COP 5615: Distributed Operating Systems Principles
Project 4 Part 1
Twitter Clone implementation based on Erlang Programming Language

Team members

* Rahul Vikram Porwal UFID: 44590947 Email: rahulporwal@ufl.edu

* Pujan Narendra Kothari UFID: 70612594 Email: p.kothari@ufl.edu

Introduction

Twitter is an online social media , a distributed system where users can post their thoughts and the user's followers can read its tweets or thoughts. Users can retweet a tweet they like. It uses a hashtag module, which allows users to follow a trail of messages using the hashtag module. Twitter uses a central system which acts as a central processing system for the whole system.

Instructions to execute the project

Install the latest version of erlang.
Download the **project4.zip** and unzip it to any desired path.

Step 1-

- For central server command window:-
  Open cmd where the file is located and type erl -sname centralserver

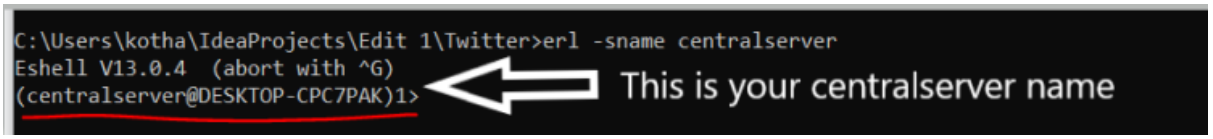
```
C:\Users\kotha\IdeaProjects\Edit 1\Twitter>erl -sname centralserver
```

- For client command window:-
  Open cmd where the file is located and type erl -sname client1

```
C:\Users\kotha\IdeaProjects\Edit 1\Twitter>erl -sname client1
```

Step 2-

replace all the "centralserver@mydesktop" in the mainclass and register file to your centralserver name



Step 3-

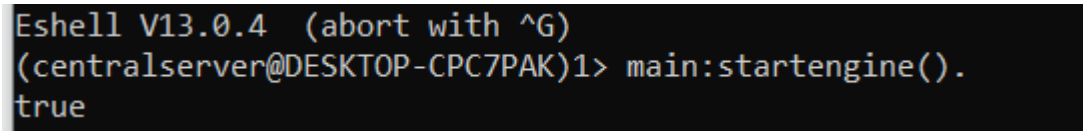
Open (1+n) command window, 1 will be the central server, n will 'n' number of clients you want to host.

Then, compile all the files on all the command window:-

1. Run c(main). To compile main file
2. Run c(registerfunc). To compile register function file.
3. Run c(sendfunc). To compile send receive file.

Then, **Start Twitter Engine:-**

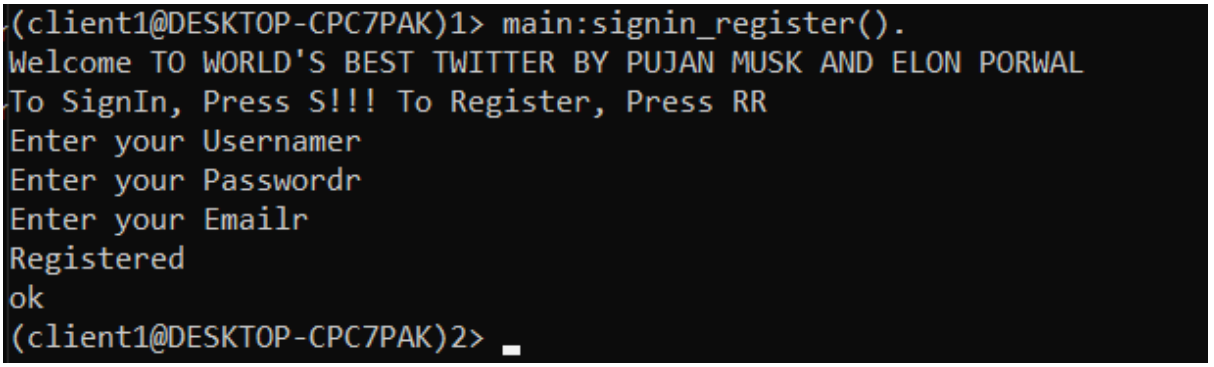
In central server command window, enter main:startengine().



