

## Tugas 5

Dengan memanfaatkan program pada progjar6  
(<https://github.com/rm77/progjar/tree/master/progjar6>),

- Implementasikan arsitektur load balancer dengan
  - Mode asynchronous
  - Mode server pool
- Buatlah perbandingan kinerja web server
- Buatlah gambar dari arsitektur percobaan
- Untuk pengukuran kinerja, gunakan tool *wrk* dengan jumlah request/koneksi 1000, dengan parameter concurrency 10,50,100,150,200

### 1. Implementasi

#### a) mode asynchronous

saya melakukan perubahan kode pada `il_async.py` pada bagian class `BackendList` untuk menggantikan port agar sesuai dengan environment yang ingin dijalankan. Pada `runserver.sh` akan dijalankan server dengan port berikut

```
1 #jalankan 5 async_server
2
3 python3 async_server.py 9002 &
4 python3 async_server.py 9003 &
5 python3 async_server.py 9004 &
6 python3 async_server.py 9005 &
```

Maka saya mengganti `il_async.py` agar sesuai port.

```
8 class BackendList:
9     def __init__(self):
10         self.servers=[]
11         self.servers.append(('127.0.0.1',9002))
12         self.servers.append(('127.0.0.1',9003))
13         self.servers.append(('127.0.0.1',9004))
14         self.servers.append(('127.0.0.1',9005))
15         self.current=0
```

Pada percobaan nanti akan ada lebih banyak data yang diterima dari client, sehingga saya menggantikan ukuran maksimum jumlah byte yang diterima dari client. Awalnya hanya 32 byte saya menggantinya menjadi 4 kb supaya tidak terjadi error nantinya.

```
32 def handle_read(self):
33     if self.client_socket:
34         try:
35             data = self.recv(4096) # meningkatkan penerimaan data dari client sebanyak 4 kb
36             if data:
37                 self.client_socket.send(data)
38         except Exception as e:
39             logging.warning("Backend handle_read error: {}".format(e))
40             self.close()
```

Untuk melakukan percobaan, saya menjalankan `python3 il_async.py` di terminal dan ini hasilnya.

```
(base) jovyan@248c9e3fbd59:~/work/progjar/progjar6$ python3 lb_async.py
/home/jovyan/work/progjar/progjar6/lb_async.py:4: DeprecationWarning: The asyncore module is deprecated and will be removed in Python 3.12. The recommended replacement is asyncio
import asyncore
WARNING:root:load balancer running on port 55555
```

Nama : Amsal Herbert  
NRP : 5025201182

## b) mode process

untuk kode mode process saya hanya mengganti portnya saja agar sesuai. Untuk mencoba menjalankan mode process saya menjalankan python3 il\_process.py di terminal dan ini hasilnya.

```
(base) jovyan@248c9e3fbd59:~/work/progjar/progjar6$ python3 lb_process.py
```

## Perbandingan kinerja web server

### a) Mode asynchronous

#### 1) tanpa load balancer

Pada soal diminta untuk pengukuran kinerja, gunakan tool wrk dengan jumlah request/koneksi 1000, dengan parameter concurrency 10,50,100,150,200 untuk itu saya menginstall wrk terlebih dahulu dengan perintah *sudo apt-get install wrk*.

Untuk menjalankan testing bisa menggunakan perintah

```
wrk -c 1000 -t {n} http://localhost:8887/
```

nilai c seribu sesuai dengan jumlah koneksi yang diminta pada soal

nilai n akan diubah sesuai soal yaitu dengan 10,50,100,150, dan 200

localhost 8887 sesuai dengan port yang ada di async\_server.py

```
52 def main():  
53     portnumber=8887
```

Hasil dari python3 async\_server.py

```
(base) jovyan@248c9e3fbd59:~/work/progjar/progjar6$ python3 async_server.py  
/home/jovyan/work/progjar/progjar6/async_server.py:4: DeprecationWarning: The asyncore module is deprecated and will be removed in Python 3.12. The recommended replacement is a  
syncio  
import asyncore  
WARNING:root:running on port 8887
```

