**COMPUTER NETWORKS**

**HTTP & PROXY SERVER PROJECT REPORT**

We developed our http server in python with socket, sys, os, datetime and thread packages. We create thread for every request. We write create\_headers\_errors(status, filetype, size) method that takes status, filetype, size and create header and error status and if status is 200 return headers otherwise, method returns headers with error status.

create\_response(request) is method which takes request such that GET http://localhost:8080/500 HTTP/1.0 and splits the request by spaces and take request code such that GET, POST etc. and URI of request without ‘/’. After separating the request, Method checks the size of request. Size bounds should be between 100 and 20000 for HTTP server. If size bound is okay, method search the ‘ctl’ file to return chai tea latte recipe to client as a response with desired length. After that, method checks the request for errors and call create\_headers\_errors() with error codes.

server\_thread(cc) is method which take connection client object. Firstly, method takes request of client as 1024 byte. Method calls reg.decode() that converts binary code to string and method prints first row of request. After that, Method calls create\_response() with response and split by ‘\r\n’ and print first row of response .Method converts response from string to binary code with response.encode() call and send the response with calling sendall() method on connection client object. After that, method unlocked the thread with release call. Finally, method closes the connection.

We write main function to manage the all process. Firstly, method checks the command line argument if user enters special port number to connection, method accepts this port number to connect. If user doesn’t enter anything, default port number is 8080 for our http server. We assign host number is '127.0.0.1'. After that, we use socket programming with socket.socket() method that takes socket.AF\_INET, socket.SOCK\_STREAM parameters .These parameters is used for socket type IPv4 option and TCP connection. Another method for socket programming is setsockopt that is set the options of socket with reusable port. socket.bind() is used for binding socket to host and port. Finally, we used socket.listen() to listen requests that comes to socket.

We provide client connection to server with socket.accept(). We locked the thread after connection to server. We call start\_new\_thread(server\_thread, (client\_connection,)) from thread package .This method creates new connection with request.

For proxy server implementation, we use socket, hashlib, thread packages. We create thread for every request. We use the same method that is create\_headers\_errors(status, filetype, size). This method returns different error codes that is only difference from http server.

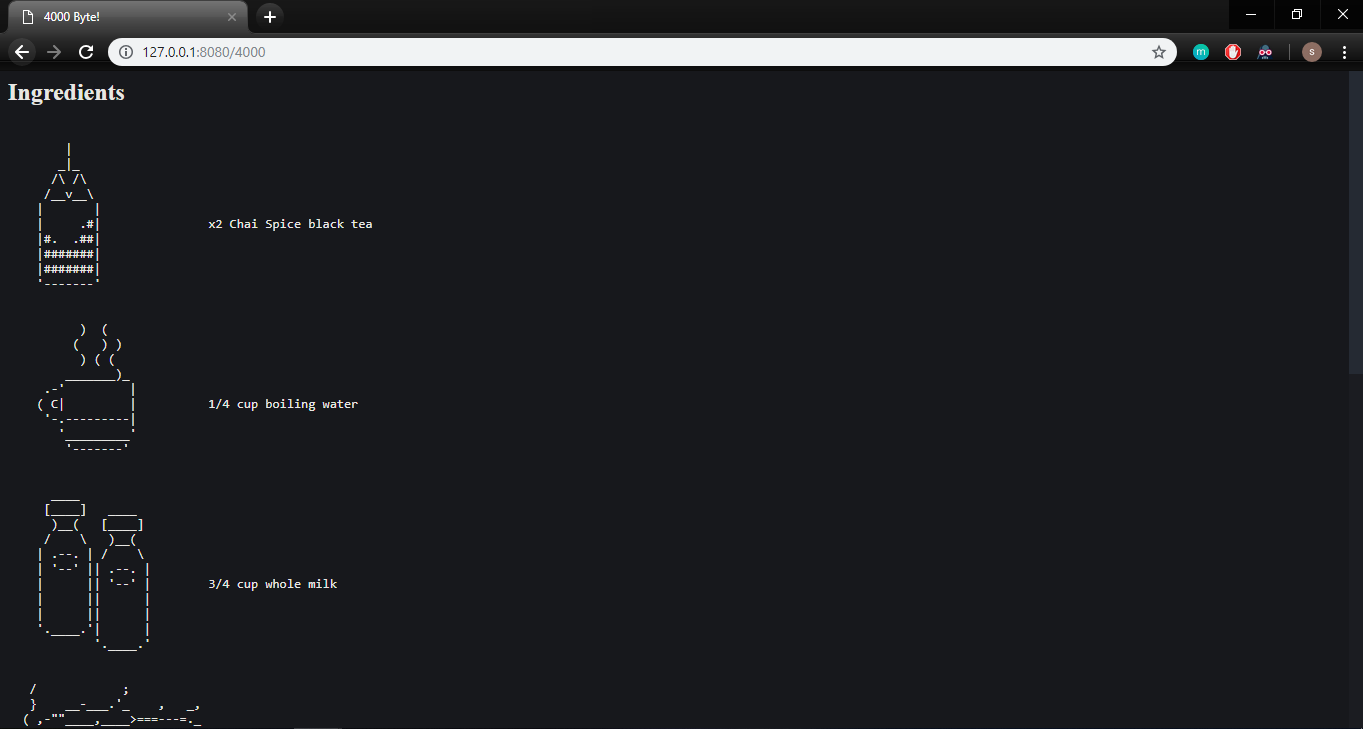
We write create\_response(request) method again with some differences. In this method, we split the request by space and we get the request code from split request. If code is GET, methods checks the URI to be digit only. If size is less than 9999, method selects the encryption type according to hashlib.md5(). Method encrypts the URI according to size to hold cache name. With this cache name, method checks the existence of this cache in the path. If cache exists, method opens cache file and send the data from cache file. If cache is miss, methods connects the http server and get data from server and write to new cache file and send to client.