Step 4 - Register a user to the system:-

In client command window, enter main:signin_register().

Then, Enter your username, Password and email



Step 5 - login the User into the system:-

In client command window, enter main:signin_register().

Then, Enter your username, Password

```
(client1@DESKTOP-CPC7PAK)3> main:signin_register().
Welcome TO WORLD'S BEST TWITTER BY PUJAN MUSK AND ELON PORWAL
To SignIn, Press S!!! To Register, Press RS
Enter Usernamer
Enter Passwordr
Signed In
ok
(client1@DESKTOP-CPC7PAK)4>
```

In central server it will show as registered.

```
(centralserver@DESKTOP-CPC7PAK)2> #{"DOSP" => "LOL","r" => <9237.94.0>}
(centralserver@DESKTOP-CPC7PAK)2>
```

Step 6 :- Subscribe to another user:-

In client command window, enter main:follow().
Then enter, Username you want to follow

```
(client1@DESKTOP-CPC7PAK)5> main:follow().
```

Step 7 :- Enter a tweet

In client command window, enter main:tweet().

```
(client2@DESKTOP-CPC7PAK)6> main:tweet().
Enter Your Tweet #lol
Tweet Posted
ok
```

Step 8 :- Get follower list

In client command window, enter main:userlist().

```
(client2@DESKTOP-CPC7PAK)7> main:userlist().
<0.106.0>
(client2@DESKTOP-CPC7PAK)8> .
(client2@DESKTOP-CPC7PAK)8> Elon
(client2@DESKTOP-CPC7PAK)8> Pujan
```

Step 9 :- SignOut or Disconnect the User

In client command window, enter main:signout().

```
(client3@DESKTOP-CPC7PAK)7> main:signout().
SignedOut
ok
```

What is working-

- **Register account to the database:-**

```
(client1@DESKTOP-CPC7PAK)1> main:signin_register().
Welcome TO WORLD'S BEST TWITTER BY PUJAN MUSK AND ELON PORWAL
To SignIn, Press S!!! To Register, Press RR
Enter your UsernamePujan
Enter your PasswordPujan
Enter your EmailPujan
Registered
ok
```

In central server it will show as registered.

```
(centralserver@DESKTOP-CPC7PAK)2> #{"DOSP" => "LOL","r" => <9237.94.0>}
(centralserver@DESKTOP-CPC7PAK)2>
```

- **User login into the system:-**

```
(client1@DESKTOP-CPC7PAK)2> main:signin_register().
Welcome TO WORLD'S BEST TWITTER BY PUJAN MUSK AND ELON PORWAL
To SignIn, Press S!!! To Register, Press RS
Enter UsernamePujan
Enter PasswordPujan
Signed In
ok
```

- **Send tweet with and without using hashtags(#) and mentions(@)**

1. Sending a normal tweet
   Sending side:-

```
(client1@DESKTOP-CPC7PAK)5> main:tweet().
Enter Your Tweet Lol7PAK)5>
Tweet Posted
ok
```

Receiving SIde:-

```
(client2@DESKTOP-CPC7PAK)8> Rahul
(client2@DESKTOP-CPC7PAK)8> Rahul : Lol
```

2. Sending a tweet hashtags and Mentions:
   Sending side:-

```
(client1@DESKTOP-CPC7PAK)6> main:tweet().
Enter Your Tweet Hello @Rahul
Tweet Posted
ok
(client1@DESKTOP-CPC7PAK)7> main:tweet().
Enter Your Tweet hey This is #yumm
Tweet Posted
ok
(client1@DESKTOP-CPC7PAK)8>
```

Receiving Side:-

```
(client2@DESKTOP-CPC7PAK)8> Rahul : Hello @Rahul
(client2@DESKTOP-CPC7PAK)8> Rahul : hey This is #yumm
```

