

DOS PROJECT 4

PART 1

Shubham Saoji 26364957
Himavanth Boddu 32451847

Project Description:

The goal of this (and the next project) is to implement a Twitter-like engine and (in part 2) pair up with Web Sockets to provide full functionality.

The Twitter Engine acts as server and users are clients. We have used GenServer actors for server as well as clients and simulated functionalities like create account, login, logout, subscribe, tweet etc. Overview of all functionalities is provided later in this report.

Multiple ets tables are used as server storage or database for server/ twitter engine. Test cases have been written and successfully tested for each functionality.

Overview of functionalities:

The functionalities that have been implemented in this project are as follows -

1. Register account –

Client can register into twitter engine. This has been simulated in project by spawning actor for client. The entry of this new user is stored in ets table, 'registered_user'. Each entry has another field having values "C" if user is connected, "D" if user is disconnected i.e after user logs out and "X" if user has deleted account/profile. After register, as user is connected, we set this field to "C" as user is currently connected.

2. Login -

Client can login into account and then is eligible to send tweets, receive live feeds or perform any other operation. Value in ets table 'registered_user' is set to "C" after user is connected.

3. Logout -

After user logs out value in ets table 'registered_user' is set to "D" and no longer receives live feed.

4. Delete account -

After user account is deleted, value in ets table 'registered_user' is set to "X" which corresponds to entry being obsolete.

5. Tweet -

User when connected can send tweet. Hashtags and mentions may be included in tweet. This tweet is sent to Twitter Engine and engine stores it in 'tweets_by_user' table corresponding to that user. Also, tweets are stored in 'hashtag_table' and 'mention_table' as per the hashtag and/or mention used in tweet.

6. Subscribe to other users -

User can subscribe to other user, so that it can receive tweets by that user. For simulation purpose in this project, we have chosen random subscriptions for a user. 'subscriptions' table is used in Twitter Engine to store the subscriptions on a user.

7. Re-tweet -

A user can retweet a tweet made by one of its subscriptions so that user's followers also get this tweet.

8. Query tweets subscribed to -

This function returns all the tweets made by subscriptions of a particular user.

9. Tweets with specific hashtag -

This function returns all the tweets containing that hashtag.

10. Tweets with specific mention -

This function returns all the tweets containing that mention.

11. Live feed -

In case a user is online, he should receive, without quering, tweet containing his mention or made by the user whom he subscribed to.

All these functionalities are working as intended and test cases for these functionalities have also been included.

We have written function called "operation" which performs an operation requested by client. This will ensure, if client wishes to perform any operation like disconnect, query results by a particular hashtag, results for particular mention etc then these results are rendered as per operation requested.

Execution:

To execute test cases, write in CLI,
\$ mix test

To execute project, write in CLI,
\$ mix run project4.exs 500 30

where 500 is number of clients to be simulated and 30 is number of messages each client sends.

Output:

When we try to run project with input of client number and message number, the clients are first registered into system(Engine) and then logged in. Each user sends number of tweets provided as input and displays live feeds if any.

This function call is asynchronous and displays output which is not in sequence, as display of all clients is encompassed into one display.

Performance:

Number of clients	Tweet count	Time(sec)
10	2	1.49
20	5	1.51
100	10	1.58
1000	20	7.84
1000	200	8.68
2000	200	39.4

Time includes time for register, login, send tweet and live_feed operation.