

Análisis de performance de e-commerce API

Coderhouse 32105, Muñoz Ignacio

Autocannon en modo standalone usando console.log(data)

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

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	307 ms	362 ms	517 ms	573 ms	375.4 ms	52.25 ms	699 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	100	100	279	295	264.15	44.42	100
Bytes/Sec	77.4 kB	77.4 kB	216 kB	229 kB	205 kB	34.4 kB	77.4 kB

Autocannon en modo standalone sin usar console.log(data)

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

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	181 ms	203 ms	274 ms	343 ms	209.93 ms	28.23 ms	485 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	283	283	492	537	475.05	56.47	283
Bytes/Sec	219 kB	219 kB	381 kB	416 kB	368 kB	43.8 kB	219 kB

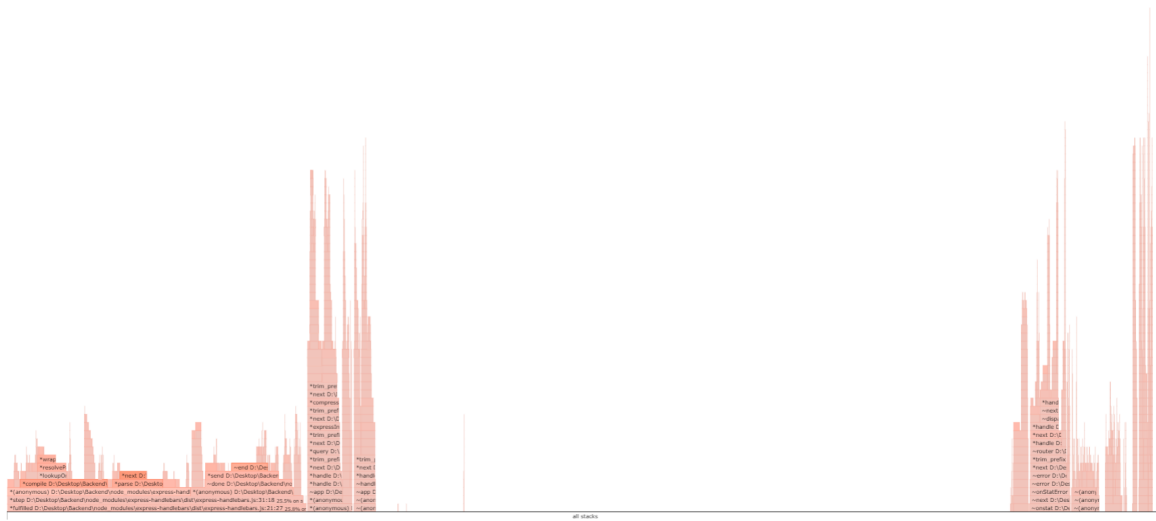
node –inspect standalone usando console.log(data)

