

Resultados con Artillery

Bloqueante

Started phase 0, duration: 1s @ 13:22:57(-0500) 2021-07-20

Report @ 13:23:00(-0500) 2021-07-20

Elapsed time: 3 seconds

Scenarios launched: 50

Scenarios completed: 50

Requests completed: 1000

Mean response/sec: 330.03

Response time (msec):

min: 3

max: 222

median: 95

p95: 136.5

p99: 148.5

Codes:

200: 1000

All virtual users finished

Summary report @ 13:23:00(-0500) 2021-07-20

Scenarios launched: 50

Scenarios completed: 50

Requests completed: 1000

Mean response/sec: 328.95

Response time (msec):

min: 3

max: 222

median: 95

p95: 136.5

p99: 148.5

Scenario counts:

0: 50 (100%)

Codes:

200: 1000

No Bloqueante

Started phase 0, duration: 1s @ 13:23:19(-0500) 2021-07-20

Report @ 13:23:21(-0500) 2021-07-20

Elapsed time: 1 second

Scenarios launched: 50

Scenarios completed: 50

Requests completed: 1000

Mean response/sec: 666.67

Response time (msec):

min: 0

max: 24

median: 8

p95: 16

p99: 19.5

Codes:

200: 1000

All virtual users finished

Summary report @ 13:23:21(-0500) 2021-07-20

Scenarios launched: 50

Scenarios completed: 50

Requests completed: 1000

Mean response/sec: 662.25
Response time (msec):
min: 0
max: 24
median: 8
p95: 16
p99: 19.5
Scenario counts:
0: 50 (100%)
Codes:
200: 1000

CONCLUSION

Todos los procesos fueron ejecutados sin inconveniente, en el caso del no bloqueante(Mean response/sec: 662.25), puede realizar el doble de peticiones en el mismo tiempo que en el caso del bloqueante(Mean response/sec: 328.95)

En prácticamente todos los valores, la media de respuesta del bloqueante es muy superior al no bloqueante, confirmando que el bloqueo del console.log es bastante significativo.

AUTOCANNON

La prueba se ejecuta para 100 conexión cada una de 20 segundos, obteniendo los siguientes resultados

```
Running
Running 20s test @ http://localhost:8080/infobloq
10 connections
```

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	21 ms	24 ms	49 ms	52 ms	26.67 ms	7.79 ms	66 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	282	282	379	412	367.9	36.07	282
Bytes/Sec	709 kB	709 kB	952 kB	1.03 MB	924 kB	90.6 kB	708 kB

Req/Bytes counts sampled once per second.

7k requests in 20.03s, 18.5 MB read

Running 20s test @ http://localhost:8080/info
10 connections

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	21 ms	23 ms	49 ms	52 ms	26.51 ms	8.13 ms	66 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	277	277	384	414	370.1	36.55	277
Bytes/Sec	696 kB	696 kB	965 kB	1.04 MB	929 kB	91.8 kB	696 kB

Req/Bytes counts sampled once per second.

7k requests in 20.07s, 18.6 MB read

Por la sencillez de los procesos los retardos, desviaciones y promedios, son muy parecidos en los dos métodos. Por cada 7k requeridos el tiempo y cantidad de datos es casi similar, no muestra gran diferencia con respecto al console.log