Millstone -3

(TCP/UDP) iterative server

By: Sarah Alyousefi, Ryan Bomalaski

Instructor: Marius Silaghi

1. Introduction:

In millstone 2 we built a iterative server that recognize commands over TCP and UDP. In this millstone we will build a client that can communicate with our server. Also, we will change to a concurrent server.

2. Objective:

In this millstone we implemented a client program that allows users to send gossip messages to known peers, and to add addresses of peers. Options that must be supported by the client are:

- -m: "Tom eats Jerry"'.
- -t :"2017-01-09-16-18-20-001Z"';
- -T: or -U for TCP or UDP; and '
- -s: "183.116.10.43"';
- port : '-p 2334'

Also, in this millstone we changed the iterative server to concurrent one. Furthermore, after the server connected to the client it should foreword the messages to the peer witness server that running in another terminal.

3. Technology

For this project, we have used below technologies:

- C and C++ languages
- SQLite for our database
- Bash script to compile our server and client

4. Architecture:

The following figure explains the main architecture:

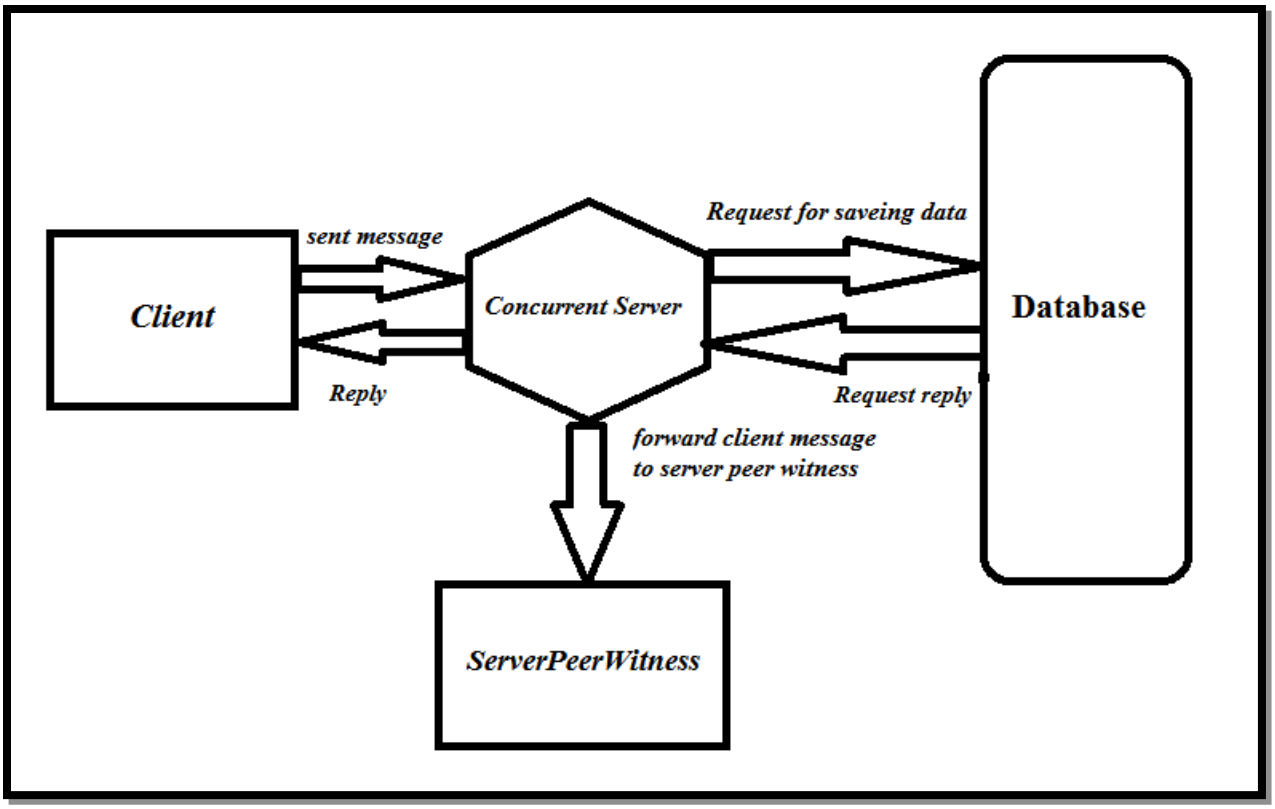

Figure (1.1) Server client communication mechanism

5. User manual:

- To run the server , first compile with compile.sh.
- Then run_server.sh with the appropriate port number
- Then you can type ./client and use the following options:
  - -p xxxx Assigns for the port number.
  - -h HostName (Default is 'localhost').
  - -U uses UDP instead of the TCP.
  - -T uses TCP instead of the UDP.
  - -m <Message> Assigns for messages
  - -s<Address> Assigns for the Address for example: 183.116.10.43.
  - -t<MassgesTime> Assigns for the time for example: 2017-01-09-16-18-20-001Z."
- Or you can use run.sh and specify a port to run the client on

6. Conclusion:

In this millstone we built a client program that communicate with a concurrent server and allow users to send gossip messages to known peers, and to add addresses of peers. Also we made the server able to forward the client message to a third terminal which is Server Peer Witness.

5. References:

[1]:http://cs.fit.edu/~msilaghi/Sp2017/2017sprNPG/Projects2015/P3/elmane_shakr/

[2]: http://cs.fit.edu/~msilaghi/pages/texts/examples/hallo/
[3]: http://www.zedwood.com/article/cpp-sha256-function
[4]: http://www.cplusplus.com/