

Laporan Peta Spice IT13

Nomor [8]

A. Laporan

Kami menggunakan [script](#) untuk mendapatkan rata-rata hasil dari 10 kali analisis

- Round Robin

```
ulbert@LAPTOP-DT0BBQ4H: ~  
root@Dmitri:~# bash benchmark.sh  
Benchmarking completed. Results saved in round_robin.txt.  
root@Dmitri:~# cat round_robin.txt  
3001.51  
3290.45  
3788.05  
3076.39  
3520.06  
3491.62  
3687.51  
3378.47  
2546.02  
3327.32  
Mean: 3310.74  
root@Dmitri:~#
```

- Least Connection

```
ulbert@LAPTOP-DT0BBQ4H: ~  
root@Dmitri:~# bash benchmark.sh  
Benchmarking completed. Results saved in least_connection.txt.  
root@Dmitri:~# cat least_connection.txt  
3235.58  
3275.17  
3700.85  
3536.74  
3149.88  
2589.18  
3547.23  
3770.43  
3601.92  
2753.24  
Mean: 3316.01  
root@Dmitri:~#
```

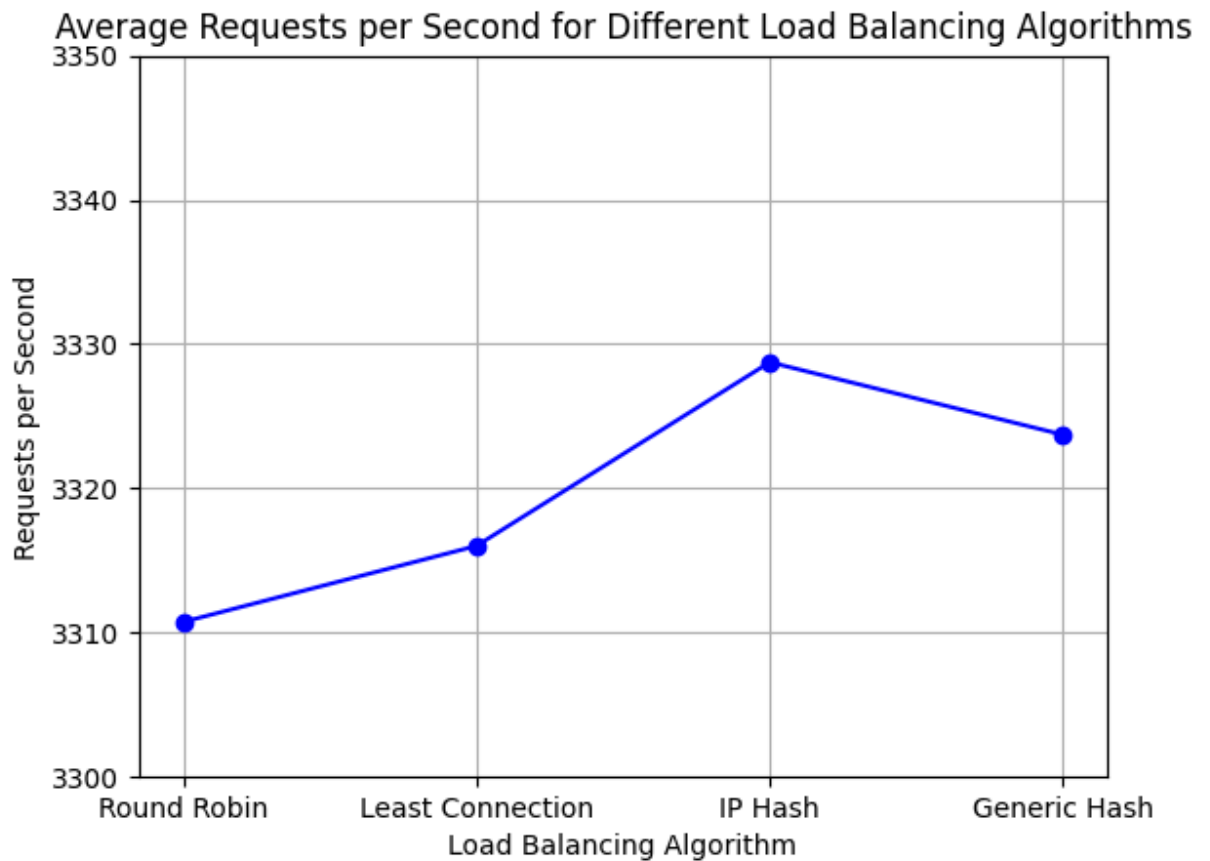
- IP Hash

```
ulbert@LAPTOP-DT0BBQ4H: ~  
root@Dmitri:~# bash benchmark.sh  
Benchmarking completed. Results saved in ip_hash.txt.  
root@Dmitri:~# cat ip_hash.txt  
3336.94  
3584.95  
3394.85  
3729.40  
2945.75  
3492.35  
3428.63  
3249.41  
2948.46  
3176.84  
Mean: 3328.74  
root@Dmitri:~#
```

- Generic Hash

```
ulbert@LAPTOP-DT0BBQ4H: ~  
root@Dmitri:~# bash benchmark.sh  
Benchmarking completed. Results saved in generic_hash.txt.  
root@Dmitri:~# cat generic_hash.txt  
2865.97  
3285.17  
3724.15  
3152.70  
2744.54  
3638.67  
2738.95  
3394.62  
3230.00  
3462.27  
Mean: 3223.71  
root@Dmitri:~#
```

B. Graph



C. Analisis

Berdasarkan hasil percobaan, IP Hash adalah algoritma load balancing yang paling efisien. Ini menyelesaikan tugas lebih cepat dibandingkan dengan yang lain. Jika sistem memiliki beban kerja yang stabil dan server memiliki kemampuan yang sebanding, IP Hash dapat mendistribusikan beban dengan cukup seimbang.

