Distributed Operating Systems Project Report 4.1

Author 1 : Sitanshu Sukhmandir Lamba UFID: 5190-8991
Author 2 : Darshan Dilip Kakwani UFID: 5311-7117

This project implements a twitter clone. It consists of a Client, Client Parent, Server and Simulator.
It consists of almost all functionalities that would be present on a twitter engine such as tweet, retweet, followers, following, register new user etc.

We have even shown the timelines of users and simulation of search queries on twitter.

How to run the code :

Run the code through the command line with the following command:

"dotnet fsi --langversion:preview proj4.fsx Arg1"

Arg1: number of users to be simulated

Main Components:

The code consists of 4 main components:

1. Server: This is the part of code that implements the literal functionality of a server. Some examples of the functions in a server actor are distributing tweets, receiving tweets. Server also handles the querying part of this project, where we can search tweets according to hashtags and mentions.
2. Client: This part of the code is multiple actors that all hit the server and behave as "users" of twitter. The clients can tweet, retweet, perform queries and can view their tweets/ timelines if they are online.
3. Simulator: This is the main twitter simulator as it randomly assigns followers to users and follows other users. It also sends out random tweets with hashtags and/or mentions to the server and to show the accurate working of the twitter engine.
4. Client Parent: This spawns all clients and creates their respective actors.

Format of Sample Output:

The output is printed in the console in the following order :

- The network (who follows who) is printed after network initialisation.
- The users start tweeting.
- Feeds of the users who receive the tweets in live time are shown.
- Retweeting is also performed and the updated feed is shown similarly.
- The query to show the entire timeline of all users is run.
- Hashtag, Mention searches (queries) are run and the results are shown.

- Connection/disconnection simulation is run last to show how users timelines are updated when a user disconnects and then reconnects again.
- In the end the time taken for each simulation is printed.

Performance/Testing:

The table below shows the time taken for the tasks performed. The first column indicates the number of users that were given as input during this simulation.

All the outputs are in milliseconds (ms).

Number of users	Tweeting	Re-tweeting	Printing home timeline (of all users)	Hashtag (#) search query	Mentions (@) search query	Connection/disconnection simulation
10	23.5	7.5	1023.9	1036.96	1032.96	3006.5
100	146.47	134.29	10328.00	1036.464	1034.106	8041.62
500	2172.57	2722.05	51459.11	1039.72	1041.90	56333.18
1000	12114.54	12600.23	102905.01	1028.9	1037.03	71418.169
2500	82594.91	143103.9	257955.09	1038.48	1036.20	176146.07