DOS Project 4: Twitter Clone Part 1

Team Members

1. Aaditya Prakash - 54154358

2. Prasann Pradeep Patil - 66037075

Status

Basic requirement: Finished.

1. SERVER

   Serves backend for our processes. The remote requests and client processes are handled by the server. Akka cluster creates a TCP cluster. To simulate multiple threads, we have created 10 instances of server and actors to handle incoming requests
   Register, Follow, Login, Tweet, Retweet reply, and Query are the functionalities.

2. CLIENT

   The client is our front end. Here we simulate the request that the user will be making. We use AKKA framework to create TCP client to connect to server component

Instructions to run the simulation

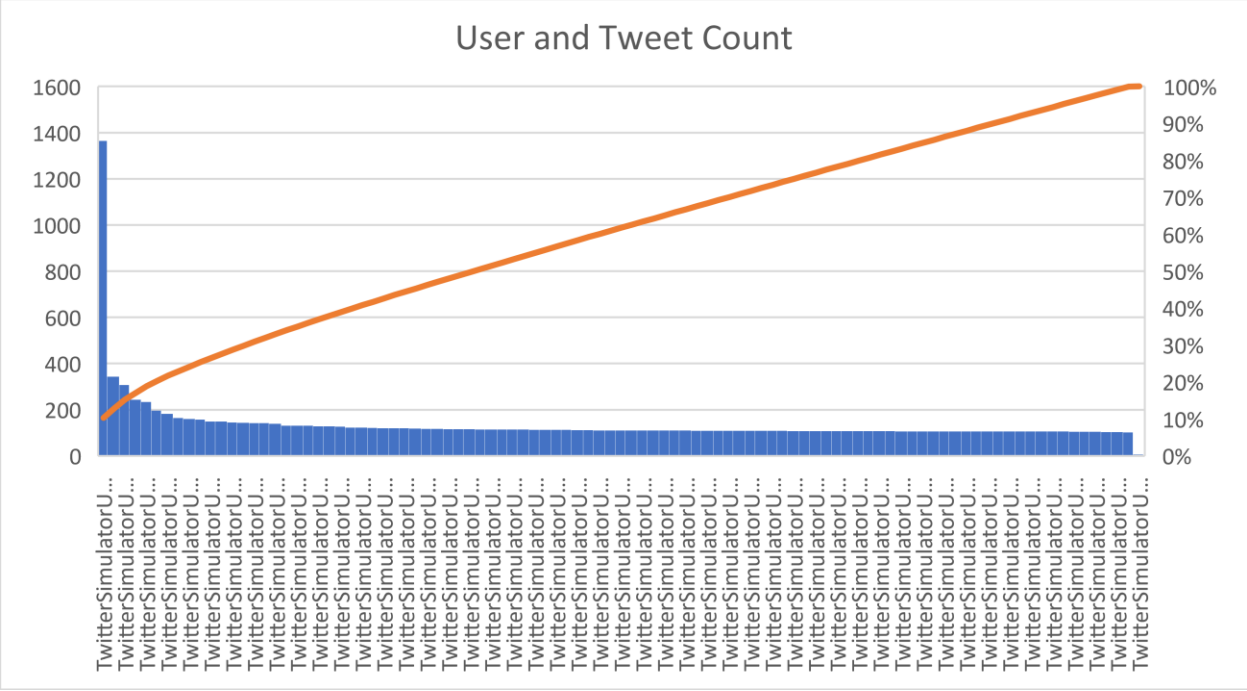
1. Run Server file with command **dotnet run**.
2. Run Client file with command **dotnet run <no. of users>**

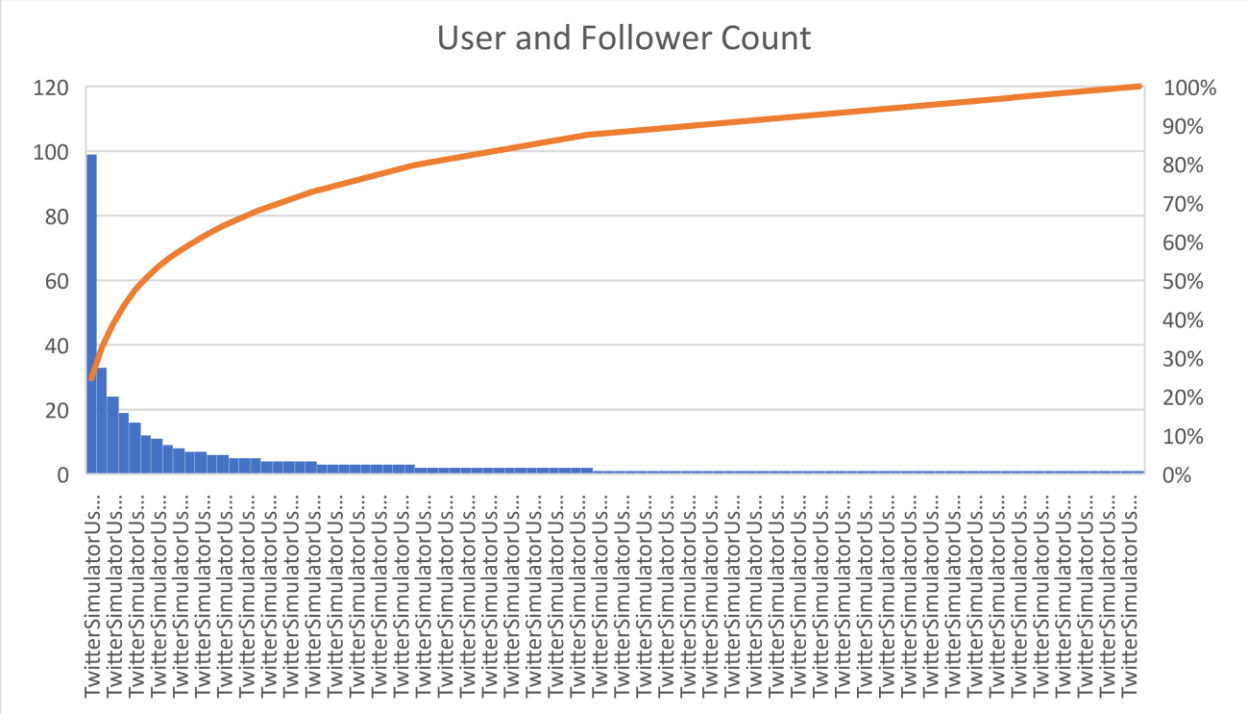
Output

When running for 100 users, the recorded **performance statistics** were

```
 Performance Stats
Number of requests per second: 32
Average response time for Tweet requests: 30
Average response time for Query to fetch mentioned tweets: 14
Average response time for Query to fetch hashtag tweets: 6
```

Zipf Distribution: - (100 Users)





Performance

Number of Users	Requests/Second	Tweet (ms)	Query for Hashtag (ms)	Query for Mentions (ms)
100	2050	4	1	1
200	1300	5	3	2
500	860	13	25	20
800	490	30	83	78

The above table shows the performance of different services in our simulator for different number of users.

1. Request/Second – The number of requests processed by the server component in one second.
2. Tweet – The average response time for TWEET requests in milliseconds.
3. Query for hashtag – The average response time for the QUERY to get the tweets with specific hashtags.
4. Query for mentions – The average response time for the QUERY request to get all the tweets that a user is mentioned in.

Note that we have rounded up the response time to have the integer values instead of float.

Conclusion

The performance of simulation is inversely proportional to the number of users. This is evident from the above table. When we changed the number of users from 100 to 800 the requests processed per seconds were reduces. The average response time of various services also increased with increase in user numbers.