Prueba Autocannon 0x

Sin console.log()

Windows PowerShell Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS D:\CODER HOUSE\BACKEND\BACKEND-VSC\desafioBackend-9> npm test

> desafiobackend-9@1.0.0 test
> node benchmark.js

Running all benchmarks in parallel ...
Running 20s test @ http://localhost:8080/info

100 connections

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	1062 ms	2004 ms	2866 ms	3025 ms	1999.02 ms	373.77 ms	3997 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	8	8	38	88	47.05	28.07	8
Bytes/Sec	4.93 kB	4.93 kB	23.5 kB	54.3 kB	29 kB	17.3 kB	4.93 kB

Req/Bytes counts sampled once per second.

of samples: 20

Con console.log()

Req/Bytes counts sampled once per second. # of samples: 20

1k requests in 20.19s, 580 kB read Running 20s test @ http://localhost:8080/info-gzip 100 connections

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	1814 ms	2008 ms	2858 ms	3006 ms	2102.36 ms	273.28 ms	3859 ms

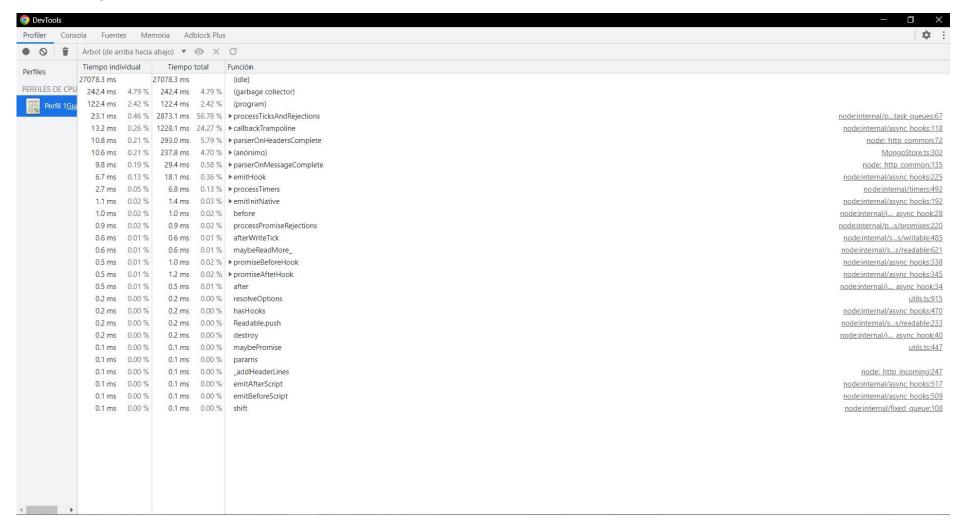
Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	0	0	37	88	44.55	28.63	7
Bytes/Sec	0 B	0 B	22.8 kB	54.3 kB	27.5 kB	17.7 kB	4.32 kB

Req/Bytes counts sampled once per second. # of samples: 20

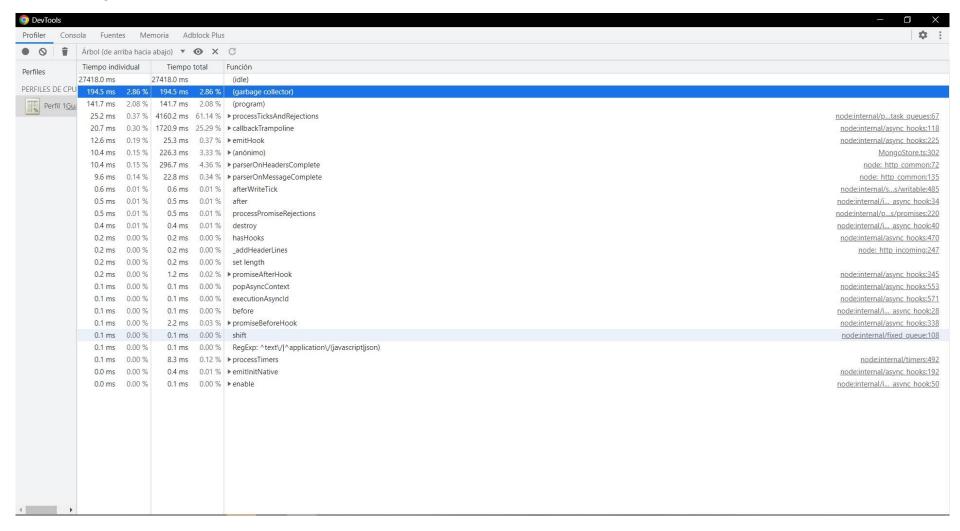
991 requests in 20.21s, 549 kB read
PS D:\CODER HOUSE\BACKEND\BACKEND-VSC\desafioBackend-9>

Prueba node inspect devtools

Sin console.log()



Con console.log()



Conclusión:

Carga más rápido la versión sin console.log(), las pruebas ayudan a visualizar los procesos del servidor y ayuda a optimizar. En producción al aumentar el stress de las conexiones al servidor, cada milisegundo optimizado vale oro.