

Distributed Operating Systems, Project 4 Part II: Twitter Simulator

Date: December 13, 2017

Group Members:

1. Sanket Achari, UFID: 71096329, sanketachari@ufl.edu
2. Sushmit Dharurkar, UFID: 14969578, sushmitdharurkar@ufl.edu

Instructions to RUN:

1. Make sure epmd daemon is running. Run `epmd -daemon`
2. Run following commands from the directory which has `mix.exs`
`mix deps.get`
`mix phx.server`
3. After printing of user details is completed, enter 'ctrl + c' to stop the execution.

Working:

1. Simulation of Twitter Engine.
2. Tweet, Retweet, Subscribe to tweets, hashtag, mention, live connection
3. Query hashtags, mentions
4. Use of WebSockets for communication between server and client.

Simulation:

100 active users each tweeting at least 1 tweet:

Time Required: 50s

Sample Outputs:

Example 1: Considering 100 live users

Total tweets with hashtag #DOS : 27

Tweets (Showing Max 10):

```
["Another tweet, study #DOS and do #projects @user36 @user23",  
"Another tweet, study #DOS and do #projects @user65 @user87",  
"Another tweet, study #DOS and do #projects @user87 @user26",  
"Another tweet, study #DOS and do #projects @user82 @user69",  
"Another tweet, study #DOS and do #projects @user11 @user89",  
"Another tweet, study #DOS and do #projects @user95 @user40",  
"Another tweet, study #DOS and do #projects @user41 @user3",  
"Another tweet, study #DOS and do #projects @user14 @user75",  
"Another tweet, study #DOS and do #projects @user37 @user35",  
"Another tweet, study #DOS and do #projects @user36 @user63"]
```

Total tweets with mention @user1: 0

Tweets (Showing Max 10):

```
[]
```

Printing user1 details:

```
%{"followers" => ["51", "42", "20"],  
"notifications" => %{"13" => ["This is #mytweet some tweet #sometweet @user34 @user86"],  
"20" => ["Another tweet, study #DOS and do #projects @user22 @user31"],  
"24" => ["Another tweet, study #DOS and do #projects @user82 @user31"],  
"29" => ["This is #mytweet some tweet #sometweet @user74 @user73"],  
"33" => ["This is #mytweet some tweet #sometweet @user18 @user17"],  
"38" => ["This is #mytweet some tweet #sometweet @user97 @user57"],  
"43" => ["Some random tweet #randomtweet #newtweet @user43 @user64"],  
"70" => ["This is #mytweet some tweet #sometweet @user74 @user66"],  
"77" => ["Some random tweet #randomtweet #newtweet @user56 @user15"]},
```

```
"retweets" => ["Another tweet, study #DOS and do #projects @user75 @user21"],
"subscribed" => [{"Some random tweet #randomtweet #newtweet @user68 @user80",
"7"}], "tweets" => [], "username" => "user1"}
```

Implementation Details:

1. We have used Phoenix Channels for communication between server and client. These channels internally implement WebSockets and transmit messages in JSON format.
2. We have used **phoenix_gen_socket_client** library to implement clients.

Client Calls to Gen Server through Engine Channel:

- a. **Add follower:** [debug] INCOMING "add_follower" on "engine" to TwitterWeb.EngineChannel
Transport: Phoenix.Transports.WebSocket
Parameters: %{"follower" => "71", "id" => "63"}
- b. **Get usernames:** [debug] INCOMING "get_usernames" on "engine" to TwitterWeb.EngineChannel
Transport: Phoenix.Transports.WebSocket
Parameters: %{}
- c. **Send tweet:** [debug] INCOMING "send_tweet" on "engine" to TwitterWeb.EngineChannel
Transport: Phoenix.Transports.WebSocket
Parameters: %{"hashtags" => ["#mytweet", "#sometweet"], "id" => "71", "isRetweet" => false, "mentions" => ["@user16", "@user81"], "tweet" => "This is #mytweet some tweet #sometweet @user16 @user81"}
- d. **Subscribe to tweet:** [debug] INCOMING "subscribe_to_tweet" on "engine" to TwitterWeb.EngineChannel
Transport: Phoenix.Transports.WebSocket
Parameters: %{"id" => "83", "tweetId" => "BA99555950ECD8DB0A756B35F842B08A0A4987D94E99CFF8A5859325E378A55F"}

Here, the parameters sent over the channel are in JSON format done internally by Phoenix so we don't need to explicitly convert them in JSON.

3. We have used Gen Server for twitter engine. This server is the single process which will keep the track of each & every user, tweet, subscription, retweet, follower, hashtag and mention.
4. This gen server acts as a backend for the twitter simulation.
Gen server's state is as following

State of Engine:

```
%{ "users" => %{id1 => true/false, id2 => true/false,...},
  "usernames" => %{ "username1" => id1, "username2" => id2,...}
  "tweets" => %{ "tweetId1" => {tweet1, id1}, "tweetId2" => {tweet2, id2}....}
  "user_details" => %{ id =>
    %{ "username" => "Unique name"
      "tweets" => [],
      "retweets" => [],
      "followers" => [],
      "notifications" => %{source_pid => [tweet1, ...]},
      "subscribed" => [tweetId1, tweetId2, ....]
    }
  }
}
```

```

"hashtags" => %{ "hashtag1" => %{id => [tweet1, tweet2, ...]}, ...}
"mentions" => %{ "mention1" => %{ "username1" => [tweet1, tweet2, ...]}, ...}
}

```

In this state table, we maintain collection of hashtags, mentions, total tweets, users, usernames and each user's details. Each user has status true or false which indicates **online** or **offline** status.

5. Initially we create total number of users. After this we add followers for each user randomly.
6. We have randomly kept all the users' status as online or offline. In either case of tweet or retweet, we update the state table of twitter engine.
7. Also, each user can subscribe to a random tweet. And this subscription is also random based on *Enum.random([true, false])*
8. Then each user can send tweet and retweet randomly. We have taken 3 sample tweets as follows:

```

tweet1 = "Some random tweet #randomtweet #newtweet @randomuser1 @randomuser2
tweet2 = "This is #mytweet some tweet #sometweet @randomuser1 @randomuser2
tweet3 = "Another tweet, study #DOS and do #projects @randomuser1 @randomuser2

```

Note that, we have total 6 hashtags:

#randomtweet, #newtweet, #mytweet, #sometweet, #DOS, #projects

And 2 mentions:

@randomuser1, @randomuser2

These mentions are usernames of 2 users which are selected randomly.

In this simulation, we have considered above 3 tweets with hashtags and mentions. Out of these 3 tweets 1 tweet will be selected randomly and that tweet is broadcasted to the followers. Retweet is also based on the same logic.