Assignment 4

Abhisek Panda

December 25, 2020

1 Problem Statement

Below is the list of deliverable

- A simple hello message from server with client ID.
- Request from Client for index.html from the server
- Simultaneously connect multiple users to the server.
- Create more than max_connection_threshold connections, and on overflow close connections in FIFO order.
- Transmission Time Comparison of various size of data on both HTTP 1.0 and HTTP 1.1 connections.
- Build a chat application using the multi-server.
- Design a bidding game using the multi server to test the thread synchronization.

2 Implementation

1. Server Setup
   - Initially we create a socket on port 8888, so that the client can connect to the server using the accept system call and can use send and recv system calls for communication with the server.
   - We have a request queue in the server which stores the client's socket id and also stores the associated thread id of the connection.
2. Handle of Requests
   - When a request is served from the server, Initially we check the request line and parse the method, request uri and the http version.
   - Based on the HTTP version, the client socket close operation is handled. Similarly we serve the GET, POST and HEAD requests in the server.
   - In case a requested uri doesn't exists we send 404 Error Response.
   - The server has a designated max_connection_threshold value, if it is reached then we terminate the connections in FIFO order.
3. Chat Application
   - The user needs to provide the name, and then he/she may connect to the chat room where they can chat. They can view other people's messages in the format <sender : message>.
   - The user needs to connect to /chat.html relative URI and then we have set up a timer function which polls data from the server in every 1 second and whenever the user sends a message we issue a post request towards the server.

- The chats are saved in the text file. So that the user can get the messages whenever he/she joins to chat room again.

4. Biding Game
   - The user needs to connect to /index.html and there we have 2 boxes one to take the bid value from the user and another to show the highest bid value among the people who had bid.
   - The server has a global variable which gets changed by the bid value of the user and we finally shows the highest bid value.

3 Use Cases

Below are the use cases of the software:

1. Server: To start the HTTP server, we perform below steps in project directory:

```
$make
$pwd
$export LD_LIBRARY_PATH=<O/P of pwd>/lib
$make server
```

2. connoverflow: Make multiple client requests to the server, where number of requests is greater than the max_connection_threshold value perform below steps in project directory:

```
$make
$pwd
$export LD_LIBRARY_PATH=<O/P of pwd>/lib
$make connoverflow
```

3. client1: To send a HTTP1.0 request from client, perform below steps in project directory:

```
$make
$pwd
$export LD_LIBRARY_PATH=<O/P of pwd>/lib
$make client1
```

4. client1_1: To send a HTTP1.1 request from client, perform below steps in project directory:

```
$make
$pwd
$export LD_LIBRARY_PATH=<O/P of pwd>/lib
$make client1_1
```

5. plot: To view the transmission Time comparison graph of various size of data on both HTTP 1.0 and HTTP 1.1 connections. , perform below steps in project directory:

```
$make
$make plot
```

4 Test Cases

To test the correctness of the software, below tests where done:

- Make a HTTP-1.0 request towards the server.
- Make a HTTP-1.1 request towards the server.
- Make multiple connections towards the server.
- Test of Chat Application using the designed multi-server.
- Test of Biding Game using the designed multi-server.

5 Observation

We have drawn the below observations about the server:

1. When we where testing the server we have observed that the time of data transmission from the server to client is more in HTTP-1.0 compared to HTTP-1.1. This is due to the fact that in HTTP-1.0 we create a new connection every time we serve a request, which also includes the thread creation overhead and the connection setup overhead.

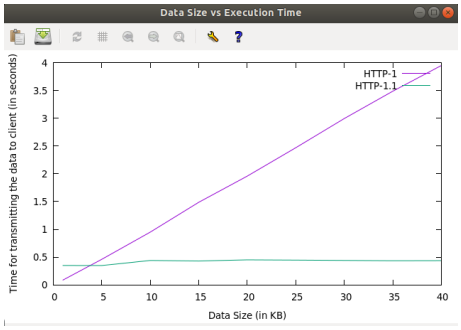

Figure 1: Transmission Time of Data in HTTP-1.0 and HTTP-1.1

6 Cites

1. Socket programming : http://beej.us/guide/bgnet/
2. https://tools.ietf.org/html/rfc1945
3. https://tools.ietf.org/html/rfc2616