

Nama : Muhammad Rizky Anditama

NPM : 1706044080

Kelas : LAW - A

Stress Testing

Read Idempotent

Dalam memanggil READ untuk transaction, nginx menerapkan *caching*, sehingga READ ini jika telah dipanggil sekali, maka nginx akan menggunakan cache ketika dikembalikan ke client, tidak lagi memanggil service READ. Jika dilihat pada web server yang saya gunakan yang melogging setiap request yang masuk, terlihat bahwa pemanggilan ini hanya terjadi sekali.

```
2021/06/23 15:22:21 C:/Users/Asus/Documents/Work/LAW/Tugas5/Read/main.go:128  
[1.003ms] [rows:1] SELECT * FROM `transactions` WHERE user_id = 1 AND id = "782" LIMIT 1  
[GIN] 2021/06/23 - 15:22:21 | 200 | 12.0077ms | :1 | GET | "/read/1706044080/782"
```

Kemudian, kita akan menjalankan stress test menggunakan Apache Benchmark dengan konfigurasi berikut:

- m : Method HTTP, method yang digunakan adalah GET
- c: Jumlah client concurrent, untuk tes kali ini 10 concurrent users
- n: Jumlah request, 10.000 request

```
C:\Users\Asus\Documents\Work\Performance Tester\Apache24\bin>ab -m GET -c 10 -n 10000 http://localhost/read/1706044080/782  
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 1000 requests  
Completed 2000 requests  
Completed 3000 requests  
Completed 4000 requests  
Completed 5000 requests  
Completed 6000 requests  
Completed 7000 requests  
Completed 8000 requests  
Completed 9000 requests  
Completed 10000 requests  
Finished 10000 requests
```

Hasilnya adalah seperti berikut:

Nama : Muhammad Rizky Anditama

NPM : 1706044080

Kelas : LAW - A

```
Concurrency Level:      10
Time taken for tests:    18.138 seconds
Complete requests:      10000
Failed requests:        0
Total transferred:      5240000 bytes
HTML transferred:       620000 bytes
Requests per second:    551.34 [#/sec] (mean)
Time per request:       18.138 [ms] (mean)
Time per request:       1.814 [ms] (mean, across all concurrent requests)
Transfer rate:          282.13 [Kbytes/sec] received
```

Connection Times (ms)

	min	mean[+/-sd]	median	max
Connect:	0	0 0.5	0	4
Processing:	5	16 28.7	15	2045
Waiting:	2	15 28.7	15	2045
Total:	6	16 28.7	15	2045

Percentage of the requests served within a certain time (ms)

50%	15
66%	16
75%	17
80%	17
90%	18
95%	19
98%	21
99%	22
100%	2045 (longest request)

Read Non-Idempotent

Dalam memanggil READ untuk user, nginx tidak menerapkan *caching*, sehingga READ ini akan terus memanggil webserver. Jika dilihat pada web server yang saya gunakan yang melogging setiap request yang masuk, terlihat bahwa pemanggilan ini terjadi berkali-kali.

Nama : Muhammad Rizky Anditama

NPM : 1706044080

Kelas : LAW - A

```
[0.000ms] [rows:1] SELECT * FROM `users` WHERE npm = "1706044080" LIMIT 1
[GIN] 2021/06/23 - 15:23:12 | 200 | 3.0032ms | 127.0.0.1 | GET | "/read/1706044080"

2021/06/23 15:23:20 Open file in editor (ctrl + click) ork/LAW/Tugas5/Read/main.go:97
[0.993ms] [rows:1] SELECT * FROM `users` WHERE npm = "1706044080" LIMIT 1
[GIN] 2021/06/23 - 15:23:20 | 200 | 1.9956ms | ::1 | GET | "/read/1706044080"

2021/06/23 15:48:00 C:/Users/Asus/Documents/Work/LAW/Tugas5/Read/main.go:97
[22.000ms] [rows:1] SELECT * FROM `users` WHERE npm = "1706044080" LIMIT 1
[GIN] 2021/06/23 - 15:48:00 | 200 | 59.9995ms | 127.0.0.1 | GET | "/read/1706044080"

2021/06/23 15:48:01 C:/Users/Asus/Documents/Work/LAW/Tugas5/Read/main.go:97
[0.000ms] [rows:1] SELECT * FROM `users` WHERE npm = "1706044080" LIMIT 1
[GIN] 2021/06/23 - 15:48:01 | 200 | 1.0018ms | ::1 | GET | "/read/1706044080"

2021/06/23 15:48:03 C:/Users/Asus/Documents/Work/LAW/Tugas5/Read/main.go:97
[0.000ms] [rows:1] SELECT * FROM `users` WHERE npm = "1706044080" LIMIT 1
[GIN] 2021/06/23 - 15:48:03 | 200 | 1.0025ms | 127.0.0.1 | GET | "/read/1706044080"
```

Kemudian, kita akan menjalankan stress test menggunakan Apache Benchmark dengan konfigurasi berikut:

- m : Method HTTP, method yang digunakan adalah GET
- c: Jumlah client concurrent, untuk tes kali ini 10 concurrent users
- n: Jumlah request, 10.000 request

```
C:\Users\Asus\Documents\Work\Performance Tester\Apache24\bin>ab -m GET -c 10 -n 10000 http://localhost/read/1706044080
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 1000 requests
Completed 2000 requests
Completed 3000 requests
Completed 4000 requests
Completed 5000 requests
Completed 6000 requests
Completed 7000 requests
Completed 8000 requests
Completed 9000 requests
Completed 10000 requests
Finished 10000 requests
```

Hasilnya adalah seperti berikut:

Nama : Muhammad Rizky Anditama

NPM : 1706044080

Kelas : LAW - A

```
Concurrency Level:      10
Time taken for tests:    73.758 seconds
Complete requests:      10000
Failed requests:        0
Total transferred:      5320000 bytes
HTML transferred:       700000 bytes
Requests per second:    135.58 [#/sec] (mean)
Time per request:       73.758 [ms] (mean)
Time per request:       7.376 [ms] (mean, across all concurrent requests)
Transfer rate:          70.44 [Kbytes/sec] received
```

Connection Times (ms)

	min	mean[+/-sd]	median	max
Connect:	0	1 0.7	1	14
Processing:	7	68 520.2	20	9809
Waiting:	7	68 520.1	20	9809
Total:	8	69 520.2	21	9815

Percentage of the requests served within a certain time (ms)

50%	21
66%	23
75%	25
80%	27
90%	37
95%	50
98%	85
99%	133
100%	9815 (longest request)

Analisis perbandingan

Dari hasil pengujian yang dilakukan, jelas bahwa pemanggilan READ yang menggunakan *cache* memberikan hasil yang lebih baik.

- Requests per second pada pemanggilan yang menggunakan *cache* berada pada angka 551 dibandingkan dengan 135 yang tidak menggunakan *cache*
- Time per requestnya pun pasti lebih kecil dimana bernilai 18ms dengan *cache* dan 73ms tanpa *cache*
- Longest request pada pemanggilan *cache* juga hanya sebesar 2045 ms dibandingkan 9815 ms tanpa *cache*

Melalui *assignment* ini dapat disimpulkan bahwa *caching* benar-benar memberikan dampak yang signifikan terhadap web service.