DISTRIBUTED OPERATING SYSTEMS (COP 5615)

Project 4 - Part I [Twitter Simulator]

Swara Lawande [6081-3129]

Vikash Pandian Kasipandi [1305-9614]

Contents

Team Members

Swara Lawande [6081-3129]

Vikash Pandian Kasipandi [1305-9614]

How to Run Code

- Unzip file 'Project4.zip'.
- Our program consists of two parts, the server and the client. The server needs to be up and running before we start the clients. To start the server, run the following command:
  dotnet fsi twitter_engine.fsx
- Once the server is up, we can run multiple clients that will query the server. To run the client program, use the command:
  dotnet fsi twitter_simulator.fsx <client-port> <clientID> <numUsers>
  Where,
  client-port: random port number where we want to host the client process
  clientID: ID of the current client. clientID should always be sequential, for example user1, user2, user3…
  numUsers: total number of users hosted by each client. If multiple clients are started, then numUsers should be same across all.

  An example to run 1500 clients across 3 clients is illustrated below:
  i. Run **dotnet fsi twitter_simulator.fsx 1234 user1 500** in terminal 1
  ii. Run **dotnet fsi twitter_simulator.fsx 1235 user2 500** in terminal 2
  iii. Run **dotnet fsi twitter_simulator.fsx 1236 user3 500** in terminal 3

What is Working

Twitter-Engine

- Register account

```
[SIGNUP] User user1_3150 registered with server
```

- Send tweets with hashtags(#trend) and mentions(@user1)

```
[TWEET] user1_4509tweeted -> tweet_5 with hashtag #powerpuffGirls
[TWEET] user1_2466 tweeted -> tweet_8
[TWEET] user1_341tweeted -> tweet_32 with hashtag #weLovePizza and mentioned - @4014
```

- Subscribe to other user's tweets

```
[FOLLOW] User @user1_2794 followed @user1_1011
```

- Re-tweets

```
[TWEET] user1_4814 tweeted -> tweet_6
```

- Query tweets in which a user is mentioned and is subscribed to

```
[QUERY_MENTION] User @4706 mentioned in [tweet7, tweet17]
```

- Once user is connected, tweets to which user is subscribed to or is mentioned in are delivered.

```
[TWEET] user1_2739tweeted -> tweet_13 with hashtag #weLovePizza and mentioned - @1421
[QUERY_HASHTAG] Hashtag cybermonday mentioned in [tweet8, tweet28, tweet4, tweet5, tweet25]
[QUERY_MENTION] User @2215 mentioned in [tweet9, tweet26, tweet11]
[QUERY_HASHTAG] Hashtag sellingSunset mentioned in [tweet2, tweet17, tweet19, tweet14]
[SIGNUP] User user1_4560 registered with server
[QUERY_MENTION] User @894 mentioned in [tweet17]
```

- Users can go offline and online

```
[OFFLINE] User user1_2990 is going offline
[ONLINE] User user1_1020 is online..No feeds yet!!!
```

Twitter-Simulator

- Multiple users who can register, connect and disconnect to the engine.
- Users can tweet, re-tweet, query mentions and hashtags and follow other users.
- Multiple clients can be run simultaneously on separate machines which host specified number of users.

Largest Network

The maximum number of users we simulated was 99999 across 3 clients. We haven't put a limitation on the number of tweets or any other parameter. The simulator will continue to randomly simulate twitter users until manually terminated.

Implementation

- Twitter-Engine

  The twitter-engine consists of designated actors to carry out all the actions. It handles requests from the clients (twitter-simulator) as described below:

  UsersActor: takes care of user registration, online/offline users, subscribe to other users, render feed)

  MentionsActor: looks after storing new mentions and querying all the mentions of the users. Certain mentions are already added to the mentionsMap.

  HashtagsActor: some predefined hashtags are stored for the user to query on startup. New hashtags are stored in-memory and querying of hashtags is enabled by this actor.

  RetweetActor: allows users to retweet a certain tweet and pushes that tweet the feed of all the subscribers of the user.

ShowFeedActor: takes care of rendering user's feed once online and refreshing the feed.

TweetActor: manages all tweets that a user makes and all the subscribed tweets.

