Laporan Ujian Akhir Semester

Pemrograman Jaringan C

Kelompok 2

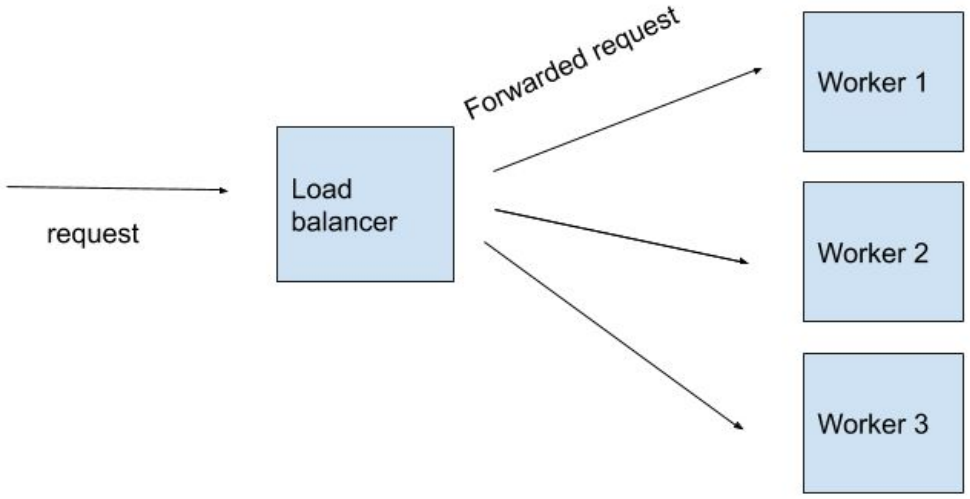
05111740000049	Yemima Sutanto
05111740000093	Ahmad Syauqi
05111740000127	Elkana Hans W
05111740000135	I Gede Agung K P
05111740000154	Raehan

Departemen Teknik Informatika
Fakultas Teknik Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya

2020

SOAL

Sistem yang diminta seperti pada gambar berikut.



Dalam sebuah sistem yang melayani banyak request, terkadang sebuah server tidak mampu melayani, diperlukan mekanisme sistem terdistribusi agar beban dan request dapat dilayani secara berbagi oleh worker yang ada. Buatlah sistem seperti pada gambar, untuk melayani request dalam jumlah besar. Gunakan performance software tester ab (https://httpd.apache.org/docs/2.4/programs/ab.html) untuk melakukan test

Buatlah tabel eksperimen yang menunjukan kemampuan sistem anda dalam melayani request yang besar dan juga konkurensi yang banyak. Model processing server (entah itu asynchronous / thread based) ditentukan oleh kelompok. Diperbolehkan teknik-teknik yang memungkinkan worker bisa bertambah secara otomatis jika dibutuhkan.

Buatlah juga sebuah video penjelasan yang memperlihat langkah-langkah menjalankan program, percobaan penambahan worker otomatis jika dibutuhkan, dan juga dalam video diberi penjelasan singkat code yang dibuat.

Buat laporan dalam bentuk PDF yang berisi penjelasan mengenai code, model processing server, mekanisme penambahan worker otomatis jika dibutuhkan, dan hasil tabel eksperimen dilengkapi dengan kesimpulan dari hasil percobaan.

Source code lengkap dapat diakses pada:
https://github.com/yemimasutanto/uas-progjar-c-2020

A. Penjelasan Kode

```
8   port_num = 9005
9
10  class BackendList:
11      def __init__(self):
12          self.servers=[]
13          self.current = 0
14          self.request = 0
15      def running_async(self,port):
16          bw = BackWorker(port)
17      def setserver(self,portnumber):
18          self.servers.append(('127.0.0.1',portnumber))
19      def getserver(self):
20          global port_num
21          s = self.servers[self.current]
22          self.current += 1
23          self.request += 1
24          print(self.request)
25          if (self.current>=len(self.servers)):
26              self.current=0
27          if(self.request % 100 == 0):
28              port_num += 1
29              self.running_async(port_num)
30              self.setserver(port_num)
31          return s
```

Gambar 1 Source Code lb.py (class BackendList)

Disini kita membuat variabel global port_num yang nantinya akan bertambah sesuai jumlah worker.

Class BackendList: untuk mendefinisikan dan menambah worker

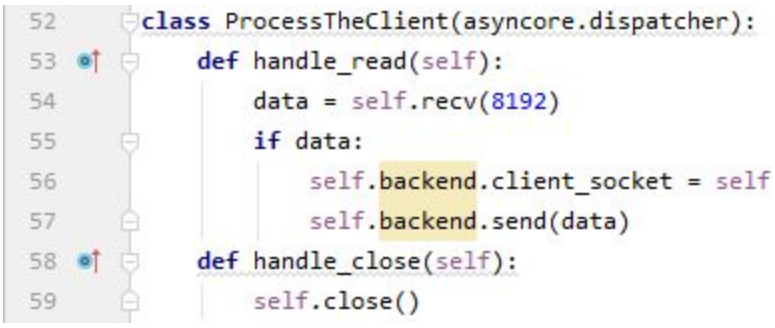
- def __init__(self): untuk inisialisasi variabel yang dibutuhkan
- def running_async(self, port): untuk menjalankan async_server
- def setserver(self, portnumber): untuk menyimpan worker
- def getserver(self): untuk melakukan penerusan koneksi dari worker satu ke worker yang lain, dan juga bila request sudah kelipatan 100 maka akan menambah worker

```
class Backend(asyncore.dispatcher_with_send):
    def __init__(self,targetaddress):
        asyncore.dispatcher_with_send.__init__(self)
        self.create_socket(socket.AF_INET, socket.SOCK_STREAM)
        self.connect(targetaddress)
        self.connection = self

    def handle_read(self):
        try:
            self.client_socket.send(self.recv(8192))
        except:
            pass
    def handle_close(self):
        try:
            self.close()
            self.client_socket.close()
        except:
            pass
```

