COP5615 - DOSP Project 4

Twitter Clone

Team Members:

1. Rashi Pandey - 88124446
2. Prerana Venugopal - 18274557

Objective

This project intends to implement a twitter server using Erlang along with multAs part of this project, an Erlang Twitter server will be developed, along with a number of clients that can each run independently. In order to allow full functionality, it is also necessary to link this Twitter engine with WebSockets. The servers distribute the tweets, which are sent and received between clients that have subscribed. Separate processes must be used to execute each of these.iple clients that can run on separate systems. There is also a requirement to pair this twitter engine with WebSockets to enable complete functionality. The clients are used to send and receive tweets to/from other subscribed clients and the server is used to distribute the tweets. Both of these have to be run in separate processes.

Project Requirements

The following features must be added to the project:

This project's objective is to:

- We need to implement a Twitter like engine with the following functionalities:-

- Registering an account

-Sending tweets where the tweets can have hashtags (e.g. #COP5615isgreat) and mentions (@bestuser)

- Subscribing to the user's tweets

- Re-tweeting (for your subscribers to get an interesting tweet you got by other means)

- Allowing querying tweets subscribed to, tweets with specific hashtags, tweets in which the user is mentioned (your mentions)

- If the user is connected, delivering the above types of tweets live (without querying)

- Implementing a tester/simulator to test the above

- Simulating as many users as possible

- Simulating periods of live connection and disconnection for users

- Simulating a Zipf distribution on the number of subscribers. For accounts with a lot of subscribers, we will be increasing the number of tweets by making some of these messages re-tweets

