Marmara University
Faculty of Engineering

Computer Science & Engineering Department

# CSE4074 - Computer Networks

150114050 - Berkay YAMAN
150115048 - Muhammed KILIÇ

# Apache Benchmark Test Results Structure

*without-with proxy,  GET /100-GET /20000,  Requests: 10 - 100,  Concurrency: 1-5-10*

```
ab/
├── without
│   ├── 100
│   │   ├── 10
│   │   │   ├── ok
│   │   │   │   ├──
n10c10.txt
│   │   │   │   ├──
n10c1.txt
│   │   │   │   └──
n10c5.txt
│   │   │   └── wk
│   │   │       ├──
n10kc10.txt
│   │   │       ├──
n10kc1.txt
│   │   │       └──
n10kc5.txt
│   │   └── 100
│   │       ├── ok
│   │       │   ├──
n100c10.txt
│   │       │   ├──
n100c1.txt
│   │       │   └──
n100c5.txt
│   │       └── wk
│   │           ├──
n100kc10.txt
│   │           ├──
n100kc1.txt
│   │           └──
n100kc5.txt
│   └── 20000
│       ├── 10
│       │   ├── ok
│       │   │   ├──
n10c10.txt
│       │   │   ├── n10c1.txt
│       │   │   └── n10c5.txt
│       │   └── wk
│       │       ├──
n100c10.txt
│       │       ├──
n100c1.txt
│       │       └──
n100c5.txt
│       └── 100
│           ├── ok
│           │   ├──
n10kc10.txt
│           │   ├──
n10kc1.txt
│           │   └──
n10kc5.txt
│           └── wk
│               ├──
n100kc10.txt
│               ├──
n100kc1.txt
│               └──
n100kc5.txt
└── with-proxy
    ├── 100
    │   ├── 10
    │   │   ├── ok
    │   │   │   ├──
n10c10.txt
    │   │   │   ├── n10c1.txt
    │   │   │   └── n10c5.txt
    │   │   └── wk
    │   │       ├──
n10kc10.txt
    │   │       ├──
n10kc1.txt
    │   │       └──
n10kc5.txt
    │   └── 100
    │       ├── ok
    │       │   ├──
n100c10.txt
    │       │   ├──
n100c1.txt
    │       │   └──
n100c5.txt
    │       └── wk
    │           ├──
n100kc10.txt
    │           ├──
n100kc1.txt
    │           └──
n100kc5.txt
    └── 20000
        ├── 10
        │   ├── ok
        │   │   ├──
n10c10.txt
        │   │   ├── n10c1.txt
        │   │   └── n10c5.txt
        │   └── wk
        │       ├──
n10kc10.txt
        │       ├── n10kc1.txt
        │       └── n10kc5.txt
        └── 100
            ├── ok
            │   ├──
n100c10.txt
            │   ├── n100c1.txt
            │   └── n100c5.txt
            └── wk
                ├──
n100kc10.txt
                ├── n100kc1.txt
                ├── n100kc5.txt
                ├── n10kc10.txt
                ├── n10kc1.txt
                └── n10kc5.txt
```

*30 directories, 51 files*

## HTTP Server Response Behaviour (Python3)

```python
def get_file(request):
   headers = request.split('\n')
   filename = headers[0].split()[1]
   method = headers[0].split()[0]
   if method == 'GET':
       if filename == '/':
           filename = '/200.html'
       else:
           path = filename[1:]
           try:
               file_size = int(path)
               in_limit = 100 <= file_size <= 20000
               if(in_limit):
                   create_file(file_size)
                   filename = '/index.html'
               else:
                   filename = '/400.html'
           except:
               filename = '/400.html'
   else: # if not GET show 501
       filename = '/501.html'
   return filename
```

