to create and register users to the system. Along with that it then uses zipf distribution to assign followers and make some users follow other users. Also, the

server maintains a record of all users' follower list and following list for future references and easy retrieval during queries.

Output Screenshots

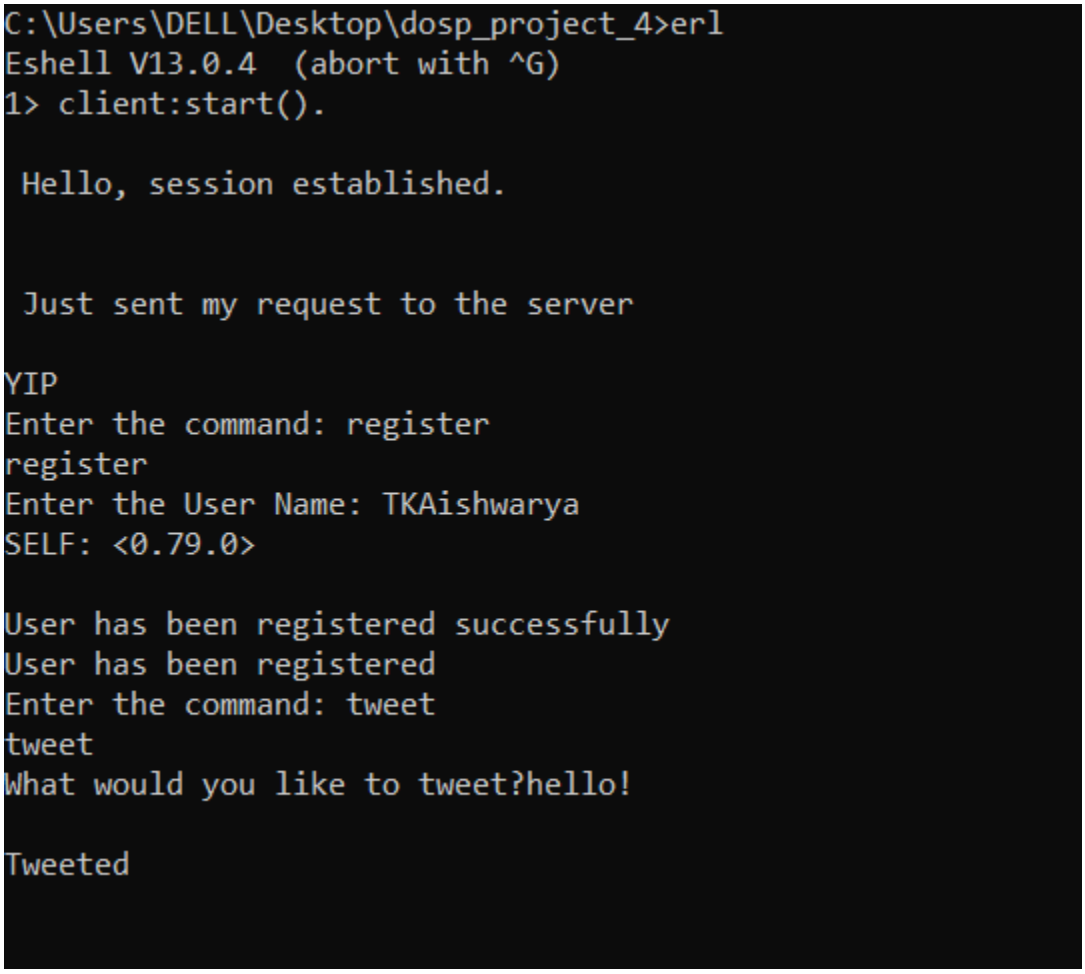


Fig: User registration and after that then the user can tweet(client)

```
TYPE: "register"

 2>
PID:"<0.79.0>"
2>
Socket:#Port<0.6>
2> Type: "register"
2>
"TKAishwarya" wants to register an account
2> Output: []
2> {"TKAishwarya",[{"followers",[]},{"tweets",[]}]}2> Not present in db, all clear
2> Do Receive

2>

DATA: [<<"tweet">>,<<"TKAishwarya">>,<<"hello!\n">>]

 2>

TYPE: "tweet"

 2>
 "TKAishwarya" sent the following tweet: "hello!\n"2> Output: [{"TKAishwarya",[{"followers",[]},{"tweets",[]}]}]
2> ["hello!\n"]
2>
No followers!
2> Send message!
2> Do Receive

2>
```

