COMPUTER NETWORKS CSE4074 PROJECT REPORT

GROUP MEMBERS
MURAT ŞENOL 150117039
EMRE YİĞİT 150116056

## Design

```python
def handle_client(conn, addr):
    print(f"[NEW CONNECTION] {addr} connected.")
    connected = True
    # while connected:
    msg = conn.recv(2048).decode(FORMAT)# Receive message from client
    if msg:
        msg = msg.split('\r\n')[0]  # first row
        print(f"Received Message: '{msg}' from {addr[1]}")

        if msg == DISCONNECT_MESSAGE:# if message is disconnect message disconnect client and return the message
            conn.sendall(DISCONNECT_MESSAGE.encode(FORMAT))
            connected = False
            # continue

        response_line = create_response_line(msg)
        response_first_line = response_line.split("\r\n")[0]# first line of response for printing
        print(f"Send Message: '{response_first_line}' to {addr[1]}")
        conn.sendall(response_line.encode(FORMAT))# sendall to client

    conn.close()


def start():
    server.listen()
    print(f"[LISTENING] Server is listening on {SERVER}")
    while True:
        conn, addr = server.accept() #Server Starts to listen on this connection and adress
        thread = threading.Thread(target=handle_client, args=(conn, addr))
        thread.start()
        print(f"[ACTIVE CONNECTIONS] {threading.activeCount() - 1}") # # of active connections (Active threads - 1 [This thread])
```

*Şekil 1:Server.py - Client Handling*

Here we start to listen to specified address and port , and start the handling client requests. We check the first line of the request if it is a disconnect message , if not send it to create_response_line method to generate response line according to received request. After response line is generated we send it to the client.

```python
def main():
    try:
        client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        client.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
        client.bind(ADDR)# bind to connection on ADDR = {server_ip , port}
        client.listen()
        print("[*] Initializing socket. Done.")
        print("[*] Socket bound successfully...")
        print(f"[*] Server started successfully [{PORT}]")
    except Exception as e_socket:
        print(e_socket)
    while True:
        conn, addr = client.accept()# accept new client connection
        thread = threading.Thread(target=handle_client, args=(conn, addr))
        thread.start()


def handle_client(conn, addr):
    connected = True

    # while connected:

    data = conn.recv(BUFFER).decode(FORMAT)# Recieve Data

    # if len(data) == 0:
        # continue

    if data == DISCONNECT_MESSAGE:# If received Data is disconnect message
        connected = False
        conn.sendall("\nDisconnected !".encode(FORMAT))# return message to client
        # continue

    response_line = connection_string(conn, addr, data)# get the response line from method
    if type(response_line) is bytes:
```

*Şekil 2: Proxy Server - Client Handling*

Similarly same thing happens in the proxy server. We listen to the specified address and port. Receive requests from client and pass it to the server. Then we receive response line from the server then we pass on it to the client back.

```python
try:# Cache hit , but file was modified , request new file from server
    server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server.connect(proxy_addr)# connect to the server
    msg = http_method + " /" + uri + " " + http_ver + "\r\n"
    server.sendall(msg.encode(FORMAT))# send requested uri to the server
    print(f"Send Message: {msg} to {proxy_addr}")
    reply = server.recv(BUFFER)# receive message from server
    response_line = reply
    reply = reply.decode(FORMAT)
    reply_first_line = reply.split("\r\n")[0]
    print(f"Received Message: {reply_first_line} from {proxy_addr}")

    # write to cache
    html_data = response_line.decode(FORMAT).split("\r\n\r\n")[1]

    f = open(cache_dir, "wb")
    f.write(html_data.encode())
    f.close()
except Exception as e_connect:
    print(f"[CONNECTION ERROR]:{e_connect}")
    response_line = create_response_line(404, 0)# 404 error not found , Server is not running
```

*Şekil 3:Caching*

If the requested file is not on cache , we write it to the cache to reduce the traffic on server.