## Proxy Server Multithreading (Java)

```java
public class ProxyMain {
   static Map cache = new HashMap();
   public static void main(String args[]) throws Exception {
       int port = 8888;
       ServerSocket server = new ServerSocket(port);
       System.out.println("\nPort " + port + " is listening...\n");
       try {while(true){
               Socket connected = server.accept();
               System.out.println("Connection Request Accepted From
"+connected.getRemoteSocketAddress().toString());
               Proxy proxy = new Proxy(connected);
               //proxy.run();
               Thread thread = new Thread(proxy);
               thread.start();
           }} catch (Exception e) {
           e.printStackTrace();
       }}}
```

# Analysis

**ab/with-proxy/100/100/ok/n100c10-short.txt**

```
Time taken for tests:   0.085 seconds
Total transferred:      18000 bytes
HTML transferred:       10000 bytes
Time per request:       8.488 [ms] (mean)
Time per request:       0.849 [ms] (mean, across all concurrent requests)
Transfer rate:          207.09 [Kbytes/sec] received
```

**ab/with-proxy/100/100/wk/n100kc10-short.txt**

```
Time taken for tests:   0.078 seconds
Total transferred:      18000 bytes
HTML transferred:       10000 bytes
Time per request:       7.821 [ms] (mean)
Time per request:       0.782 [ms] (mean, across all concurrent requests)
Transfer rate:          224.75 [Kbytes/sec] received
```

For localhost:5000/100 when we send 100 requests with 10 concurrency time taken for tests is shorter for keep alive parameter.

**ab/with-proxy/20000/100/wk/n100kc10-short.txt**

```
Time taken for tests:   1.755 seconds
Total transferred:      2008200 bytes
HTML transferred:       2000000 bytes
Time per request:       175.539 [ms] (mean)
Time per request:       17.554 [ms] (mean, across all concurrent requests)
Transfer rate:          1117.21 [Kbytes/sec] received
```

**ab/with-proxy/20000/100/ok/n100c10-short.txt**

```
Time taken for tests:   1.755 seconds
Total transferred:      2008200 bytes
HTML transferred:       2000000 bytes
Time per request:       175.548 [ms] (mean)
Time per request:       17.555 [ms] (mean, across all concurrent requests)
Transfer rate:          1117.15 [Kbytes/sec] received
```

For localhost:5000/20000 when we send 100 requests with 10 concurrency time taken for tests this time we see there is not a remarkable change and then when we check transfer rate it makes sense and we can say all client, proxy and http servers are on the same computer so that the transfer depends on file transfer rate. Which makes sense.

**ab/with-proxy/100/10/ok/n10c1-short.txt**

```
Time taken for tests:   0.010 seconds
Total transferred:      1800 bytes
HTML transferred:       1000 bytes
Time per request:       1.015 [ms] (mean)
Time per request:       1.015 [ms] (mean, across all concurrent requests)
Transfer rate:          173.17 [Kbytes/sec] received
```

**ab/without/100/10/ok/n10c1.txt**

```
Time taken for tests:   0.008 seconds
Total transferred:      1800 bytes
HTML transferred:       1000 bytes
Time per request:       0.808 [ms] (mean)
Time per request:       0.808 [ms] (mean, across all concurrent requests)
Transfer rate:          217.66 [Kbytes/sec] received
```

**ab/with-proxy/20000/10/ok/n10c1-short.txt**

```
Time taken for tests:   0.185 seconds
Total transferred:      200820 bytes
HTML transferred:       200000 bytes
Time per request:       18.468 [ms] (mean)
Time per request:       18.468 [ms] (mean, across all concurrent requests)
Transfer rate:          1061.91 [Kbytes/sec] received
```

**ab/without/20000/10/ok/n10c1.txt**

```
Time taken for tests:   0.173 seconds
Total transferred:      200820 bytes
HTML transferred:       200000 bytes
Time per request:       17.317 [ms] (mean)
Time per request:       17.317 [ms] (mean, across all concurrent requests)
Transfer rate:          1132.49 [Kbytes/sec] received
```

In both lower and upper limits we can see that if the proxy is open the transfer rate is a little bit slower which is reasonable because all requests get to the proxy server and then http server. By that reason time per requests are also a little smaller according to the results when proxy is online.

