DOS Project 4 Part I

Mohammed Haroon Rasheed Kalilur Rahman
6751 2967
mkalilurrahman@ufl.edu

Ariz Ahmad
7111 2167
ariz.ahmad@ufl.edu

December 4, 2020

1 Implementation Details

- We have one server engine (Server.fsx) and N number client actors given by the user in Client.fsx. Simulation is performed by client actors in Client.fsx. All the requests are handled by the server engine. Server maintains tables for data storage - tweets, hashtags, mentions and subscribers. These values are maintained for each client in the Server engine. The number of subscribers are calculated using zipf distribution.
- Both the Server engine and client actors are remote processes, they are running in different processes.
- Client.fsx file takes input of the number of clients as input, spawns a remote client actor for each user, which is then registered with the server. Server engine will keep listening to requests from the client actors, in localhost:9000.
- Once the registration is done, the subscription process begins, where the number of subscribers are set for each user using zipf distribution.
- For each client actor, the subscribers are added randomly through the server engine.
- After this the clients start sending tweets to the server. Server then broadcasts these tweets to each of the individual subscribers of the client who tweeted.
- The tweets, hashtags, and mentions are also broadcast similarly.
- Tweets with mentions are only broadcast to people who are mentioned in the tweet, and not to anyone else.

- Hashtag is generated randomly from a list of five hashtags, and added to the tweet and sent.
- For re-tweets, client will receive the tweets he has subscribed to from the server. Then he would re-tweet few of these tweets to his subscribers.
- Live connection and disconnection is simulated by disconnecting few of the clients, and those clients will not receive live feed once they are disconnected.
- Each connected client will perform queries where he can request all the tweets he has subscribed to, or tweets with a particular hashtag, or tweets in which he has been mentioned (my mentions).
- These requests are sent to the server, and the server sends all the requested data only if the client is connected and it will be displayed in the client's homepage.
- Time for total simulation is calculated by taking the start time when the simulation started and is subtracted from the system time once all the functionality has been simulated. This is then printed in the client console window.

2 Results

- Total simulation time for the entire process is done for three tweets per user.
- For 2000 users, with three to five tweets per user was the best our system performed, in terms of running time.
- Keeping number of tweets per client to 3-5, we could increase the number of clients to without much decrease in performance.
- When we tried to increase the number of tweets per client, our performance decreased.
- The largest number of clients supported for our system was 3000, with execution time 319780 ms, and 19780 total requests.

Number of Clients	Total time of execution (in ms)	Total number of Requests
100	11824	674
250	29827	1671
500	58311	3329
750	87913	4977
1000	117946	6628
1500	179067	9924
2000	237707	13210

Figure 1: Total time of execution and total number of requests for number of clients

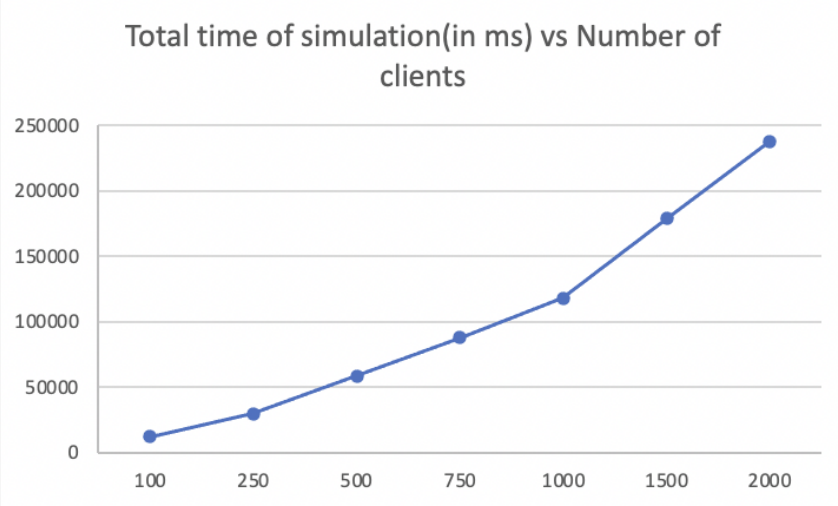

Figure 2: Total time of execution for number of clients