**PROJECT 4.1**

**Twitter Simulator**

**Team Members:**

**Dikshant Gupta (*UFID:6840-1523*)**

**Lavish Mehta (*UFID:7981-8557*)**

**Project is based to simulate the architecture of Twitter engine and give the performance analyses on various test cases performed.**

**How to run**

1.Unzip the folder Gupta\_Mehta.zip

2.From terminal go to path Gupta\_Mehta\twittersim.

3. Run command “mix run project4.exs num\_of\_users num\_of\_Msg”.

Number of users are the users in the network and number of messages are the message requests made in the network.

**Functions provided in the project are:**

1 Registration of a user.

2 Login of a user while check for duplicity.

3 Sending tweets.

4 Tweets can include predefined Hashtags (eg; #UFL) and mentions of other users (like tagging someone in a tweet).

5 Retweeting a tweet which is liked by a user from a different user.

6 Following of a user (subscribing one user by others using Zipf distribution).

7 Delete an account while checking for the user.

**Working of the project:**

A Genserver is made which acts as a backend server which handles all the requests from the client side. It works to handle all the functionalities to run the services.

The server maintains all the repositories (ets tables for the several functions) like:

1 **Register Account:** As soon as the server starts, it registers all the users into the ETS database with incremental user ids.

2. **Authentication:** Apart from the user ids and usernames, passwords are also stored in the ETS database. Passwords are used as an authentication parameter to authenticate the user and refuse the connection if the wrong password is entered.

3. **Send Tweet:** This functionality sends tweet to all the followers of a user. It also saves the tweet in the ETS database with username as the key. It also extracts hashtags and user mentions from the tweet and store them in different tables ie. Send tweets with hashtags/ with mentions.

4. **Re-tweet**: A user can retweet a tweet and all its followers will receive that tweet. The retweeted tweets will have the owner username appended with it to identify the username who initiated the tweet.

5. **Hashtags & User Mentions Query**: The user can query for tweets with specific hashtags and user mentions. User gets the list of all the tweets that has the hashtag and mentioned user id specified by the user.

6. **Delete User:** The user can delete its account. All the data stored in the system gets deleted

Functions on the client side:

**On the Client side**, we are creating multiple Client GenServers, based on the input. The twitter client contains the state where we are tracking several actions of the user.

**Zipf Distribution for Followers:** The simulator follows zipf distribution scheme to create followers for each user. The zipf distribution will make sure that some of the users are have large number of followers due to good fame, while most of the users are normal users with a smaller number of followers.

In the Client side every user does all the services sequentially send a tweet, query the tweets based on the hashtags, based on mentioned user.

We set the user online(connected) and offline(disconnected) by adding and removing from the active user table in the server class, which gives the live feed from server from offline messages.

**Test Case implementation:**

**1 Register new user:** It will true for new user wanted to get registered into the system.

**2 Login correctly:** Once the user is registered this test case will check and return true if the password matched with the correct user id.

**3 Followers check:** This case will check the followers for all users in their followers table according to zipf distribution.

**4 Send tweet:** This test will check and return true for the tweets which are able to be sent.

**5 Receive tweet:** This test will check if the sent tweet is added to the correct user\_id and will return true.

**6 Testing wrong login:**  This function will check if the login is made by an authenticate user or not , also is he registered or not.

**7 testing logout:**  This will return true on successful logout

**8 Testing hashtag:** If the hash tags sent are correct will be tested by this case.

**9 Retweet:** This will return true if the retweeting is successful.

**10 Query by mention:**  This function will check for the query by the user based on mentions.

**11 Query by hashtag:**  This case will check for the queries by the users based on the hashtags.

Test cases are working fine, sometimes test cases fail due to randomness in setting the followers the active users as we are only making few users online. Please re-run again, the test cases execute perfectly.

Performance Observations:

We have performed the simulation on maximum. 1500 nodes and 100 requests it took around 4 minutes for entire simulation.

|  |  |  |
| --- | --- | --- |
| No of Nodes | No of Requests | Time taken |
| 100 | 10 | 2 secs |
| 100 | 100 | 10 secs |
| 500 | 10 | 12 secs |
| 500 | 100 | 40 secs |
| 1000 | 10 | 1 minute |
| 1000 | 100 | 1 minute 30 sec |
| 1200 | 10 | 1 minute 20 sec |
| 1200 | 100 | 2 minutes |
| 1500 | 10 | 2 minutes 40 sec |
| 1500 | 100 | 4 minutes |

From the above table, the time for simulation is proportional to the number of Nodes and Number of requests.