COP 5615 – Project 4(Part I)

Facebook Simulator

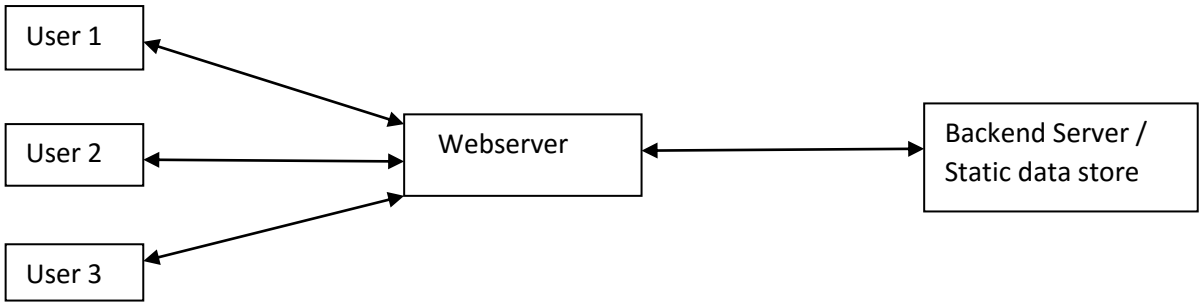
Priyanshu Pandey (UFID: 21081358)
Sourav Kumar Parmar (UFID: 86511933)

Instructions to run the program Build system: Linux (Ubuntu):

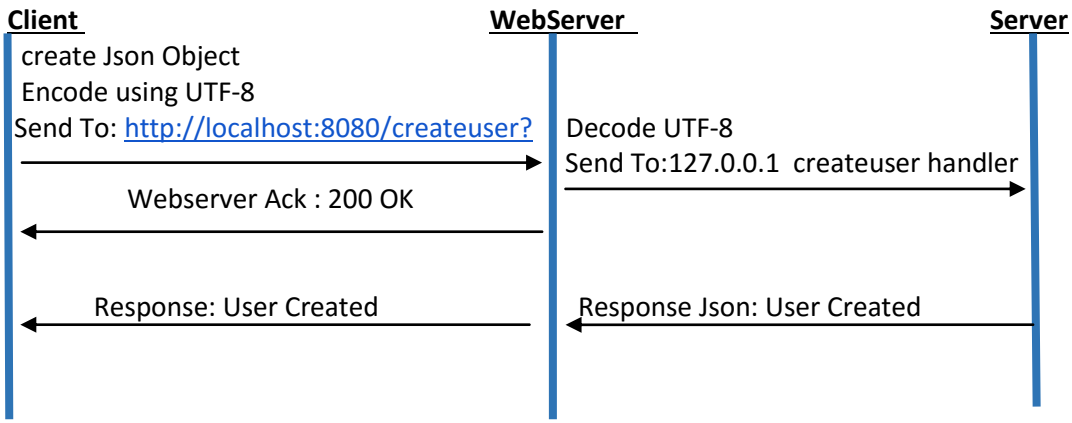
- Unzip the **PriyanshuPandey_SouravParmar_Project4.tar.bz2**
- Open Three terminal window
- Change directory to the following
  Terminal 2: Path: PriyanshuPandey_SouravParmar_Project4/Facebooksimulator/Server
  Terminal 3: Path: PriyanshuPandey_SouravParmar_Project4/Facebooksimulator/Webserver
  Terminal 1: Path: PriyanshuPandey_SouravParmar_Project4/Facebooksimulator/Client
- Execute the command sbt run in each of the terminal in following sequence.
  1st Server 2nd Webserver 3rd Client

Framework Used : Spray-can, Akka

Project Architecture :
Three layered Architecture
1st layer : User Interface: Client layer – This layer is used to simulate the behaviour of end users.
2nd layer: Webserver: Interface layer between Client and Backend server. This take cares of routing of client request to backend server and viceversa.
3rd layer: Backend Server: This is a database layer that maintains all the datastuctures and handles request from client.



Command Flow:
Example: Createuser



Implementation Details:

Client: Worker actors are generated which simulate client behaviour. Handling for createuser , createpost, pageview ,addprofilepic, addfriend , , addalbum etc are simulated.

Image is simulated using Byte array.

Three types of user are simulated

1) 50 percent Active users: These actors call handlers more frequently.

2)33.33 Percent Moderately Active users: Call handlers with frequency half of that of type 1

3) 16.67 Percent Inactive Users: Call handlers rarely.

Two type of methods are used to communicate to Backend Server:

1) Post : This is used while creation of activities eg createuser , createpost etc.

2) Get: This is used while fetching existing data from server eg viewpost, viewprofilepic

Webserver: It accepts the Http request from client and call the respective handlers of backend server. It acknowledges the request with error codes (200 ok , 400 handler not found)

Backend Server: This acts like a static Database store and provides handler to fetch data corresponding to the request from client.

For Post request from client entries are updated in corresponding data structures.

For get request from client corresponding value is fetched if exist and is sent back to client.

Observation:

Server Logs

```
Total Requests Served so Far = 191096
 Max Average Requests PerSecond(Over period of 10 sec)= ***** 4093 ********
Total Requests Served so Far = 484305
 Max Average Requests PerSecond(Over period of 10 sec)= ***** 8166 ********
```

Client side : Logs for viewpage are enabled. It shows the complete profile if person is in friend list else it shows restricted view.


```
{
  "loginid":"Ama",
  "firstname":"Amanda",
  "lastname":"Amanda",
  "gender":"M",
  "country":"England",
  "city":"London",
  "profession":"Politician",
  "interestedin":"Coding",
  "friendidlist":[
    {
      "firstname":"Fred",
      "lastname":"Fred"
    },
    {
      "firstname":"Melissa",
      "lastname":"Melissa"
    },
    {
      "firstname":"Carol",
      "lastname":"Carol"
    },
  ],
  "postlist":[
    "Be yourself; everyone else is already taken.",
    "In three words I can sum up everything I've learned about life: it goes on.",
  ],
 "albumidlist":[
    1,
    2,
    4,
    5
  ],
  "profilepic":"-7 51 111 103 50 -120 -76 78 -103 -60"
`
```

Conclusion:

From observed logs it can be concluded that on average 6000 + request are handled when all users are active.

Further Work:

Implementation of security for message passed

References:

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

https://github.com/spray/spray-json

http://www.javacodegeeks.com/2014/11/first-steps-with-rest-spray-and-scala.html