3. Answers to the questions

a)

```
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:   38.026 seconds
Complete requests:      10
Failed requests:        0
Total transferred:      52431510 bytes
HTML transferred:       52428800 bytes
Requests per second:    0.26 [#/sec] (mean)
Time per request:       3802.645 [ms] (mean)
Time per request:       3802.645 [ms] (mean, across all concurrent requests)
Transfer rate:          1346.50 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:       71   82  15.3     78     125
Processing:  2204 3721 1678.0   3897    7069
Waiting:       63   83  20.8     78     141
Total:       2282 3803 1678.9   4022    7146

Percentage of the requests served within a certain time (ms)
  50%   4022
  66%   4040
  75%   4321
  80%   6040
  90%   7146
  95%   7146
  98%   7146
  99%   7146
 100%   7146 (longest request)

C:\Apache24\bin>
```

*Şekil 4:10 Requests - Concurrency Level 1*

i) Time taken for tests (seconds) : 38.026

ii) Total transferred (bytes) and HTML transferred (bytes) : 52431510

iii) Time per request (ms) : 3802.645 (mean , across all concurrent requests)

iv) Requests per second (#/sec) : 0.26

v) Transfer rate (Kbytes/sec) : 1346.50

vi) Connection times : can be seen from figure 4.

b)

```
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:   27.187 seconds
Complete requests:      10
Failed requests:        0
Total transferred:      52431510 bytes
HTML transferred:       52428800 bytes
Requests per second:    0.37 [#/sec] (mean)
Time per request:       13593.455 [ms] (mean)
Time per request:       2718.691 [ms] (mean, across all concurrent requests)
Transfer rate:          1883.36 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:       70   86  12.9     85     119
Processing: 10078 12737 2084.8  12424   16174
Waiting:       70  126  74.4    104     296
Total:      10163 12823 2080.5  12507   16251

Percentage of the requests served within a certain time (ms)
  50%  12507
  66%  14148
  75%  14768
  80%  14769
  90%  16251
  95%  16251
  98%  16251
  99%  16251
 100%  16251 (longest request)

C:\Apache24\bin>
```

*Şekil 5:10 Requests - Concurrency Level 5*

i) Time taken for tests (seconds) : 27.187
ii) Total transferred (bytes) and HTML transferred (bytes) : 52431510
iii) Time per request (ms) : 2718.691 (mean , across all concurrent requests)
iv) Requests per second (#/sec) : 0.37
v) Transfer rate (Kbytes/sec) : 1883.36
vi) Connection times : can be seen from figure 5

c)

```
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:   29.459 seconds
Complete requests:      10
Failed requests:        0
Total transferred:      52431510 bytes
HTML transferred:       52428800 bytes
Requests per second:    0.34 [#/sec] (mean)
Time per request:       29458.930 [ms] (mean)
Time per request:       2945.893 [ms] (mean, across all concurrent requests)
Transfer rate:          1738.10 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:       76   84   5.5     83      92
Processing: 21467 25595 2223.6  25748   29054
Waiting:       78  330 218.9    349     683
Total:      21550 25679 2223.9  25832   29143

Percentage of the requests served within a certain time (ms)
  50%  25832
  66%  26171
  75%  27211
  80%  28076
  90%  29143
  95%  29143
  98%  29143
  99%  29143
 100%  29143 (longest request)

C:\Apache24\bin>
```

*Şekil 6:10 Requests - Concurrency Level 10*

i) Time taken for tests (seconds) : 29.459
ii) Total transferred (bytes) and HTML transferred (bytes) : 52431510
iii) Time per request (ms) : 2945.893 (mean , across all concurrent requests)
iv) Requests per second (#/sec) : 0.34
v) Transfer rate (Kbytes/sec) : 1738.10
vi) Connection times : can be seen from figure 6

As Concurrency Level goes up ;

Time taken for tests firstly goes down , after concurrency level 10 it does not change much .

Total transferred (bytes) does not change , because data on question does not change , it's still 5mb.

Time per request goes up , since parallel requests goes up , server should spent more time on requests on total

Requests per second goes up . When concurrency level goes up , server will handle more requests at the same time , as a result more requests will be handled per second. Although this is true until some threshold. After that server will be overloaded and Requests per second should go down.