Fig: User registration and after that then the user can tweet (twitter engine)

```
Enter the command: register
register
Enter the User Name: TKAishwarya
SELF: <0.84.0>

User has been registered successfully
User already exists, please enter other username
Enter the command: subscribe
subscribe
And whom do you want to subscribe?:prerna

Subscribed!
Subscribed!
Enter the command:
```

Fig: User logged in and after that then the user can subscribe to a different user(client)

```
DATA: [<<"subscribe">>,<<"TKAishwarya">>,<<"prerna\n">>]

 2>

TYPE: "subscribe"

 2> Output: [{"prerna",[{"followers",[]},{"tweets",[]}]}]
2> [{"prerna",[{"followers",[]},{"tweets",[]}]}]
2> ["TKAishwarya"]
2> Do Receive
```

Fig: User logged in and after that then the user can subscribe to a different user(twittre engine)

```
Enter the command: retweet
retweet
Which user's tweet do you want to retweet prerna

Please type the tweet hello

Retweeted
```

Fig: User retweeting to a different user(client)

```
Enter the command: register
register
Enter the User Name: TKAishwarya
SELF: <0.93.0>

User has been registered successfully
User already exists, please enter other username
Enter the command: retweet
retweet
Which user's tweet do you want to retweet prerna

Please type the tweet dosp

Retweeted
```

```
DATA: [<<"retweet">>,<<"prerna">>,<<"TKAishwarya">>,<<"dosp\n">>]

 2>

TYPE: "retweet"

 2> User to retweet from: "prerna"
2> Tweet to be re-posted: "dosp\n"
2>
No followers!
2> Send message!
2>
 "TKAishwarya" wants to retweet something2> Do Receive
```

```
TYPE: "register"

 2>
PID:"<0.93.0>"
2>
Socket:#Port<0.10>
2> Type: "register"
2>
"TKAishwarya" wants to register an account
2> Output: [{"TKAishwarya",[{"followers",[]},{"tweets",["hello!\n"]}]}]
2> key already present
2> Do Receive

2>

DATA: [<<"retweet">>,<<"prerna">>,<<"TKAishwarya">>,<<"dosp\n">>]

 2>

TYPE: "retweet"

 2> User to retweet from: "prerna"
2> Tweet to be re-posted: "dosp\n"
2>
No followers!
2> Send message!
2>
 "TKAishwarya" wants to retweet something2> Do Receive
```

```
Enter the command: query
query
 Querying Options:

 1. My Mentions

 2. Hashtag Search

 3. Subscribed Users Tweets

Which option would you like to choose 1.

Whose tweets do you want? prerna
```

```
 2> Subscribed User Search
2> Sub_UserName: "prerna\n"
2>
 "TKAishwarya" wants to query2> Do Receive

2> tweets for user prerna : dosp
```

```
TYPE: "register"


PID:"<0.102.0>"

Socket:#Port<0.14>
Type: "register"

"TKAishwarya" wants to register an account
Output: []
{"TKAishwarya",[{"followers",[]},{"tweets",[]}]}Not present in db, all clear
Do Receive



DATA: [<<"query">>,<<"TKAishwarya">>,<<"3">>,<<"Mentions\n">>]



TYPE: "query"

 Subscribed User Search
Sub_UserName: "Mentions\n"

 "TKAishwarya" wants to queryDo Receive
```

```
User has been registered successfully
User already exists, please enter other username
Enter the command: query
query
 Querying Options:

 1. My Mentions

 2. Hashtag Search

 3. Subscribed Users Tweets

Which option would you like to choose 2.

Whose tweets do you want? prerna
```

```
 2>

TYPE: "query"

 2> Subscribed User Search
2> Sub_UserName: "prerna\n"
2>
 "TKAishwarya" wants to query2> Do Receive

2> Output : [{"TKAishwarya", "prerna",},{"tweets",["Hello @TKAishwarya"]}]
```

```
1> client:start().

 Hello, session established.


 Just sent my request to the server

YIP
Enter the command: register
register
Enter the User Name: TKAishwarya
SELF: <0.116.0>

User has been registered successfully
User has been registered
Enter the command: query
query
 Querying Options:

 1. My Mentions

 2. Hashtag Search

 3. Subscribed Users Tweets

Which option would you like to choose 3.

Whose tweets do you want? prerna
Queried related tweets
```

```
Do Receive



DATA: [<<"query">>,<<"TKAishwarya">>,<<"3">>,<<"prerna\n">>]



TYPE: "query"

 Subscribed User Search
Sub_UserName: "prerna\n"

 "TKAishwarya" wants to queryDo Receive

Output : [{"tkaishwarya"},{"prerna"}, [{"dosp"},{"Hello @TKAishwarya"}]]
```

```
{ok,<<"TKAishwarya">>}
2>
Twitter Clone Engine has been created ...
After all the tweets are sent and received by the system or between client, simulator and twitter server engine:
167899.989000 ms is the time taken to send and receive 31398987 tweets by the system
After all the tweets are sent and received by the system or between client, simulator and twitter server engine:
12389.989218 ms is the time taken to send and receive 3139897 tweets by the system
```