**ab/with-proxy/20000/100/wk/n100kc1.txt**

```
Requests per second:    56.52 [#/sec] (mean)
Time per request:       17.693 [ms] (mean)
Time per request:       17.693 [ms] (mean, across all concurrent requests)
```

**ab/with-proxy/20000/100/wk/n100kc5.txt**

```
Requests per second:    57.43 [#/sec] (mean)
Time per request:       87.070 [ms] (mean)
Time per request:       17.414 [ms] (mean, across all concurrent requests)
```

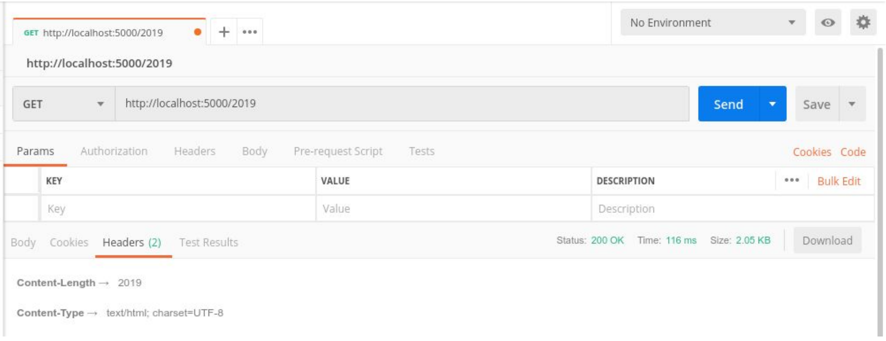
**ab/with-proxy/20000/100/wk/n100kc10.txt**

```
Requests per second:    57.40 [#/sec] (mean)
Time per request:       174.224 [ms] (mean)
Time per request:       17.422 [ms] (mean, across all concurrent requests)
```

The tests above show that almost all concurrent requests take the same time mean 1.7 seconds and the total time per requests increases when concurrent request number increases.

## Screen Shots

localhost:5000/2019 -> 200 OK



GET http://localhost:5000/2019

http://localhost:5000/2019

GET http://localhost:5000/2019

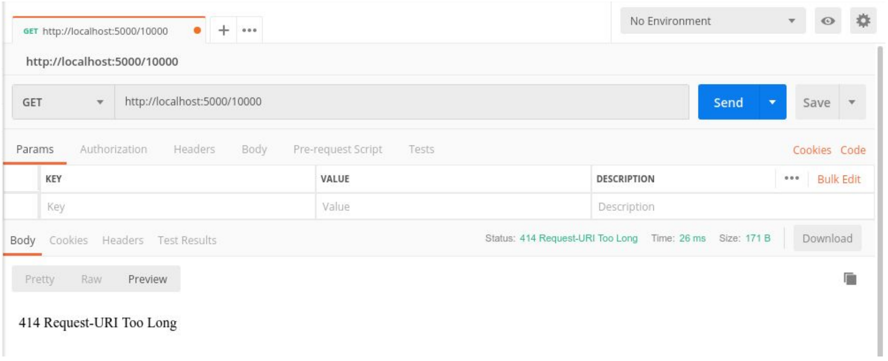
| KEY | VALUE | DESCRIPTION |
|---|---|---|
| Key | Value | Description |

Status: 200 OK Time: 116 ms Size: 2.05 KB

Content-Length → 2019

Content-Type → text/html; charset=UTF-8

localhost:5000/10000 -> 414 URI Too Long Error



GET http://localhost:5000/10000

http://localhost:5000/10000

GET http://localhost:5000/10000

| KEY | VALUE | DESCRIPTION |
|---|---|---|
| Key | Value | Description |

Status: 414 Request-URI Too Long Time: 26 ms Size: 171 B

414 Request-URI Too Long