Pada terminal lain akan dicoba dijalankan wrk

Untuk t=10

```
(base) jovyan@248c9e3fbd59:~/work/progjar/progjar6$ wrk -c 1000 -t 10 http://localhost:8887/  
Running 10s test @ http://localhost:8887/  
10 threads and 1000 connections  
Thread Stats Avg Stdev Max +/- Stdev  
Latency 12.10ms 35.86ms 1.78s 99.81%  
Req/Sec 163.55 100.67 545.00 77.41%  
6977 requests in 10.08s, 0.98MB read  
Socket errors: connect 0, read 0, write 0, timeout 1  
Requests/sec: 692.39  
Transfer/sec: 99.40KB
```

Untuk t = 50

Nama : Amsal Herbert  
NRP : 5025201182

```
(base) jovyan@248c9e3fbd59:~/work/progjar/progjar6$ wrk -c 1000 -t 50 http://localhost:8887/
Running 10s test @ http://localhost:8887/
 50 threads and 1000 connections
   Thread Stats   Avg      Stdev     Max   +/-  Stdev
   Latency    11.53ms   23.56ms   1.76s   99.86%
   Req/Sec    119.06    74.46   510.00   80.37%
7126 requests in 10.07s, 1.00MB read
Socket errors: connect 0, read 0, write 0, timeout 4
Requests/sec:   707.61
Transfer/sec:   101.58KB
```

Untuk t = 100

```
(base) jovyan@248c9e3fbd59:~/work/progjar/progjar6$ wrk -c 1000 -t 100 http://localhost:8887/
Running 10s test @ http://localhost:8887/
 100 threads and 1000 connections
   Thread Stats   Avg      Stdev     Max   +/-  Stdev
   Latency    14.96ms   64.84ms   1.78s   99.64%
   Req/Sec    101.80    58.64   400.00   77.46%
6449 requests in 10.07s, 0.90MB read
Socket errors: connect 0, read 0, write 0, timeout 8
Requests/sec:   640.65
Transfer/sec:    91.97KB
```

Untuk t = 150

```
(base) jovyan@248c9e3fbd59:~/work/progjar/progjar6$ wrk -c 1000 -t 150 http://localhost:8887/
Running 10s test @ http://localhost:8887/
 150 threads and 1000 connections
   Thread Stats   Avg      Stdev     Max   +/-  Stdev
   Latency    17.98ms   70.87ms   1.85s   99.42%
   Req/Sec     66.78    33.60   270.00   73.00%
5564 requests in 10.07s, 798.74KB read
Socket errors: connect 0, read 0, write 0, timeout 7
Requests/sec:   552.38
Transfer/sec:    79.30KB
```

Untuk t = 200

```
(base) jovyan@248c9e3fbd59:~/work/progjar/progjar6$ wrk -c 1000 -t 200 http://localhost:8887/
Running 10s test @ http://localhost:8887/
 200 threads and 1000 connections
   Thread Stats   Avg      Stdev     Max   +/-  Stdev
   Latency    15.18ms   68.80ms   1.93s   99.54%
   Req/Sec     84.74    39.17   282.00   80.50%
6689 requests in 10.09s, 0.94MB read
Socket errors: connect 0, read 0, write 0, timeout 7
Requests/sec:   662.86
Transfer/sec:    95.16KB
```

Pada server hanya ditampilkan dari alamat ip yang sama

```
WARNING:root:connection from ('127.0.0.1', 47012)
WARNING:root:connection from ('127.0.0.1', 47014)
WARNING:root:connection from ('127.0.0.1', 60002)
WARNING:root:connection from ('127.0.0.1', 60000)
WARNING:root:connection from ('127.0.0.1', 59876)
```

