CSE 4074 – Programming Assignment

Socket Programming – HTTP Server and Proxy Server

Kevser İLDEŞ – 150116048

Melisa DÖNMEZ – 150116030

1-) Implementing a multithreaded web server

In this part of the project, we implemented multithreaded http web server with using Python. We created a thread for each request. If the request is not against any rules, we created a html document of the desired size. However, if there is any violation of the rules, we printed the required error message.

Detailed explanation of code and methods:

First, we define host and take port number argument from user in the main. After that we setup a socket connection with client, bind it to the port and host. We accept client request in the forever loop. Inside this while loop, we get client request. We split request from lines and get first line. For example, first line of request something like that GET /500 HTTP/1.1. Then, we remove the leading slash from this line and reach the requested file. After splitting request, we called RequestThread class. We created e thread for each request in this class. Also, we checked whether the requested method is GET. If the requested method is GET and the size of the requested file is between 100 and 20000 bytes, we created an html file with the given size in the create_document method. Then we called ok_req method to write OK request header. If the requested file is not a digit or is outside the desired size, we called the bad_req method. In addition to these, we called not_impl method if the requested method is not GET. After checking all factors, we send the header to notify client and we closed the connection. These operations are repeated in a while loop for each request if there is no keyboard interrupt. If there is a keyboard interrupt, we close the socket.

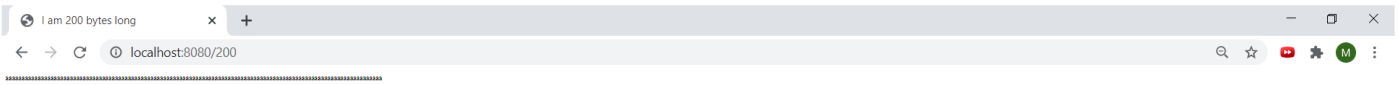
Example output:

1-) 200 OK – File size 500

```
GET /500 HTTP/1.1
New Thread created to handle a request 500 at 127.0.0.1:8080

GET /500 HTTP/1.1
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 500

Thread No: 0
```

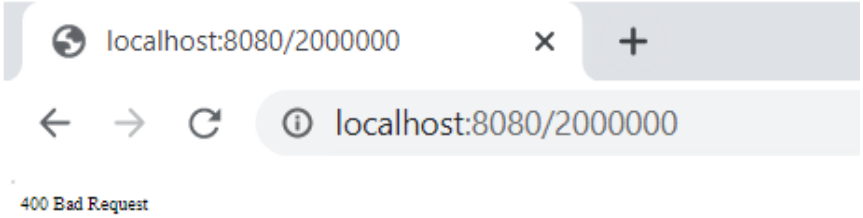
2-) 200 OK – File size 200

```
GET /200 HTTP/1.1
New Thread created to handle a request 200 at 127.0.0.1:8080

GET /200 HTTP/1.1
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 200

Thread No: 2
```



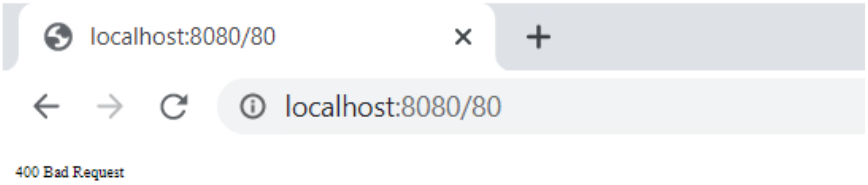
3-) 400 Bad Request – File size outside the range

```
GET /2000000 HTTP/1.1
New Thread created to handle a request 2000000 at 127.0.0.1:8080

GET /2000000 HTTP/1.1
HTTP/1.1 400 Bad Request
Content-Type: text/html
Content-Length: 15

Thread No: 4
```



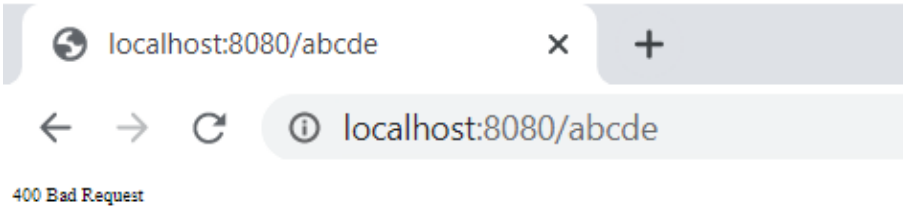
4-) 400 Bad Request – File size outside the range

```
GET /80 HTTP/1.1
New Thread created to handle a request 80 at 127.0.0.1:8080

GET /80 HTTP/1.1
HTTP/1.1 400 Bad Request
Content-Type: text/html
Content-Length: 15

Thread No: 6
```



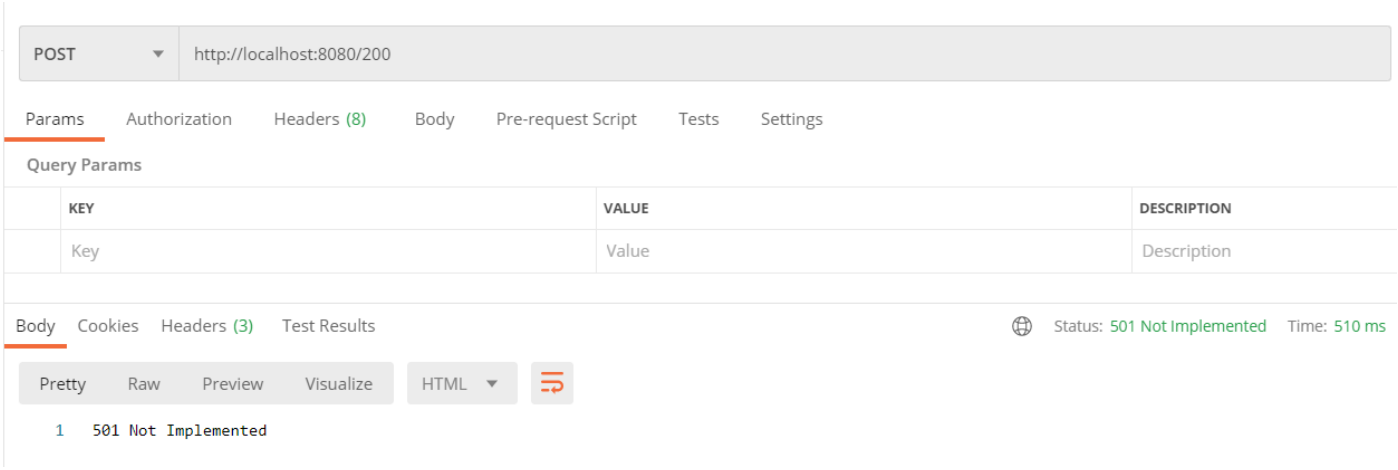
5-) 400 Bad Request – Requested file is not a digit.

```
GET /abcde HTTP/1.1
New Thread created to handle a request abcde at 127.0.0.1:8080

GET /abcde HTTP/1.1
HTTP/1.1 400 Bad Request
Content-Type: text/html
Content-Length: 15

Thread No: 8
```



