

Project 4 – Part 1

Twitter Simulator using Elixir

Members

1: Manjary Modi ,UFID: 38408368, manjary.modi@ufl.edu

2: Rameshwari Obla Ravikumar,UFID: 16161302, rameshwari.oblar@ufl.edu

What is working?

For this project, we have created a twitter engine and a client test/simulator which has functionalities such as the ability to create multiple clients, send tweets from various clients, use of hashtag in tweets for easy collection and storing of data related to a particular topic, use of mentions in tweets in order to specify a certain user. A user can subscribe to other users by using the 'subscribe' option. This will allow users to get tweets directly from the subscribed user who he/she is following. Our Twitter engine supports 'retweeting' so as to allow sharing of interesting tweets. A user can also search for previous tweets by using the 'QuerySimulator' in our program. This will retrieve tweets from the data store of the engine and display the required tweets to the user querying the system. The query is also delivered to the client if the client process is alive.

Implementation Details

- *project4.ex*

This file contains the main method of our program. Based on the different input choice given by the user - server,client,live or query - different interface is created.

1. Server

Server which is an individual kernel process, creates the main server - our twitter engine. The server is a genserver node named as 'server@127.0.0.1' and it is also responsible for creating subservers. The use of subservers plays an important part and has been explained further in the document.

2. Client Simulator

This is also kernel process that creates multiple independent client process and connects them to the respective subservers which in turn are managed by the server created above.

3. Query

This is used to query the tweets based on hashtags(#), mentions(@) and subscriber information.

4. LiveQuery

This is used to display the tweets live into a client terminal if that client is alive. This will display the tweets from that user and all the tweets from the users whom the user has subscribed to.

- *mainserver.ex*

This file contains methods that perform the work of the main server or the central server. This simulates the Twitter Server.

1. Handle_cast method - 'noOfClients'

This method is used to create subservers, datacenters and hashtag centers. To support huge number of clients, the server is distributed. The mainserver creates several subservers based on 'loadfactor'. The datacenters are also split based on the loadfactor. This is essential to improve the performance of our system.

2. Handle_cast method - 'setUser'

This method is responsible for simulating 'Registering an account' by an user in our Twitter system. This method creates as many number of clients as specified in the program. This is responsible for creating users, linking the newly created client process to the server, to assign necessary states and to specify which subserver it reports to.

3. Handle_call method - 'Tweet'

This method redirectes the tweet from clients to different subservers to process and store

4. Handle_cast method - 'setFollowers'

This method is used to build the user-subscriber relation. It builds a list that maps every user to all the users to which he/she has subscribed to. This is used to query all the tweets tweeted by the subscribed users.

5. Handle_call method - 'queryHashtag'

This method is used to query all the tweets with a given hashtag

6. Handle_call method - 'getsubscribertweet'

This method uses the information created in the setFollowers method to get the tweets that is being queried for. 'name' refers to the user name or userid

- *server.ex*

This file contain methods that performs the work of the subservers. This is used to improve the performance of our system.

1. Handle_cast method - 'initialize'

This method is used to initialize and fill the details of tweets in the datacenters, hashtag centers and the subserver.

2. Handle_cast method - 'tweet'

This method is used in actual processing of tweets. It adds timestamp to the generated tweet, stores the tweets in appropriate hashtagcenter and datacenter. It also updates who subscribes to whom.

3. Handle_cast method - 'deliver_tweet'

This method is used to display the tweets on the client process.

4. Handle_cast method - 'getsubscribertweet'

This method is used to send subscriber tweets to the datacenter.

- *HashtagCenter.ex*

This file is used to create a hashtag center which can hold information about the hashtags that appear in each tweet. It groups all the tweets having a particular hashtag so that it can be returned while querying

1. Handle_cast method - 'link_tweet'

This method is used to group all the tweets with a particular hashtag into a list of maps.

2. Handle_cast method - 'gethashtag'

This method is used to get all the tweets containing the given hashtag.