Gambar 2 Class Backend(asyncore.dispatcher_with_send)

Class Backend(asyncore.dispatcher_with_send): class yang membuat asynchronous socket

- def __init__(self, targetaddress): melakukan inisialisasi
- def handle_read(self): mengirim data ke client
- def handle_close(self): memutuskan koneksi ke client



```
class ProcessTheClient(asyncore.dispatcher):
    def handle_read(self):
        data = self.recv(8192)
        if data:
            self.backend.client_socket = self
            self.backend.send(data)
    def handle_close(self):
        self.close()
```

Gambar 3 Class ProcessTheClient(asyncore.dispatcher)

Class ProcessTheClient(asyncore.dispatcher): melakukan proses ke client

- def handle_read(self) mengirim data ke client
- def handle_close(self): memutuskan koneksi ke client

```
61  class Server(asyncore.dispatcher):
62      def __init__(self,portnumber):
63          asyncore.dispatcher.__init__(self)
64          self.create_socket(socket.AF_INET, socket.SOCK_STREAM)
65          self.set_reuse_addr()
66          self.bind(('',portnumber))
67          self.listen(1)
68          self.bservers = BackendList()
69          self.bservers.running_async(9002)
70          self.bservers.running_async(9003)
71          self.bservers.running_async(9004)
72          self.bservers.running_async(9005)
73          self.bservers.setserver(9002)
74          self.bservers.setserver(9003)
75          self.bservers.setserver(9004)
76          self.bservers.setserver(9005)
77
78          logging.warning("load balancer running on port {}" . format(portnumber))
79
80      def handle_accept(self):
81          pair = self.accept()
82          if pair is not None:
83              sock, addr = pair
84              logging.warning("connection from {}" . format(repr(addr)))
85
86              #menentukan ke server mana request akan diteruskan
87              bs = self.bservers.getserver()
88              logging.warning("koneksi dari {} diteruskan ke {}" . format(addr, bs))
89              backend = Backend(bs)
90
91              #mendapatkan handler dan socket dari client
92              handler = ProcessTheClient(sock)
93              handler.backend = backend
```

Gambar 4 Class Server(asyncore.dispatcher)

Class Server(asyncore.dispatcher): Melakukan proses server

- def __init__(self.portnumber): Melakukan inisialisasi 4 worker, menjalankan server
- def handle_accept(self): Melakukan penerusan koneksi, dan mendapatkan handler dari client

B. Model Processing Server

Kami menggunakan model asynchronous server. Cara kerjanya yaitu Server berjalan pada satu thread melakukan never ending loop, dan memonitor event/kejadian pada socket tersebut. Event ada socket bisa berupa Accept, Read, dan Write.

C. Mekanisme Penambahan Worker

Untuk mekanisme penambahan worker, pada awalnya mempunyai 5 worker. Setelah request mencapai kelipatan 100, maka worker akan bertambah satu.

D. Hasil Performance Test

Menggunakan ab testing, dengan syntax:
ab -n 1000 -c 1,50,100,500,1000 http://127.0.0.1:4444/
Dengan ketentuan:

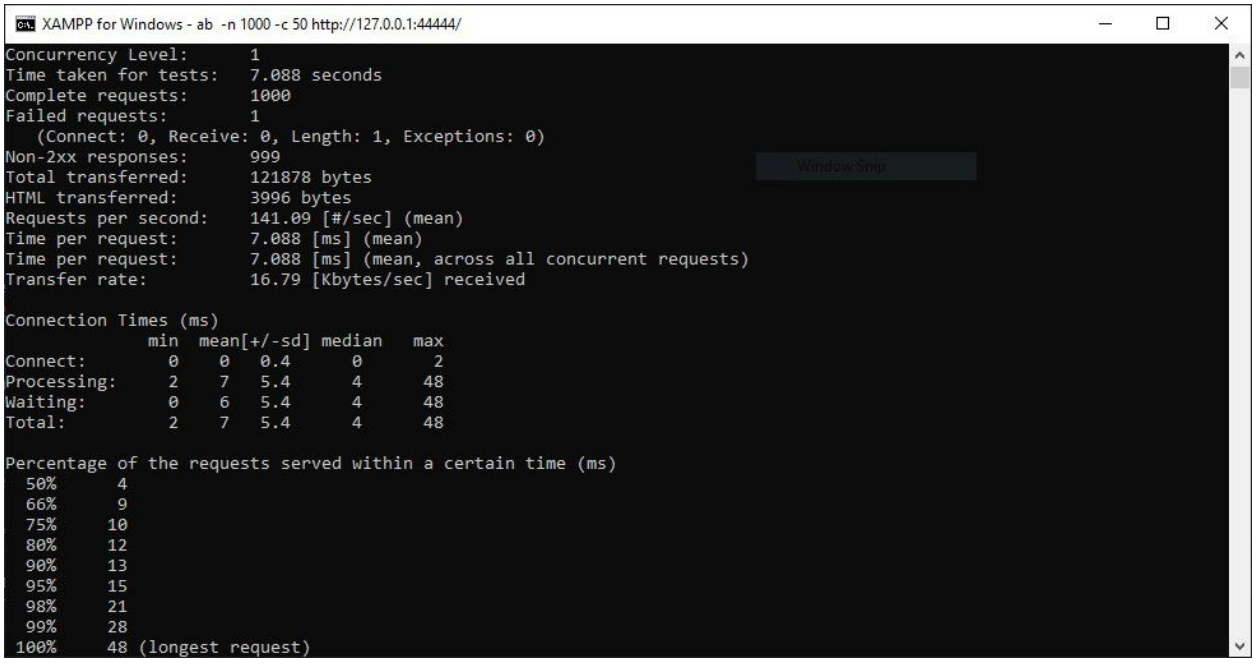
1. Jumlah Request: 1000
2. Concurrency: 1,50,100,500,1000

