Project 4
Project Report – Twitter Clone
Ammar Amjad 5992-1730, Mohammad Anas 5981-5998
November 14th, 2022

Task:
To implement a simulation of Twitter with clients, server, supervisor/tester and measure different performance metrics like time taken for tasks along with random client disconnection and reconnection i.e.. dropout.

What is Working?
- When the user is connected, deliver tweets live (without querying)
- Registration of user accounts,
- Send tweet with or without hashtags and mentions,
- Subscribing to user's tweets,
- Re-tweeting
- Allow querying tweets subscribed to, tweets with specific hashtags, tweets in which the user is mentioned.
- Simulate of live connection and disconnection of users.
- Zipf Distribution. The more the subscribers a user has, the more it tweets with respect to the Zipf distribution.
- A supervisor sends instructions to clients
- Client listens to both supervisor and server. Then acts according to the instructions from supervisor i.e.. To send tweet, retweet etc.
- Server executes commands from Clients and returns results.

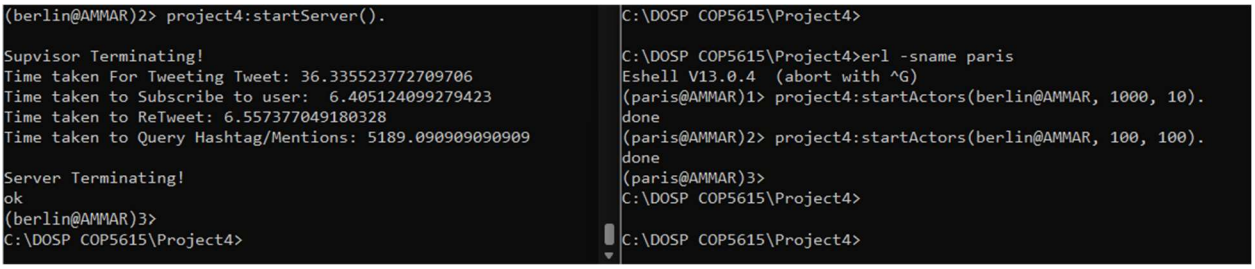
Steps for execution are given below:

Open up two terminals in the directory of code file.
Run the project by using commands in terminal:
-> erl -sname paris
-> c(project4).
-> project4:startServer().

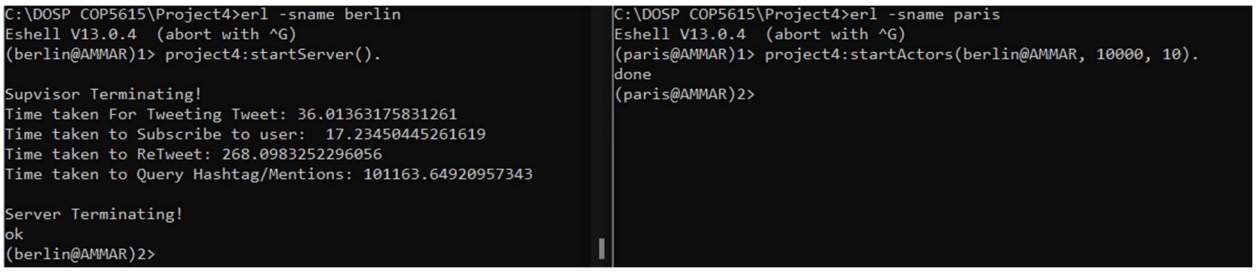
Now in 2nd terminal type.
-> erl -sname berlin
-> project4:startActors(paris@USER, NumActors, NumRequests)

where NumActors = number of Actors/Clients, NumRequests = number of requests sent by each actor/client.

Sample execution:
Of project4.erl : This shows time taken for each commands to execute on average.



What is the largest number of actors you managed to deal with for Twitter Clone?
For NumActors = 10000 and NumRequests = 10 for each actor.



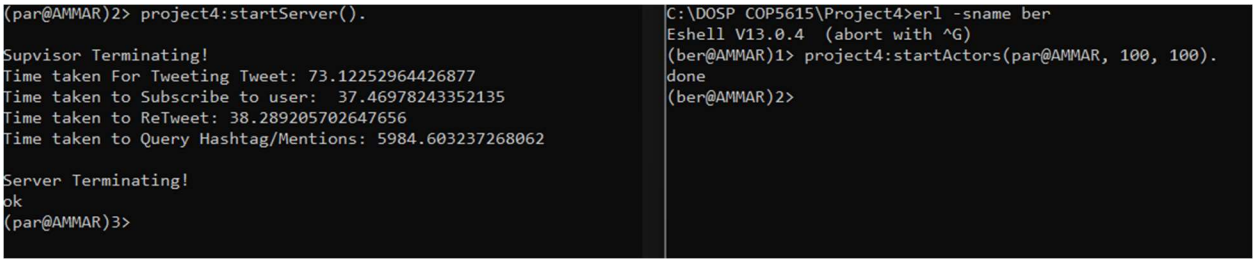
For NumReqeusts = 10 and NumActors = 10000 for each actor.

Dropout – Disconnection and Reconnection:

Disconnecting and reconnecting random actors leads to.