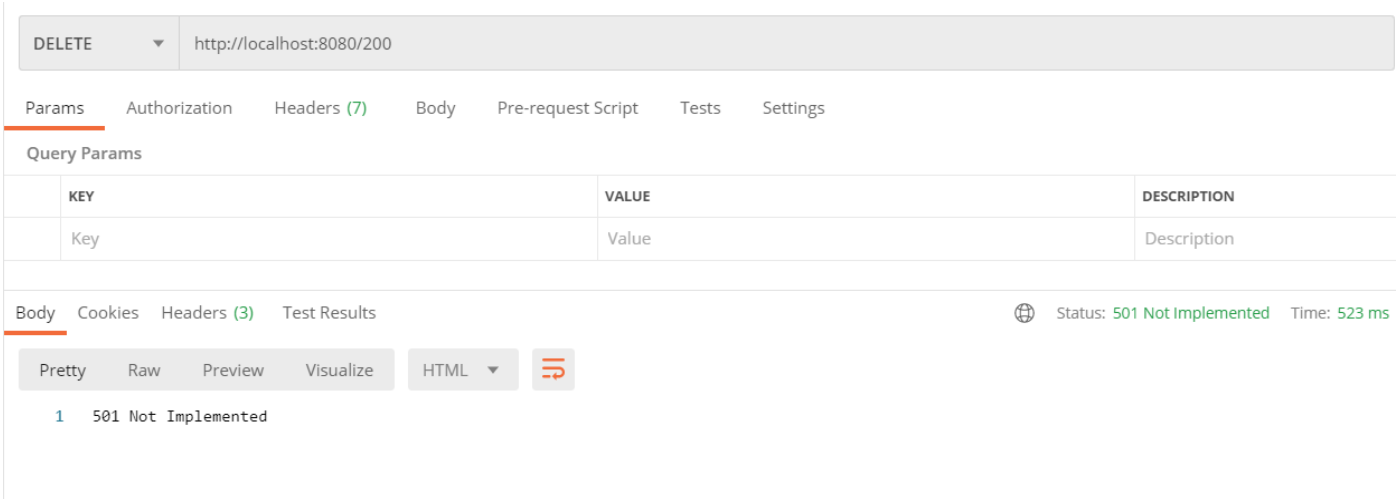
6-) 501 Not Implemented – Requested method is POST

```
POST /200 HTTP/1.1
New Thread created to handle a request 200 at 127.0.0.1:8080

POST /200 HTTP/1.1
HTTP/1.1 501 Not Implemented
Content-Type: text/html
Content-Length: 19

Thread No: 9
```



7-) 501 Not Implemented – Requested method is DELETE

```
DELETE /200 HTTP/1.1
New Thread created to handle a request 200 at 127.0.0.1:8080

DELETE /200 HTTP/1.1
HTTP/1.1 501 Not Implemented
Content-Type: text/html
Content-Length: 19

Thread No: 10
```



2-) Implementing a proxy server

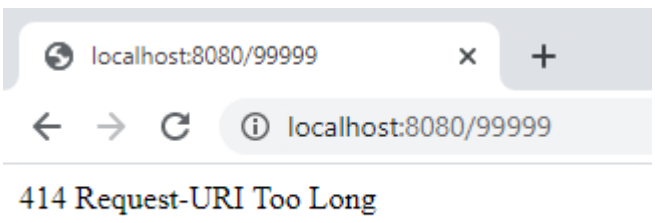
A multithreaded proxy server which is able to do cache with conditional get control implemented in this part using Python. A thread is created for each request and then if it obeys rules, request received from client is redirected to web server implemented in first part. Then receiving response from webserver or cache, proxy sends it to client.

In detail:

First, we define host and port number as localhost and 8888. Then we setup a socket connection with client, bind it to the port and host. We accept client request in the forever loop. Inside this while loop, we get client request. After that, we make necessary splits and arrangements including URL to URI conversion. Next we called RequestThreadPr class. We created a thread for each request in this class. It tries to connect to webserver and if it fails, that is, if webserver is not running currently it returns not found message with not_found method. If webserver is available, firstly, it checks size of the requesting file and if greater than 9999, it does not redirect to server instead returns long request message using long_req method. If size is in range, it firstly searches cache and check condition. If length is odd, that is, not modified, and if it exists in cache it sends response from cache. If it does not exist or even length which means it is modified, it sends request to webserver and receive response. After that it adds the response to cache as named according to size and then sends response to client. In addition, there is an expire time for each file in cache and when time is up, cached file is removed. After all necessary operations are done, we close connection. These operations are repeated in a while loop for each request.

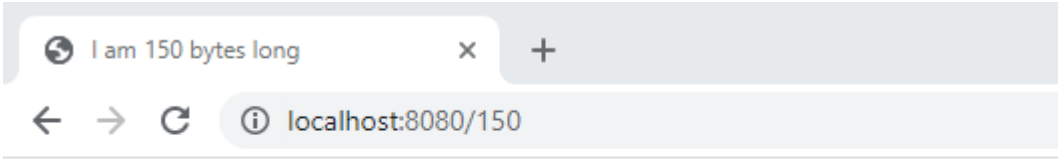
Example Outputs:

1) 414 Request-URI Too Long – responded from proxy without redirecting webserver



```
Proxy Server running on 127.0.0.1 using port 8888
********************** HTTP Server **********************
HTTP Server running on 127.0.0.1 using port 8080

------------------------------------------------------------
New thread is created in proxy side

HTTP/1.0 414 Request-URI Too Long
Content-Type: text/html
Content-Length: 24
```

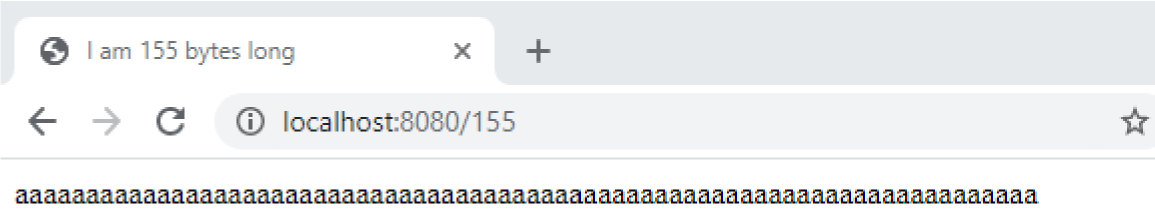
2) Get request for even-length file twice – in first, request redirected to webserver; in second, response although it is in cache, as it is modified, it is again received from webserver



aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

```
New thread is created in proxy side

New Thread created to handle a request 150 at 127.0.0.1:8080

GET /150 HTTP/1.1Host: localhost:8080
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 150
Server: HTTPServer/1.1

Thread No: 4

------------------------------------------------------------
Received from server and added to cache:  150

New thread is created in proxy side

The cached version is old, so receive modified response from server
New Thread created to handle a request 150 at 127.0.0.1:8080

GET /150 HTTP/1.1Host: localhost:8080
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 150
Server: HTTPServer/1.1

Thread No: 5

------------------------------------------------------------
Received from server and added to cache:  150
```

3) Get request for odd-length file twice – in first, request redirected to webserver; in second, response is send from proxy cache without connecting to webserver




```
New thread is created in proxy side

New Thread created to handle a request 155 at 127.0.0.1:8080

GET /155 HTTP/1.1Host: localhost:8080
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 155
Server: HTTPServer/1.1

Thread No: 0

----------------------------------------------------------
Received from server and added to cache:  155

New thread is created in proxy side

Send from cache:  155
```