No test	Concurrency level	Time taken for test	Complete request	Failed request	Total transferred	Request per second	Time per request	Transfer rate
1	1	7.088	1000	1	121878	141.09	7.088	16.79
2	50	250.704	1000	2	121756	3.99	12535.197	0.47
3	100	254.408	1000	0	115000	3.93	25440.757	0.44
4	500	253.213	1000	0	122000	3.95	126606.448	0.47
5	1000	257.6	1000	0	115000	3.88	257599.627	0.44

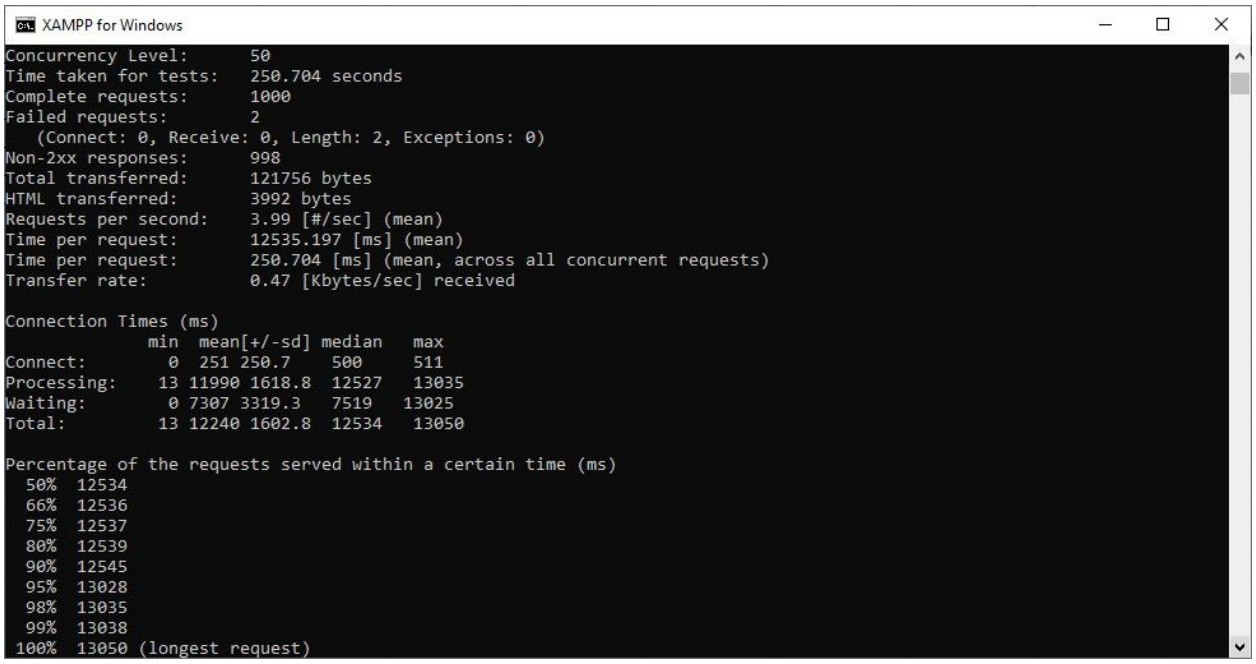
Kesimpulan:

Dari hasil uji coba yang telah kami lakukan, penggunaan asynchronous dengan load balancing menyebabkan waktu eksekusi program lebih cepat dan program kami dapat menangani suatu proses dengan menggunakan *multi worker*.