- Measuring various aspects of the simulator and reporting the performance

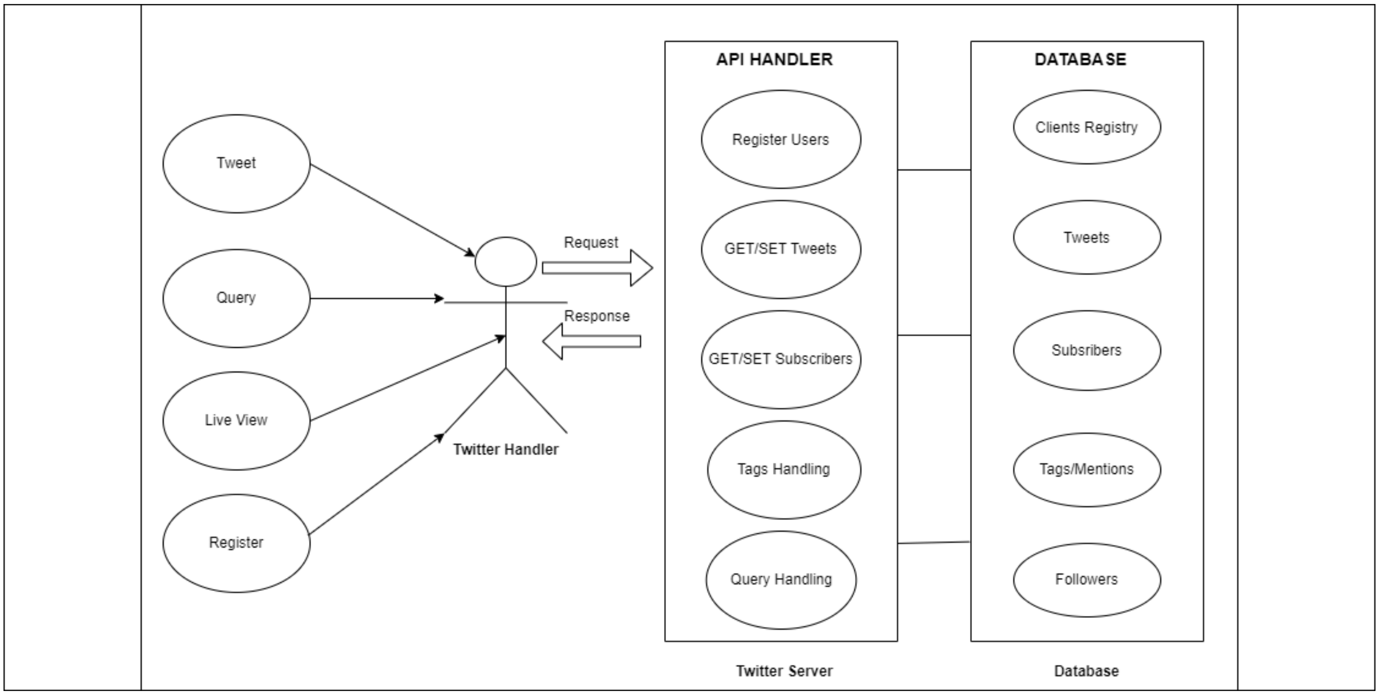
INSTRUCTIONS/COMMANDS FOR EXECUTION

- We first compile all the zip files consisting of engine, server, test and user.
- Open the erlang terminal and run the engine using the command test:delegate() followed by the IP address.

- Open a new terminal and run the client by typing the command test:delegate(No. of users we need to simulate for, Maximum no. of followers of a user). These are the two parameters that should be passed for the client simulation. The seond parameter should be one less than the maximum users.
- We have formulated the results obtained from the above 2 simulations in a table and built graphs for them.
- For manual testing, the user can refer to the screenshots attached in the SCREENSHOTS section.

ARCHITECTURE

The Twitter engine acts as an interface between the twitter handler and the database and retrieves the data on the request of the user. The engine manages multiple API calls and queries specified by the user and interacts with the database i.e. ETS tables and returns the required values. The engine supervises the registration of the user, processing of the tweets and handling of queries. Information about clients and users is kept in the database, along with a client's list of followers and subscribers and their tweets and mentions. Sending requests to the server for tasks to be completed based on features requested by the client is the responsibility of the client module. Following each request, the engine responds to the client with a response.

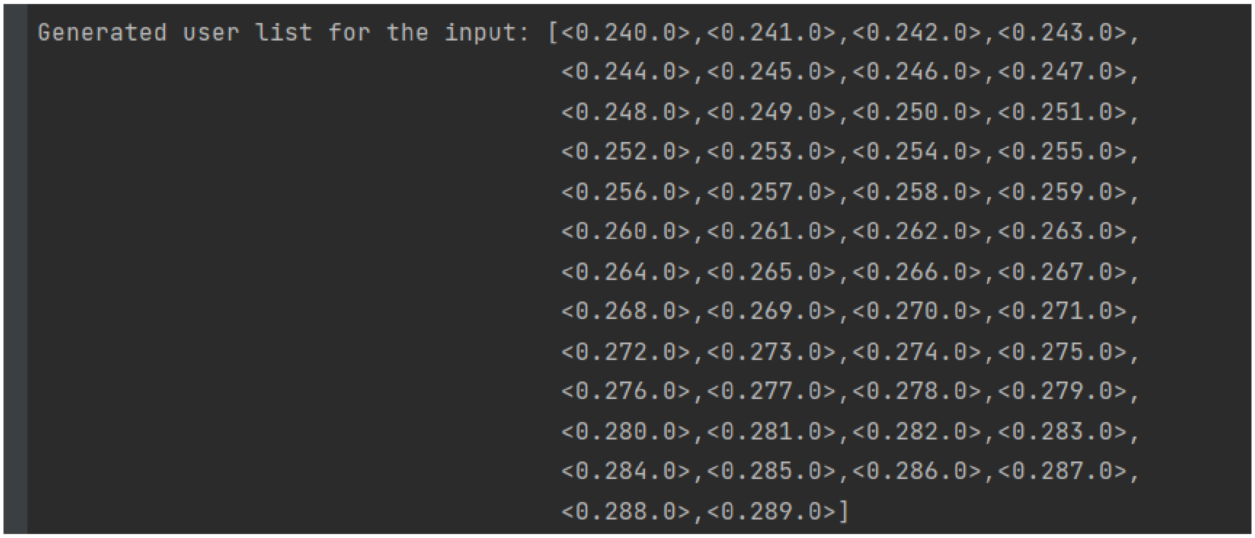


PRE-REQUISITES

The system must have Erlang installed, and running Twitter server and client simulators independently calls for the use of two terminals.

