Twitter Simulator

Ghodkari Chowdary, Raghunatha Rao UFID: 6218-1051

Aim of this project is to develop a Twitter clone and measure the performance of this simulator on various aspects.

How to run

1. Unzip the folder GhodkariChowdaryMaroke.zip
2. From terminal go to folder twitterengine
3. Run command "mix run proj4.exs numofusers noofRequests".
   Ex. mix run proj4.exs 100 5, where the project takes 100 users and each user sends 5 tweets

Following Services have implemented and Available

1. Register account and delete account for user
2. Send tweet, where tweets can have hashtags and mentions
3. Users can subscribe to other user's tweets, (followers)
4. Retweet, a user can re tweet an interesting tweet received
5. Querying of tweets with hashtags and mentions
6. Live feed of tweets available to user without querying if he is online

Implementation Details

- A GenServer process is created TwitterEngine. It acts like a server whose job is to handle all the services the project described.
- The GenServer TwitterEngine maintains all the ETS tables for storing the info
    1. User table: To maintain all the users registered
    2. Mentions table: To maintain all the tweets containing a specific mentioned user
    3. HashTag table: To maintain all the tweets containing a specific HashTag
    4. Tweets table: To maintain all the tweets send by user to twitter engine
- In each element in users table we are storing the followers list, following list and list of offline messages need to send to user when he comes online
- The TwitterEngine State contains fields to maintain total number of users required as per the input, number of registered users, a list to maintain active number of users and counter to maintain the number of tweets the engine is handling

- We are creating multiple TwitterClient GenServers, based on the input. The twitter client contains the state where we are tracking several actions of the user.
    - Tweet feed which contains the tweets received by the user.
    - A list of tweets which stores the tweets the user posts.
    - List of hashtags to store the hashtags in the tweets the user posts. Lists of mentions to store the users mentioned in different tweets.
    - A counter to maintain the number of tweets the user is posting and a field to maintaining the followers count
- We have a main GenServer whose job is to create multiple users and build followers for each user and then it kickstart the simulation of the project
- In the simulation each user does all the services sequentially send a tweet, query the tweets based on the hashtags, query tweets based on mention user.
- We set the user online and offline by adding and removing from the active list in the engine state, which gives the live feed from server from offline messages
- We also invoke retweet operation in the end, where a user can retweet from his tweetfeed
- We measure the performance of our simulator by changing the parameters of number of followers for each user, number of active users available initially at start of the simulator
- Output on the console shows the live feed of every action occurring where users send tweet, user query based on hashtags and mentions, user goes online and offline for specific amount of time and retweet operation

Test Cases

For each functionality there are 2 test cases available

- Test cases for user registration where the user registered is stored in the user ets table and assert is performed
- Test cases to check if the followers and following are build for the user
- Test cases to send the tweet where a user sends the tweet and the followers of that user receives the tweets
- Test cases where a user queries the tweets based on the hashtags and mentions
- Test cases where we set user online and offline
- Test cases for user deletion where the user is delete in the table, from user active list
- Test cases for retweet the user from the tweets received

- Test cases to check the live feed where the user comes online and gets the tweets without the querying

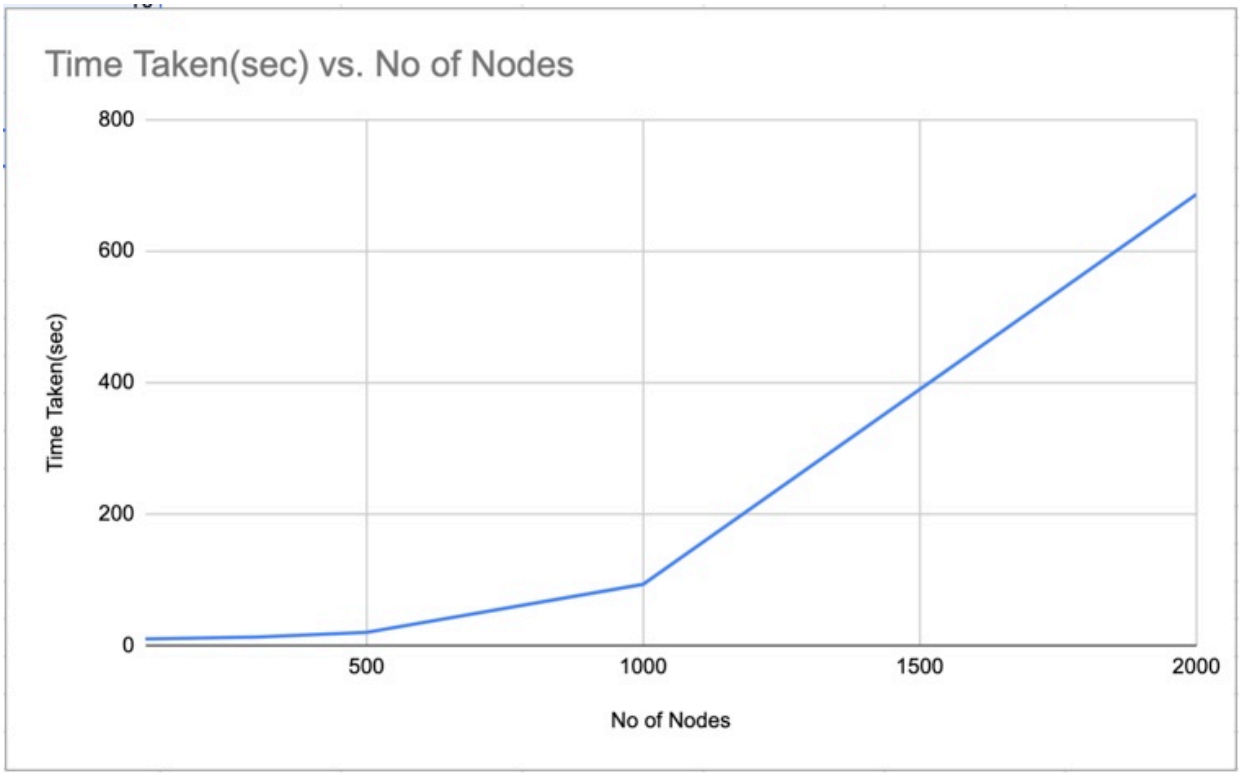
Note: Test cases are working fine, sometimes test cases fail due to randomness in setting the followers the active users as we are only making few users online. Please re-run again, the test cases execute perfectly