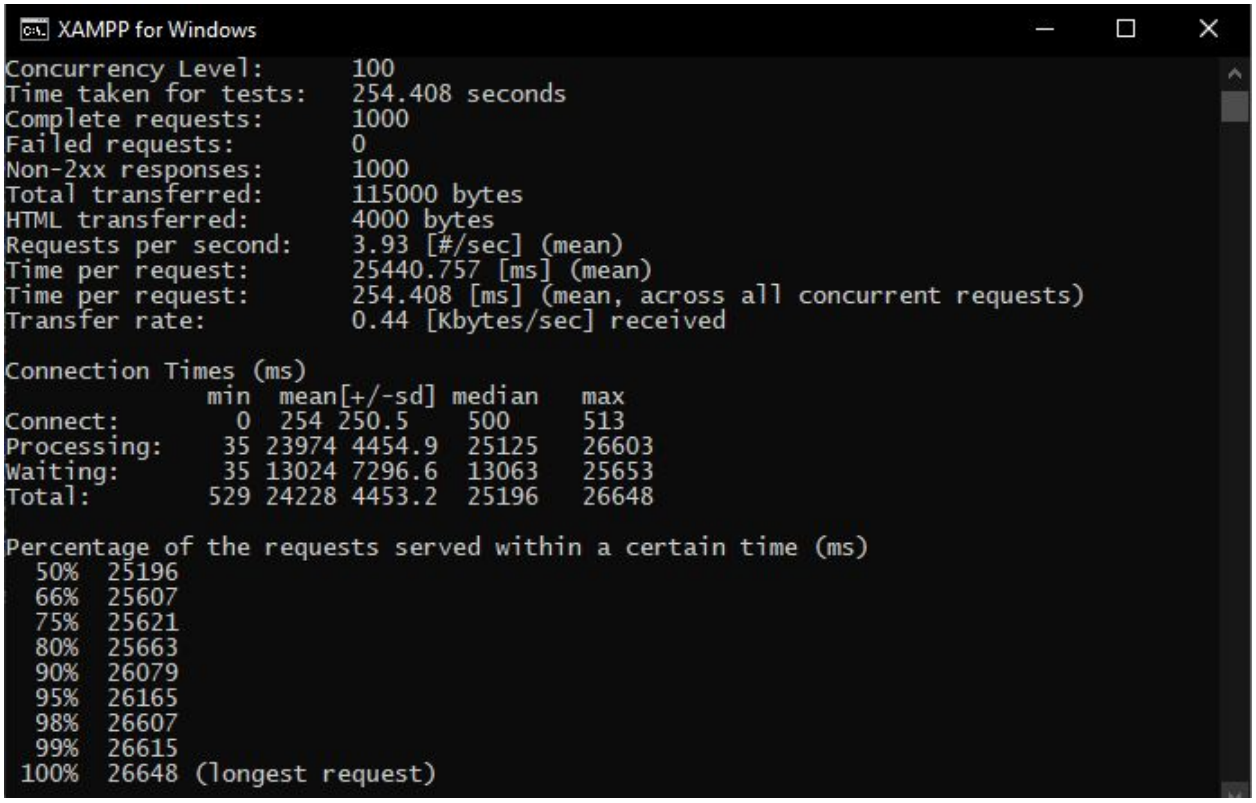
E. Lampiran

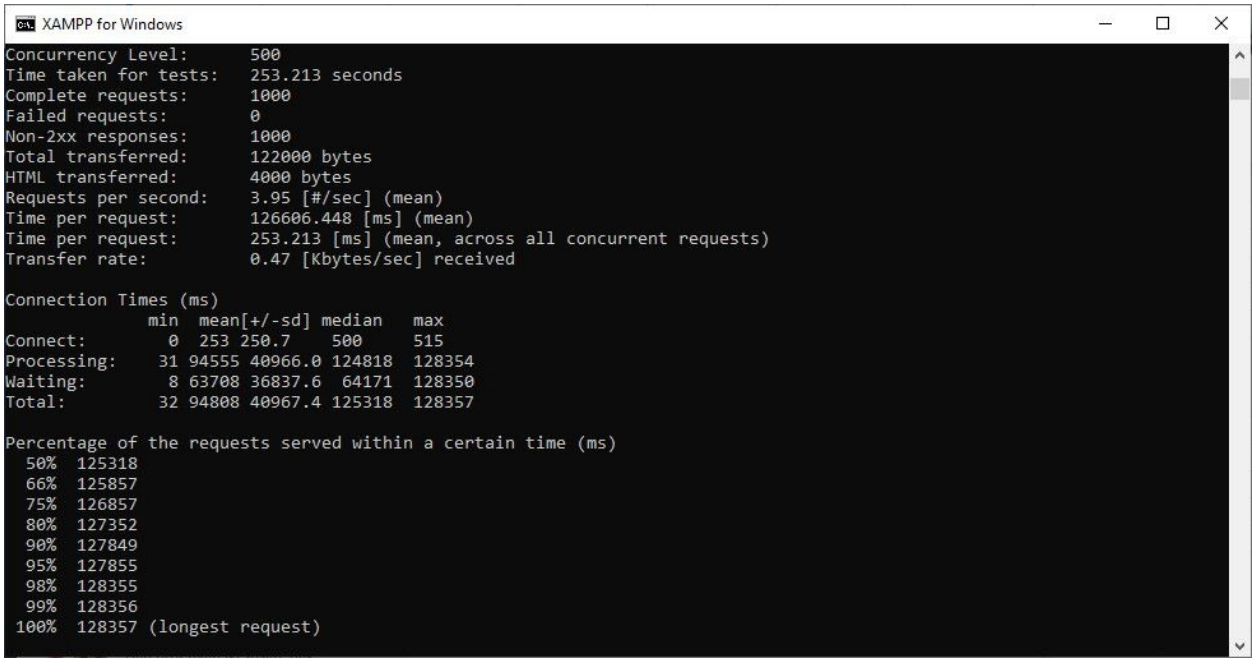


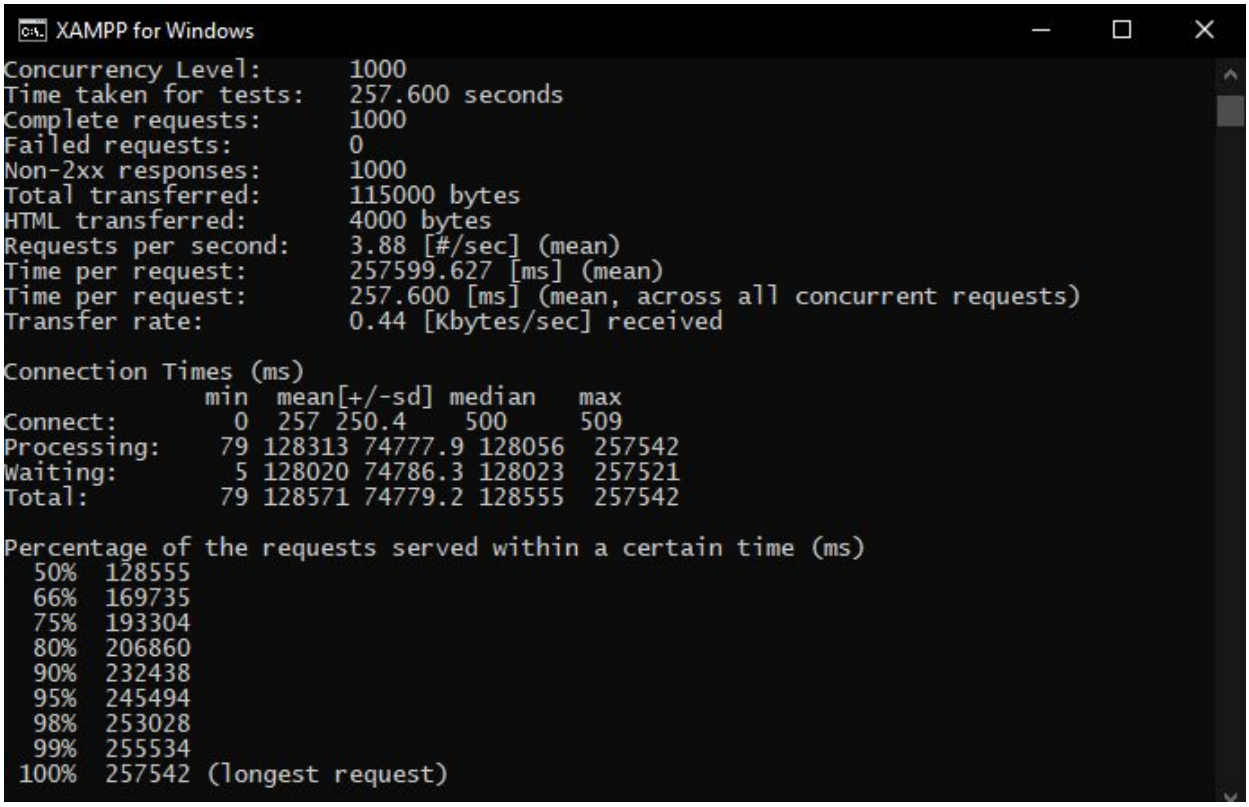
```
XAMPP for Windows - ab  -n 1000 -c 50 http://127.0.0.1:44444/
Concurrency Level:      1
Time taken for tests:   7.088 seconds
Complete requests:      1000
Failed requests:        1
   (Connect: 0, Receive: 0, Length: 1, Exceptions: 0)
Non-2xx responses:      999
Total transferred:      121878 bytes
HTML transferred:       3996 bytes
Requests per second:    141.09 [#/sec] (mean)
Time per request:       7.088 [ms] (mean)
Time per request:       7.088 [ms] (mean, across all concurrent requests)
Transfer rate:          16.79 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   0.4      0       2
Processing:     2    7   5.4      4      48
Waiting:        0    6   5.4      4      48
Total:          2    7   5.4      4      48

Percentage of the requests served within a certain time (ms)
  50%      4
  66%      9
  75%     10
  80%     12
  90%     13
  95%     15
  98%     21
  99%     28
 100%     48 (longest request)
```