3-) Using ApacheBench (ab) Program

a- <u>Send single request at a time (concurrency level = 1)</u>

ab -n 10 -c 1 http://ipv4.download.thinkbroadband.com/5MB.zip
We repeat the tests 3 times:

```
C:\Apache24\bin>ab -n 10 -c 1 http://ipv4.download.thinkbroadband.com/5MB.zip
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking ipv4.download.thinkbroadband.com (be patient).....done


Server Software:        nginx
Server Hostname:        ipv4.download.thinkbroadband.com
Server Port:            80

Document Path:          /5MB.zip
Document Length:        5242880 bytes

Concurrency Level:      1
Time taken for tests:   40.972 seconds
Complete requests:      10
Failed requests:        0
Total transferred:      52431510 bytes
HTML transferred:       52428800 bytes
Requests per second:    0.24 [#/sec] (mean)
Time per request:       4097.222 [ms] (mean)
Time per request:       4097.222 [ms] (mean, across all concurrent requests)
Transfer rate:          1249.69 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:       62   99  57.8     78     233
Processing:  2139 3998 1349.2   4340    6228
Waiting:       47   84  31.4     78     153
Total:       2233 4097 1346.5   4403    6290

Percentage of the requests served within a certain time (ms)
  50%   4403
  66%   4575
  75%   4965
  80%   5332
  90%   6290
  95%   6290
  98%   6290
  99%   6290
 100%   6290 (longest request)
```

```
C:\Apache24\bin>ab -n 10 -c 1 http://ipv4.download.thinkbroadband.com/5MB.zip
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking ipv4.download.thinkbroadband.com (be patient).....done


Server Software:        nginx
Server Hostname:        ipv4.download.thinkbroadband.com
Server Port:            80

Document Path:          /5MB.zip
Document Length:        5242880 bytes

Concurrency Level:      1
Time taken for tests:   34.285 seconds
Complete requests:      10
Failed requests:        0
Total transferred:      52431510 bytes
HTML transferred:       52428800 bytes
Requests per second:    0.29 [#/sec] (mean)
Time per request:       3428.541 [ms] (mean)
Time per request:       3428.541 [ms] (mean, across all concurrent requests)
Transfer rate:          1493.42 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:       51   72  28.1     63     150
Processing:  1359 3355 4474.4   2094   16021
Waiting:       59  302 746.9     63    2428
Total:       1422 3427 4473.0   2164   16084

Percentage of the requests served within a certain time (ms)
  50%   2164
  66%   2254
  75%   2453
  80%   3019
  90%  16084
  95%  16084
  98%  16084
  99%  16084
 100%  16084 (longest request)
```

```
C:\Apache24\bin>ab -n 10 -c 1 http://ipv4.download.thinkbroadband.com/5MB.zip
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking ipv4.download.thinkbroadband.com (be patient).....done


Server Software:        nginx
Server Hostname:        ipv4.download.thinkbroadband.com
Server Port:            80

Document Path:          /5MB.zip
Document Length:        5242880 bytes

Concurrency Level:      1
Time taken for tests:   34.536 seconds
Complete requests:      10
Failed requests:        0
Total transferred:      52431510 bytes
HTML transferred:       52428800 bytes
Requests per second:    0.29 [#/sec] (mean)
Time per request:       3453.632 [ms] (mean)
Time per request:       3453.632 [ms] (mean, across all concurrent requests)
Transfer rate:          1482.57 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:       55   76  33.4     64     141
Processing:  1352 3377 1571.4   3594    6697
Waiting:       55   66   7.2     66      79
Total:       1411 3453 1575.7   3658    6758

Percentage of the requests served within a certain time (ms)
  50%   3658
  66%   3890
  75%   4120
  80%   4399
  90%   6758
  95%   6758
  98%   6758
  99%   6758
 100%   6758 (longest request)
```

Result:

i) Time taken for test

(40.972 + 34.285 + 34.536) / 3 = 36.597 seconds

ii) Total transferred (bytes) and HTML transferred (bytes)

Total = 52431510

HTML = 524288000

iii) Time per request

(4097.222 + 3428.541 + 3453.632) / 3 = 3659.798 ms

iv) Requests per second (#/sec)

(0.24 + 0.29 + 0.29) / 3 = 0.273

v) Transfer rate (Kbytes / sec)

(1249.69 + 1493.42 + 1482.57) / 3 = 1408.56

vi) Connection times

Mean:

(4097 + 3427 + 3453) / 3 = 3659 ms

b- Send 5 at a time (concurrency level = 5)

ab -n 10 -c 5 http://ipv4.download.thinkbroadband.com/5MB.zip
We repeat the tests 3 times:

```
C:\Apache24\bin>ab -n 10 -c 5 http://ipv4.download.thinkbroadband.com/5MB.zip
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking ipv4.download.thinkbroadband.com (be patient).....done


Server Software:        nginx
Server Hostname:        ipv4.download.thinkbroadband.com
Server Port:            80

Document Path:          /5MB.zip
Document Length:        5242880 bytes

Concurrency Level:      5
Time taken for tests:   18.567 seconds
Complete requests:      10
Failed requests:        0
Total transferred:      52431510 bytes
HTML transferred:       52428800 bytes
Requests per second:    0.54 [#/sec] (mean)
Time per request:       9283.600 [ms] (mean)
Time per request:       1856.720 [ms] (mean, across all concurrent requests)
Transfer rate:          2757.69 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:       57   63   4.9     64      73
Processing:  5733 7914 1894.9   7961   11629
Waiting:       58   83  40.9     68     187
Total:       5792 7977 1895.2   8024   11688

Percentage of the requests served within a certain time (ms)
  50%   8024
  66%   8504
  75%   8784
  80%  10240
  90%  11688
  95%  11688
  98%  11688
  99%  11688
 100%  11688 (longest request)
```

```
C:\Apache24\bin>ab -n 10 -c 5 http://ipv4.download.thinkbroadband.com/5MB.zip
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking ipv4.download.thinkbroadband.com (be patient).....done


Server Software:        nginx
Server Hostname:        ipv4.download.thinkbroadband.com
Server Port:            80

Document Path:          /5MB.zip
Document Length:        5242880 bytes

Concurrency Level:      5
Time taken for tests:   20.639 seconds
Complete requests:      10
Failed requests:        0
Total transferred:      52431510 bytes
HTML transferred:       52428800 bytes
Requests per second:    0.48 [#/sec] (mean)
Time per request:       10319.700 [ms] (mean)
Time per request:       2063.940 [ms] (mean, across all concurrent requests)
Transfer rate:          2480.82 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:       62   65   3.7     66      72
Processing:  7251 9053 1218.7   9545   10859
Waiting:       64   88  42.8     69     195
Total:       7314 9119 1220.7   9611   10925

Percentage of the requests served within a certain time (ms)
  50%   9611
  66%   9644
  75%   9788
  80%  10536
  90%  10925
  95%  10925
  98%  10925
  99%  10925
 100%  10925 (longest request)
```

```
C:\Apache24\bin>ab -n 10 -c 5 http://ipv4.download.thinkbroadband.com/5MB.zip
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking ipv4.download.thinkbroadband.com (be patient).....done


Server Software:        nginx
Server Hostname:        ipv4.download.thinkbroadband.com
Server Port:            80

Document Path:          /5MB.zip
Document Length:        5242880 bytes

Concurrency Level:      5
Time taken for tests:   19.810 seconds
Complete requests:      10
Failed requests:        0
Total transferred:      52431510 bytes
HTML transferred:       52428800 bytes
Requests per second:    0.50 [#/sec] (mean)
Time per request:       9904.930 [ms] (mean)
Time per request:       1980.986 [ms] (mean, across all concurrent requests)
Transfer rate:          2584.71 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:       55   66   7.5     66      77
Processing:  4845 7611 1671.7   7554   10604
Waiting:       58   88  42.7     71     194
Total:       4900 7677 1675.0   7628   10665

Percentage of the requests served within a certain time (ms)
  50%   7628
  66%   7668
  75%   8972
  80%   9587
  90%  10665
  95%  10665
  98%  10665
  99%  10665
 100%  10665 (longest request)
```