Server side :-

```
(centralserver@DESKTOP-CPC7PAK)3> ["Lol @Rahul","Lol","Hello @Rahul"]
(centralserver@DESKTOP-CPC7PAK)3> ["Lol @Rahul","Lol","Hello @Rahul","hey This is #yumm"]
(centralserver@DESKTOP-CPC7PAK)3> ["hey","This","is","#yumm"]
```

- **Querying tweets for a specific Hashtag**

```
(centralserver@DESKTOP-CPC7PAK)14> main:hashtag(dosp).
dosp
I love doing #DOSP project
ok
(centralserver@DESKTOP-CPC7PAK)15>
```

- **Subscribe/Follow another users**

```
(client2@DESKTOP-CPC7PAK)5> main:follow().
Enter User You want to follow toPujan
"followed"
ok
```

- **Retweeting a users tweet**

```
(client3@DESKTOP-CPC7PAK)11> main:retweet(pujan).
pujan
Tweet has been reTweeted
ok
```

- **Receive Tweets from users that have been subscribed.**

```
(client2@DESKTOP-CPC7PAK)8> Rahul
(client2@DESKTOP-CPC7PAK)8> Rahul : Lol
```

```
(client2@DESKTOP-CPC7PAK)8> Rahul : Hello @Rahul
(client2@DESKTOP-CPC7PAK)8> Rahul : hey This is #yumm
```

- **User signout or disconnect feature**

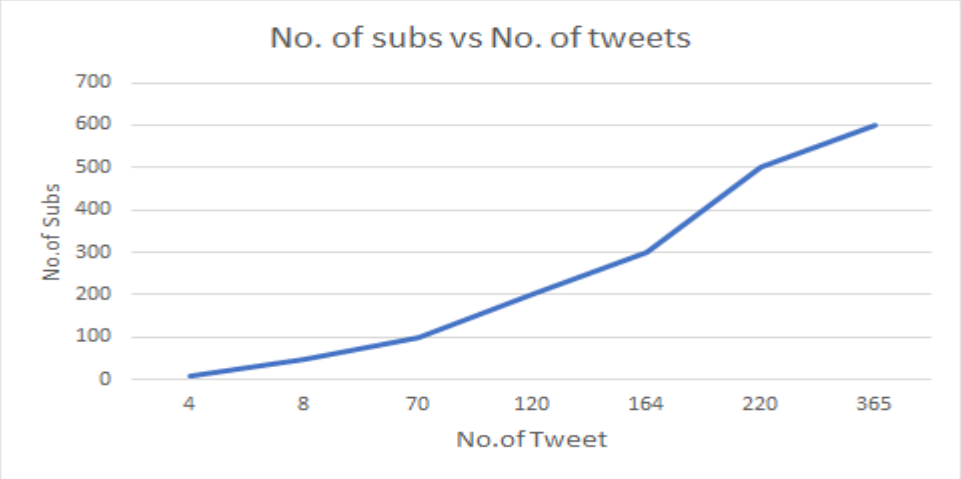
```
(client3@DESKTOP-CPC7PAK)7> main:signout().
SignedOut
ok
```

