**Project 4 Part 1 ReadMe**

Nikhil Tiwari UFID 91507670 ([nikhiltiware@ufl.edu](mailto:nikhiltiware@ufl.edu))

Parikshit Tiwari UFID 79218564 ([pariksh1tatiwari@ufl.edu](mailto:pariksh1tatiwari@ufl.edu))

**How to run:**

To run the application the Server needs to be run first. Followed by the Client.

**Running the Server (No arguments need to be passed)**

Navigate to the FacebookServer folder in terminal/commandline

For example navigate to “/home/Documents/ project4\_part1 /FacebookServer” or “C:\Desktop\ project4\_part1 \FacebookServer”

Run the command: sbt compile

Run the command: sbt run

**Running the Client (Number of users need to be passed)**

Navigate to the FacebookClient folder in terminal/commandline

For example navigate to “/home/Documents/ project4\_part1 /Facebook Client” or “C:\Desktop\ project4\_part1 \Facebook Client”

Run the command: sbt compile

Run the command: sbt “run number\_of\_users”

For example if you want to run 1000 users the command would be

sbt “run 1000”

The output is printed on the screen after 60 seconds showing the statistics of the simulation

**API Implemented**

The project closely implements the Page, Post, Profile, Friend List, Picture and Album APIs. The following functionalities were implemented.

1. Register Users and Pages: All users and pages are created and then registered.
2. User can visit any page look at the posts made by that page, the number of likes the page has. The users can then like the page. They can dislike a page they had already liked.
3. Pages can Post texts or picture albums on their own walls. They can also delete the text posts from their own wall. Any User can read the Page’s feed.
4. Any User can send a Friend Request to any other User. The recipient User then has the option to accept or decline the request.
5. The User can visit the profiles of other Users.
6. A User can Post texts and pictures on their own feeds or on the feed of the Friends in the list.
7. Users can get their Friend Lists and see the list of Friend Requests sent to them.

**User Simulation**

The paper we referred to while deciding the user simulation was “**CHARACTERIZING USER BEHAVIOR AND INFORMATION PROPAGATION ON A SOCIAL MULTIMEDIA NETWORK**” by the Boston Fusion Corp and the Palo Alto Research Center. The paper analysed the online behaviour of a set of users and classified them based on common characteristics. The paper collected the Facebook usage data from a set of users. While deciding the user simulation parameters we have roughly used a similar data model. The user are classified into 5 categories.

1. **Multimedia-Savvy & Engaged Users (C1):** These users are highly active with a high number of text and multimedia posts. Their activity is high in general. They are of type User and comprise of 17% of the total number of user and pages.
2. **Low Engagement Users (C2):** These users make very little post of either text or picture type. They are of type User and comprise of 22.5% of total users.
3. **Text-savvy Pages (C3):** These users make a lot of text type Posts and little of picture type Posts. This user is of type Page and comprise of 18% of the total users
4. **High Engagement Users (C4):** These users frequently post text type Posts, but very rarely post pictures. They represent the largest user group comprising of about 40% of the total population. They are of type User.
5. **Multimedia Specialists Pages (C5):** These user are of type page and post lots of pictures compared to text type posts. They comprise of 2.5% of total users.

The frequency of the reading of posts and pictures is generally higher for each category of user, this is normally the case in real life where we read more posts than write. Similarly the frequency of liking a page or post is also roughly in ratio with the posting characteristics of the category. The frequency of likes is generally lower than post frequency. The frequency of unliking a page or post is kept very low as has the frequency of deleting a post.

**Working**

The user enters the number of users to be created. The client side code then create Users and Pages in accordance to the percentages mentioned above. The Users and Pages register themselves. The registered users are stored in memory lists on the server side. The registered users then hit the server with various post and get requests in accordance to the user category mentioned above. On the server side 1000 actors are created to handle the large flow of incoming requests. Futures are used on the server side to handle the requests asynchronously and in a non-blocking manner. The response to the client is then printed on the client side.

**Statistics**

|  |  |  |  |
| --- | --- | --- | --- |
| Number of Users Simulated  For 60 seconds | 1000 | 5000 | 10000 |
| Number of pictures posted | 966 | 4809 | 9611 |
| Number of Friend Requests Sent | 3314 | 16542 | 33069 |
| Number of Posts made by User | 33076 | 165340 | 330678 |
| Number of Posts made by Post | 5207 | 26063 | 52129 |
| Total Number of Posts | 38283 | 191403 | 382807 |

**References**

* Francis T. O’Donovan , Connie Fournelle , Steve Gaffigan , Oliver Brdiczka , Jianqiang Shen , Juan Liu , and Kendra E. Moore , “CHARACTERIZING USER BEHAVIOR AND INFORMATION PROPAGATION ON A SOCIAL MULTIMEDIA NETWORK” (May 2013)

<http://arxiv.org/ftp/arxiv/papers/1305/1305.2091.pdf>

* Spray Tutorials Referenced
  + <http://engineering.monsanto.com/2015/08/11/simple-spray/>
  + <http://engineering.monsanto.com/2015/11/12/simple-spray-part-2/>
  + <https://www.youtube.com/watch?v=XPuOlpWEvmw&list=LLQkF9RWBrusNtV_YFdjPPXQ&index=3>