Result:

i) Time taken for test

(18.567 + 20.639 + 19.810) / 3 = 19.672 seconds

ii) Total transferred (bytes) and HTML transferred (bytes)

Total = 52431510

HTML = 52428800

iii) Time per request

(1856.720 + 2063.940 + 1980.986) / 3 = 1967.215 [ms] (mean, across all concurrent requests)

iv) Requests per second (#/sec)

(0.54 + 0.48 + 0.50) / 3 = 0.507

v) Transfer rate (Kbytes / sec)

(2757.69 + 2480.72 + 2584.71) / 3 = 2607.71 [Kbytes/sec] received

vi) Connection times

Mean:

(7977 + 9119 + 7677) / 3 = 8257.7 ms

c- Send 10 at a time (concurrency level = 10)

ab -n 10 -c 10 http://ipv4.download.thinkbroadband.com/5MB.zip

We repeat the tests 3 times:

```
C:\Apache24\bin>ab -n 10 -c 10 http://ipv4.download.thinkbroadband.com/5MB.zip
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking ipv4.download.thinkbroadband.com (be patient).....done


Server Software:        nginx
Server Hostname:        ipv4.download.thinkbroadband.com
Server Port:            80

Document Path:          /5MB.zip
Document Length:        5242880 bytes

Concurrency Level:      10
Time taken for tests:   19.213 seconds
Complete requests:      10
Failed requests:        0
Total transferred:      52431510 bytes
HTML transferred:       52428800 bytes
Requests per second:    0.52 [#/sec] (mean)
Time per request:       19212.986 [ms] (mean)
Time per request:       1921.299 [ms] (mean, across all concurrent requests)
Transfer rate:          2665.00 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:       61   70   7.8     70      85
Processing: 13515 15572 1567.8  15771   18813
Waiting:       62  280 191.1    292     579
Total:      13577 15642 1570.8  15838   18898

Percentage of the requests served within a certain time (ms)
  50%  15838
  66%  16112
  75%  16472
  80%  16760
  90%  18898
  95%  18898
  98%  18898
  99%  18898
 100%  18898 (longest request)
```

```
C:\Apache24\bin>ab -n 10 -c 10 http://ipv4.download.thinkbroadband.com/5MB.zip
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking ipv4.download.thinkbroadband.com (be patient).....done


Server Software:        nginx
Server Hostname:        ipv4.download.thinkbroadband.com
Server Port:            80

Document Path:          /5MB.zip
Document Length:        5242880 bytes

Concurrency Level:      10
Time taken for tests:   21.326 seconds
Complete requests:      10
Failed requests:        0
Total transferred:      52431510 bytes
HTML transferred:       52428800 bytes
Requests per second:    0.47 [#/sec] (mean)
Time per request:       21325.779 [ms] (mean)
Time per request:       2132.578 [ms] (mean, across all concurrent requests)
Transfer rate:          2400.97 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:       60   67   5.0     65      76
Processing: 16428 18045 1619.9  17476   20851
Waiting:       71  259 174.5    269     537
Total:      16493 18112 1622.5  17541   20924

Percentage of the requests served within a certain time (ms)
  50%  17541
  66%  17720
  75%  19511
  80%  20597
  90%  20924
  95%  20924
  98%  20924
  99%  20924
 100%  20924 (longest request)
```

```
C:\Apache24\bin>ab -n 10 -c 10 http://ipv4.download.thinkbroadband.com/5MB.zip
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking ipv4.download.thinkbroadband.com (be patient).....done


Server Software:        nginx
Server Hostname:        ipv4.download.thinkbroadband.com
Server Port:            80

Document Path:          /5MB.zip
Document Length:        5242880 bytes

Concurrency Level:      10
Time taken for tests:   18.402 seconds
Complete requests:      10
Failed requests:        0
Total transferred:      52431510 bytes
HTML transferred:       52428800 bytes
Requests per second:    0.54 [#/sec] (mean)
Time per request:       18402.318 [ms] (mean)
Time per request:       1840.232 [ms] (mean, across all concurrent requests)
Transfer rate:          2782.40 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:       55   71  11.5     72      92
Processing: 10787 14966 2217.1  15461   18200
Waiting:       67  282 187.3    322     574
Total:      10848 15037 2223.8  15552   18272

Percentage of the requests served within a certain time (ms)
  50%  15552
  66%  16116
  75%  16761
  80%  17074
  90%  18272
  95%  18272
  98%  18272
  99%  18272
 100%  18272 (longest request)
```

Result:

i) Time taken for test
(19.213 + 21.326 + 18.402) / 3 = 19.647 seconds

ii) Total transferred (bytes) and HTML transferred (bytes)
Total = 52431510
HTML = 52428800

iii) Time per request
(1921.299 + 2132.578 + 1840.232) / 3 = 1964.703 [ms] (mean, across all concurrent requests)

iv) Requests per second (#/sec)
(0.52 + 0.47 + 0.54) / 3 = 0.51

v) Transfer rate (Kbytes / sec)
(2665.00 + 2400.97 + 2782.40) / 3 = 2616.12 [Kbytes/sec] received

vi) Connection times
Mean:
(15642 + 18112 + 15037) / 3 = 16263.67 ms

Comments:

Time taken for test:

When the concurrency level increased from 1 to 5, there was a noticeable decrease in the time taken for test value. But when the c value increased from 5 to 10, we did not observe a huge change. However, we run it 3 times and get an average. Finally, we realized that time taken for test decrease when concurrency level increases.

Total transferred (bytes) and HTML transferred (bytes):

Total transferred and HTML transferred values did not change according to concurrency value. We got same values at every c value and every run.

Time per request:

Although there is not a huge difference between time per request values when concurrency level is 5 and 10, the time per request value decreased when the concurrency value increased.

Requests per second (#/sec):

Total number of requests per second decreased when concurrency level increased.

Transfer rate (Kbytes / sec):

The transfer rate calculated using the formula totalread/1024/timetaken. Since the time taken values decrease when the concurrency level increases, the transfer rate also increases.