- Twitter-Simulator

  This is to simulate the actual users of the twitter engine. We can run multiple client processes on multiple machines by running the twitter-simulator program as described above. It uses the following actors to emulate the activities on twitter:

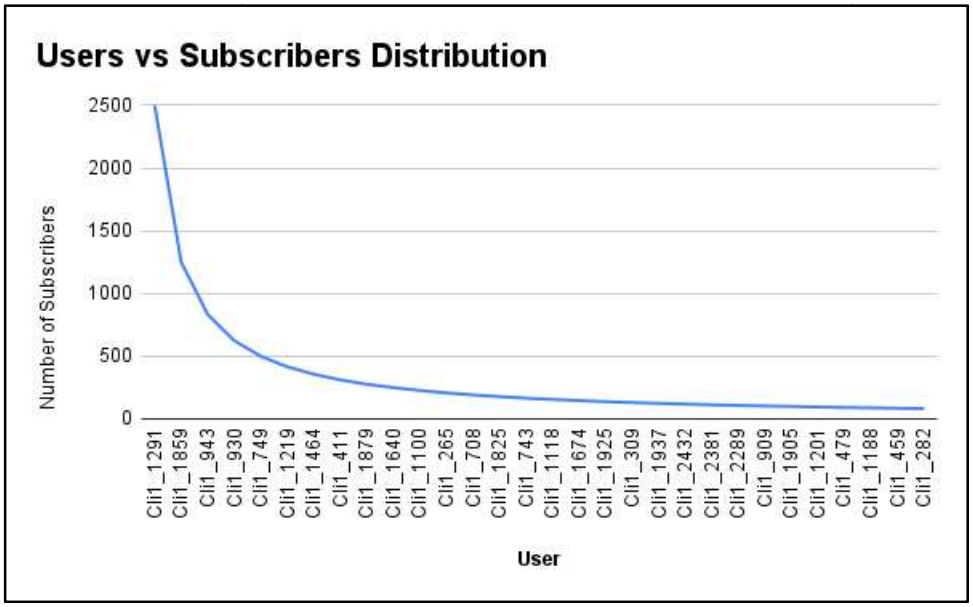
  User_actor: All actions that users can perform such as tweets, retweets, follow(subscribe), query hastags and mentions are handled by the user_actor.

  Client_supervisor: The client_supervisor simulates user activity such as going offline/online at random and manages registration of new users.

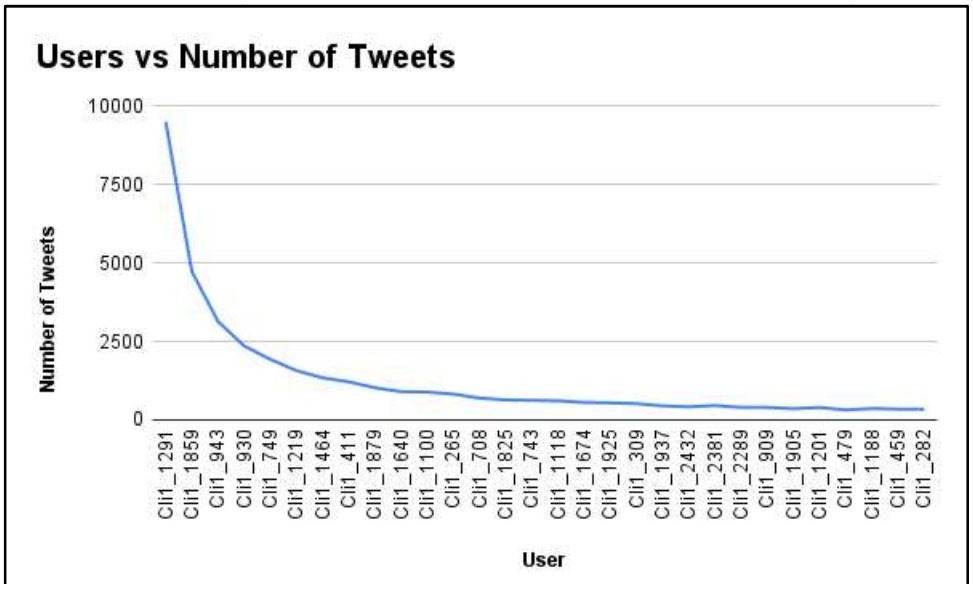
  Logger_actor: is responsible for logging the messages received from the various actors.

