

Distributed Operating system and principles

Twitter Clone Part 2

Problem Statement:

In Part 1 implementation, we made a Twitter clone project using AKKA messaging, which allows client-server implementation with functionalities like register, login, tweet, and subscribe. Now we need to upgrade our Part 1 implementation using Web Sharper web framework to design a proper WebSocket interface, including designing a JSON-based API that represents all messages and their replies along with errors. Along with this we also need to re-write some parts of the engine as well as the client Using web sharper to use Web Sockets.

Implementation:

In this project, we implemented a Twitter clone using the web socket. We have used Cowboy web socket, which uses ranch protocol. We store data using the ETS tables. Below are the steps to run the Twitter Clone. A call-back interface is also defined by cowboy WebSocket for managing WebSocket connections. JSON-type data gets printed on the server every time we send a tweet, subscribe or mention.

- 1) Open erlang server using the `erl -sname "Your name"`.
- 2) Run both the erlang files with the code `c(server).` and `c(client).`
- 3) Start the server with the command `server:start()`.
- 4) Open two other terminals and start the client-server with the command `client:start()`.
- 5) Then you can post a tweet with the command `tweet`, and then it asks you to type a message to post.
- 6) Also, we can subscribe to our friends with the `subscribe` command.
- 7) We can mention friends with the `"@name"` command and hashtag.
- 8) All the users can search for hashtags and mentions with the `query` command.

In Part 2, we have updated the client server implementation using web socket interface. Once we start the server, it starts listening on port 8080 using cowboy WebSocket. Once the client is started using gun, it sends all the requests to the web socket. This is where, the connection is established between client and server using client server communication. We also designed a JSON based API which displays all the messages and their replies along with error messages. Now, for example, if the client wants to register to the application, then once the connection is created between client and server, it creates a JSON object containing two keys one is username as input and the other is query as register. Which is then encoded as a JSON object and send to the server using web sockets. Once it is sent to the server, it is decoded into a JSON object which stores the required registration details in ETS table. Similarly, all the

remaining options like tweet, retweet and subscribe are handled by client and sent to the server and if there are any errors, then it sends back to the client.

Snapshots:

1. JSON Data displayed in the server.

```
Word to be searched: "#Vamsi"
Current Row key: "Yashas"
CurrentTweets: ["Hi im Yashas\n", "Hi @Vamsi #Vamsi\n"]
FilteredTweets: ["Hi @Vamsi #Vamsi\n"]
Word to be searched: "#Vamsi"
Found tweets: ["Hi @Vamsi #Vamsi\n"]

"Vamsi" wants to queryDo Receive

DATA: [<<"query">>,<<"Vamsi">>,<<"3">>,<<"Yashas">>]

TYPE: "query"

Query: The current username is -> "Vamsi"
Subscribed User Search
Sub_UserName: "Yashas"

"Vamsi" wants to queryDo Receive
```

2. Created a client server and registered a new user.

```
(base) vamsipachamatla@vamsis-MacBook-Air project-4-2 2 % erl -sna
me Vamsi
Erlang/OTP 25 [erts-13.1] [source] [64-bit] [smp:8:8] [ds:8:8:10] [async-t
hread:1] [jit] [dtrace]

Eshell V13.1 (abort with ^G)
(Vamsi@vamsis-MacBook-Air)1> client:start().

Hii, I am a new client

Just sent my request to the server

Received message from server
YIP
Enter the command: register
register
Enter the User Name: Vamsi
SELF: <0.91.0>

Account has been Registered

Received message from server

User has been registered
```

3. In the query command, you check mentions, hashtags and subscribed tweets.

```
Enter the command: query
query
Querying Options:

1. My Mentions

2. Hashtag Search

3. Subscribed Users Tweets

Specify the task number you want to perform: 1

Received message from server

New tweet received!
Yashas:Hi @Vamsi #Vamsi
```

4. Subscribing to the other user.

```
Enter the command: login
login
Enter the User Name: Vamsi
SELF: <0.91.0>

Account has been Signed in

Enter the command: subscribe
subscribe
Who do you want to subscribe to?:Yashas

Subscribed!

Received message from server

Subscribed!
Enter the command: tweet
tweet
What's on your mind?:Hi im Vamsi

Tweet Sent

Received message from server

Your tweet has been sent
Server processed tweet

Received message from server

New tweet received!
Yashas:Hi im Yashas
```

Video Link:

https://youtu.be/6oOt_VsntSc