**How to run** :

Go into the project directory:

*First, run the Server:*

$ cd server

$ sbt compile

$ sbt run

*Next, run the Client:*

$ cd client

$ sbt compile

$ sbt “run n”

**Note:** Here the ‘n’ refers to the number of users (clients) to simulate e.g. $ sbt “run 10”

Note: The maximum number of users that we could simulate is 10,000

**Working** : The client program spawns ‘n’ number of client actors (users), and for every user, the client program initializes and maps some default information such as profile, page, friend list, user, picture and album. This information is AES-128 encrypted using the random key (generated using secure random number generator).

Then, for every ‘x’ seconds, client actors (users) sends get requests (profile, page, friend list, post, picture, and album) to the other client actors (users).

In addition, if any user makes a Post, all the user’s friends (actors) will be notified, so that the user’s friends can make an appropriate request (get) to see the posted data.

For e.g. if user2 actor makes a get request (profile) by sending its public key to user1 actor, then user1 actor will fetch its own information from server, decrypts it using the random key (stored with the user1 actor).

Now to send this information to user2, user1 actor will encrypt the information by AES-128 using the random key (generated using secure random number generator) and encrypts this random key as well with the user2 actor’s public key.

When user2 actor receives this encrypted information and encrypted key from user1 actor, the user2 actor first decrypts the key using its private key. Thus, using the key, user2 actor decrypts the encrypted information.

The simulation strategy is decided based on the real statistics from the references mentioned at the end of this report.

E.g., if the total number of Facebook users (clients) are 1000 then,

*Initialization:*

Populate all the 1000 users’ profiles, populate each user with 1 default picture.

Create 1 album for each user with 1 default picture in the album.

Average number of friends per user: 10. Connect each user with 10 random users (friends).

To summarize, after populating all the users’ memory, the following simulation is achieved by random actors

1. FRIENDLIST: Friend List is requested (Get) by random user for every 8 seconds and a Post friend request for every 15 seconds.
2. POST: Get, Post requests for POST for every 5 seconds.
3. PAGE: Get Page Posts request for every 10 seconds and Post requests for a PAGE for every 5 seconds.
4. PICTURE: Get pictures for every 10 seconds and Post pictures for every 7 seconds.
5. ALBUM: Get album requests for 10 seconds and Post album requests for every 6 seconds
6. PROFILE: Get profile requests for every 10 seconds and Put profile requests for every 7 seconds.   
     
   This simulation is achieved by the following global variables in the client program

|  |  |
| --- | --- |
| Frequency of Requests | Time in seconds |
| CREATE\_POST\_FREQ | 5 |
| GET\_POSTS\_FREQ | 5 |
| CREATE\_PAGE\_POSTS\_FREQ | 5 |
| GET\_PAGE\_FREQ | 10 |
| PUT\_PROFILE\_FREQ | 7 |
| GET\_PROFILE\_FREQ | 10 |
| POST\_FRIEND\_FREQ | 8 |
| GET\_PICTURES\_FREQ | 10 |
| POST\_ALBUM\_PICTURE\_FREQ | 6 |
| GET\_ALBUM\_FREQ | 10 |

Statistics References:

<http://www.statisticbrain.com/facebook-statistics/>

<https://zephoria.com/top-15-valuable-facebook-statistics/>

<https://blog.kissmetrics.com/facebook-statistics/>

<http://www.statista.com/statistics/267745/frequency-of-checking-of-facebook-accounts-by-us-users/>