

Twitter Clone with WebSockets

Project 4 Part 2

Team Members:

Kasiviswanathan Srikant Iyer (UFID: 5222-2519)

Swaathi Reena Velavan (UFID: 1230-8520)

YouTube Link - https://youtu.be/NuL_K4zxqx8

Problem Statement -

A JSON based API has been designed. Project 4 Part 1 has been rewritten using WebSharper to implement WebSocket interface.

Encryption and Decryption has been implemented using AES (Advanced Encryption Standard) along with the Diffie Hellman Protocol.

All functionality required by the problem statement has been implemented.

To run the code -

There are three files to run -

- server.fs
- client.fs
- performance.fs

The code can be run on JetBrains Rider or Visual Studio. We require the following nuget dependencies in order to run the code.

server.fs -

FSharp.Core 5.0.0
WebSharper 4.7.2.445
Akka.FSharp 1.4.29
Akka.Remote 1.4.29
WebSharper.Suave 4.7.1.280

client.fs -

FSharp.Core 5.0.0
Akka 1.4.29

Akka.FSharp 1.4.29
Akka.Remote 1.4.29
FSharp.Data 4.2.6

performance.fs -

FSharp.Core 5.0.0
Akka 1.4.29
Akka.FSharp 1.4.29
Akka.Remote 1.4.29
FSharp.Data 4.2.6

Implementation -

The client should enter a command of the following format - new is used to register a new user, tweet is used to publish a tweet, follow indicates a user subscribing to another and twitterfeed is used to query the tweets of an user. # is used to query hashtags and @ is used to query mentions. Examples can be seen below.

Supposing this is the command written in client.fs,

```
new,abc,!23456Abc
```

This command is stored as a JSON string.

```
{"command":"new,abc,!23456Abc"}
```

Further we encrypt the value of the command and store it in the encrypted JSON string.

This is decrypted by server.fs.

We use the **AES encryption API** along with the diffie hellman protocol as required.

The following commands are a example of commands that can be used in the simulation -

```
new,abc,!23456Abc  
new,xyz,BUBBLES  
tweet,xyz,BUBBLES,message1 @Adam  
tweet,abc,!23456Abc,message1 @Adam  
tweet,abc,AAAAAAA,message1 @Adam  
follow,xyz,BUBBLES,abc  
twitterfeed,xyz,BUBBLES
```

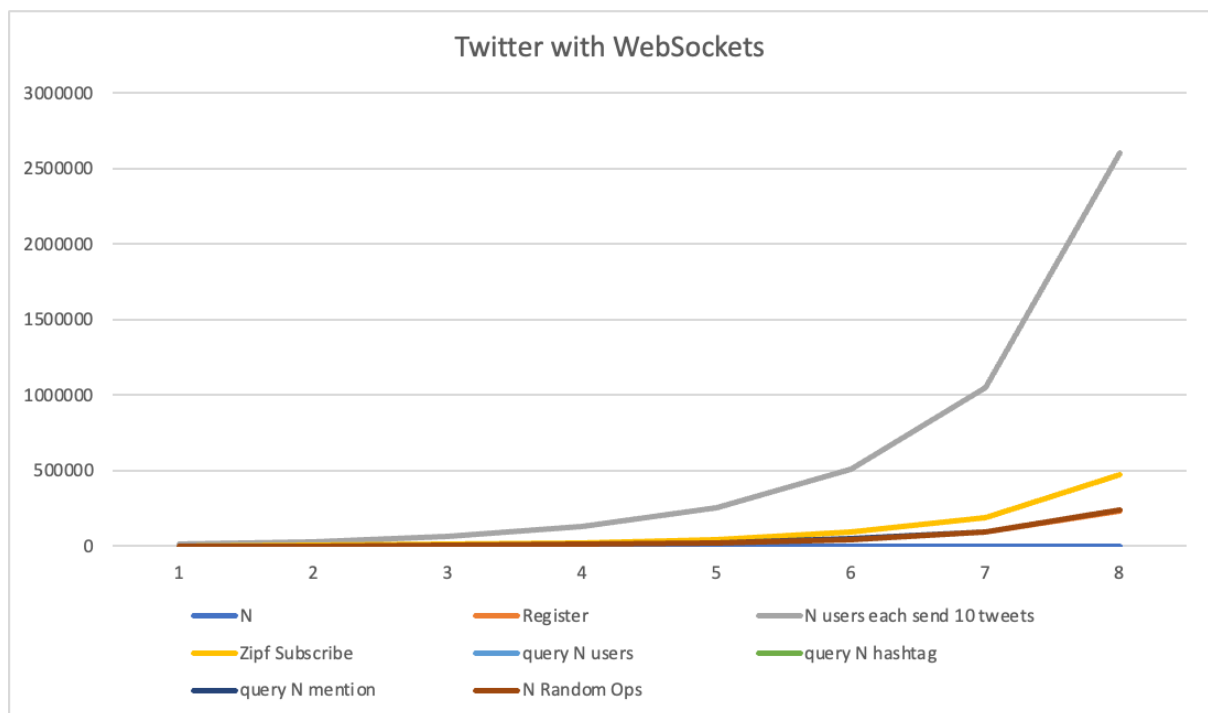
#,#UFL
@,@Adam
retweet,abc,!23456Abc,message3 @Adam #UFL

Largest Network Dealt with -

The largest network that we handled was 1000 users. The program got significantly slower after we added encryption and decryption.

Graphs -

Time taken by users to perform operations -



Analysis -

From the above graph, we see that the time taken for N users to send tweets takes the most time, as it involves a lot of string processing. Zipf subscribe stands next in line in time consumption, as it requires $2*N$ operations.