Performance Metrics

60% Users active initially and 30% followers for each User
We are measuring time taken for simulator

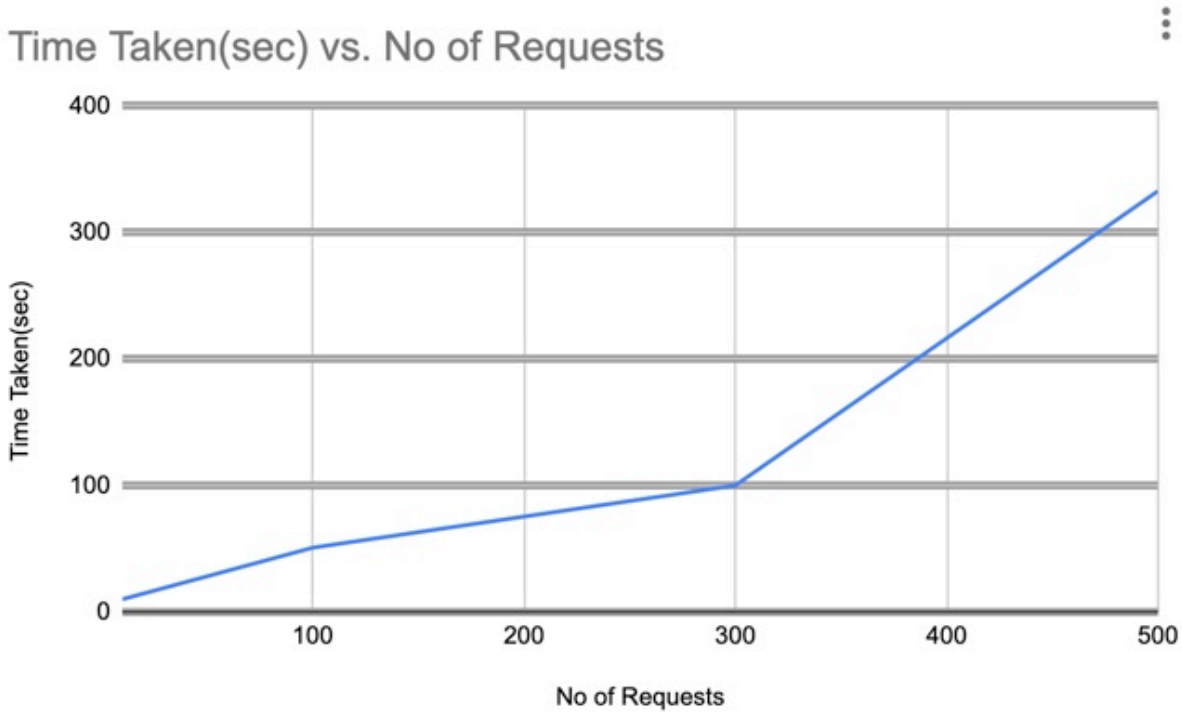
No of Nodes	No of Requests	Time taken
100	10	10 secs
300	10	12.9 secs
500	10	20 secs
1000	10	1 min 33 sec
2000	10	11 min 26 sec

We have performed the simulation on maximum. 2000 nodes it took around 11 min for entire simulation. For nodes greater than 2000 it will take more time.



60% Users active initially and 30% followers for each User
We are measuring time taken for simulator for 100 Nodes

No of Nodes	No of Requests	Time taken
100	10	10 secs
100	100	50.8 secs
100	300	1min 40 secs
100	500	5 min 33 sec



- The simulator time is proportional to the number of Nodes and Number of requests
- Our computer was able to handle 2000 nodes which took around 10 min and for increase in number of nodes the system will take enormous time as time is increasing exponentially
- For 100 users the number of requests to server was around 1.9 per second while for 500 users it was 1.1 requests per second.