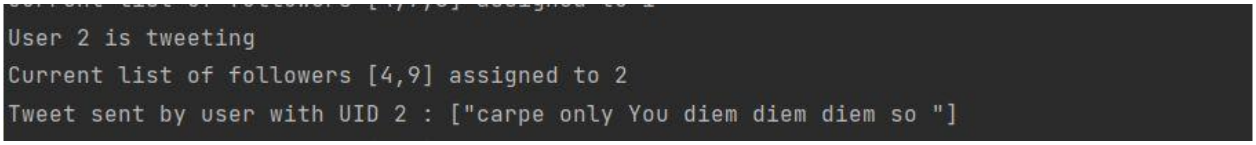
Features Implemented

- Twitter Engine start: enables the program to go online by launching the server
- Register user: Enables the user to create an account on the social media clone, without which the user cannot utilize any of the application's features.

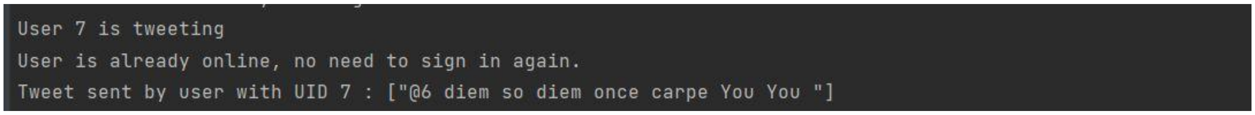


```
Generated user list for the input: [<0.240.0>,<0.241.0>,<0.242.0>,<0.243.0>,
                                    <0.244.0>,<0.245.0>,<0.246.0>,<0.247.0>,
                                    <0.248.0>,<0.249.0>,<0.250.0>,<0.251.0>,
                                    <0.252.0>,<0.253.0>,<0.254.0>,<0.255.0>,
                                    <0.256.0>,<0.257.0>,<0.258.0>,<0.259.0>,
                                    <0.260.0>,<0.261.0>,<0.262.0>,<0.263.0>,
                                    <0.264.0>,<0.265.0>,<0.266.0>,<0.267.0>,
                                    <0.268.0>,<0.269.0>,<0.270.0>,<0.271.0>,
                                    <0.272.0>,<0.273.0>,<0.274.0>,<0.275.0>,
                                    <0.276.0>,<0.277.0>,<0.278.0>,<0.279.0>,
                                    <0.280.0>,<0.281.0>,<0.282.0>,<0.283.0>,
                                    <0.284.0>,<0.285.0>,<0.286.0>,<0.287.0>,
                                    <0.288.0>,<0.289.0>]
```

- Send a tweet: This enables the user to express his emotions and current thoughts by posting tweets. A check has been incorporated to check that a registered user cannot tweet unless he has created an account.

