Report(Bonus Part)

Twitter simulator

Rohan Nitin Gandhi (UF ID 1791 0958)
Deep Gosalia (UF ID 5697 9299)

Brief Description:

The goal of this project is to implement a Twitter Clone and a client simulator. We have implemented multiple process which act as clients, one server and one engine. In this the server is a process which does all the preprocessing works like parsing a tweet or giving a user a valid user id, etc. The engine distributes the tweets to the designated clients. The server and the engine can be thought of as one single process. Just to increase the efficiency of the project they are divided into two. The main of distribution of tweets is still handle by the engine, but the pre-processing stage is handled by the server.

The Twitter engine (Server) has been implemented with the following functionality:

- Register account
- Send tweet. Tweets can have hashtags and mentions
- Subscribe to user's tweets
- Re-tweets (so that your subscribers get an interesting tweet you got by other means)
- Allow querying tweets subscribed to, tweets with specific hashtags, tweets in which the user is mentioned (my mentions)
- If the user is connected, deliver the above types of tweets live (without querying)

How to run:

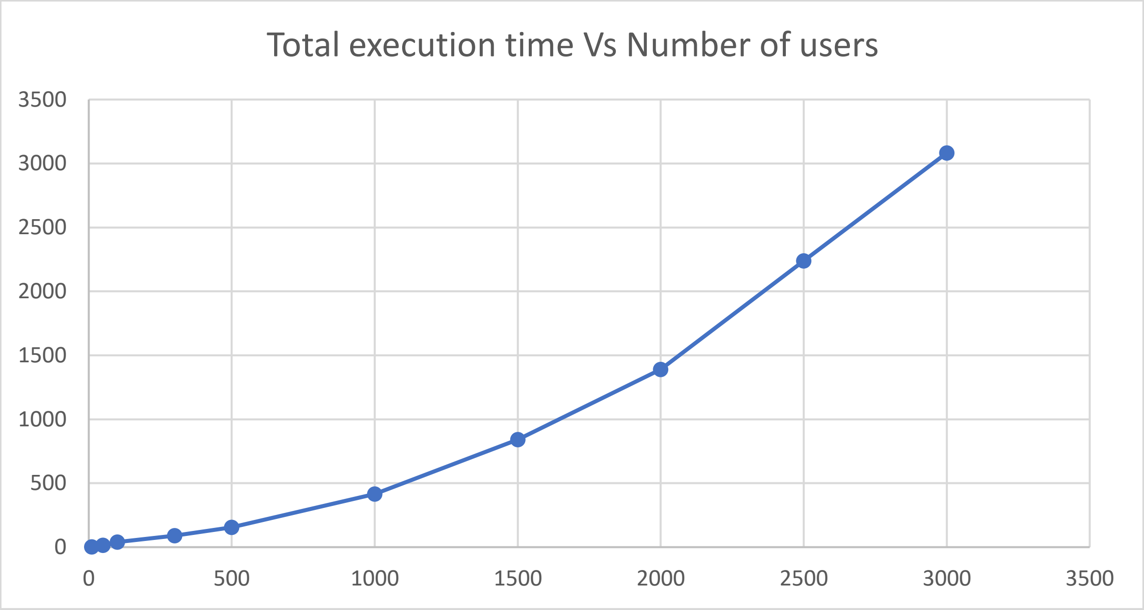
```
Mix run proj4.ex numuser numtweets
```

Implementation details:

Distribution the subscribers (zipf distribution): Here, we have taken the user id as decimal digits
If the user ID is divisible by 10, then those users will get 70% of the total number of users. This user will send 70% of the total tweets. This user will also retweet more frequently.
If the user ID is divisible by 5, then those users will get 15% of the total number of users. This user will send 15% of the total tweets
If the user ID is an even number, then those users will get 10% of the total number of users. This user will send 10% of the total tweets
For all the other users the subscribers will be 5% of the total number of users. This user will send 5% of the total tweets

Live connection or disconnection of user: we randomly give users status as online or offline. This information is stored in an ETS table. The value can be 0(offline) or 1(online). The engine checks whether the user is online or offline. If the user is online then the engine will send the tweet in the users live feed. If the user is offline, then when the user come online, then he will immediately receive all the tweets. The users are randomly logged out or logged in while sending a tweet.

Performance Result:


Total execution time Vs Number of users