Connection times:

Connection time increases when concurrency level increases.

d- Repeat a), b) and c) using –k argument. This argument will send a “Connection: Keep- Alive” header to the web server

C = 1 with -k

```
C:\Apache24\bin>ab -n 10 -c 1 -k http://ipv4.download.thinkbroadband.com/5MB.zip
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking ipv4.download.thinkbroadband.com (be patient).....done


Server Software:        nginx
Server Hostname:        ipv4.download.thinkbroadband.com
Server Port:            80

Document Path:          /5MB.zip
Document Length:        5242880 bytes

Concurrency Level:      1
Time taken for tests:   45.661 seconds
Complete requests:      10
Failed requests:        0
Keep-Alive requests:    10
Total transferred:      52431560 bytes
HTML transferred:       52428800 bytes
Requests per second:    0.22 [#/sec] (mean)
Time per request:       4566.117 [ms] (mean)
Time per request:       4566.117 [ms] (mean, across all concurrent requests)
Transfer rate:          1121.36 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0   10  30.1      0      95
Processing:  3743 4557 523.6   4626    5307
Waiting:       70   77  15.6     72     121
Total:       3743 4566 525.5   4699    5307

Percentage of the requests served within a certain time (ms)
  50%   4699
  66%   4738
  75%   4819
  80%   5190
  90%   5307
  95%   5307
  98%   5307
  99%   5307
 100%   5307 (longest request)
```

Result:

i) Time taken for test
   45.661 seconds

ii) Total transferred (bytes) and HTML transferred (bytes)
   Total = 52431560
   HTML = 52428800

iii) Time per request
   4566.117 [ms] (mean, across all concurrent requests)

iv) Requests per second (#/sec)
   0.22

v) Transfer rate (Kbytes / sec)
   1121.36 [Kbytes/sec] received

vi) Connection times
   Mean:
   4566 ms

C = 5 with -k

```
C:\Apache24\bin>ab -n 10 -c 5 -k http://ipv4.download.thinkbroadband.com/5MB.zip
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking ipv4.download.thinkbroadband.com (be patient).....done


Server Software:        nginx
Server Hostname:        ipv4.download.thinkbroadband.com
Server Port:            80

Document Path:          /5MB.zip
Document Length:        5242880 bytes

Concurrency Level:      5
Time taken for tests:   24.988 seconds
Complete requests:      10
Failed requests:        0
Keep-Alive requests:    10
Total transferred:      52431560 bytes
HTML transferred:       52428800 bytes
Requests per second:    0.40 [#/sec] (mean)
Time per request:       12493.803 [ms] (mean)
Time per request:       2498.761 [ms] (mean, across all concurrent requests)
Transfer rate:          2049.12 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0  103 168.9     67     541
Processing:  8410 10488 2055.1   9920   15520
Waiting:       59  249 297.3    130     867
Total:       8410 10591 2096.1   9920   15587

Percentage of the requests served within a certain time (ms)
  50%   9920
  66%  10412
  75%  11786
  80%  12072
  90%  15587
  95%  15587
  98%  15587
  99%  15587
 100%  15587 (longest request)
```

Result:

i) Time taken for test
   24.988 seconds

ii) Total transferred (bytes) and HTML transferred (bytes)
   Total = 52431560
   HTML = 52428800

iii) Time per request
   2498.761 [ms] (mean, across all concurrent requests)

iv) Requests per second (#/sec)
   0.40

v) Transfer rate (Kbytes / sec)
   2049.12 [Kbytes/sec] received

vi) Connection times
   Mean:
   10591 ms

C = 10 with -k

```
C:\Apache24\bin>ab -n 10 -c 10 -k http://ipv4.download.thinkbroadband.com/5MB.zip
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking ipv4.download.thinkbroadband.com (be patient).....done


Server Software:        nginx
Server Hostname:        ipv4.download.thinkbroadband.com
Server Port:            80

Document Path:          /5MB.zip
Document Length:        5242880 bytes

Concurrency Level:      10
Time taken for tests:   19.915 seconds
Complete requests:      10
Failed requests:        0
Keep-Alive requests:    10
Total transferred:      52431560 bytes
HTML transferred:       52428800 bytes
Requests per second:    0.50 [#/sec] (mean)
Time per request:       19915.420 [ms] (mean)
Time per request:       1991.542 [ms] (mean, across all concurrent requests)
Transfer rate:          2571.01 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:       54   65   9.4     65      88
Processing: 13319 16308 2032.3  16075   19670
Waiting:       56  248 174.9    277     526
Total:      13373 16373 2037.3  16140   19736

Percentage of the requests served within a certain time (ms)
  50%  16140
  66%  16927
  75%  18312
  80%  18399
  90%  19736
  95%  19736
  98%  19736
  99%  19736
 100%  19736 (longest request)
```

Result:

i) Time taken for test
   19.915 seconds

ii) Total transferred (bytes) and HTML transferred (bytes)
   Total = 52431560
   HTML = 52428800

iii) Time per request
   1991.542 [ms] (mean, across all concurrent requests)

iv) Requests per second (#/sec)
   0.50

v) Transfer rate (Kbytes / sec)
   2571.01 [Kbytes/sec] received

vi) Connection times
   Mean:
   16373 ms

Comments (Test of concurrency level with -k):

Time taken for test:

Time taken for test increases for all concurrency level according to with keep alive.

Total transferred (bytes) and HTML transferred (bytes):

We realized that total transferred value increased with keep alive argument. But HTML transferred did not change.

Time per request:

With keep alive, time per request values increased. We can say that server is running slower.

Requests per second (#/sec):

With keep alive, request per second values decreased, it gave better results.

Transfer rate (Kbytes / sec):

Transfer rate decreased with keep alive.

Connection times:

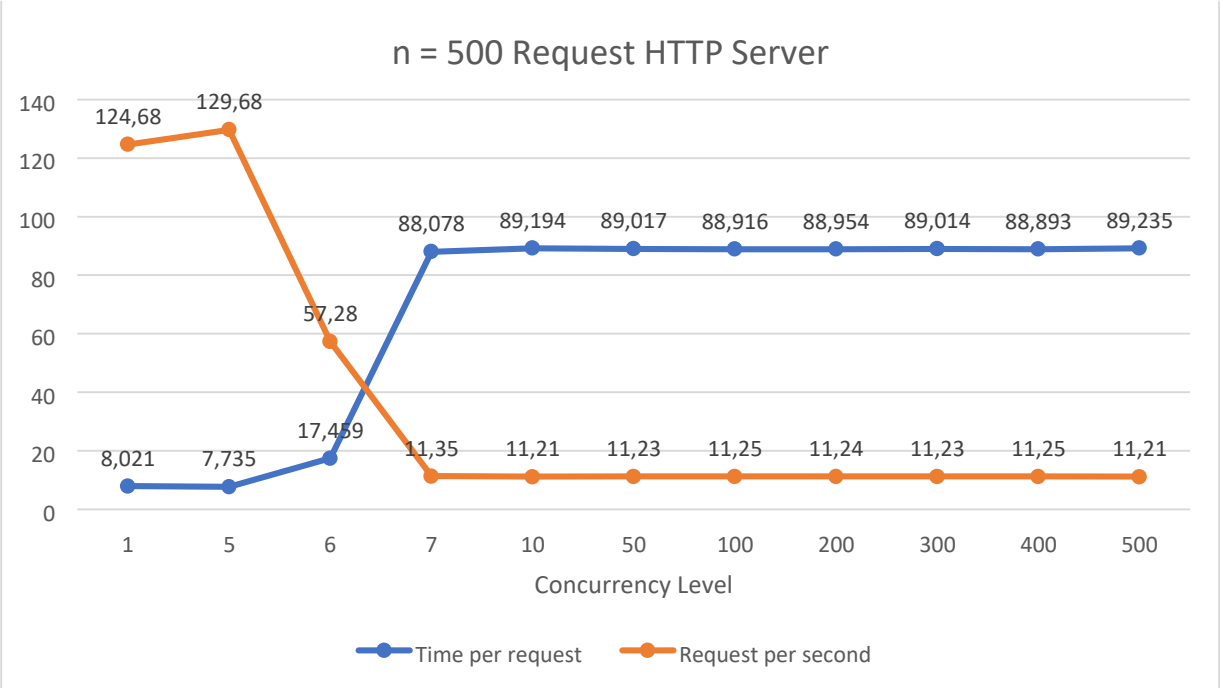
With keep alive connection times increased.