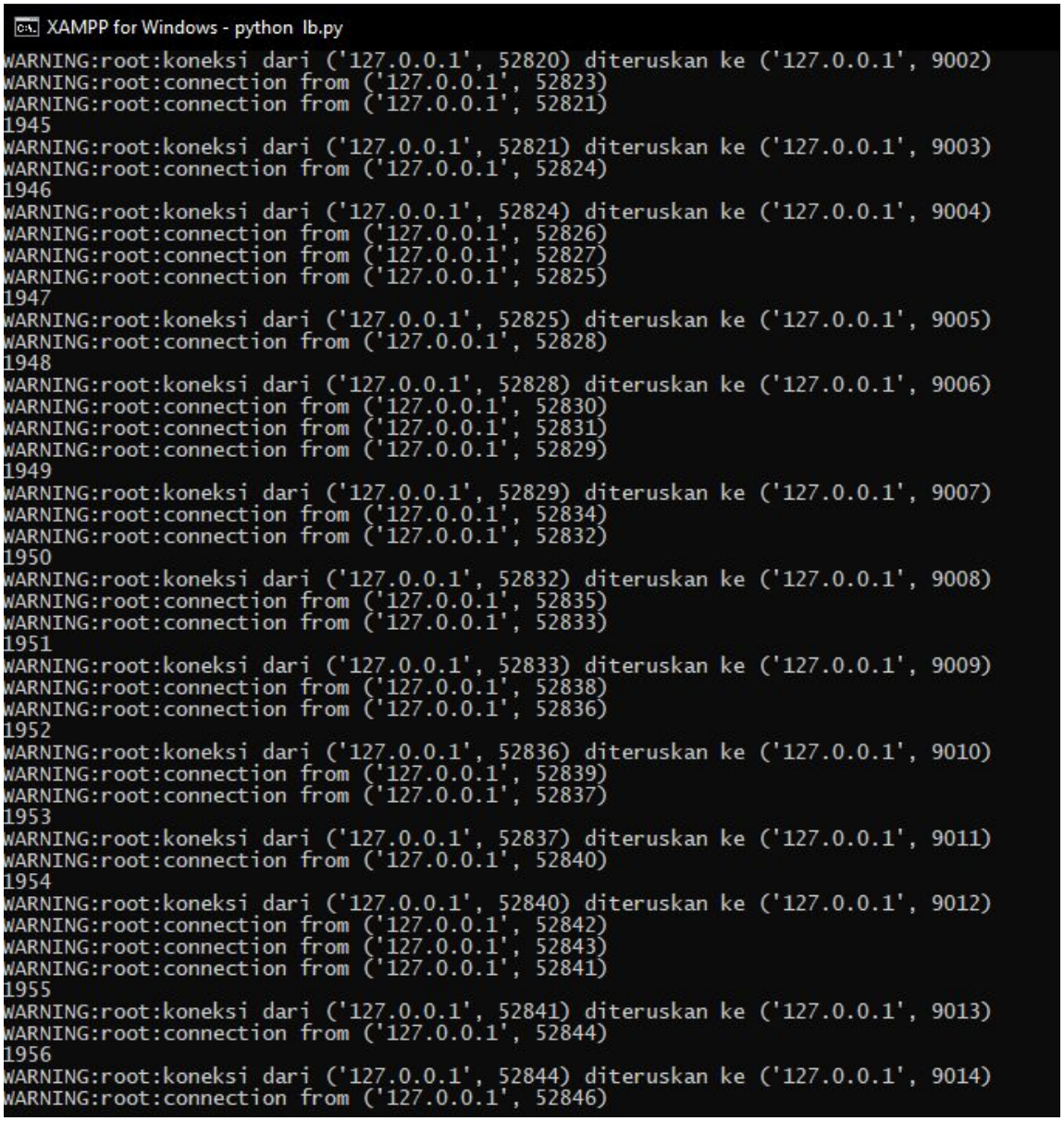


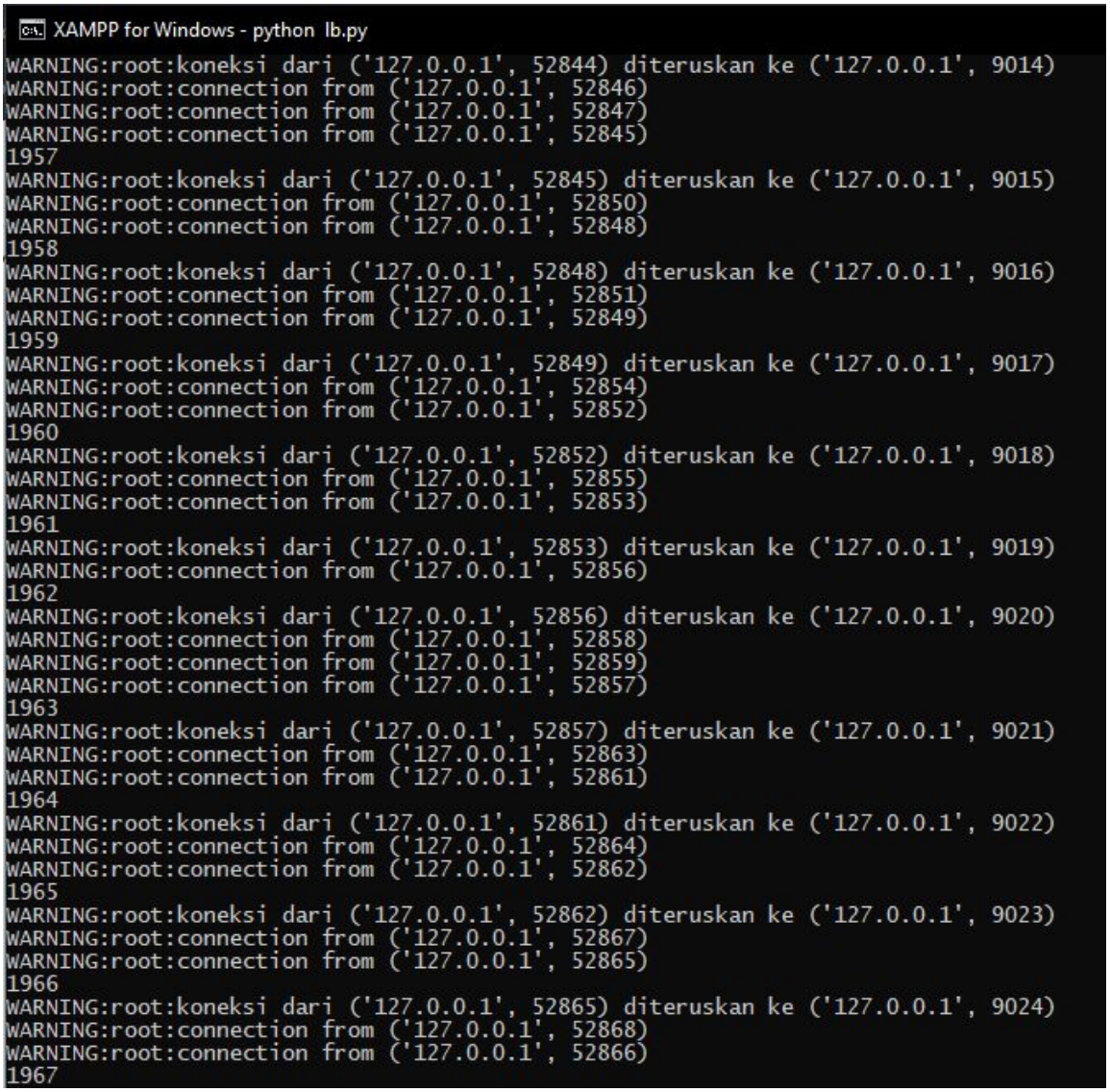
```
XAMPP for Windows
Concurrency Level:      50
Time taken for tests:   250.704 seconds
Complete requests:      1000
Failed requests:        2
   (Connect: 0, Receive: 0, Length: 2, Exceptions: 0)
Non-2xx responses:      998
Total transferred:      121756 bytes
HTML transferred:       3992 bytes
Requests per second:    3.99 [#/sec] (mean)
Time per request:       12535.197 [ms] (mean)
Time per request:       250.704 [ms] (mean, across all concurrent requests)
Transfer rate:          0.47 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0  251 250.7    500     511
Processing:    13 11990 1618.8  12527   13035
Waiting:        0 7307 3319.3   7519   13025
Total:         13 12240 1602.8  12534   13050

Percentage of the requests served within a certain time (ms)
  50%  12534
  66%  12536
  75%  12537
  80%  12539
  90%  12545
  95%  13028
  98%  13035
  99%  13038
 100%  13050 (longest request)
```