- *datacenter.ex*

This file is used for storing subscriber and 'mentions' information. Mainly used in querying.

1. Handle_cast method - 'store_tweet'

This method is used to store tweets with tweet_id, tweeter info and a flag to check if it is a tweet or a retweet.

2. Handle_cast method - 'mentioned'

This method is used to store all the tweets in which a user is mentioned.

- *client.ex*

This file contains methods that performs client functionalities.

1. Handle_cast method - 'startFollowers'

This method is used to create follower list. The followers are created by randomly picking variable amounts of followers for each user.

2. Handle_cast method - 'tweet'

This method is used to generate and send the tweet. It also defines the structure of the hashtags and mentions.

3. Handle_info method - 'startTweeting'

This method is used to send periodic tweets. More the followers, more the frequency of generating tweets. This facilitates the zipf distribution formulation in sending the tweets and is achieved by using rank of the user.

4. Handle_cast method - 'subscriberTweeted'

This method is used to display tweets of the user and the subscribed users. Live Simulation is called from this method.

- *quersimulator.ex*

This file contains methods that are used in querying of tweets by mentions, hashtag and subscriber. All the methods are self-explanatory

- *livesimulator.ex*

This file contains methods that displays subscribed tweets live on the user when it is alive.

1. Handle_cast method - 'subscriberTweet'

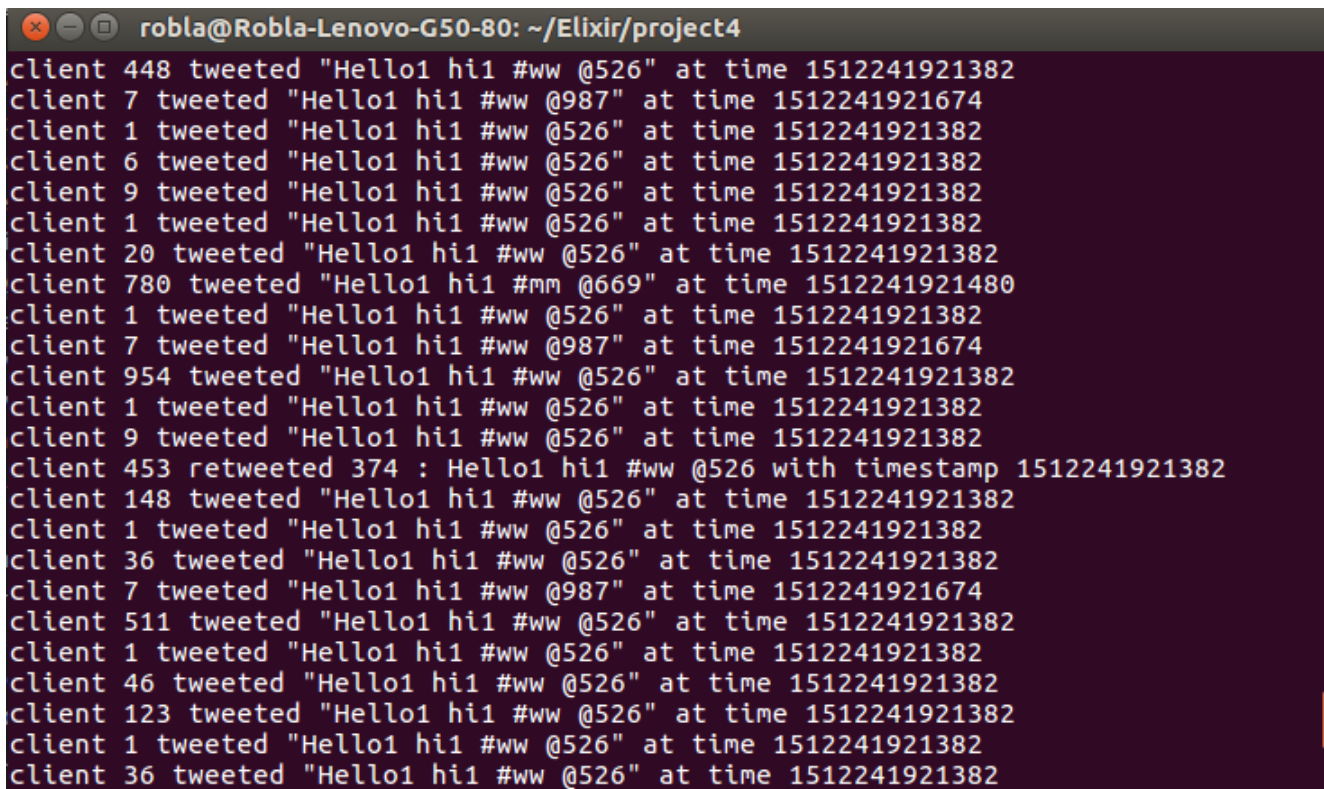
This method displays the tweet from the user and the tweets tweeted by it subscribers.

