Twitter_Clone PART-1

Srinadh Kakkera (0514-0863)

Neharika Khera (8950-0993)

Brief Description:

The goal of this project is to implement a Twitter Clone and a client tester/simulator. The client part(send/receive tweets) and the engine (distribute tweets) are simulated in different processes.

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

- Register account
- Send tweet. Tweets can have hashtags (e.g. #COP5615isgreat) and mentions (@bestuser)
- 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)

Implementation Details:

Main.ex: This elixir file is the entry point and hosts the Mainmodule method. Our implementation takes 4 parameters in the order - noOfUsers, noOfTweets, maxSubsc and noOfDelUsers

- noOfUsers: the number of users to simulate (eg. 100)
- noOfTweets: the number of tweets a user has to make
- maxSubsc: the maximum number of subscribers a Twitter account can have in the simulation (eg. 50)
- noOfDelUsers: the percentage of clients to disconnect to simulate periods of live connection and disconnection (eg. 10)

Server.ex: This file contains the code for Twitter engine implementation that is responsible for processing and distributing tweets. The engine directly communicates with the database (implemented using ETS tables) to handle subscribers, tweets, queries, etc.

Client.ex: This file consists of the information pertaining to a single user participant that stands for a single twitter user. This includes the information of its userName which is stored as a numeric string passed to it upon initiation.

Re-tweets are handled by making every client randomly pick a tweet from one of its subscribers and retweet it to its own subscribers. A “-RT” is appended to the end of a retweet to differentiate it from normal tweets as in original Twitter.

All the queries with tweets subscribed to, tweets with specific hashtags, tweets in which the user is mentioned (my mentions) were handled successfully by message passing to the server and displaying the list of tweets received on the client side. The tweets will be delivered live to a user that is online by means of live view. User ID is prefixed to this output to identify which user's live view is getting updated.

Zipf distribution:

Zipf distribution is basically used for the number of Subscribers. For any user the number of subscribers are as (A constant Value)/ userNumber. userNumber is interpreted as username in the program. So, the constant value is the reciprocal of (1/1+1/2+1/3……1/noOfUsers).

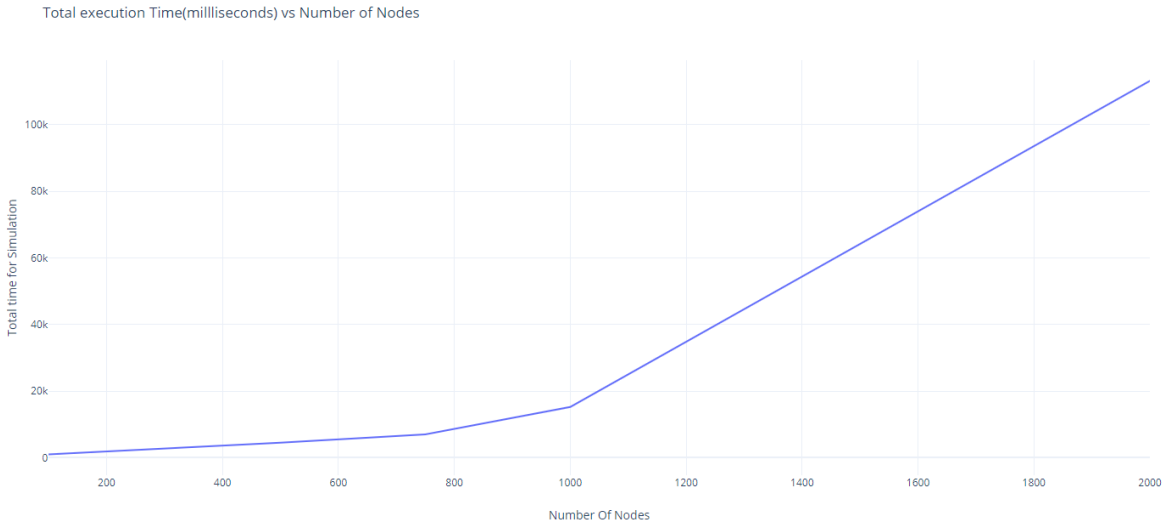
Periods of live connection and disconnection for users – The third parameter taken by the program is disconnect clients which takes in the percentage of clients to disconnect to simulate periods of live connection and disconnection. If <noOfDelUsers> parameter is 0, the client's simulator console displays the performance statistics at the end. Otherwise, it prints the statistics and continues to simulate recurring periods of live connection and disconnection.

Performance Results:

Time for the simulation vs number of Clients.

Some Process.sleep() functions were used for keeping the whole simulator alive. This might affect the time . But for comparison this may not be a factor.

Number of Clients	Max Subscribers	Time to Tweet, Retweet (milliseconds)	Query Tweets Subscribed To (milliseconds)	Query Tweets by Hashtag (milliseconds)	Query Tweets by Mention (milliseconds)	Query All Relevant Tweets (milliseconds)	Complete Execution (milliseconds)
100	99	48.74	129.63	40.86	18.93	8.22	996
500	499	276.68	2412.12	329.56	340.94	152.29	4563
750	749	319.80	3690.56	686.83	428.97	226.25	7023
1000	999	1280.98	6823.45	1359.84	763.56	203.54	12531
2000	1999	5901.21	54480.38	7486.94	17156.25	2496.34	113141



Performance for varying Subscribers.

Number of Clients	Max Subscribers / Max Tweets per Account	Time to Tweet, Retweet (milliseconds)	Query Tweets Subscribed To (milliseconds)	Query Tweets by Hashtag (milliseconds)	Query Tweets by Mention (milliseconds)	Query All Relevant Tweets (milliseconds)	Complete Execution (milliseconds)
1000	100	39.56	186.25	380.63	321.93	173.235	1852
1000	200	103.26	402.19	396.236	356.57	123.89	2265
1000	500	419.08	1536.54	1387.59	465.98	175.77	5756
1000	750	145.22	7107.60	3089.58	1786.60	279.67	13658
1000	999	1264.29	8269.56	1356.82	729.46	223.36	16896


Total Execution Time(milliseconds) vs Max Subscribers
16k
14k
12k
10k
8k
6k
4k
2k
Total Execution Time
100
200
300
400
500
600
700
800
900
Total Execution Time