All the process is built with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) , connect() and sendall() method calls. Here, if request is done from proxy server directly, method redirect the request to http server as default.

We use same main() and server\_thread(cc) logic in Proxy server code.

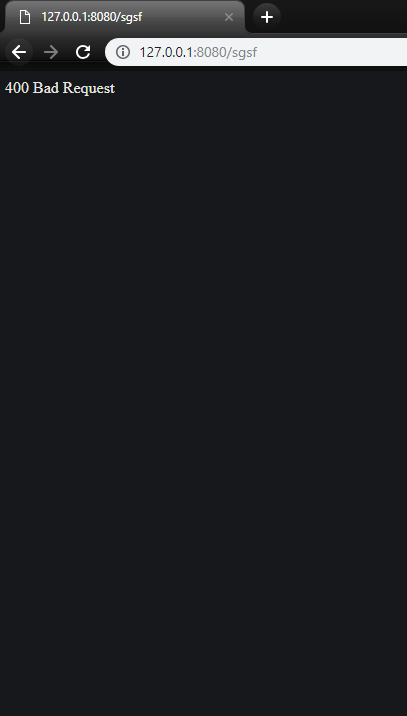
Screenshots from execution process:

## 1. 200 OK



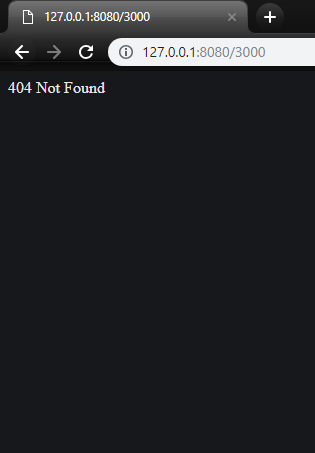
## 2. 400 Bad Request

We try to enter URI with non-digit.