Nama : Amsal Herbert  
NRP : 5025201182

## 2) Dengan load balancer

Sama seperti percobaan diatas, tapi saya menambahkan load balancer dengan menjalankan runserver.sh dengan perintah ./runserver.sh

```
(base) jovyan@248c9e3fbd59:~/work/progjar/progjar6$ ./runserver.sh
(base) jovyan@248c9e3fbd59:~/work/progjar/progjar6$ /home/jovyan/work/progjar/progjar6/async_server.py:4: DeprecationWarning: The asyncore module is deprecated and will be removed in Python 3.12. The recommended replacement is asyncio
import asyncore
/home/jovyan/work/progjar/progjar6/async_server.py:4: DeprecationWarning: The asyncore module is deprecated and will be removed in Python 3.12. The recommended replacement is a syncio
import asyncore
/home/jovyan/work/progjar/progjar6/async_server.py:4: DeprecationWarning: The asyncore module is deprecated and will be removed in Python 3.12. The recommended replacement is a syncio
import asyncore
/home/jovyan/work/progjar/progjar6/async_server.py:4: DeprecationWarning: The asyncore module is deprecated and will be removed in Python 3.12. The recommended replacement is a syncio
import asyncore
WARNING:root:running on port 9002
WARNING:root:running on port 9003
WARNING:root:running on port 9005
WARNING:root:running on port 9004
```

Disini akan menerima koneksi yang akan diteruskan server

Untuk melakukan percobaan saya melakukan dengan perintah

`wrk -c 1000 -t 10 http://localhost:55555/`

local host saya sesuaikan dengan yang ada di lb\_async.py

```
97 def main():
98     portnumber = 55555
```

Pada percobaan t = 10

```
(base) jovyan@248c9e3fbd59:~/work/progjar/progjar6$ wrk -c 1000 -t 10 http://localhost:55555/
Running 10s test @ http://localhost:55555/
 10 threads and 1000 connections
   Thread Stats   Avg     Stdev     Max   +/-  Stdev
   Latency    82.17ms  212.60ms   1.97s   95.55%
   Req/Sec    86.87    68.25   676.00   78.47%
 7993 requests in 10.10s, 1.12MB read
Socket errors: connect 0, read 0, write 0, timeout 57
Requests/sec:   791.08
Transfer/sec:   113.56KB
```

Tampilan di terminal lb\_async.py

```
WARNING:root:koneksi dari ('127.0.0.1', 50614) diteruskan ke ('127.0.0.1', 9003)
WARNING:root:connection from ('127.0.0.1', 50604)
WARNING:root:koneksi dari ('127.0.0.1', 50604) diteruskan ke ('127.0.0.1', 9004)
WARNING:root:connection from ('127.0.0.1', 47214)
WARNING:root:koneksi dari ('127.0.0.1', 47214) diteruskan ke ('127.0.0.1', 9005)
```

Tampilan di terminal ./runserver.sh

```
WARNING:root:connection from ('127.0.0.1', 50034)
WARNING:root:connection from ('127.0.0.1', 42882)
WARNING:root:connection from ('127.0.0.1', 60668)
```

Saya melanjutkan untuk percobaan yang lain

Nama : Amsal Herbert  
NRP : 5025201182

Untuk t = 50

```
(base) jovyan@248c9e3fbd59:~/work/progjar/progjar6$ wrk -c 1000 -t 50 http://localhost:55555/
Running 10s test @ http://localhost:55555/
 50 threads and 1000 connections
   Thread Stats   Avg      Stdev     Max   +/-  Stdev
   Latency    60.65ms  187.79ms   1.95s   96.05%
   Req/Sec    52.97    41.65   360.00   73.32%
 11507 requests in 10.10s, 1.61MB read
Socket errors: connect 0, read 0, write 0, timeout 48
Requests/sec: 1139.37
Transfer/sec: 163.56KB
```

Untuk t = 100

```
(base) jovyan@248c9e3fbd59:~/work/progjar/progjar6$ wrk -c 1000 -t 100 http://localhost:55555/
Running 10s test @ http://localhost:55555/
 100 threads and 1000 connections
   Thread Stats   Avg      Stdev     Max   +/-  Stdev
   Latency    89.54ms  223.42ms   2.00s   95.15%
   Req/Sec    29.04    25.89   252.00   75.41%
  7894 requests in 10.10s, 1.11MB read
Socket errors: connect 0, read 0, write 0, timeout 73
Requests/sec:  781.44
Transfer/sec:  112.18KB
```