4-) Testing Our Server Using Apache Bench

- **HTTP Server:**

  We applied Apache Bench test to our http server. We sent 500 requests with different concurrency level. This table shows our Apache Bench test result for http server.

Concurrency Level	Time per request	Request per second
1	8.021	124.68
5	7.735	129.68
6	17.459	57.28
7	88.078	11.35
10	89.194	11.21
50	89.017	11.23
100	88.916	11.25
200	88.954	11.24
300	89.014	11.23
400	88.893	11.25
500	89.235	11.21



As seen on the table, our http server performance is high when concurrency level less than 6. When the concurrency level is greater than 6, our http server slowed down. In addition, we also observed very small changes in Time per request and request per second values when concurrency level greater than 6.

Example Apache Bench test result screen shots for http server:

Concurrency level = 5:

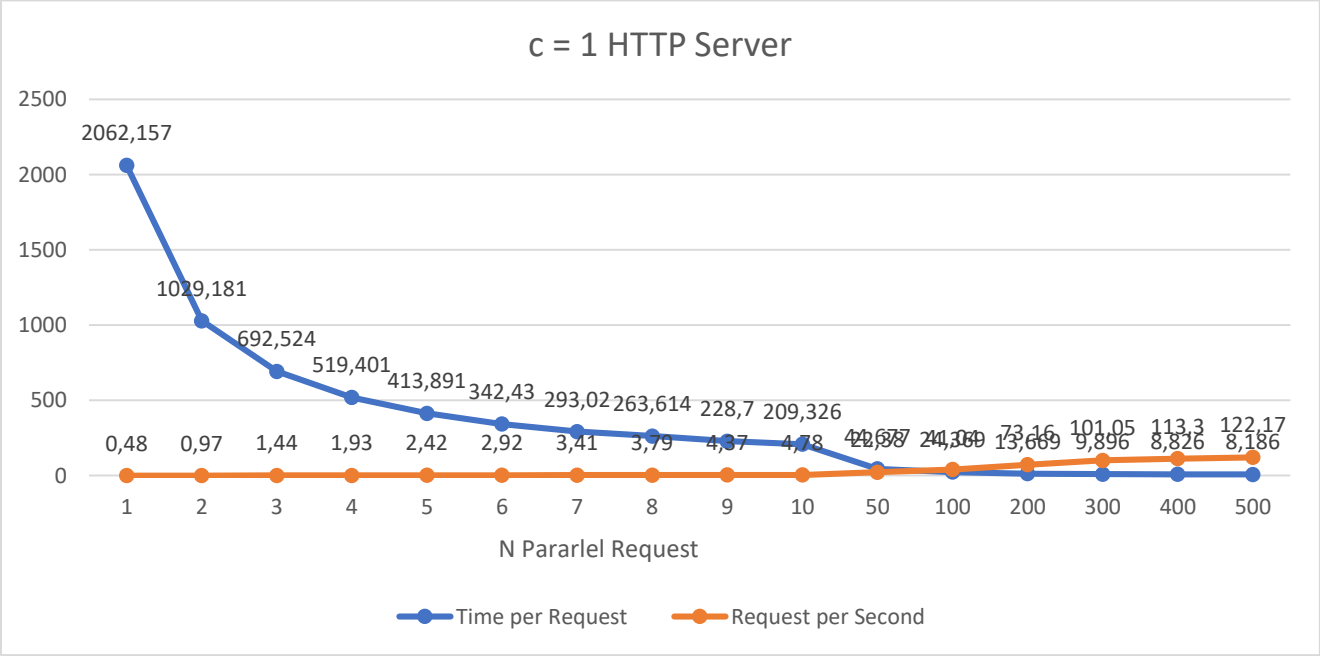
```
C:\Apache24\bin>ab -n 500 -c 5 http://localhost:8080/150
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Finished 500 requests


Server Software:        HTTPServer/1.1
Server Hostname:        localhost
Server Port:            8080

Document Path:          /150
Document Length:        150 bytes

Concurrency Level:      5
Time taken for tests:   3.868 seconds
Complete requests:      500
Failed requests:        0
Total transferred:      135500 bytes
HTML transferred:       75000 bytes
Requests per second:    129.28 [#/sec] (mean)
Time per request:       38.677 [ms] (mean)
Time per request:       7.735 [ms] (mean, across all concurrent requests)
Transfer rate:          34.21 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   2.4      0      16
Processing:     0   18   7.5     16      31
Waiting:        0   18   7.5     16      31
Total:          0   18   7.7     16      31

Percentage of the requests served within a certain time (ms)
  50%     16
  66%     16
  75%     16
  80%     24
  90%     31
  95%     31
  98%     31
  99%     31
 100%     31 (longest request)
```

Concurrency level = 100:

```
C:\Apache24\bin>ab -n 500 -c 100 http://localhost:8080/150
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Finished 500 requests


Server Software:        HTTPServer/1.1
Server Hostname:        localhost
Server Port:            8080

Document Path:          /150
Document Length:        150 bytes

Concurrency Level:      100
Time taken for tests:   44.458 seconds
Complete requests:      500
Failed requests:        0
Total transferred:      135500 bytes
HTML transferred:       75000 bytes
Requests per second:    11.25 [#/sec] (mean)
Time per request:       8891.638 [ms] (mean)
Time per request:       88.916 [ms] (mean, across all concurrent requests)
Transfer rate:          2.98 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0   85 190.4      0     531
Processing:     0 7627 1962.0   8188    8726
Waiting:        0 4205 2406.9   4092    8200
Total:        509 7712 1958.7   8664    8726

Percentage of the requests served within a certain time (ms)
  50%   8664
  66%   8681
  75%   8689
  80%   8694
  90%   8703
  95%   8716
  98%   8723
  99%   8726
 100%   8726 (longest request)
```

Then, we applied Apache Bench test to our server once again with same concurrency level (1) and different request numbers. The table and graphic below show our Apache Bench test results.

n	Time per Request	Request per Second
1	2062,157	0,48
2	1029,181	0,97
3	692,524	1,44
4	519,401	1,93
5	413,891	2,42
6	342,430	2,92
7	293,020	3,41
8	263,614	3,79
9	228,700	4,37
10	209,326	4,78
50	44,677	22,38
100	24,369	41,04
200	13,669	73,16
300	9,896	101,05
400	8,826	113,30
500	8,186	122,17