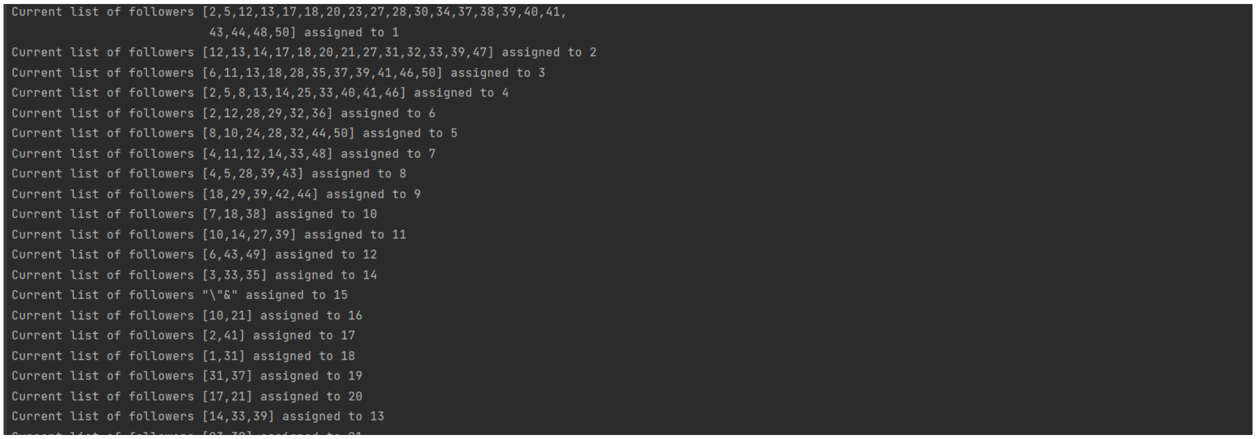


```
User 2 is tweeting
Current list of followers [4,9] assigned to 2
Tweet sent by user with UID 2 : ["carpe only You diem diem diem so "]
```



```
User 7 is tweeting
User is already online, no need to sign in again.
Tweet sent by user with UID 7 : ["@6 diem so diem once carpe You You "]
```

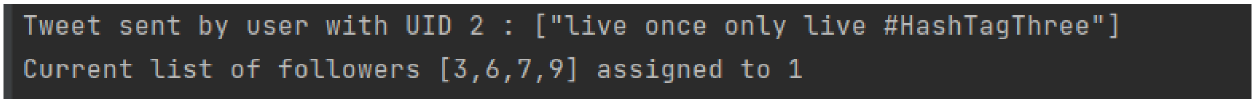
- Subscription to users: Users who wish to connect with other users whose tweets they find interesting and would like to follow can do so by subscribing to their profiles. All of a user's tweets become part of the user's personal tweet list when they subscribe to them.



```
Current list of followers [2,5,12,13,17,18,20,23,27,28,30,34,37,38,39,40,41,
                           43,44,48,50] assigned to 1
Current list of followers [12,13,14,17,18,20,21,27,31,32,33,39,47] assigned to 2
Current list of followers [6,11,13,18,28,35,37,39,41,46,50] assigned to 3
Current list of followers [2,5,8,13,14,25,33,40,41,46] assigned to 4
Current list of followers [2,12,28,29,32,36] assigned to 6
Current list of followers [8,10,24,28,32,44,50] assigned to 5
Current list of followers [4,11,12,14,33,48] assigned to 7
Current list of followers [4,5,28,39,43] assigned to 8
Current list of followers [18,29,39,42,44] assigned to 9
Current list of followers [7,18,38] assigned to 10
Current list of followers [10,14,27,39] assigned to 11
Current list of followers [6,43,49] assigned to 12
Current list of followers [3,33,35] assigned to 14
Current list of followers "\"&" assigned to 15
Current list of followers [10,21] assigned to 16
Current list of followers [2,41] assigned to 17
Current list of followers [1,31] assigned to 18
Current list of followers [31,37] assigned to 19
Current list of followers [17,21] assigned to 20
Current list of followers [14,33,39] assigned to 13
```

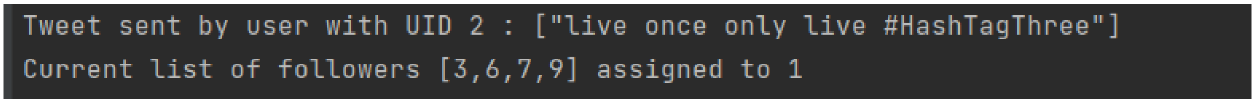
- Retweeting: Enables users to repost tweets of their liking and interest.
- Querying:

1. Hashtags: Displays a list of tweets that include a particular hashtag that was input by the user. if no such match is found, returns nothing.