Untuk t = 150

```
(base) jovyan@248c9e3fbd59:~/work/progjar/progjar6$ wrk -c 1000 -t 150 http://localhost:55555/
Running 10s test @ http://localhost:55555/
 150 threads and 1000 connections
   Thread Stats   Avg      Stdev     Max   +/-  Stdev
   Latency    82.15ms  223.02ms   1.97s   95.38%
   Req/Sec    28.21    23.02   220.00   73.20%
  8574 requests in 10.10s, 1.20MB read
Socket errors: connect 0, read 0, write 0, timeout 48
Requests/sec:  848.80
Transfer/sec:  121.85KB
```

Untuk t = 200

```
(base) jovyan@248c9e3fbd59:~/work/progjar/progjar6$ wrk -c 1000 -t 200 http://localhost:55555/
Running 10s test @ http://localhost:55555/
 200 threads and 1000 connections
   Thread Stats   Avg      Stdev     Max   +/-  Stdev
   Latency    87.90ms  234.35ms   2.00s   94.71%
   Req/Sec    25.31    20.76   200.00   66.30%
  8953 requests in 10.10s, 1.26MB read
Socket errors: connect 0, read 0, write 0, timeout 64
Requests/sec:  886.36
Transfer/sec:  127.24KB
```

b) Mode Pool

1) tanpa Load Balancer

Pertama saya menjalankan web server dengan perintah

```
python3 server_process_pool_http.py
```

lalu menjalankan seperti cara sebelumnya pada port 8000

Nama : Amsal Herbert  
NRP : 5025201182

Untuk t = 10

```
(base) jovyan@248c9e3fbd59:~/work/progjar/progjar6$ wrk -c 1000 -t 10 http://localhost:8000/
Running 10s test @ http://localhost:8000/
 10 threads and 1000 connections
   Thread Stats   Avg     Stdev     Max   +/-  Stdev
   Latency    72.16ms   181.88ms   1.95s   93.38%
   Req/Sec    41.60     36.60   210.00   74.65%
2929 requests in 10.02s, 420.47KB read
Socket errors: connect 0, read 0, write 0, timeout 27
Requests/sec: 292.36
Transfer/sec: 41.97KB
```

Untuk t = 50

```
(base) jovyan@248c9e3fbd59:~/work/progjar/progjar6$ wrk -c 1000 -t 50 http://localhost:8000/
Running 10s test @ http://localhost:8000/
 50 threads and 1000 connections
   Thread Stats   Avg     Stdev     Max   +/-  Stdev
   Latency   267.65ms   228.38ms   1.99s   79.00%
   Req/Sec   10.75     12.88   100.00   91.45%
1679 requests in 10.07s, 241.03KB read
Socket errors: connect 0, read 0, write 0, timeout 31
Requests/sec: 166.69
Transfer/sec: 23.93KB
```

Untuk t = 100

```
(base) jovyan@248c9e3fbd59:~/work/progjar/progjar6$ wrk -c 1000 -t 100 http://localhost:8000/
Running 10s test @ http://localhost:8000/
100 threads and 1000 connections
   Thread Stats   Avg     Stdev     Max   +/-  Stdev
   Latency   169.23ms   219.58ms   1.99s   94.35%
   Req/Sec   11.15     11.93   101.00   91.42%
1228 requests in 10.06s, 176.29KB read
Socket errors: connect 0, read 0, write 0, timeout 24
Requests/sec: 122.01
Transfer/sec: 17.52KB
```

Untuk t = 150

```
(base) jovyan@248c9e3fbd59:~/work/progjar/progjar6$ wrk -c 1000 -t 150 http://localhost:8000/
Running 10s test @ http://localhost:8000/
150 threads and 1000 connections
   Thread Stats   Avg     Stdev     Max   +/-  Stdev
   Latency   284.61ms   243.06ms   2.00s   83.48%
   Req/Sec    6.18     6.67    50.00   93.33%
962 requests in 10.08s, 138.10KB read
Socket errors: connect 0, read 0, write 0, timeout 30
Requests/sec: 95.47
Transfer/sec: 13.71KB
```