XAMPP for Windows
Concurrency Level: 100
Time taken for tests: 254.408 seconds
Complete requests: 1000
Failed requests: 0
Non-2xx responses: 1000
Total transferred: 115000 bytes
HTML transferred: 4000 bytes
Requests per second: 3.93 [#/sec] (mean)
Time per request: 25440.757 [ms] (mean)
Time per request: 254.408 [ms] (mean, across all concurrent requests)
Transfer rate: 0.44 [Kbytes/sec] received
Connection Times (ms)
min mean[+/-sd] median max
Connect: 0 254 250.5 500 513
Processing: 35 23974 4454.9 25125 26603
Waiting: 35 13024 7296.6 13063 25653
Total: 529 24228 4453.2 25196 26648
Percentage of the requests served within a certain time (ms)
50% 25196
66% 25607
75% 25621
80% 25663
90% 26079
95% 26165
98% 26607
99% 26615
100% 26648 (longest request)


XAMPP for Windows
Concurrency Level: 500
Time taken for tests: 253.213 seconds
Complete requests: 1000
Failed requests: 0
Non-2xx responses: 1000
Total transferred: 122000 bytes
HTML transferred: 4000 bytes
Requests per second: 3.95 [#/sec] (mean)
Time per request: 126606.448 [ms] (mean)
Time per request: 253.213 [ms] (mean, across all concurrent requests)
Transfer rate: 0.47 [Kbytes/sec] received
Connection Times (ms)
min mean[+/-sd] median max
Connect: 0 253 250.7 500 515
Processing: 31 94555 40966.0 124818 128354
Waiting: 8 63708 36837.6 64171 128350
Total: 32 94808 40967.4 125318 128357
Percentage of the requests served within a certain time (ms)
50% 125318
66% 125857
75% 126857
80% 127352
90% 127849
95% 127855
98% 128355
99% 128356
100% 128357 (longest request)


XAMPP for Windows
Concurrency Level: 1000
Time taken for tests: 257.600 seconds
Complete requests: 1000
Failed requests: 0
Non-2xx responses: 1000
Total transferred: 115000 bytes
HTML transferred: 4000 bytes
Requests per second: 3.88 [#/sec] (mean)
Time per request: 257599.627 [ms] (mean)
Time per request: 257.600 [ms] (mean, across all concurrent requests)
Transfer rate: 0.44 [Kbytes/sec] received
Connection Times (ms)
min mean[+/-sd] median max
Connect: 0 257 250.4 500 509
Processing: 79 128313 74777.9 128056 257542
Waiting: 5 128020 74786.3 128023 257521
Total: 79 128571 74779.2 128555 257542
Percentage of the requests served within a certain time (ms)
50% 128555
66% 169735
75% 193304
80% 206860
90% 232438
95% 245494
98% 253028
99% 255534
100% 257542 (longest request)