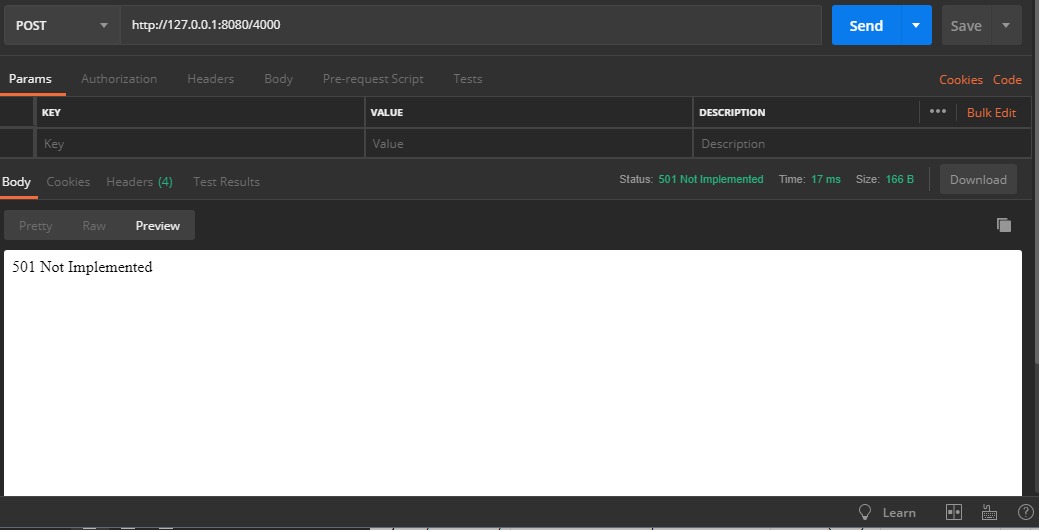
## 3.404 Not Found

We try to reach http server from Proxy server when http server is closed.



## 4. 501 Not Implemented

We send **post** request to http server.



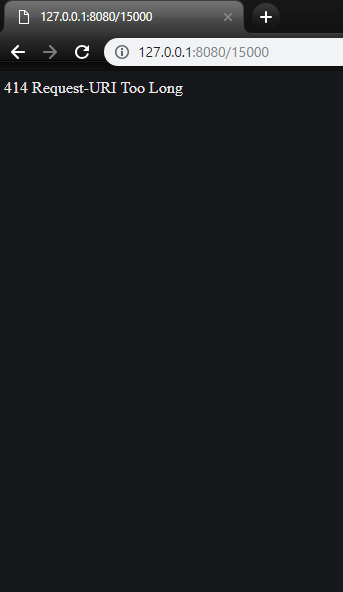
## 5.400 Bad Request

We try to enter under 100 length.