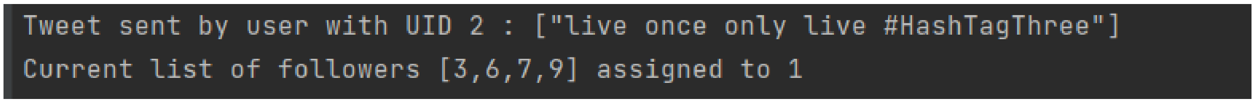


```
Tweet sent by user with UID 2 : ["live once only live #HashTagThree"]
Current list of followers [3,6,7,9] assigned to 1
```

2. Mentions: Returns the mentions of the tweets of the person requested and if the person isn't present, it returns no match.

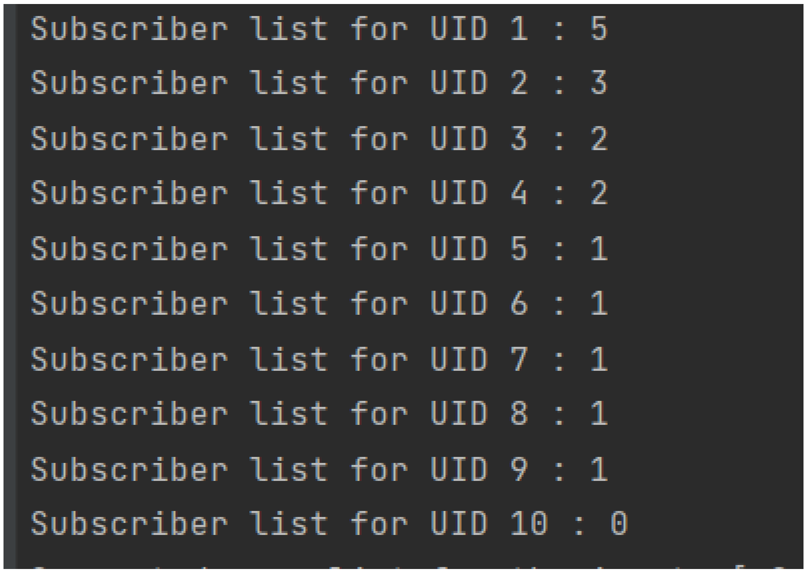


```
Tweet sent by user with UID 2 : ["live once only live #HashTagThree"]
Current list of followers [3,6,7,9] assigned to 1
```