Simulation Results

1. Tweets with Hashtag, Mentions and Retweeting.

Hashtags are prefixed by '#' and Mentions are prefixed by '@'

Retweeting has information about original user who created the tweet.

A terminal window with a dark background and light-colored text. The title bar reads 'robla@Robla-Lenovo-G50-80: ~/Elixir/project4'. The terminal output consists of 25 lines of log messages. Each line follows the format: 'client [ID] [action] "[text]" at time [timestamp]'. The actions include 'tweeted' and 'retweeted'. The text content of the tweets includes 'Hello1 hi1 #ww @526', 'Hello1 hi1 #ww @987', and 'Hello1 hi1 #mm @669'. One line shows a retweet: 'client 453 retweeted 374 : Hello1 hi1 #ww @526 with timestamp 1512241921382'. The timestamps are either 1512241921382 or 1512241921480.

```
robla@Robla-Lenovo-G50-80: ~/Elixir/project4
client 448 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 7 tweeted "Hello1 hi1 #ww @987" at time 1512241921674
client 1 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 6 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 9 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 1 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 20 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 780 tweeted "Hello1 hi1 #mm @669" at time 1512241921480
client 1 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 7 tweeted "Hello1 hi1 #ww @987" at time 1512241921674
client 954 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 1 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 9 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 453 retweeted 374 : Hello1 hi1 #ww @526 with timestamp 1512241921382
client 148 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 1 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 36 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 7 tweeted "Hello1 hi1 #ww @987" at time 1512241921674
client 511 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 1 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 46 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 123 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 1 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 36 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
```

2. Max. No.of Users

We were able to simulate 100,000 clients in our system. It was using full 4 core power(Tested in a quadcore system). We believe more users can be simulated if more powerful CPU is used.

(The below screenshot also shows the registration process.)

```
robla@Robla-Lenovo-G50-80: ~/Elixir/project4
robla@Robla-Lenovo-G50-80: ~/Elixir/proj... x robla@Robla-Lenovo-G50-80: ~/Elixir/proj... x +
Registering client 99978 at server
Registering client 99979 at server
Registering client 99980 at server
Registering client 99981 at server
Registering client 99982 at server
Registering client 99983 at server
Registering client 99984 at server
Registering client 99985 at server
Registering client 99986 at server
Registering client 99987 at server
Registering client 99988 at server
Registering client 99989 at server
Registering client 99990 at server
Registering client 99991 at server
Registering client 99992 at server
Registering client 99993 at server
Registering client 99994 at server
Registering client 99995 at server
Registering client 99996 at server
Registering client 99997 at server
Registering client 99998 at server
Registering client 99999 at server
Registering client 100000 at server
```