N = 5

```
C:\Apache24\bin>ab -n 5 -c 1 http://localhost:8080/150
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient).....done


Server Software:        HTTPServer/1.1
Server Hostname:        localhost
Server Port:            8080

Document Path:          /150
Document Length:        150 bytes

Concurrency Level:      1
Time taken for tests:   2.069 seconds
Complete requests:      5
Failed requests:        0
Total transferred:      1355 bytes
HTML transferred:       750 bytes
Requests per second:    2.42 [#/sec] (mean)
Time per request:       413.891 [ms] (mean)
Time per request:       413.891 [ms] (mean, across all concurrent requests)
Transfer rate:          0.64 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   0.0      0       0
Processing:     0    3   6.9      0      15
Waiting:        0    3   6.9      0      15
Total:          0    3   6.9      0      15

Percentage of the requests served within a certain time (ms)
  50%      0
  66%      0
  75%      0
  80%     15
  90%     15
  95%     15
  98%     15
  99%     15
 100%     15 (longest request)
```

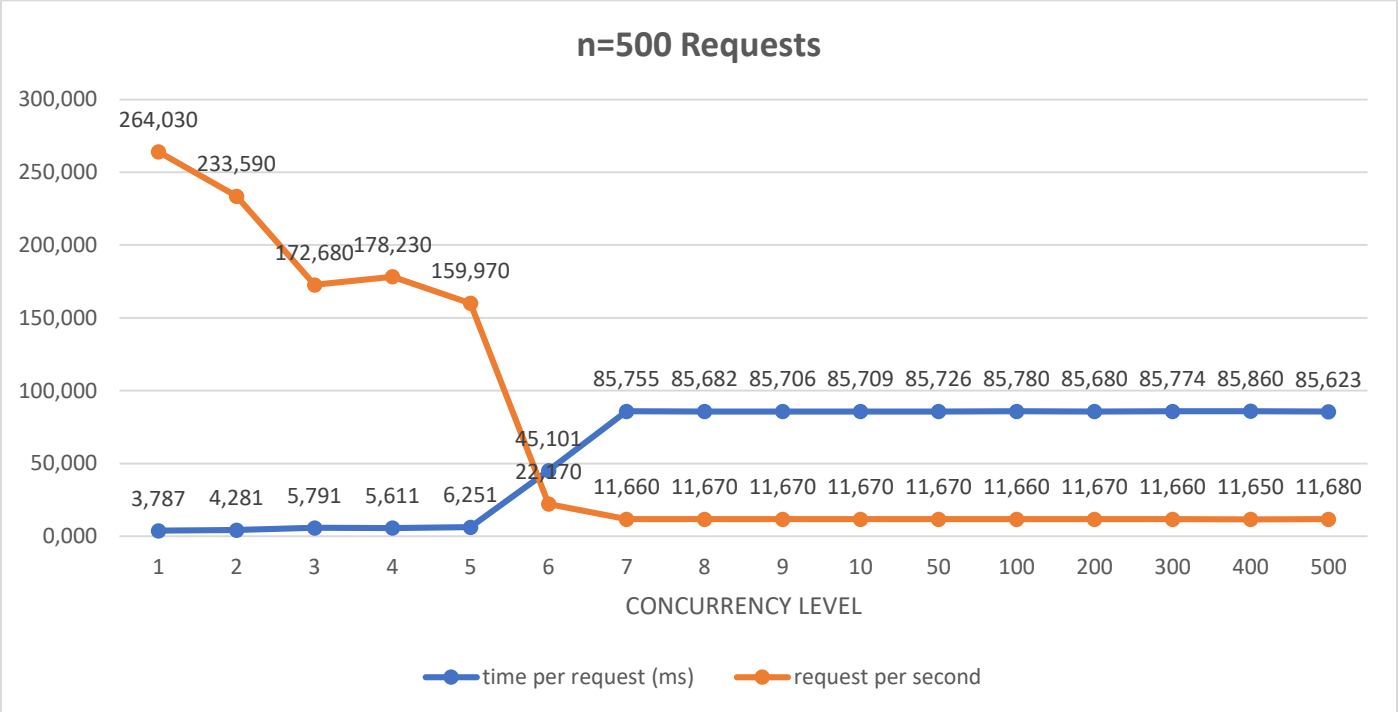
N = 100

```
C:\Apache24\bin>ab -n 100 -c 1 http://localhost:8080/150
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient).....done


Server Software:        HTTPServer/1.1
Server Hostname:        localhost
Server Port:            8080

Document Path:          /150
Document Length:        150 bytes

Concurrency Level:      1
Time taken for tests:   2.437 seconds
Complete requests:      100
Failed requests:        0
Total transferred:      27100 bytes
HTML transferred:       15000 bytes
Requests per second:    41.04 [#/sec] (mean)
Time per request:       24.369 [ms] (mean)
Time per request:       24.369 [ms] (mean, across all concurrent requests)
Transfer rate:          10.86 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   2.2      0      16
Processing:     0    3   6.4      0      16
Waiting:        0    3   6.3      0      16
Total:          0    4   6.6      0      16

Percentage of the requests served within a certain time (ms)
  50%      0
  66%      0
  75%      0
  80%     16
  90%     16
  95%     16
  98%     16
  99%     16
 100%     16 (longest request)
```

- **Proxy Server:**

  We applied Apache Bench test to our proxy server. We sent 500 requests with different concurrency levels. The table and graphic below show our Apache Bench test results for proxy server.

Concurrency Level	Time per Request	Request per Second
1	3,787	264,030
2	4,281	233,590
3	5,791	172,680
4	5,611	178,230
5	6,251	159,970
6	45,101	22,170
7	85,755	11,660
8	85,682	11,670
9	85,706	11,670
10	85,709	11,670
50	85,726	11,670
100	85,780	11,660
200	85,680	11,670
300	85,774	11,660
400	85,860	11,650
500	85,623	11,680



As seen on the table and the graphic, our proxy server performance is high when concurrency level less than 6. There is a significant change between c=5 – c=6 and c=6 – c=7. Besides, when the concurrency level is greater than 6, proxy server slowed down and we observed very little changes between results.