XAMPP for Windows - python lb.py
WARNING:root:koneksi dari ('127.0.0.1', 52820) diteruskan ke ('127.0.0.1', 9002)
WARNING:root:connection from ('127.0.0.1', 52823)
WARNING:root:connection from ('127.0.0.1', 52821)
1945
WARNING:root:koneksi dari ('127.0.0.1', 52821) diteruskan ke ('127.0.0.1', 9003)
WARNING:root:connection from ('127.0.0.1', 52824)
1946
WARNING:root:koneksi dari ('127.0.0.1', 52824) diteruskan ke ('127.0.0.1', 9004)
WARNING:root:connection from ('127.0.0.1', 52826)
WARNING:root:connection from ('127.0.0.1', 52827)
WARNING:root:connection from ('127.0.0.1', 52825)
1947
WARNING:root:koneksi dari ('127.0.0.1', 52825) diteruskan ke ('127.0.0.1', 9005)
WARNING:root:connection from ('127.0.0.1', 52828)
1948
WARNING:root:koneksi dari ('127.0.0.1', 52828) diteruskan ke ('127.0.0.1', 9006)
WARNING:root:connection from ('127.0.0.1', 52830)
WARNING:root:connection from ('127.0.0.1', 52831)
WARNING:root:connection from ('127.0.0.1', 52829)
1949
WARNING:root:koneksi dari ('127.0.0.1', 52829) diteruskan ke ('127.0.0.1', 9007)
WARNING:root:connection from ('127.0.0.1', 52834)
WARNING:root:connection from ('127.0.0.1', 52832)
1950
WARNING:root:koneksi dari ('127.0.0.1', 52832) diteruskan ke ('127.0.0.1', 9008)
WARNING:root:connection from ('127.0.0.1', 52835)
WARNING:root:connection from ('127.0.0.1', 52833)
1951
WARNING:root:koneksi dari ('127.0.0.1', 52833) diteruskan ke ('127.0.0.1', 9009)
WARNING:root:connection from ('127.0.0.1', 52838)
WARNING:root:connection from ('127.0.0.1', 52836)
1952
WARNING:root:koneksi dari ('127.0.0.1', 52836) diteruskan ke ('127.0.0.1', 9010)
WARNING:root:connection from ('127.0.0.1', 52839)
WARNING:root:connection from ('127.0.0.1', 52837)
1953
WARNING:root:koneksi dari ('127.0.0.1', 52837) diteruskan ke ('127.0.0.1', 9011)
WARNING:root:connection from ('127.0.0.1', 52840)
1954
WARNING:root:koneksi dari ('127.0.0.1', 52840) diteruskan ke ('127.0.0.1', 9012)
WARNING:root:connection from ('127.0.0.1', 52842)
WARNING:root:connection from ('127.0.0.1', 52843)
WARNING:root:connection from ('127.0.0.1', 52841)
1955
WARNING:root:koneksi dari ('127.0.0.1', 52841) diteruskan ke ('127.0.0.1', 9013)
WARNING:root:connection from ('127.0.0.1', 52844)
1956
WARNING:root:koneksi dari ('127.0.0.1', 52844) diteruskan ke ('127.0.0.1', 9014)
WARNING:root:connection from ('127.0.0.1', 52846)


XAMPP for Windows - python lb.py
WARNING:root:koneksi dari ('127.0.0.1', 52844) diteruskan ke ('127.0.0.1', 9014)
WARNING:root:connection from ('127.0.0.1', 52846)
WARNING:root:connection from ('127.0.0.1', 52847)
WARNING:root:connection from ('127.0.0.1', 52845)
1957
WARNING:root:koneksi dari ('127.0.0.1', 52845) diteruskan ke ('127.0.0.1', 9015)
WARNING:root:connection from ('127.0.0.1', 52850)
WARNING:root:connection from ('127.0.0.1', 52848)
1958
WARNING:root:koneksi dari ('127.0.0.1', 52848) diteruskan ke ('127.0.0.1', 9016)
WARNING:root:connection from ('127.0.0.1', 52851)
WARNING:root:connection from ('127.0.0.1', 52849)
1959
WARNING:root:koneksi dari ('127.0.0.1', 52849) diteruskan ke ('127.0.0.1', 9017)
WARNING:root:connection from ('127.0.0.1', 52854)
WARNING:root:connection from ('127.0.0.1', 52852)
1960
WARNING:root:koneksi dari ('127.0.0.1', 52852) diteruskan ke ('127.0.0.1', 9018)
WARNING:root:connection from ('127.0.0.1', 52855)
WARNING:root:connection from ('127.0.0.1', 52853)
1961
WARNING:root:koneksi dari ('127.0.0.1', 52853) diteruskan ke ('127.0.0.1', 9019)
WARNING:root:connection from ('127.0.0.1', 52856)
1962
WARNING:root:koneksi dari ('127.0.0.1', 52856) diteruskan ke ('127.0.0.1', 9020)
WARNING:root:connection from ('127.0.0.1', 52858)
WARNING:root:connection from ('127.0.0.1', 52859)
WARNING:root:connection from ('127.0.0.1', 52857)
1963
WARNING:root:koneksi dari ('127.0.0.1', 52857) diteruskan ke ('127.0.0.1', 9021)
WARNING:root:connection from ('127.0.0.1', 52863)
WARNING:root:connection from ('127.0.0.1', 52861)
1964
WARNING:root:koneksi dari ('127.0.0.1', 52861) diteruskan ke ('127.0.0.1', 9022)
WARNING:root:connection from ('127.0.0.1', 52864)
WARNING:root:connection from ('127.0.0.1', 52862)
1965
WARNING:root:koneksi dari ('127.0.0.1', 52862) diteruskan ke ('127.0.0.1', 9023)
WARNING:root:connection from ('127.0.0.1', 52867)
WARNING:root:connection from ('127.0.0.1', 52865)
1966
WARNING:root:koneksi dari ('127.0.0.1', 52865) diteruskan ke ('127.0.0.1', 9024)
WARNING:root:connection from ('127.0.0.1', 52868)
WARNING:root:connection from ('127.0.0.1', 52866)
1967