3. Live Tweeting(without querying)

We were able to simulate live tweeting by displaying the tweets of the user and all the tweets tweeted by users whom the user is subscribed to.

```
robla@Robla-Lenovo-G50-80: ~/Elixir/project4
robla@Robla-Lenovo-G50... x robla@Robla-Lenovo-G50... x robla@Robla-Lenovo-G50... x +
robla@Robla-Lenovo-G50-80:~/Elixir/project4$ ./project4 live
Enter your id28
I am live
"I am subscribed to: [835, 829, 737, 577, 305, 405, 943, 422, 445, 408, 940, 123, 259, 526, 967, 259, 393, 766, 672, 122, 177, 768, 642, 414, 314, 364, 373, 888, 446, 827, 218, 427, 549, 659, 949, 518]"
client 445 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 393 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 408 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 28 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 28 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 28 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 28 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 393 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 393 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 967 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 28 tweeted "Hello1 hi1 #nn @644" at time 1512241924076
client 28 tweeted "Hello1 hi1 #nn @644" at time 1512241924076
client 177 tweeted "Hello1 hi1 #mm @669" at time 1512241921480
client 445 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 672 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 829 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 177 tweeted "Hello1 hi1 #ww @526" at time 1512241921382
client 305 tweeted "Hello1 hi1 #ll @68" at time 1512241921574
```

4. Querying of tweets by hashtag/mentions or subscriber.

#Hashtag Querying

```
rob1a@Robla-Lenovo-G50-80: ~/Elixir/project4
rob1a@Robla-Lenovo-G50-80:~/Elixir/project4$ ./project4 query
#PID<8382.77.0>
connected
*****
1.Mentions
2.Hashtag
3.Subscriber tweet
*****
what is ur choice 1 2 or 3 ?2
hashtag
which hashtag ... pls enter without #aa
[%{origin: nil, retweet: false, tweet: "Hello1 hi1 #aa @772", user: 27},
 %{origin: 27, retweet: true, tweet: "Hello1 hi1 #aa @772", user: 964},
 %{origin: 27, retweet: true, tweet: "Hello1 hi1 #aa @772", user: 982},
 %{origin: 964, retweet: true, tweet: "Hello1 hi1 #aa @772", user: 978},
 %{origin: 27, retweet: true, tweet: "Hello1 hi1 #aa @772", user: 1},
 %{origin: nil, retweet: false, tweet: "Hello1 hi1 #aa @494", user: 56},
 %{origin: 1, retweet: true, tweet: "Hello1 hi1 #aa @772", user: 980},
 %{origin: 1, retweet: true, tweet: "Hello1 hi1 #aa @772", user: 964},
 %{origin: 1, retweet: true, tweet: "Hello1 hi1 #aa @772", user: 969},
 %{origin: 1, retweet: true, tweet: "Hello1 hi1 #aa @772", user: 954},
 %{origin: 1, retweet: true, tweet: "Hello1 hi1 #aa @772", user: 986},
 %{origin: 1, retweet: true, tweet: "Hello1 hi1 #aa @772", user: 982},
 %{origin: 1, retweet: true, tweet: "Hello1 hi1 #aa @772", user: 1000},
```

@Mentions Querying

```
rob1a@Robla-Lenovo-G50-80: ~/Elixir/project4
rob1a@Robla-Lenovo-G50-80:~/Elixir/project4$ ./project4 query
connected to clint
#PID<8382.77.0>
connected
*****
1.Mentions
2.Hashtag
3.Subscriber tweet
*****
what is ur choice 1 2 or 3 ?1
mentioned
whom you want to search234
[%{origin: nil, retweet: false, tweet: "Hello1 hi1 #ee @234", user: 6},
 %{origin: 6, retweet: true, tweet: "Hello1 hi1 #ee @234", user: 996},
 %{origin: 6, retweet: true, tweet: "Hello1 hi1 #ee @234", user: 961},
 %{origin: 6, retweet: true, tweet: "Hello1 hi1 #ee @234", user: 966},
 %{origin: 6, retweet: true, tweet: "Hello1 hi1 #ee @234", user: 956},
 %{origin: 6, retweet: true, tweet: "Hello1 hi1 #ee @234", user: 976},
 %{origin: 6, retweet: true, tweet: "Hello1 hi1 #ee @234", user: 9},
 %{origin: 6, retweet: true, tweet: "Hello1 hi1 #ee @234", user: 48},
 %{origin: 6, retweet: true, tweet: "Hello1 hi1 #ee @234", user: 26},
 %{origin: 6, retweet: true, tweet: "Hello1 hi1 #ee @234", user: 40},
 %{origin: 6, retweet: true, tweet: "Hello1 hi1 #ee @234", user: 39},
 %{origin: 6, retweet: true, tweet: "Hello1 hi1 #ee @234", user: 45},
```