Transfer Rate goes up , Because more requests are coming to the server so it is transferring more data per second.

Connection times generaly goes up , Because more requests are coming from more clients and as a result delay slightly increases and connection times goes up.

d.a)

```
C:\Apache24\bin>ab -k -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:   30.784 seconds
Complete requests:      10
Failed requests:        0
Keep-Alive requests:    10
Total transferred:      52431560 bytes
HTML transferred:       52428800 bytes
Requests per second:    0.32 [#/sec] (mean)
Time per request:       3078.354 [ms] (mean)
Time per request:       3078.354 [ms] (mean, across all concurrent requests)
Transfer rate:          1663.31 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    8  25.1      0      79
Processing:  1960 3070 897.4   3132    4552
Waiting:       79   92  15.1     88     127
Total:       1960 3078 908.3   3132    4552

Percentage of the requests served within a certain time (ms)
  50%   3132
  66%   3570
  75%   3760
  80%   4239
  90%   4552
  95%   4552
  98%   4552
  99%   4552
 100%   4552 (longest request)

C:\Apache24\bin>
```

*Şekil 7:10 Requests - Concurrency Level 1 - Keep Alive*

i) Time taken for tests (seconds) : 30.784
ii) Total transferred (bytes) and HTML transferred (bytes) : 52431560
iii) Time per request (ms) : 3078.354 (mean , across all concurrent requests)
iv) Requests per second (#/sec) : 0.32
v) Transfer rate (Kbytes/sec) : 1663.31
vi) Connection times : can be seen from figure 7

d.b)

```
C:\Apache24\bin>ab -k -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:   29.477 seconds
Complete requests:      10
Failed requests:        0
Keep-Alive requests:    10
Total transferred:      52431560 bytes
HTML transferred:       52428800 bytes
Requests per second:    0.34 [#/sec] (mean)
Time per request:       14738.413 [ms] (mean)
Time per request:       2947.682 [ms] (mean, across all concurrent requests)
Transfer rate:          1737.05 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0   41  43.9     73      93
Processing:  9516 12414 2105.5  12455   16675
Waiting:       73  112  58.1     90     258
Total:       9598 12455 2111.6  12546   16768

Percentage of the requests served within a certain time (ms)
  50%  12546
  66%  13338
  75%  13634
  80%  14053
  90%  16768
  95%  16768
  98%  16768
  99%  16768
 100%  16768 (longest request)

C:\Apache24\bin>
```

*Şekil 8:10 Requests - Concurrency Level 5 - Keep Alive*

i) Time taken for tests (seconds) : 29.477
ii) Total transferred (bytes) and HTML transferred (bytes) : 52431560
iii) Time per request (ms) : 1737.05 (mean , across all concurrent requests)
iv) Requests per second (#/sec) : 0.34
v) Transfer rate (Kbytes/sec) : 1737.05
vi) Connection times : can be seen from figure 8

d.c)

```
C:\Apache24\bin>ab -k -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:   28.799 seconds
Complete requests:      10
Failed requests:        0
Keep-Alive requests:    10
Total transferred:      52431560 bytes
HTML transferred:       52428800 bytes
Requests per second:    0.35 [#/sec] (mean)
Time per request:       28798.809 [ms] (mean)
Time per request:       2879.881 [ms] (mean, across all concurrent requests)
Transfer rate:          1777.94 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:       76   88   9.2     86     102
Processing: 18771 25454 2693.2  25642   28105
Waiting:       79  346 239.1    370     727
Total:      18853 25542 2696.0  25745   28207

Percentage of the requests served within a certain time (ms)
  50%  25745
  66%  26837
  75%  27485
  80%  28124
  90%  28207
  95%  28207
  98%  28207
  99%  28207
 100%  28207 (longest request)

C:\Apache24\bin>
```

*Şekil 9:10 Requests - Concurrency Level 10 - Keep Alive*

i) Time taken for tests (seconds) : 28.799
ii) Total transferred (bytes) and HTML transferred (bytes) : 52431560
iii) Time per request (ms) : 1777.94 (mean , across all concurrent requests)
iv) Requests per second (#/sec) : 0.35
v) Transfer rate (Kbytes/sec) : 1777.94
vi) Connection times : can be seen from figure 9

When Connection Kept Alive , numbers will improve slightly, changes will be minimal.

