

DOSP Project 4 Part-1

Sai Siddharth Upadhyayula (UFID: 7459-7449)

Pravallika Maddukuri (UFID: 5980-7014)

1) Twitter Machine:

The objective of the project is to build a Twitter-like functioning engine which can perform few applications like Authentication of the user, posting tweets, re-posting the tweets, following a twitter account, and querying any tweets. The tweets posted by the user may include names of other users and any hashtags. The tweet search can be achieved for the tweets which are from the followed accounts, tweets containing any hashtags, and for the posts which contains the name of the user.

2) Twitter Clone:

The twitter clone would be one which could simulate the users with periodical connections. The followers a user possess' must be in line with the Zipf distribution. The users having more number of followers must be tweeting and retweeting higher.

3) Considerations:

The frontend and backend of the twitter engines should be in individual processes. We must be able to use a myriad number of separate frontend processes to keep up with the large number of users and one twitter-engine process.

4) Execution:

Twitter Backend:

- The twitter engine is used as the application's server or the backend, which is capable of performing functions like Authentication, Posting and Sharing other tweets, following other twitter accounts and searching for any tweets.
- Individual features of the application are handled by an actor which responds to the request received by a particular feature.
- There is a request handler which is the lead actor where it caters to all the requests and transfers them to the concerning actors handling the features.
- There is a printer actor which outputs the number of requests handled for each 5 seconds.
- The server acts as the storage for the user's information such as the followers, following, tweets and any retweets.

- The tweets posted by the user are always secured or encrypted at the frontend by making use of the symmetric key. They stay secured at the backed. They will only be decrypted at the frontend.

Twitter Clone:

- The user must first register before using any features provided by the Twitter.
- After the user is registered, the user will be directed to the Twitter homepage where the user can make use of the features like tweeting, retweeting, following, tag users and search for tweets.
- There are some trending hashtags which are preloaded or displayed when a user decides to tweet or retweet so that it is easier for the user to look up for the recent hashtags being used around.
- A large number of processes are used to handle a large number of the users.
- Since the tweets made by the user are secured at the frontend and is only decrypted at the frontend, the printer takes a while to output the tweets.

5) Zipf Distribution:

- The Zipf distribution is made use while considering the number of followers for the user.
- Each user is given a rank to grant him the unique status and the index of the user stored in the array is considered as it is a unique value.
- The users are arranged such that the user with the first rank has the highest number of users and the arrangement continues in a descending order.
- The Zipf distribution is employed on each client even when various clients are connected.
- For instance, if there are 7 clients which are connected and each client had 150 users. In this case, 7 users will be present for each rank with equal number of followers.
- There is a small time-interval assigned after which the user will be enabled to perform some action. In our case, we have taken a time interval of 1 ms after which the user will be able to perform any action.

- The time interval assignment follows a similar pattern like Zipf distribution.
- The users with higher number of followers have less time interval to perform actions and the users with lower number of followers have slightly higher time interval to perform actions.
- If there are 200 users present and the followers are allocated to the users following the zipf distribution. Now, if the user with rank one contains 198 followers has 2 ms time interval to perform actions, then the user with the highest rank having 2 followers will have 198 ms to perform an action.
- By following the above logic, the users with more followers will be more active by implementing more actions than the one with smaller number of followers.

THE LARGEST NUMBER OF USERS WE MANAGED

Number of Users: 2000 (Almost 12000 tweets)

6) Running the project:

There are two files present:

1. client.erl
2. twitter_engine.erl

The two files have to be compiled first using the command 'c(filename).'

Now, we have to start twitter_engine.erl and client.erl in two different systems using the commands 'twitter_engine:start().' and 'client:start()'.

The server now starts running from the client terminal and we have to give command for registering then we can perform multiple tasks like tweet, retweet, mention, search for keyword, hashtag, follow, unfollow, and so on.