Untuk t = 200

```
(base) jovyan@248c9e3fbd59:~/work/progjar/progjar6$ wrk -c 1000 -t 200 http://localhost:8000/
Running 10s test @ http://localhost:8000/
200 threads and 1000 connections
   Thread Stats   Avg     Stdev     Max   +/-  Stdev
   Latency   334.48ms   319.55ms   1.99s   89.75%
   Req/Sec    6.39     11.27   121.00   91.21%
882 requests in 10.10s, 126.62KB read
Socket errors: connect 0, read 0, write 0, timeout 33
Requests/sec: 87.33
Transfer/sec: 12.54KB
```



Nama : Amsal Herbert  
NRP : 5025201182

## 2) Dengan Load Balancer

Sebelum menjalankan ini saya mengubah port pada lb\_process.py dan runserverprocess.sh agar sesuai

```
1  #jalankan 4 process_server
2
3  python3 server_process_pool_http.py 9012 &
4  python3 server_process_pool_http.py 9013 &
5  python3 server_process_pool_http.py 9014 &
6  python3 server_process_pool_http.py 9015 &
7
13  self.servers.append(('127.0.0.1',9012))
14  self.servers.append(('127.0.0.1',9013))
15  self.servers.append(('127.0.0.1',9014))
16  self.servers.append(('127.0.0.1',9015))
```

Lalu menjalankan runserverprocess.sh pada terminal satu, python3 lb\_process.py pada terminal 2, dan wrk -c 1000 -t 10 <http://localhost:44444/> pada terminal 3.

Port 44444 agar sesuai yang ada di lb\_process.py

```
64  my_socket.bind(('0.0.0.0', 44444))
65  my_socket.listen(1)
```

Untuk t = 10

```
(base) jovyan@248c9e3fbd59:~/work/progjar/progjar6$ wrk -c 1000 -t 10 http://localhost:44444/
Running 10s test @ http://localhost:44444/
 10 threads and 1000 connections
   Thread Stats   Avg      Stdev     Max   +/-  Stdev
   Latency    678.45ms  454.77ms  1.98s   68.68%
   Req/Sec    19.25    17.48   90.00   76.18%
 835 requests in 10.08s, 119.87KB read
Socket errors: connect 0, read 1, write 0, timeout 56
Requests/sec:    82.84
Transfer/sec:    11.89KB
```

Untuk t = 50

```
(base) jovyan@248c9e3fbd59:~/work/progjar/progjar6$ wrk -c 1000 -t 50 http://localhost:44444/
Running 10s test @ http://localhost:44444/
 50 threads and 1000 connections
   Thread Stats   Avg      Stdev     Max   +/-  Stdev
   Latency    1.29s   551.92ms  2.00s   57.92%
   Req/Sec     6.19     6.85   59.00   90.16%
 750 requests in 10.09s, 107.67KB read
Socket errors: connect 0, read 1, write 0, timeout 106
Requests/sec:    74.31
Transfer/sec:    10.67KB
```

Nama : Amsal Herbert  
NRP : 5025201182

Untuk t = 100

```
(base) jovyan@248c9e3fbd59:~/work/progjar/progjar6$ wrk -c 1000 -t 100 http://localhost:44444/
Running 10s test @ http://localhost:44444/
100 threads and 1000 connections
  Thread Stats   Avg      Stdev     Max   +/-  Stdev
    Latency    777.40ms   635.10ms   2.00s   70.08%
    Req/Sec     3.21     4.56    39.00   85.17%
  793 requests in 10.10s, 113.84KB read
  Socket errors: connect 0, read 0, write 0, timeout 539
Requests/sec:      78.52
Transfer/sec:      11.27KB
```