Time taken for tests goes down , because since connection is being kept alive between requests there will no more delay for reestablishing a connection.

Total transferred (bytes) does not change , because data on question does not change , it's still 5mb.

Time per request goes up , but it should go down slightly because there will be no connection reestablishment delay. These result maybe effected by other things that I'm not aware of.

Requests per second goes down , but it should go up since there is no reconnection on every request. Results were nearly the same how many times I tried.

Transfer Rate goes down , but it should not change since number of requests in total does not change.

Connection times goes down , as it should be because there is no connection delay between requests.

4. Server Stress Test

```
C:\Apache24\bin>ab -n 100 -c 1 http://192.168.1.39:8080/500
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 192.168.1.39 (be patient).....done


Server Software:        HTTPServer/1.0
Server Hostname:        192.168.1.39
Server Port:            8080

Document Path:          /500
Document Length:        500 bytes

Concurrency Level:      1
Time taken for tests:   0.071 seconds
Complete requests:      100
Failed requests:        0
Total transferred:      62300 bytes
HTML transferred:       50000 bytes
Requests per second:    1405.38 [#/sec] (mean)
Time per request:       0.712 [ms] (mean)
Time per request:       0.712 [ms] (mean, across all concurrent requests)
Transfer rate:          855.03 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   0.3      0       1
Processing:     0    0   0.5      0       2
Waiting:        0    0   0.5      0       2
Total:          0    1   0.5      1       2

Percentage of the requests served within a certain time (ms)
  50%      1
  66%      1
  75%      1
  80%      1
  90%      1
  95%      1
  98%      1
  99%      2
 100%      2 (longest request)

C:\Apache24\bin>ab -n 100 -c 1 http://192.168.1.39:8080/500_
```

*Şekil 10: 100 Requests -Concurrency Level 1*

```
C:\Apache24\bin>ab -n 100 -c 10 http://192.168.1.39:8080/500
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 192.168.1.39 (be patient).....done


Server Software:        HTTPServer/1.0
Server Hostname:        192.168.1.39
Server Port:            8080

Document Path:          /500
Document Length:        500 bytes

Concurrency Level:      10
Time taken for tests:   0.063 seconds
Complete requests:      100
Failed requests:        0
Total transferred:      62300 bytes
HTML transferred:       50000 bytes
Requests per second:    1598.82 [#/sec] (mean)
Time per request:       6.255 [ms] (mean)
Time per request:       0.625 [ms] (mean, across all concurrent requests)
Transfer rate:          972.72 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   1.6      0      16
Processing:     0    6   7.7      0      16
Waiting:        0    6   7.7      0      16
Total:          0    6   7.7      0      16

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

C:\Apache24\bin>
```

*Şekil 11:100 Requests - Concurrency Level 10*

```
C:\Apache24\bin>ab -n 100 -c 100 http://192.168.1.39:8080/500
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 192.168.1.39 (be patient).....done


Server Software:        HTTPServer/1.0
Server Hostname:        192.168.1.39
Server Port:            8080

Document Path:          /500
Document Length:        500 bytes

Concurrency Level:      100
Time taken for tests:   0.063 seconds
Complete requests:      100
Failed requests:        0
Total transferred:      62300 bytes
HTML transferred:       50000 bytes
Requests per second:    1598.54 [#/sec] (mean)
Time per request:       62.557 [ms] (mean)
Time per request:       0.626 [ms] (mean, across all concurrent requests)
Transfer rate:          972.55 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   1.6      0      16
Processing:    16   26  10.1     31      47
Waiting:        0   26  10.5     31      47
Total:         16   27  10.3     31      47

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

C:\Apache24\bin>
```

*Şekil 12:100 Requests - Concurrency Level 100*

Even with 100 requests with concurrency level 100 our server could handle the load. All numbers are on reasonable levels.