Subscriber Querying

```
robla@Robla-Lenovo-G50-80: ~/Elixir/project4
robla@Robla-Lenovo-G50-80:~/Elixir/project4$ ./project4 query
connected to clint
#PID<8382.77.0>
connected
*****
1.Mentions
2.Hashtag
3.Subscriber tweet
*****
what is ur choice 1 2 or 3 ?3
subscriber
Enter your name34
34
client 34 has tweeted: "Hello1 hi1 #ee @312" at 1512243301089
client 34 has tweeted: "Hello1 hi1 #ee @312" at 1512243301089
client 34 has tweeted: "Hello1 hi1 #ee @312" at 1512243301089
client 34 has tweeted: "Hello1 hi1 #ee @312" at 1512243301089
client 34 has tweeted: "Hello1 hi1 #ee @312" at 1512243301089
client 34 has tweeted: "Hello1 hi1 #ee @312" at 1512243301089
client 34 has tweeted: "Hello1 hi1 #ee @312" at 1512243301089
client 34 has tweeted: "Hello1 hi1 #ee @312" at 1512243301089
client 34 has tweeted: "Hello1 hi1 #ee @312" at 1512243301089
client 34 has tweeted: "Hello1 hi1 #ee @312" at 1512243301089
client 34 has tweeted: "Hello1 hi1 #ee @312" at 1512243301089
```