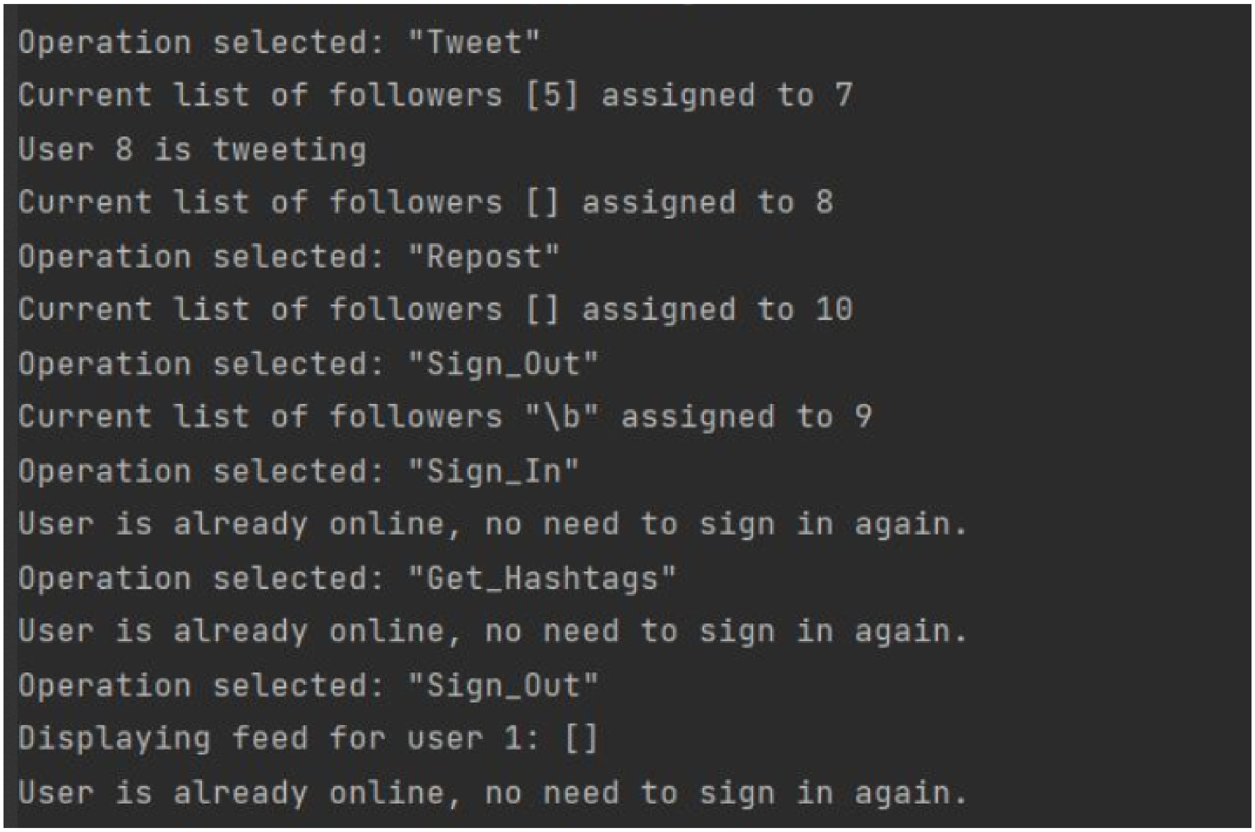


```
Tweet sent by user with UID 2 : ["live once only live #HashTagThree"]
Current list of followers [3,6,7,9] assigned to 1
```

3. Subscribe: This enables users to view the tweets of the people they have subscribed to.



```
Subscriber list for UID 1 : 5
Subscriber list for UID 2 : 3
Subscriber list for UID 3 : 2
Subscriber list for UID 4 : 2
Subscriber list for UID 5 : 1
Subscriber list for UID 6 : 1
Subscriber list for UID 7 : 1
Subscriber list for UID 8 : 1
Subscriber list for UID 9 : 1
Subscriber list for UID 10 : 0
```



```
Operation selected: "Tweet"
Current list of followers [5] assigned to 7
User 8 is tweeting
Current list of followers [] assigned to 8
Operation selected: "Repost"
Current list of followers [] assigned to 10
Operation selected: "Sign_Out"
Current list of followers "\b" assigned to 9
Operation selected: "Sign_In"
User is already online, no need to sign in again.
Operation selected: "Get_Hashtags"
User is already online, no need to sign in again.
Operation selected: "Sign_Out"
Displaying feed for user 1: []
User is already online, no need to sign in again.
```