Proxy Server Stress Test

```
C:\Apache24\bin>ab -n 250 -c 1 -X 127.0.0.1:8888 http://192.168.1.39:8080/500
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 192.168.1.39 [through 127.0.0.1:8888] (be patient)
Completed 100 requests
Completed 200 requests
Finished 250 requests


Server Software:        HTTPServer/1.0
Server Hostname:        192.168.1.39
Server Port:            8080

Document Path:          /500
Document Length:        97 bytes

Concurrency Level:      1
Time taken for tests:   0.172 seconds
Complete requests:      250
Failed requests:        0
Non-2xx responses:      250
Total transferred:      58000 bytes
HTML transferred:       24250 bytes
Requests per second:    1455.08 [#/sec] (mean)
Time per request:       0.687 [ms] (mean)
Time per request:       0.687 [ms] (mean, across all concurrent requests)
Transfer rate:          329.67 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   1.0      0      16
Processing:     0    1   3.1      0      16
Waiting:        0    1   2.9      0      16
Total:          0    1   3.2      0      16

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

C:\Apache24\bin>
```

*Şekil 13: 250 Requests - Concurrency Level 1*

```
C:\Apache24\bin>ab -n 250 -c 100 -X 127.0.0.1:8888 http://192.168.1.39:8080/500
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 192.168.1.39 [through 127.0.0.1:8888] (be patient)
Completed 100 requests
Completed 200 requests
Finished 250 requests


Server Software:        HTTPServer/1.0
Server Hostname:        192.168.1.39
Server Port:            8080

Document Path:          /500
Document Length:        97 bytes

Concurrency Level:      100
Time taken for tests:   0.122 seconds
Complete requests:      250
Failed requests:        0
Non-2xx responses:      250
Total transferred:      58000 bytes
HTML transferred:       24250 bytes
Requests per second:    2052.83 [#/sec] (mean)
Time per request:       48.713 [ms] (mean)
Time per request:       0.487 [ms] (mean, across all concurrent requests)
Transfer rate:          465.09 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   0.4      0       1
Processing:    24   33   6.7     32      55
Waiting:        1   30   8.5     29      55
Total:         24   33   6.8     32      56

Percentage of the requests served within a certain time (ms)
  50%     32
  66%     36
  75%     39
  80%     40
  90%     43
  95%     46
  98%     50
  99%     53
 100%     56 (longest request)

C:\Apache24\bin>ab -n 250 -c 100 -X 127.0.0.1:8888 http://192.168.1.39:8080/500
```

*Şekil 14:250 Requests - Concurrency Level 100*

```
C:\Apache24\bin>ab -n 250 -c 250 -X 127.0.0.1:8888 http://192.168.1.39:8080/500
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 192.168.1.39 [through 127.0.0.1:8888] (be patient)
Completed 100 requests
Completed 200 requests
Finished 250 requests


Server Software:        HTTPServer/1.0
Server Hostname:        192.168.1.39
Server Port:            8080

Document Path:          /500
Document Length:        97 bytes

Concurrency Level:      250
Time taken for tests:   0.115 seconds
Complete requests:      250
Failed requests:        0
Non-2xx responses:      250
Total transferred:      58000 bytes
HTML transferred:       24250 bytes
Requests per second:    2168.48 [#/sec] (mean)
Time per request:       115.288 [ms] (mean)
Time per request:       0.461 [ms] (mean, across all concurrent requests)
Transfer rate:          491.30 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   0.4      0       1
Processing:    32   47   7.6     47      66
Waiting:        0   44   8.6     44      63
Total:         32   47   7.6     47      66

Percentage of the requests served within a certain time (ms)
  50%     47
  66%     51
  75%     53
  80%     55
  90%     57
  95%     59
  98%     62
  99%     64
 100%     66 (longest request)

C:\Apache24\bin>ab -n 250 -c 250 -X 127.0.0.1:8888 http://192.168.1.39:8080/500
```

*Şekil 15:250 Requests - Concurrency Level 250*

As for our Proxy Server numbers can be seen from above figures. Even 250 requests with concurrency level 250 our proxy server could handle it on a reasonable levels.