Performance Measurements

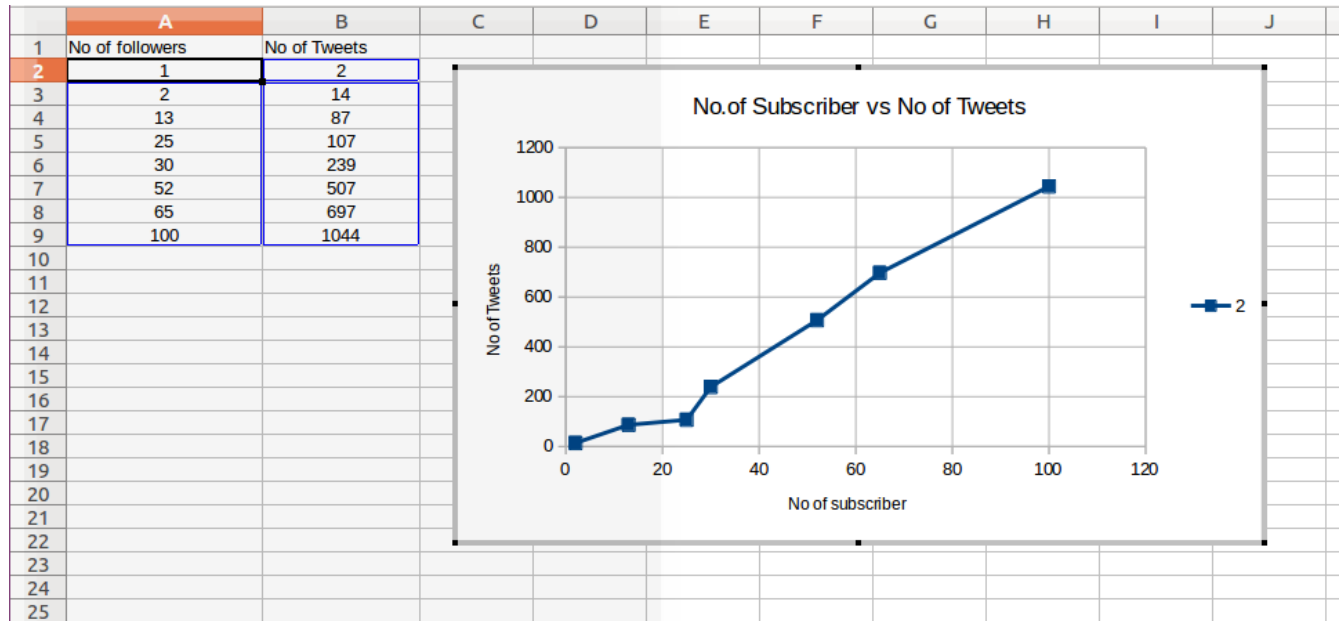
1. No.of users vs No. of tweets

Then number of tweets is showing an increasing trend when the number of users are increasing. This is because, more the number of user, the tweeting will be done more frequent. But there is a bottleneck that when the number of user increases beyond a certain threshold, there will be too much load in the server and hence the server will not be able to process tweets with the same frequency as before. This can be seen from the below table:

No of Users	No Of Tweets/second
100	4916
500	5903
1000	6183
5000	6483
10000	4756
15000	3962
20000	3348

2. No. of subscriber vs No. of tweets

This simulates a zipf distribution on the number of subscribers. More the number of subscribers or followers, more is the number of tweets tweeted.



```

client 29 has 3 subscribers , tweeted 30 so far
client 4 has 25 subscribers , tweeted 208 so far
client 21 has 5 subscribers , tweeted 42 so far
client 5 has 20 subscribers , tweeted 8 so far
client 79 has 1 subscribers , tweeted 10 so far
client 8 has 13 subscribers , tweeted 85 so far
client 4 has 25 subscribers , tweeted 209 so far
client 23 has 4 subscribers , tweeted 6 so far
client 4 has 25 subscribers , tweeted 210 so far
client 9 has 11 subscribers , tweeted 54 so far
client 91 has 1 subscribers , tweeted 9 so far
client 4 has 25 subscribers , tweeted 211 so far
client 4 has 25 subscribers , tweeted 212 so far
client 4 has 25 subscribers , tweeted 213 so far
client 46 has 2 subscribers , tweeted 19 so far
client 48 has 2 subscribers , tweeted 9 so far
client 21 has 5 subscribers , tweeted 43 so far
client 19 has 5 subscribers , tweeted 33 so far
client 29 has 3 subscribers , tweeted 31 so far
client 63 has 2 subscribers , tweeted 14 so far
client 41 has 2 subscribers , tweeted 5 so far
client 8 has 13 subscribers , tweeted 86 so far
client 4 has 25 subscribers , tweeted 214 so far
client 4 has 25 subscribers , tweeted 215 so far
client 69 has 1 subscribers , tweeted 14 so far
client 4 has 25 subscribers , tweeted 216 so far
client 38 has 3 subscribers , tweeted 5 so far
client 75 has 1 subscribers , tweeted 2 so far
client 4 has 25 subscribers , tweeted 217 so far
client 80 has 1 subscribers , tweeted 12 so far
client 21 has 5 subscribers , tweeted 44 so far
client 4 has 25 subscribers , tweeted 218 so far
client 8 has 13 subscribers , tweeted 87 so far
client 4 has 25 subscribers , tweeted 219 so far
client 29 has 3 subscribers , tweeted 32 so far
client 4 has 25 subscribers , tweeted 220 so far
client 64 has 2 subscribers , tweeted 13 so far
client 4 has 25 subscribers , tweeted 221 so far

```