Implementation Details

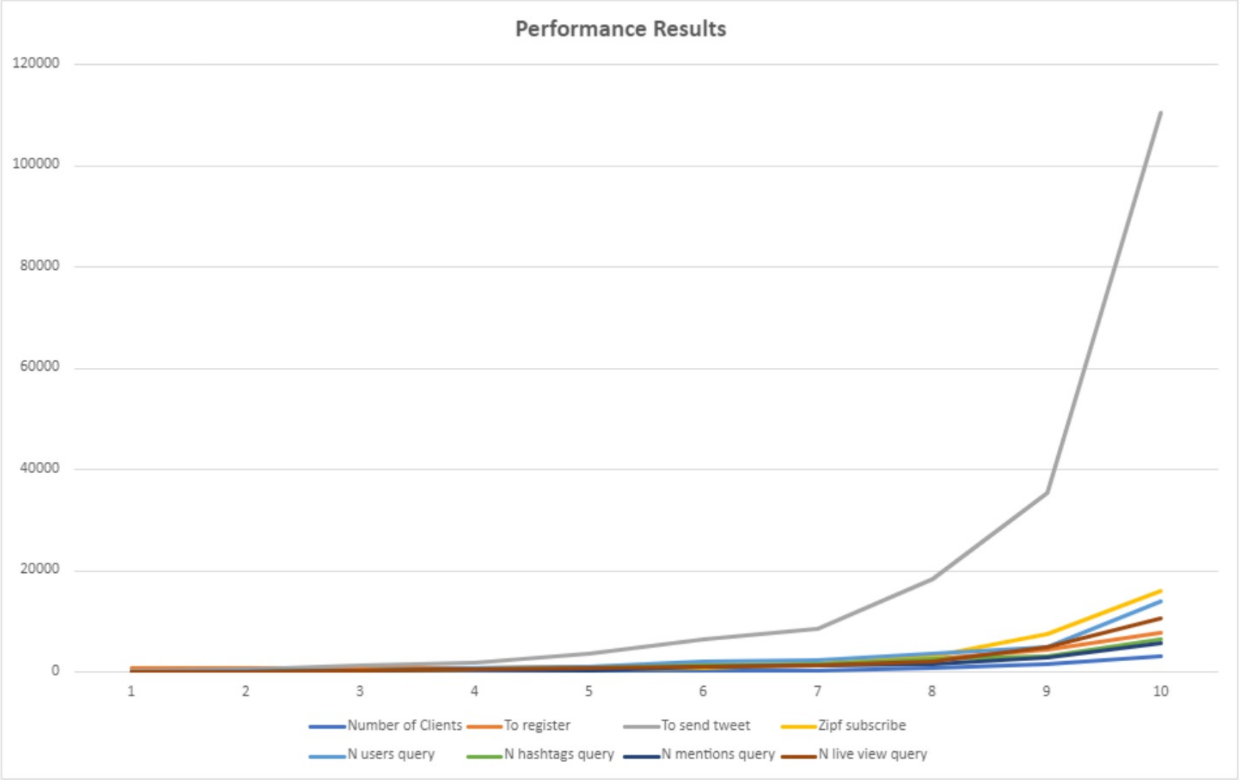
server.erl: The Twitter engine implementation code is included in this, to supervise handling and implementing tweets. To manage subscriptions, tweets, searches, etc., the engine directly interacts with the database. It also communicates directly with users via the API handler in order to relay the search queries and send the tweets to subscribers. Process states are maintained by the client's registry ETS table, which keeps track of server actors.