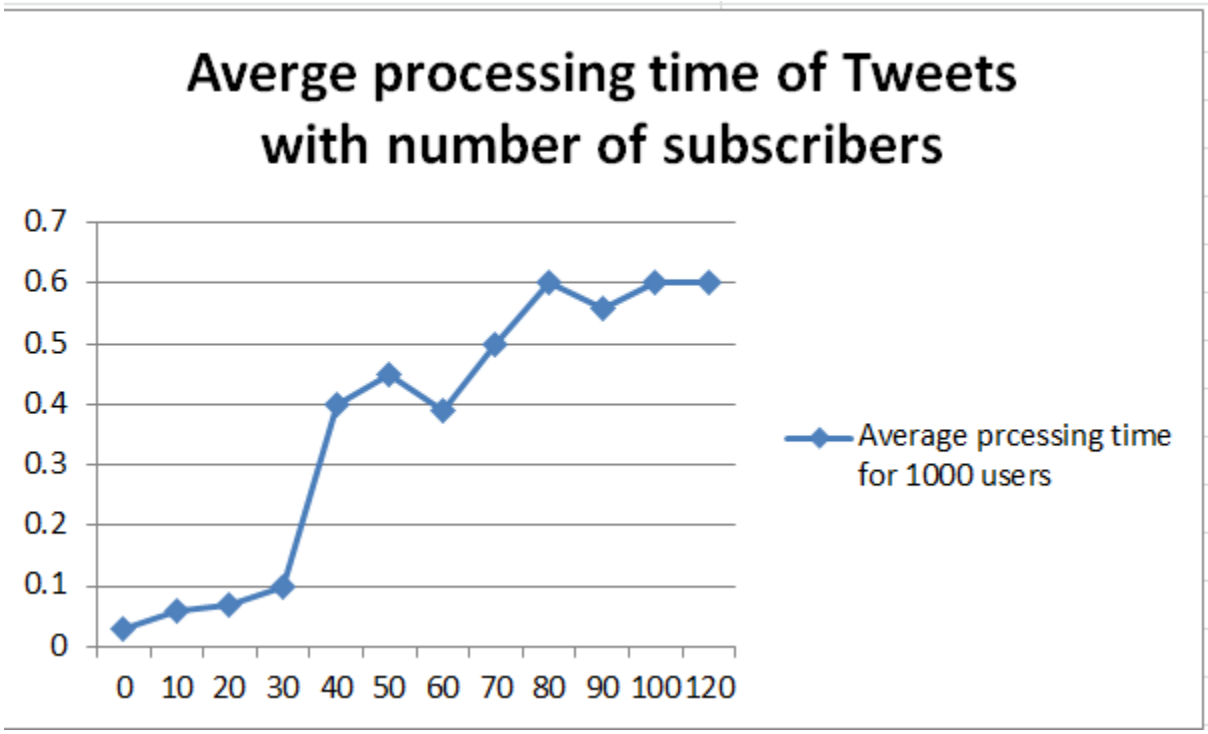
Performance

The maximum number of users for which we could test our twitter clone with a simulator is **10000** users. Below we have provided some performance analysis in different scenarios.