Untuk t = 150

```
(base) jovyan@248c9e3fbd59:~/work/progjar/progjar6$ wrk -c 1000 -t 150 http://localhost:44444/
Running 10s test @ http://localhost:44444/
150 threads and 1000 connections
  Thread Stats   Avg      Stdev     Max   +/-  Stdev
    Latency    946.04ms   534.25ms   2.00s   67.06%
    Req/Sec     3.94     4.82    30.00   78.15%
  782 requests in 10.10s, 112.26KB read
  Socket errors: connect 0, read 1, write 0, timeout 96
Requests/sec:      77.42
Transfer/sec:      11.11KB
```

Untuk t = 200

```
(base) jovyan@248c9e3fbd59:~/work/progjar/progjar6$ wrk -c 1000 -t 200 http://localhost:44444/
Running 10s test @ http://localhost:44444/
200 threads and 1000 connections
  Thread Stats   Avg      Stdev     Max   +/-  Stdev
    Latency    817.98ms   384.40ms   1.98s   66.27%
    Req/Sec     4.16     4.69    30.00   78.09%
  976 requests in 10.10s, 140.11KB read
  Socket errors: connect 0, read 2, write 0, timeout 42
Requests/sec:      96.63
Transfer/sec:      13.87KB
```

Proses yang terjadi di terminal runserverprocess.sh

```
WARNING:root:( '127.0.0.1', 37678) connecting to ( '127.0.0.1', 9015)
( '127.0.0.1', 9012)
WARNING:root:( '127.0.0.1', 37682) connecting to ( '127.0.0.1', 9012)
( '127.0.0.1', 9013)
WARNING:root:( '127.0.0.1', 42070) connecting to ( '127.0.0.1', 9013)
( '127.0.0.1', 9014)
WARNING:root:( '127.0.0.1', 42066) connecting to ( '127.0.0.1', 9014)
█
```

Proses yang terjadi di lb\_process.py

```
WARNING:root:balas ke client: b'HTTP/1.0 200 OK\r\nDate: Sun Jun 18 07:09:02 2023\r\nConnection: close\r\nServer: myserver/1.0\r\nContent-Length: 31\r\n\r\nIni Adalah web Serv
er percobaan\r\n\r\n'
WARNING:root:balas ke client: b'HTTP/1.0 200 OK\r\nDate: Sun Jun 18 07:09:02 2023\r\nConnection: close\r\nServer: myserver/1.0\r\nContent-Length: 31\r\n\r\nIni Adalah web Serv
er percobaan\r\n\r\n'
WARNING:root:data dari client: GET / HTTP/1.1
Host: localhost:44444

WARNING:root:balas ke client: b'HTTP/1.0 200 OK\r\nDate: Sun Jun 18 07:09:02 2023\r\nConnection: close\r\nServer: myserver/1.0\r\nContent-Length: 31\r\n\r\nIni Adalah web Serv
er percobaan\r\n\r\n'
WARNING:root:data dari client: GET / HTTP/1.1
Host: localhost:44444
```



Nama : Amsal Herbert  
NRP : 5025201182

Hasil dalam bentuk tabel

	jumlah concurrency	Jumlah request	request/sec	trasnfer/sec (kb)
Async	10	6977	692.39	99.4
	50	7126	707.61	101.58
	100	6449	640.65	91.97
	150	5564	552.38	79.3
	200	6689	662.86	95.16
Async+load balancer	10	7993	791.08	113.56
	50	11507	1139.37	163.56
	100	7894	781.44	112.18
	150	8574	848.8	121.85
	200	8953	886.36	127.24
Pool	10	2929	292.36	41.97
	50	1679	166.69	23.93
	100	1228	122.01	17.52
	150	962	95.47	13.71
	200	882	87.33	12.54
Pool+load balancer	10	835	82.84	11.89
	50	750	74.31	10.67
	100	793	78.52	11.27
	150	782	77.42	11.11
	200	967	96.63	13.87

Kesimpulan yang bisa saya ambil adalah pada mode asynchronous penggunaan load balancer membuat jumlah request yang diterima lebih banyak, sedangkan pada pool penggunaan load balancer membuat jumlah request yang diterima menjadi lebih sedikit. Artinya penggunaan web server mode asynchronous lebih efektif menggunakan load balancer sedangkan Pool tidak.