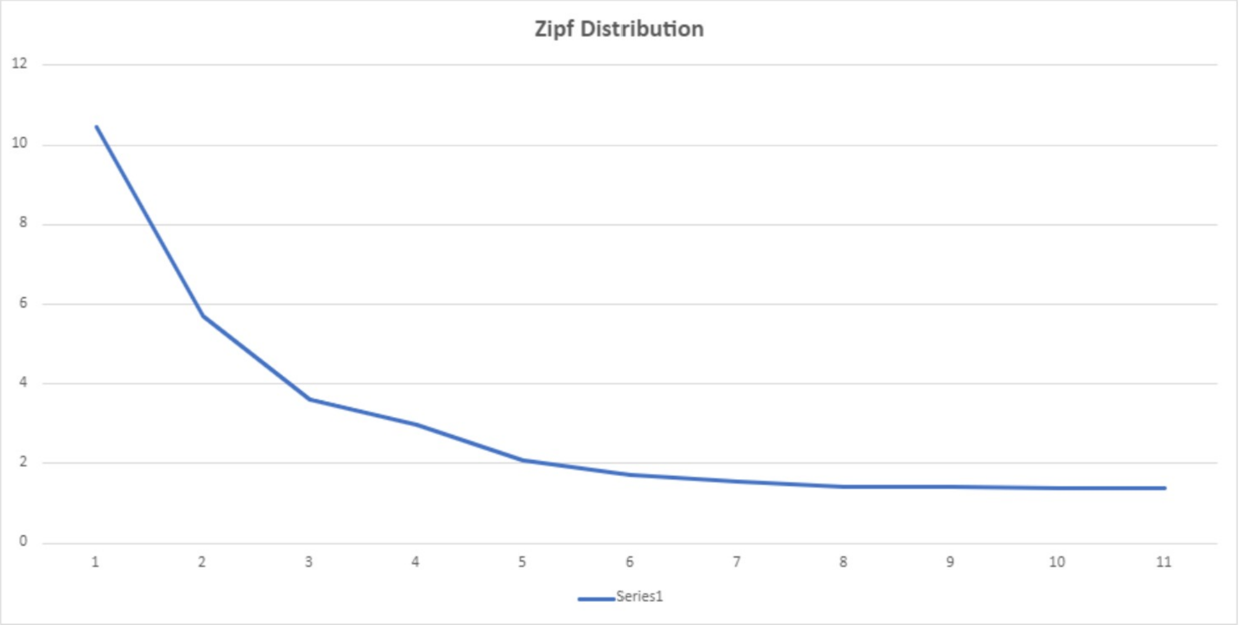
client.erl: This contains the code for the client. It sends requests to the twitter engine and tests it's working and functionality.

Zipf distribution - In accordance with the specifications, we were required to model a Zipf distribution according to the subscriber count. The user with the second-highest subscriber count had a total subscriber count of maxSubscribers/2, the user with the next-highest subscriber count had a total subscriber count of maxSubscribers/3, and so on. This was done by employing the equation noToSubscribe = round(Float.floor(totalSubscribers/(noOfClients-count+1))) - 1, where 'count' is the userId of a client, to determine the amount of peers a person should register to in order to ensure zipf distribution.

Observations and Graphs

Number of users	Register	N users : 10 tweets	Zipf Subscribe	N users query	N hashtag query	N mention query	N Random Ops
5	679.67	264.6	101.87	131.1	109.26	104.12	131.52
10	739.86	467.17	130.54	183.96	127.12	136.18	133.24
25	887.44	1,271.58	208.7	326.32	208.74	203.45	251.58
50	849.08	1,771.38	547.98	844.24	573.44	418.94	299.02
100	1,261.37	3,668.31	366.34	850.03	353.11	369.27	867.73
200	1,142.64	6,475.22	775.96	2,189.67	1,310.85	1,246.93	777.67
400	1,361.06	8,579.69	1,550.86	1,428.99	1,178.31	616.3	1,205.13
900	2,635.78	18,551.73	3,458.86	2,468.88	1,577.32	1,562.44	2,135.49
1800	4,364.85	35,759.25	7,345.58	5,024.35	3,010.32	2,971.00	5,084.73
3600	7,442.84	1,10,990.34	15,225.70	13,889.39	6575.74	5,946.99	10,498.63


Performance Results
120000
100000
80000
60000
40000
20000
0
1
2
3
4
5
6
7
8
9
10
Number of Clients
To register
To send tweet
Zipf subscribe
N users query
N hashtags query
N mentions query
N live view query


Zipf Distribution
12
10
8
6
4
2
0
1
2
3
4
5
6
7
8
9
10
11
Series1