Fall 2020

COP5615 Distributed Operating System Principles

Readme file

Project 4 Part 1 - Twitter Clone and a client tester/simulator

Group Details

Parth P. Chitroda	Saheel R. Sawant
5189-1737	1164-7923
pchitroda@ufl.edu	sawant.s@ufl.edu

Description :

In this project, we have developed a Twitter clone and a client tester/ simulator using the F# actor model. We have implemented this system based on the client and remote server architecture. We have provided users with both alternatives, to manually register and access the different functionalities or to view the simulation of the system for 120 seconds where multiple users register, tweet, retweet and have followers based on the Zip-f distribution.

On the client-side, users are provided with a list of options based on the different real-life functionalities of Twitter. Once the user enters the action he wants to perform, the user is asked to provide certain inputs and these are asynchronously sent from the client-side to the remote server-side. On the remote server-side, according to each respective functionality, data is either stored or processed, required computations are made, and accordingly, some feedback is sent back to the users to indicate the status of their actions.

We have implemented all the required twitter-like functionalities mentioned in the project description are explained as follows :

1. A user X needs to enter **option 1** to register in the system with a unique username. If the username entered already exists, then user X is accordingly notified.
2. After registration, if user X needs to login, then it needs to enter **option 2** and provide its registered username to access other functionalities.
3. User X can also log out of the system by selecting **option 3** from the choice menu. A list of total registered, online and offline users is maintained on the server-side.
4. If user X is logged-in and wants to post one or more tweets, then **option 4** should be selected and user X must enter its username and what they want to tweet about. Once the tweet is processed and stored on the server-side, user X gets informed that its tweet has been posted.
5. To follow other registered users, user X must select **option 5** and enter its username and the username who it wants to follow, who must be also registered in the system.
6. To get subscribed tweets, User X must follow one or more users. When user X selects **option 6** to get subscribed tweets, a list of tweets and retweets posted by people user X is following. User X can also retweet any tweets from the list of the received subscribed tweets and pass it onto its followers. The total number of tweets and retweets are updated on the server-side as per the user actions.
7. If user X wants to find tweets with a specific hashtag or mention, then **option 7** should be selected. The system asks to enter the hashtag or mention and based on the input a list of tweets is presented. User X can also get tweets containing both a hashtag and a mention tag.
8. A user X can also get all the live tweets posted by other users on the system by selecting **option 8**.
9. To terminate the program, user X can select **option 9**.

How to run Twitter clone implementation : **(Video: https://tinyurl.com/y2besch4)**

1. Fisrt run the remote server in cmd : dotnet fsi --langversion:preview Remote.fsx
2. To run the client in cmd : dotnet fsi --langversion:preview Client.fsx

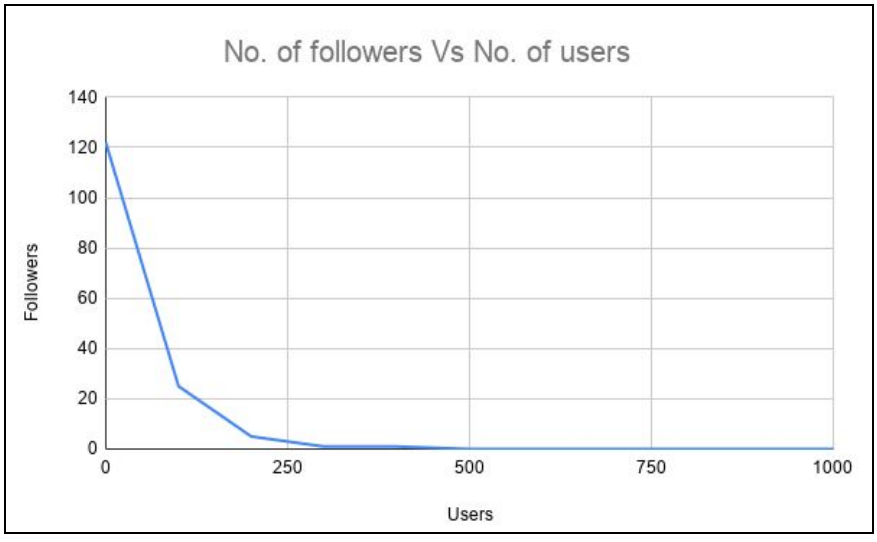
How to run the Twitter clone simulation : **(Video: https://tinyurl.com/y6dtnet6)**

1. First run the remote server in cmd : dotnet fsi --langversion:preview remoteSimulator.fsx
2. To run Client in cmd : dotnet fsi --langversion:preview clientSimulator.fsx numClients
   Example : dotnet fsi --langversion:preview clientSimulator.fsx 100

For the Twitter simulation, we have set a constant simulation duration of 120 seconds, after which the simulation stops. The maximum number of users we can simulate is 10,000. To measure the performance of the simulation, we computed the number of requests processed, and the total tweets posted each time. As the number of users increased, subsequently the total tweets posted and processed requests also increased. Also, the number of messages sent by each user is proportional to the number of followers. The obtained findings are as follows :

No. of Users	Number of Requests Processed	Total Tweets Posted
100	65546	37730
1000	79716	87254
10000	85642	156082

From the following graph for 1000 users, where the X-axis indicates the number of users based on their user IDs and the Y-axis indicates the number of followers, based on the zip-f distribution the initial 200-250 users comparatively had a higher number of followers than the rest of the total users as the computation of followers drops due to the subsequent division.



References: https://www.youtube.com/watch?v=9NvxDAUF_kI