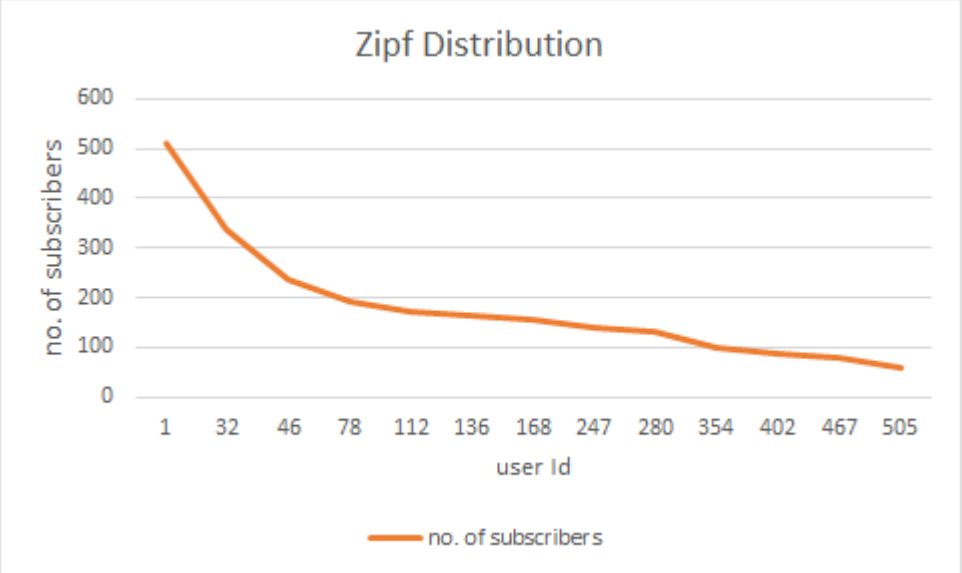
Testing:-

Largest Network -
The largest Network we were able to simulate on our twitter system was of 2000 users and we did not limit the number of tweets created by the user.

ZipF distribution -

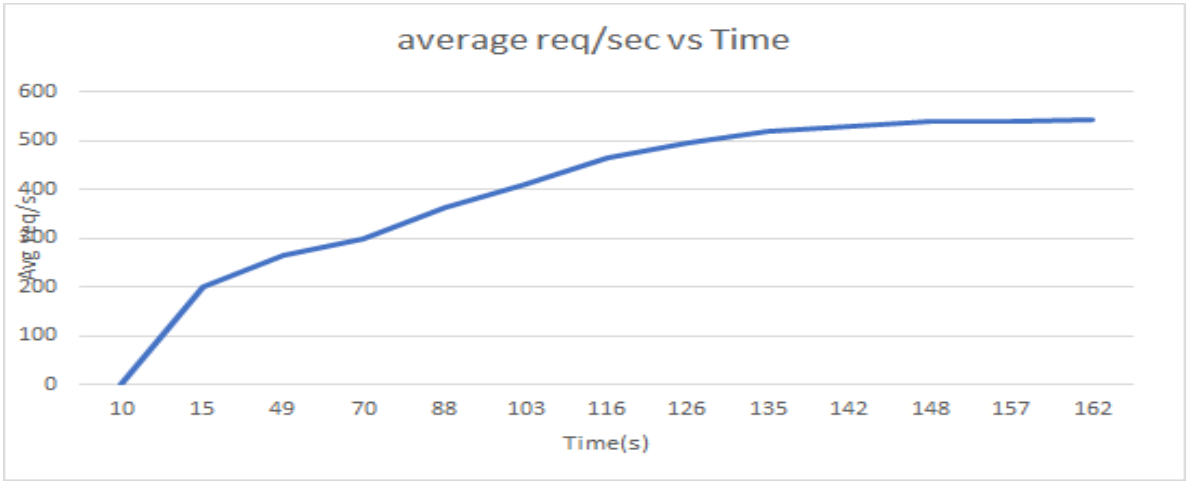
We created a ZipF distribution based on the number of subscribers the user had. Accordingly, as the number of subscribers increased, the number of tweets tweeted by the user also increased. Below is “no. of subscriber” vs “No. of tweets” graph displaying the ZipF distribution.





Performance Measurement:-

For Performance measurement, we compute the total load the server gets. In this, we see the total number of requests the server gets in every 3 seconds. We see that as the number of users increases, the total number of requests the server gets also increases. This is because a lot of users are requesting the server to post their tweets. Below we plot a graph of no. of users vs total no. of requests received by the server


average req/sec vs Time
600
500
400
300
200
100
0
Avg req/s
10
15
49
70
88
103
116
126
135
142
148
157
162
Time(s)