Nomer [9]

A. Laporan

Kami menggunakan [script](#) untuk mendapatkan rata-rata hasil dari 10 kali analisis

- 3 Worker

```
ulbert@LAPTOP-DT0BBQ4H: ~  
root@Dmitri:~# bash benchmark2.sh  
Benchmarking completed. Results saved in 3worker.txt.  
root@Dmitri:~# cat 3worker.txt  
4229.71  
4594.91  
4194.63  
4491.29  
4562.09  
4528.31  
4292.84  
4484.41  
4875.79  
4285.76  
Mean: 4453.97  
root@Dmitri:~#
```

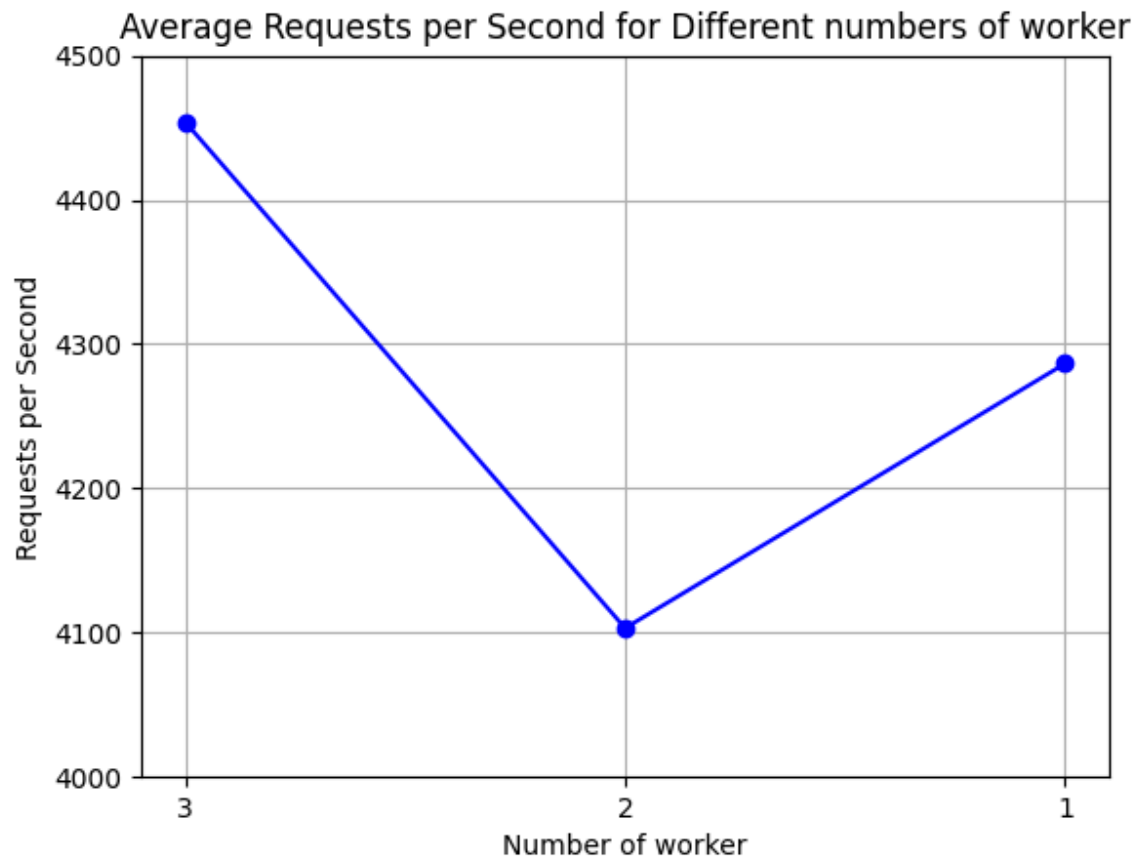
- 2 Worker

```
ulbert@LAPTOP-DT0BBQ4H: ~  
root@Dmitri:~# bash benchmark2.sh  
Benchmarking completed. Results saved in 2worker.txt.  
root@Dmitri:~# cat 2worker.txt  
4126.18  
4083.90  
4610.06  
4307.13  
4123.93  
4098.04  
3750.29  
3442.73  
4044.10  
4444.98  
Mean: 4103.12  
root@Dmitri:~#
```

- 1 Worker

```
ulbert@LAPTOP-DT0BBQ4H: ~  
root@Dmitri:~# bash benchmark2.sh  
Benchmarking completed. Results saved in 1worker.txt.  
root@Dmitri:~# cat 1worker.txt  
3879.46  
4343.31  
4188.48  
4175.73  
4451.51  
4309.36  
4298.45  
4464.19  
4435.73  
4318.52  
Mean: 4286.46  
root@Dmitri:~#
```

B. Graph



C. Analisis

Berdasarkan percobaan yang telah dilakukan, hasil dari algoritma Least Connection dengan 2 worker merupakan hasil terburuk di antara ketiga percobaan ini. Percobaan ini telah kami lakukan sebanyak 10 kali pada setiap n workernya dan diambil rata-rata dari 10 percobaan tersebut. Dapat disimpulkan bahwa 3 worker memiliki keunggulan yang jauh dibanding dengan 2 worker maupun 1 worker, dimana dengan 3 worker memiliki Request per Second yang tinggi dan lebih menguntungkan.