For NumReqeusts = 100 and NumActors = 100 for each actor.

With Dropout = 10%



Disconnecting and reconnecting random actors leads to.

For NumReqeusts = 100 and NumActors = 100 for each actor.

With Dropout = 50%

```
(par@AMMAR)2> project4:startServer().

Supvisor Terminating!
Time taken For Tweeting Tweet: 55.20751761942052
Time taken to Subscribe to user:  37.124802527646125
Time taken to ReTweet: 101.04669887278584
Time taken to Query Hashtag/Mentions: 6390.946502057613

Server Terminating!
ok
(par@AMMAR)3>
```

```
Eshell V13.0.4  (abort with ^G)
(ber@AMMAR)1> project4:startActors(par@AMMAR, 100, 100).
done
(ber@AMMAR)2>
```

Total time taken to execute code is increased, the higher the dropout value is. This is due to the fact that if a lot of actors are unresponsive, a lot of messages will not be delivered until they come back online, thus incurring extra overhead and delay.

The zipf Distribution:

Formula:

$$f(x) = \frac{1}{x^{\alpha} \sum_{i=1}^{n} (1/i)^{\alpha}} \qquad x = 1, 2, \ldots, n,$$

Sample Demonstration:

alpha	x	x^alpha	i	1/i^alpha	sum	product	Formula	Probability	
1	1	1	1	1	3.59774	3.59774	0.277952	27.79523	
	2	2	2	0.5		7.195479	0.138976	13.89761	
	3	3	3	0.333333		10.79322	0.092651	9.265077	
	4	4	4	0.25		14.39096	0.069488	6.948807	
	5	5	5	0.2		17.9887	0.05559	5.559046	
	6	6	6	0.166667		21.58644	0.046325	4.632538	
	7	7	7	0.142857		25.18418	0.039707	3.970747	
	8	8	8	0.125		28.78192	0.034744	3.474404	
	9	9	9	0.111111		32.37966	0.030884	3.088359	
	10	10	10	0.1		35.9774	0.027795	2.779523	
	11	11	11	0.090909		39.57514	0.025268	2.526839	
	12	12	12	0.083333		43.17288	0.023163	2.316269	
	13	13	13	0.076923		46.77062	0.021381	2.138095	
	14	14	14	0.071429		50.36836	0.019854	1.985374	
	15	15	15	0.066667		53.96609	0.01853	1.853015	
	16	16	16	0.0625		57.56383	0.017372	1.737202	
	17	17	17	0.058824		61.16157	0.01635	1.635014	
	18	18	18	0.055556		64.75931	0.015442	1.544179	where x are subscribers
	19	19	19	0.052632		68.35705	0.014629	1.462907	alpha controls shape of curve
	20	20	20	0.05		71.95479	0.013898	1.389761	<- Drag this line down

Note: x here is the rank of actors based on subscribers.

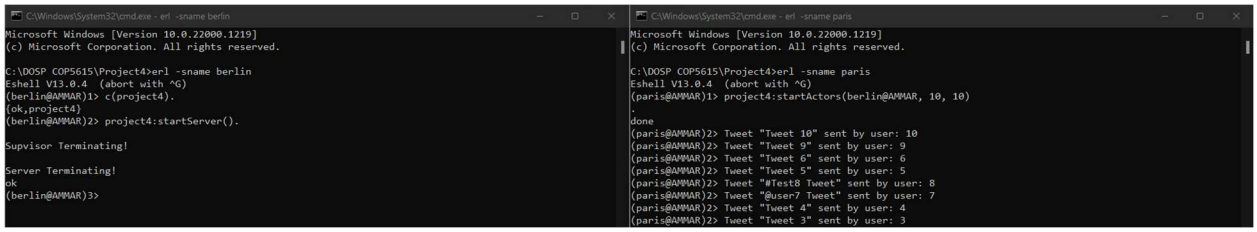
Here probabilities sum upto 100 which means the formula is working as intended.

Using the formula, The Actors with more subscribers send more tweets.

To view logs:

Sample execution:

Of project4Logs.erl : This shows the logs of commands executed.



To view Logs Execute following commands in code directory:
erl -sname w1
c(project4Logs).
project4Logs:startServer().

In 2nd terminal:
erl -sname w2
Project4Logs:startActors(w1@USER, NumActors, NumRequests)

Conclusion:

Number of Actors	Number of Requests per Actor	Average time to tweet (ms)	Average time to subscribe (ms)	Average time to Retweet (ms)	Average time to search by hashtags or mentions (ms)
10	10	36.3	6.4	6.5	5189
50	50	48.4	25.3	23.4	5532
100	100	73.6	37.1	38.4	5984

The time taken to tweet, retweet, subscribe and query hashtags or mentions is directly proportional to the number of clients and the number of requests per client.

Furthermore, increasing dropout ie. Rate at which users disconnect and reconnect, also increases total time taken for execution.

Number of Actors	Number of Requests per Actor	Dropout	Average time to tweet (ms)	Average time to subscribe (ms)	Average time to Retweet (ms)	Average time to search by hashtags or mentions (ms)
10	10	10	73.4	37.4	38.4	5984
100	100	50	75.6	37.1	101.3	6390