7) Functionalities discussed above:

1. User Registration
2. User Login
3. User Logout
4. Assigning users followers according to Zipf distribution
5. A map is developed with user as key and followers as value
6. Website Homepage
7. Adding new followers
8. Post Tweet

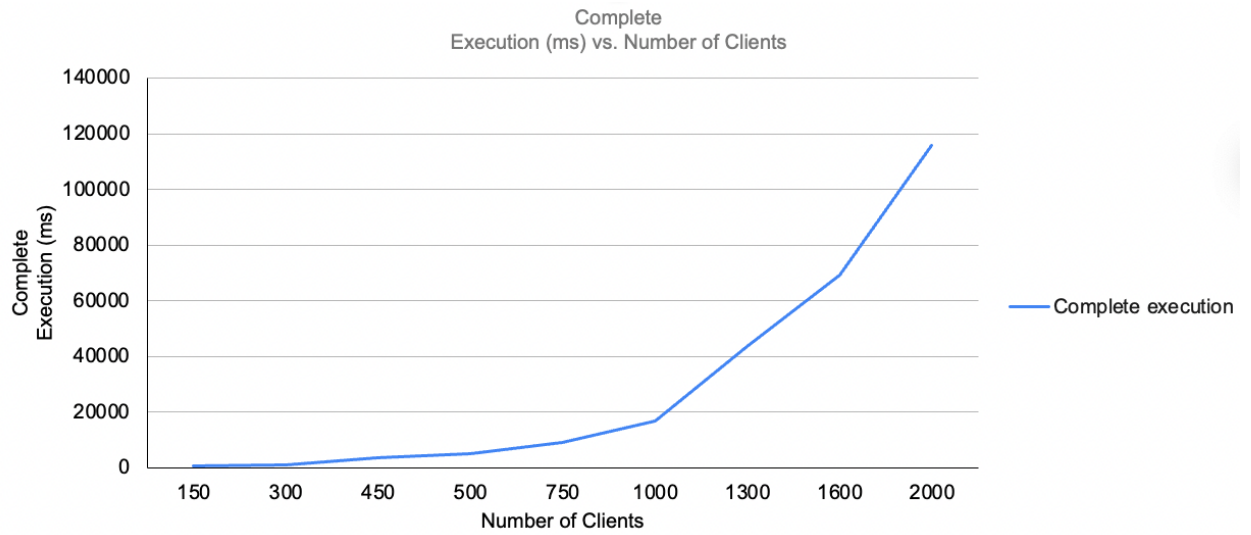
9. Retweet
10. Hashtag look-up
11. User Mention look-up
12. Keyword look-up
13. User tweets/retweets appear on the followers timeline.
14. The server adds the posted tweets/retweets to the pending map of the followers who are offline.

8) Tests Executed:

1. Ability to register users correctly – Successful
2. Ability to login correctly – Successful
3. Ability to logout correctly – Successful
4. Website homepage changes when user correctly logs in – Successful
5. The pending map should update values when the followers of the users tweets or retweets – Successful
6. The home map must not renew when the user is offline and the follower posts/shares – Successful
7. When the user logs in, the pending map must be cleared– Successful
8. When the follower of the user posts or shares and the user is online, the pending map is not renewed – Successful
9. New follower updated in the current followers list– Successful
10. All posts and shared posts include the looked-up hashtag returned– Successful
11. All posts and shared posts include the user’s mention returned – Successful
12. All posts and shared posts include the looked-up word returned – Successful

8) Statistics:

Number of Clients	Max Subscribers	to Tweet & Retweet	Subscribed To	Query Tweets by Hashtag (ms)	y Tweets by Mention	All Relevant Tweet	Complete Execution (ms)
150	149	48.57	130.65	40.99	17.21	8.09	818
300	299	56.97	572.61	114.37	57.03	99.83	1066
450	449	166.42	1635.47	260.71	184.61	125.32	3570
500	499	258.89	2313.74	323.52	337.89	149.06	5154
750	749	308.27	3577.73	648.92	428.51	215.11	9204
1000	999	1280.28	6968.66	1305.41	729.12	499.02	16702
1300	1299	2613.62	16425.63	3306.15	4277.6	1126.45	43346
1600	1599	3675.17	32177.74	2801.18	9437.07	1416.24	69270
2000	1999	5978.31	55466.31	7507.86	17135.57	2502.03	116658



Number of Clients	Max Subscribers	to Tweet & Retweet	Subscribed To	Query Tweets by Hashtag (ms)	y Tweets by Mention	All Relevant Tweet	Complete Execution (ms)
1500	150	40.78	203.14	393.79	327.81	211.98	1828
1500	300	169.18	427.51	429.63	363.11	227.26	2680
1500	600	589.46	1671.37	1697.37	#VALUE!	237.08	6130
1500	1200	1378.81	11444.89	2953.32	2135.86	285.33	18592
1500	1499	1879.24	15416.52	3131.28	2842.16	310.13	18860