Zipf Distribution

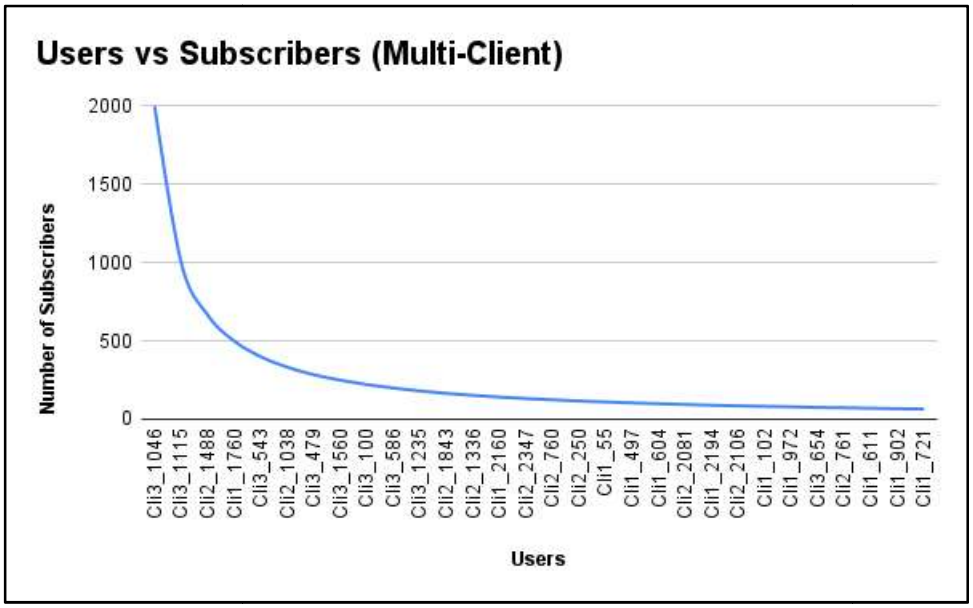
We carried out the distribution of number of subscribers to users based on the Zipf law, such that user 1 has 1000 subscribers, user 2 will have (1000/2) = 500 subscribers, user 3 has (100/3) = 333 subscribers and so on.

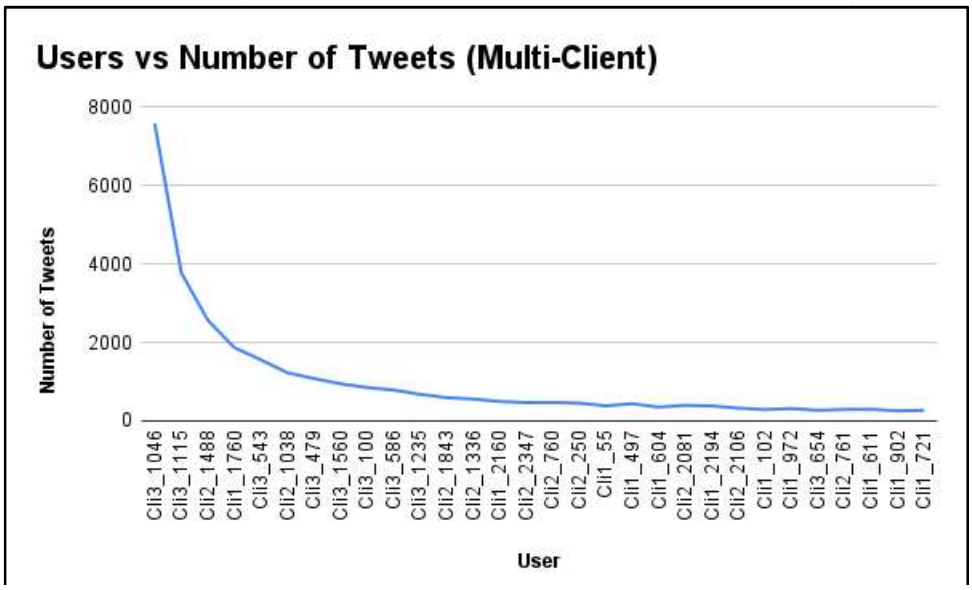


Then we plotted the number of tweets/re-tweets of same users on a graph. We observed that this graph also follows a Zipf distribution inspite of the tweets being completely randomized. That is, the more subscribers a user has, the simulator generates more activity(tweets/retweets).



Similar pattern is observed in multi-client simulation as well and is evident from the graphs below:





Performance

- We log various performance parameters in the 'Perflog.txt' file. We have measured the performance based on the time taken by our twitter engine to execute the various services. Few instances of the performance log after running our engine and simulator for 10 minutes each, by varying the number of users is shown below.

    i. The below instance is for 1000 users after running the

```
01-12-2021 19:35:21
Number of user requests handled per second = 61

Average time taken for service(s) in ms:
Follow/Offline/Online = 15.305148337687791
QueryMentions = 15.540039004090046
QueryHashtags = 13.601254426425412
Tweet/Retweet = 31.989599353312354
```

    ii. 3000 users

```
01-12-2021 19:40:17
Number of user requests handled per second = 82

Average time taken for service(s) in ms:
Follow/Offline/Online = 15.783806519310776
QueryMentions = 12.94543736910832
QueryHashtags = 14.65718649523107
Tweet = 35.167043645053
```

iii. 5000 users

```
01-12-2021 19:44:08
Number of user requests handled per second = 114

Average time taken for service(s) in ms:
Follow/Offline/Online = 19.20653644799045
QueryMentions = 12.958441101694964
QueryHashtags = 13.567112493020839
Tweet = 47.45990596314916
```

- From the above experiments, we observe that the number of requests handled by the twitter-engine increases with increase the number of users. However, increasing the number of users does not really affect the time taken by the average time taken by services.
- Along with the above performance logs, we also log the server uptime and number of requests handled by the engine as visible in the below screenshot.