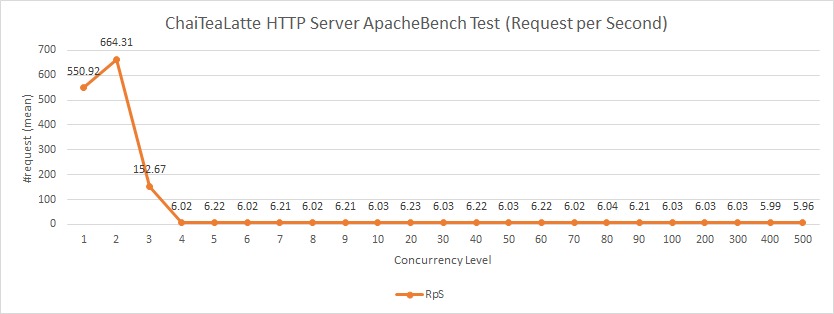


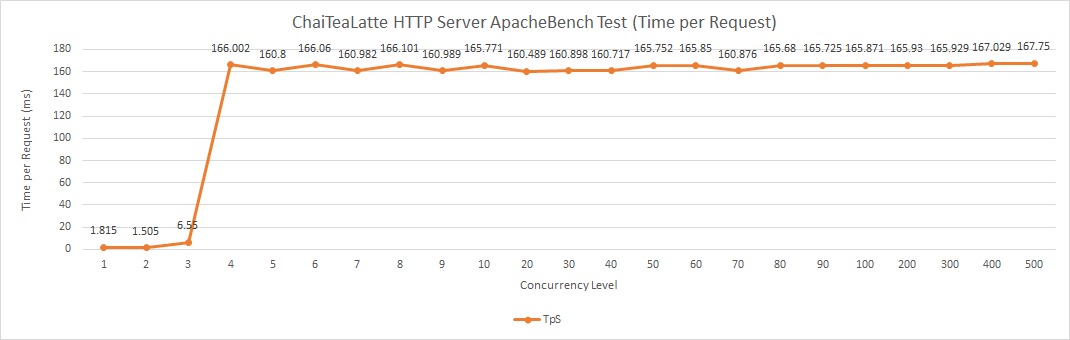
## 6. 414 – URI Too Long

We try to enter 15.000 above Proxy size bound

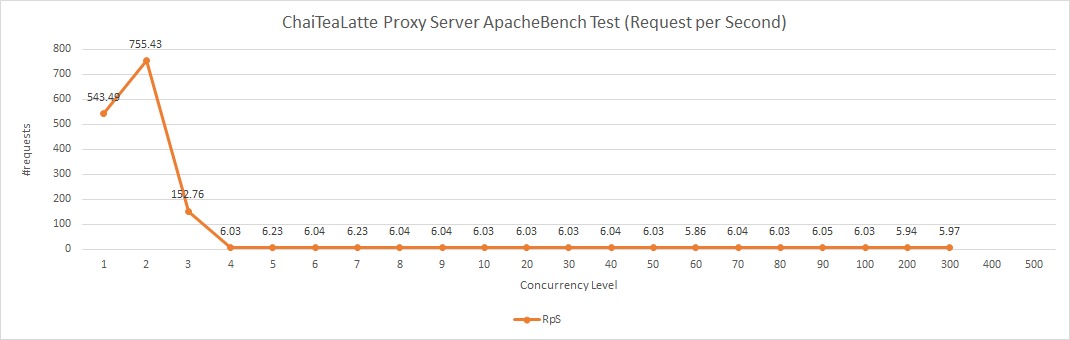


# Apache Bench Test Results:

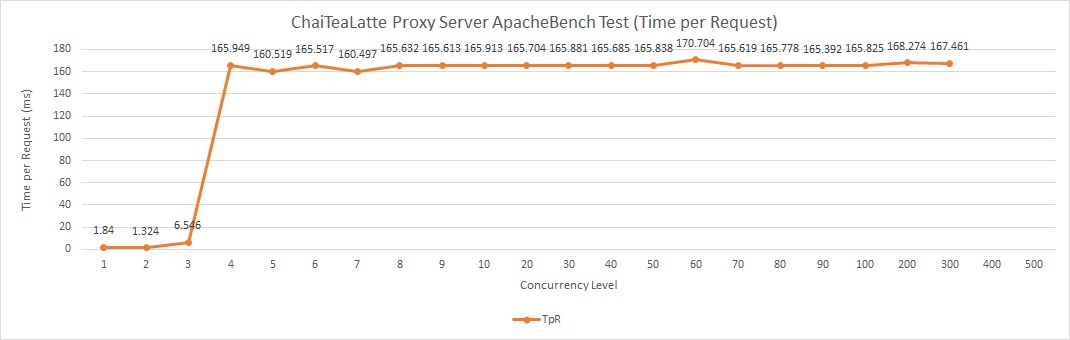


We applied Apache Bench stress test to Http server. When concurrency level is less than 4, performance of http server is very high. Performance of http server is dropped almost 6 requests after 4 concurrency level.

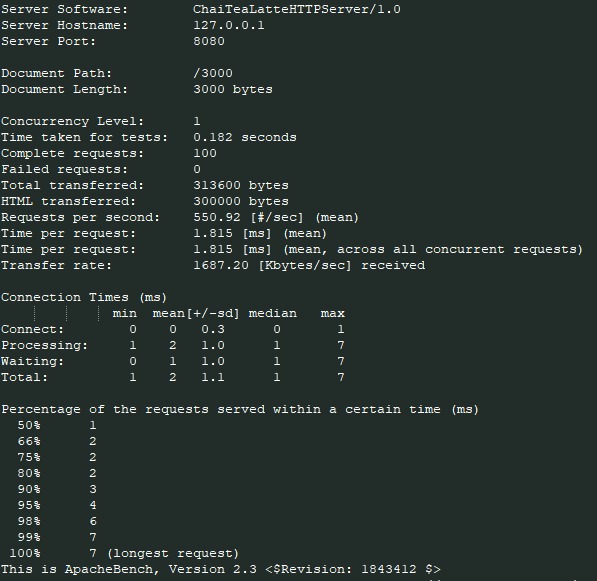
When concurrency level is less than 4, performance of http server is very high. Performance of http server is dropped almost 6 requests after 4 concurrency level.



