Assignment 4:
High Performance Service and Cache

A - 1906350774 - Dionisius Baskoro Samudra

README

1. Service dijalankan pada **Google Cloud Platform** dengan spesifikasi sebagai berikut:
    a. Machine Type : e2-micro
    b. CPU Platform : Intel Broadwell
    c. Zone : us-central1-a
    d. External IP : 34.121.239.172

2. Service dibuat menggunakan python dengan framework **Django**.

3. Testing dilakukan dan dimonitoring menggunakan **locust.** Sedangkan untuk monitoring terhadap instance menggunakan **netstat**.

4. Testing idempoten maupun non idempotent menggunakan rincian yang sama dengan total user sebanyak **10.000** users dan dengan spawn rate sebesar **100** users/s. Hal ini bertujuan untuk membandingkan secara langsung performa service ketika mengimplementasikan caching dengan tidak mengimplementasikannya.

5. Source code juga dapat dilihat pada https://github.com/FXDROS/law-assignment4

1. Non Idempotent Testing

Dalam melakukan stress test non idempotent terhadap service READ data mahasiswa, service mulai down saat melayani lebih dari **2.700** pengguna. Testing dilakukan dengan melihat pada saat kapan service mulai mengalami penurunan performa (mulai tidak mengembalikan response). Berikut merupakan grafik dari testing non idempotent.

Rincian metrics yang dicatat adalah sebagai berikut:

a. Total Request: 26274
b. Average Response Time: 3000 ms
c. Peak Response Time: 31000 ms
d. Error Rates: ~69%
e. Peak CPU Utilisation: ~59%

Performa CPU dari instance tempat service dijalankan menunjukkan angka yang cukup besar. CPU usage mencapai 59% dari kapasitas penggunaan CPU. Namun, setelah 2.700 pengguna, load kerja CPU mulai berkurang karena service tidak lagi mengembalikan response; sehingga tidak ada lagi data yang diproses. Berikut merupakan tangkapan layar dari CPU pada instance saat dilakukan non idempotent testing.

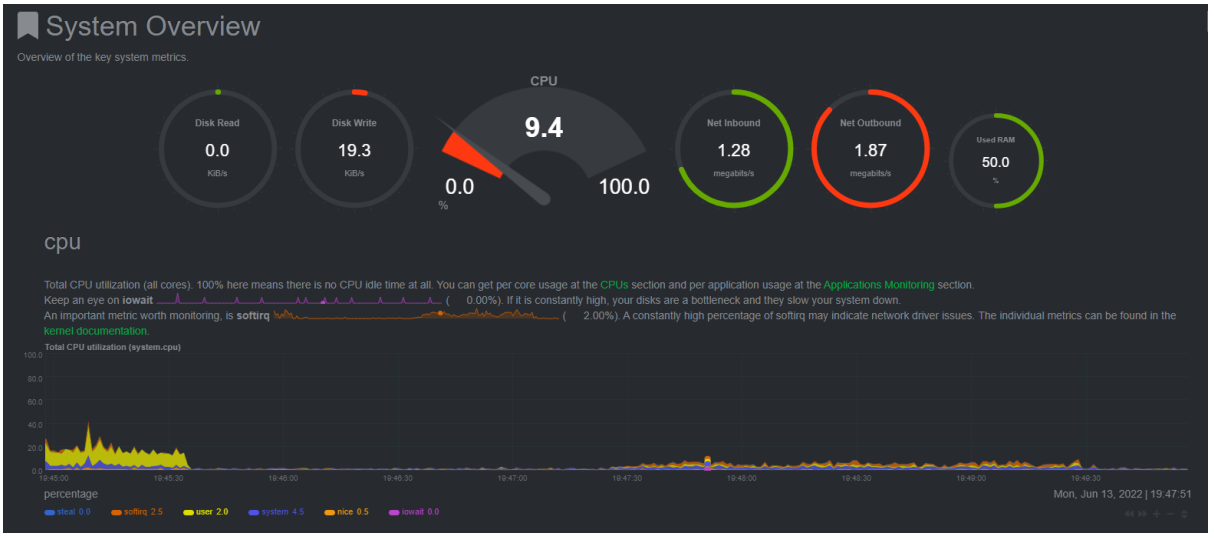
2. Idempotent Testing

Ketika melakukan testing terhadap service READ data mahasiswa secara idempotent dengan spesifikasi yang sama, service tidak down meskipun testing dijalankan lebih lama dari non idempotent testing. Hal tersebut disebabkan pada idempotent testing dimanfaatkan fitur caching, sehingga proses tidak akan dilanjutkan ke service, dan hanya sampai pada gateway. Dengan demikian, service tidak perlu memproses data yang sama. Berikut merupakan grafik dari testing idempotent.

Rincian metrics yang dicatat adalah sebagai berikut:

f. Total Request: 45883
g. Average Response Time: 600 ms
h. Peak Response Time: 57000 ms
i. Error Rates: ~4.12%
j. Peak CPU Utilisation: ~9%

Dengan melakukan caching, service dapat memberikan respon lebih cepat dan tidak terjadi overload pada CPU karena tidak dilakukan pengaksesan terhadap database dan service secara keseluruhan. Berikut merupakan tangkapan layar dari CPU pada instance saat dilakukan idempotent testing.



Lampiran

Konfigurasi NGINX

```
1. proxy_cache_path /tmp/cache keys_zone=mycache:1m levels=1:2
   inactive=180s max_size=10m;
2. server {
3.     listen 80;
4.     listen [::]:80;
5.     server_name 34.121.239.172;
6.
7.     location = /favicon.ico { access_log off; log_not_found
   off; }
8.
9.    location /update/ {
10.           include proxy_params;
11.           proxy_pass http://0.0.0.0:8000;
12.       }
13.
14.       location ~ /read/([0-9]+)/([0-9]+)$ {
15.           proxy_cache mycache;
16.           proxy_cache_valid 3m;
17.           add_header X-Proxy-Cache $upstream_cache_status;
18.           rewrite /read/([0-9]+)/([0-9]+)$ /read/$1 break;
19.           include proxy_params;
20.           proxy_cache_key $host$request_uri$2;
21.           proxy_pass http://0.0.0.0:8001;
22.       }
23.
24.       location /read/ {
25.           include proxy_params;
26.           proxy_pass http://0.0.0.0:8001;
27.       }
28.   }
```