Below, some examples of screen shots are provided for Apache Bench test results:

C = 5

```
C:\Apache24\bin>ab -n 500 -c 5 -X 127.0.0.1:8888 http://localhost:8080/125
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost [through 127.0.0.1:8888] (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Finished 500 requests


Server Software:        HTTPServer/1.1
Server Hostname:        localhost
Server Port:            8080

Document Path:          /125
Document Length:        125 bytes

Concurrency Level:      5
Time taken for tests:   3.126 seconds
Complete requests:      500
Failed requests:        0
Total transferred:      121004 bytes
HTML transferred:       62500 bytes
Requests per second:    159.97 [#/sec] (mean)
Time per request:       31.257 [ms] (mean)
Time per request:       6.251 [ms] (mean, across all concurrent requests)
Transfer rate:          37.81 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   1.2      0      16
Processing:     0   31   6.1     31      48
Waiting:        0   30   6.1     31      48
Total:          0   31   6.1     31      48
```

```
Percentage of the requests served within a certain time (ms)
  50%     31
  66%     31
  75%     31
  80%     31
  90%     31
  95%     47
  98%     47
  99%     47
 100%     48 (longest request)
```

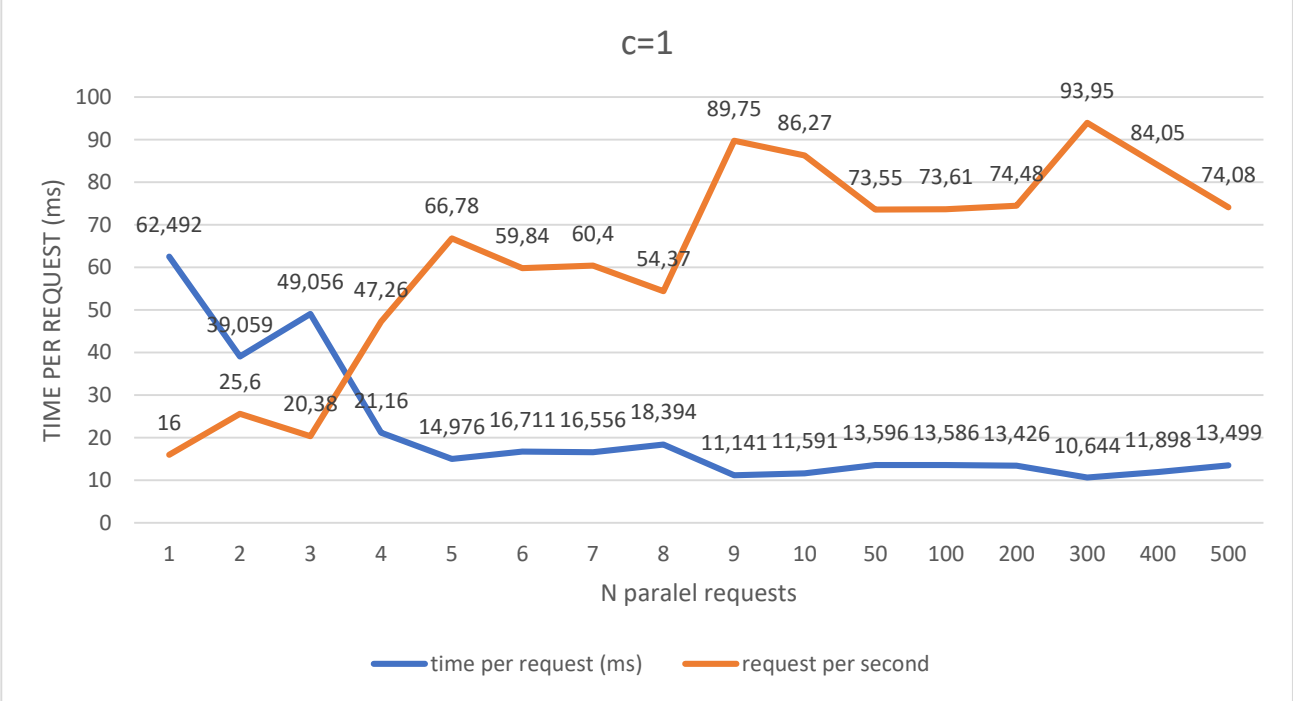
- C=100

```
C:\Apache24\bin>ab -n 500 -c 100 -X 127.0.0.1:8888 http://localhost:8080/125
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost [through 127.0.0.1:8888] (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Finished 500 requests


Server Software:        HTTPServer/1.1
Server Hostname:        localhost
Server Port:            8080

Document Path:          /125
Document Length:        125 bytes

Concurrency Level:      100
Time taken for tests:   42.837 seconds
Complete requests:      500
Failed requests:        0
Total transferred:      121004 bytes
HTML transferred:       62500 bytes
Requests per second:    11.67 [#/sec] (mean)
Time per request:       8567.428 [ms] (mean)
Time per request:       85.674 [ms] (mean, across all concurrent requests)
Transfer rate:          2.76 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0   86 192.0      0     517
Processing:    31 7697 1975.8   8251    8767
Waiting:       16 4245 2429.7   4125    8251
Total:        516 7782 1972.4   8759    8767
```

```
Percentage of the requests served within a certain time (ms)
  50%   8759
  66%   8765
  75%   8767
  80%   8767
  90%   8767
  95%   8767
  98%   8767
  99%   8767
 100%   8767 (longest request)
```

Then, we applied Apache Bench test to our proxy server once again with same concurrency level (1) and different request numbers. The table and graphic below show our Apache Bench test results.

n	Time per Request	Request per Second
1	62,492	16,00
2	39,059	25,60
3	49,056	20,38
4	21,160	47,26
5	14,976	66,78
6	16,711	59,84
7	16,556	60,40
8	18,394	54,37
9	11,141	89,75
10	11,591	86,27
50	13,596	73,55
100	13,586	73,61
200	13,426	74,48
300	10,644	93,95
400	11,898	84,05
500	13,499	74,08



N=5

```
C:\Apache24\bin>ab -n 5 -c 1 -X 127.0.0.1:8888 http://localhost:8080/125
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/
Benchmarking localhost [through 127.0.0.1:8888] (be patient).....done
Server Software:        HTTPServer/1.1
Server Hostname:        localhost
Server Port:            8080
Document Path:          /125
Document Length:        125 bytes
Concurrency Level:      1
Time taken for tests:   0.075 seconds
Complete requests:      5
Failed requests:        0
Total transferred:      1214 bytes
HTML transferred:       625 bytes
Requests per second:    66.78 [#/sec] (mean)
Time per request:       14.976 [ms] (mean)
Time per request:       14.976 [ms] (mean, across all concurrent requests)
Transfer rate:          15.83 [Kbytes/sec] received
Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   0.0      0       0
Processing:     0   15  11.9     18      31
Waiting:        0   15  11.9     18      31
Total:          0   15  11.9     18      31
Percentage of the requests served within a certain time (ms)
  50%     16
  66%     20
  75%     20
  80%     31
  90%     31
```

```
 95%   31
 98%   31
 99%   31
100%   31 (longest request)
```

n=100

```
C:\Apache24\bin>ab -n 100 -c 1 -X 127.0.0.1:8888 http://localhost:8080/125
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost [through 127.0.0.1:8888] (be patient).....done


Server Software:        HTTPServer/1.1
Server Hostname:        localhost
Server Port:            8080

Document Path:          /125
Document Length:        125 bytes

Concurrency Level:      1
Time taken for tests:   1.359 seconds
Complete requests:      100
Failed requests:        0
Total transferred:      24204 bytes
HTML transferred:       12500 bytes
Requests per second:    73.61 [#/sec] (mean)
Time per request:       13.586 [ms] (mean)
Time per request:       13.586 [ms] (mean, across all concurrent requests)
Transfer rate:          17.40 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    1   2.2      0      16
Processing:     0   13  13.5     11     100
Waiting:        0   13  13.6     11     100
Total:          0   13  13.4     12     100

Percentage of the requests served within a certain time (ms)
  50%     12
  66%     16
  75%     16
  80%     16
  90%     18
  95%     31
  98%     73
  99%    100
 100%    100 (longest request)
```