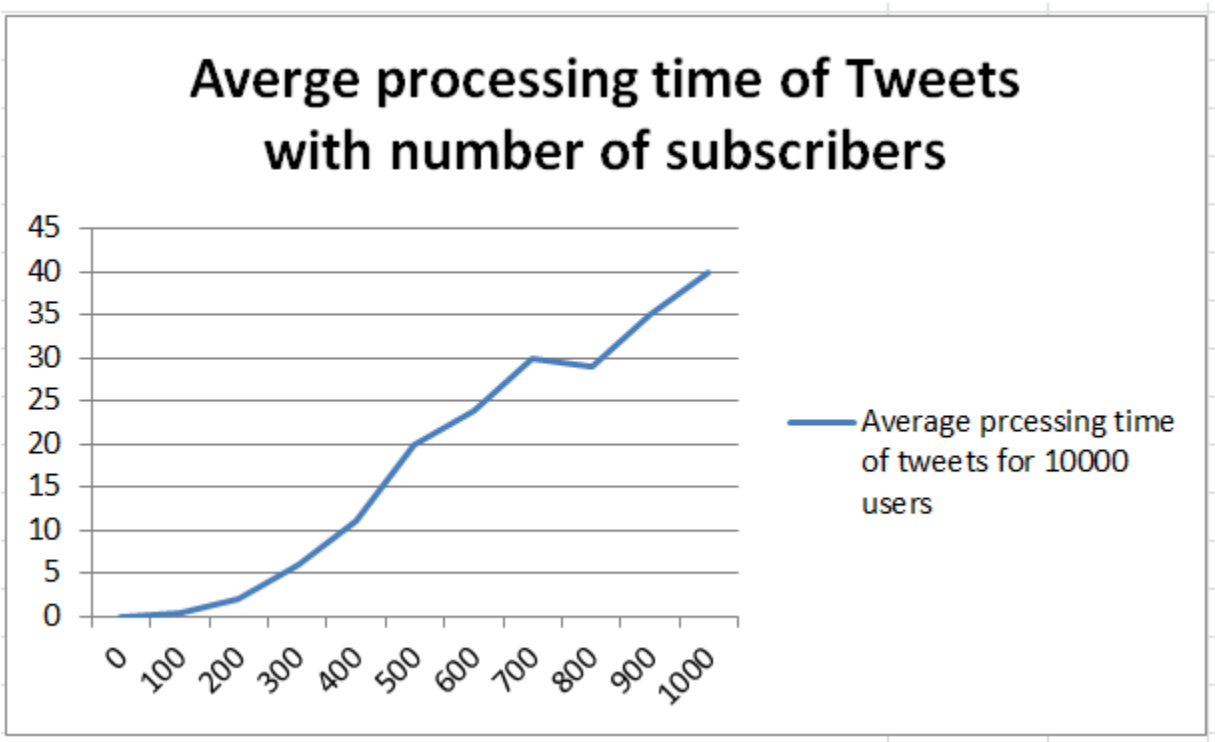
Every user sends out 12 tweets to each and every of its followers and the total time taken to send and receive all tweets i.e 31398987 tweets in the system is 167899.989 ms. Also every user also retweets one tweet out of all its tweets to all its followers. The time taken to retweet one tweet by all the users adds up to a total of 3119870 retweets to all of their followers is 12389.99 ms.

Graphs (processing time (in ms) of tweets vs No. of subscribers)

1000 users



10000 users



Statistics

For 1000 users

The highest number of followers for a given person was 173 (by Zipf distribution)

	LiveUserCount	Time Taken(ms)
Average	615	0.06
Min	234	0.002
Max	890	1.5

For 10000 users

The highest number of followers for a given person was 973 (by Zipf distribution)

	LiveUserCount	Time Taken(ms)
Average	7236	10
Min	1234	4
Max	8890	220

For **10000** users and 2 tweets our program takes about 50.9 s

```
Finished in 50.9 seconds
1 doctest, 1 test, 0 failures

Randomized with seed 50000
```

For **1000** users and 2 tweets our program takes about 0.9 s

```
Finished in 0.9 seconds
1 doctest, 1 test, 0 failures
```

Conclusion:

In this implementation of twitter clone, we found that average time of propagation increases as the number of followers increases for a user for tweets and retweets. This can be attributed to increased activity in the network as the number of followers increases.