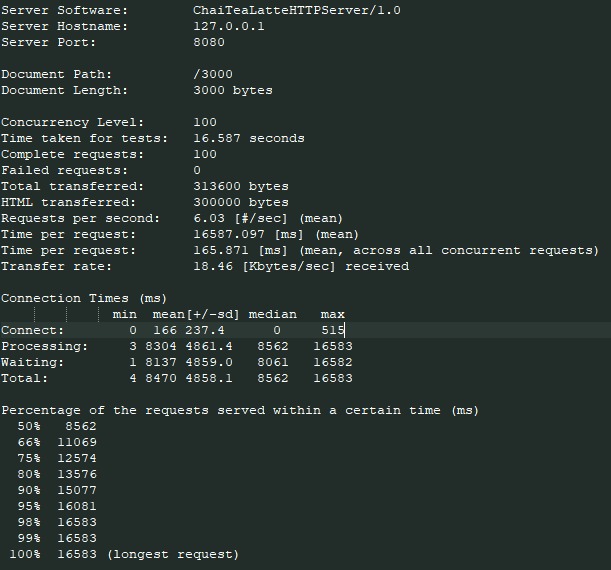
When concurrency level is less than 4, performance of http server is very high. Performance of http server is dropped almost 6 requests after 4 concurrency level.



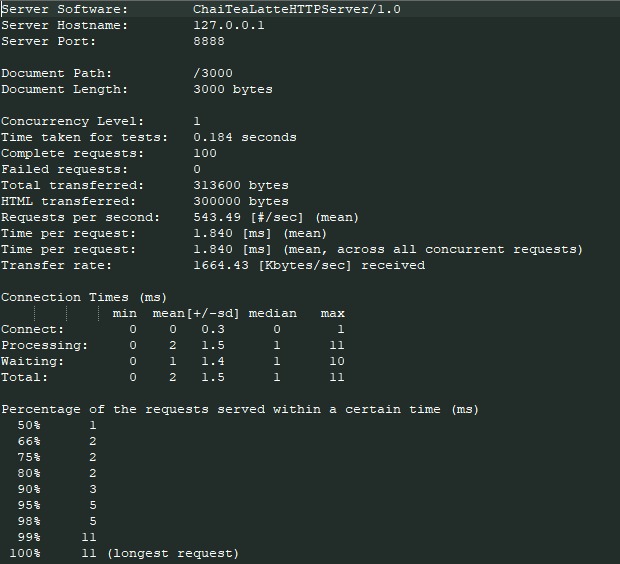
When concurrency level is less than 4, performance of http server is very high. Performance of http server is dropped almost 6 requests after 4 concurrency level.



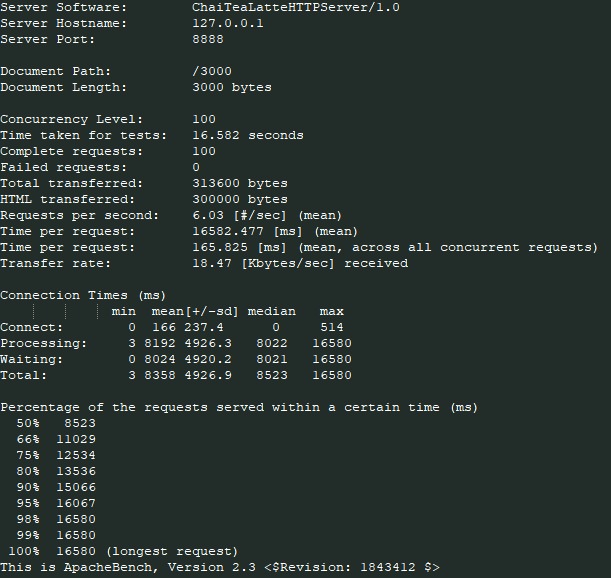
Apache Bench test for HTTP server result with concurrency level 1



Apache Bench test for Http Server result with concurrency level 100



Apache Bench test for Proxy Server result with concurrency level 1



Apache Bench test for Proxy Server result with concurrency level 100