Self Time	Total Time	Function
42.8 ms	0.55 %	17204.6 ms 222.48 % ▶ handle
16897.5 ms		(idle) layer.js:86
51.3 ms	0.66 %	14614.7 ms 188.99 % ▶ next
0.9 ms	0.01 %	14540.4 ms 188.03 % ▶ process_params
12.1 ms	0.16 %	14539.5 ms 188.02 % ▶ (anonymous)
8.8 ms	0.11 %	11835.1 ms 153.05 % ▶ trim_prefix
23.3 ms	0.30 %	4558.4 ms 58.95 % ▶ emit
12.9 ms	0.17 %	4083.6 ms 52.81 % (anonymous)
63.2 ms	0.82 %	4072.2 ms 52.66 % ▶ step
27.6 ms	0.36 %	3953.7 ms 51.13 % ▶ handle
28.2 ms	0.36 %	3067.3 ms 39.67 % ▶ callbackTrampoline
2.1 ms	0.03 %	2793.2 ms 36.12 % ▶ (anonymous)
3.3 ms	0.04 %	2790.7 ms 36.09 % ▶ onstat
1.7 ms	0.02 %	2786.9 ms 36.04 % ▶ next
1.4 ms	0.02 %	2785.2 ms 36.02 % ▶ onStatError
4.1 ms	0.05 %	2783.1 ms 35.99 % ▶ error
0.8 ms	0.01 %	2765.2 ms 35.76 % ▶ error
5.0 ms	0.06 %	2755.7 ms 35.64 % ▶ router
3.3 ms	0.04 %	2690.6 ms 34.79 % ▶ dispatch
1.3 ms	0.02 %	2687.3 ms 34.75 % ▶ next
33.6 ms	0.43 %	2679.7 ms 34.65 % ▶ (anonymous)
10.8 ms	0.14 %	2495.9 ms 32.28 % fulfilled
601.1 ms	7.77 %	2014.8 ms 26.06 % ▶ consoleCall
30.3 ms	0.39 %	1814.8 ms 23.47 % ▶ session
14.1 ms	0.18 %	1739.0 ms 22.49 % ▶ _awaiter
12.9 ms	0.17 %	1699.3 ms 21.97 % ▶ Socket_writeGeneric
11.4 ms	0.15 %	1681.6 ms 21.75 % ▶ (anonymous)
34.6 ms	0.45 %	1653.5 ms 21.38 % ▶ (anonymous)
9.2 ms	0.12 %	1581.4 ms 20.45 % ▶ Writable.write

node –inspect standalone sin usar console.log(data)

Self Time	Total Time	Function
17922.8 ms		(idle)
37.1 ms	0.72 %	9615.8 ms 186.44 % ▶ handle
46.5 ms	0.90 %	9286.6 ms 180.05 % ▶ next
0.3 ms	0.01 %	9221.6 ms 178.79 % ▶ process_params
10.8 ms	0.21 %	9221.4 ms 178.79 % ▶ (anonymous)
10.1 ms	0.20 %	8787.1 ms 170.37 % ▶ trim_prefix
12.4 ms	0.24 %	3620.0 ms 70.19 % (anonymous)
51.1 ms	0.99 %	3608.8 ms 69.97 % ▶ step
11.9 ms	0.23 %	2343.9 ms 45.44 % fulfilled
23.0 ms	0.45 %	2163.8 ms 41.95 % ▶ emit
24.3 ms	0.47 %	1656.3 ms 32.11 % ▶ session
21.7 ms	0.42 %	1583.0 ms 30.69 % ▶ handle
37.9 ms	0.73 %	1564.7 ms 30.34 % ▶ (anonymous)
10.2 ms	0.20 %	1402.8 ms 27.20 % ▶ _awaiter
7.6 ms	0.15 %	1355.3 ms 26.28 % ▶ (anonymous)
3.7 ms	0.07 %	1331.2 ms 25.81 % ▶ ExpressHandlebars_renderTemplate
6.5 ms	0.13 %	1327.6 ms 25.74 % ▶ ret
11.8 ms	0.23 %	1237.3 ms 23.99 % ▶ parserOnHeadersComplete
28.4 ms	0.55 %	1203.7 ms 23.34 % ▶ compileInput
14.5 ms	0.28 %	1198.3 ms 23.23 % ▶ parserOnIncoming
2.4 ms	0.05 %	1126.0 ms 21.83 % ▶ (anonymous)
1.2 ms	0.02 %	1123.3 ms 21.78 % ▶ (anonymous)
2.2 ms	0.04 %	1122.2 ms 21.76 % ▶ app
5.7 ms	0.11 %	1119.7 ms 21.71 % ▶ handle
22.9 ms	0.44 %	1109.0 ms 21.50 % ▶ (anonymous)
2.5 ms	0.05 %	1090.8 ms 21.15 % ▶ query
28.9 ms	0.56 %	1075.3 ms 20.85 % ▶ expressInit
39.9 ms	0.77 %	1035.3 ms 20.07 % ▶ compression
20.3 ms	0.39 %	985.8 ms 19.11 % ▶ cookieParser

0x flamegraph standalone usando console.log(data)



0x flamegraph standalone sin usar console.log(data)

node .\32-LogsnPerf\main.js

cold hot + search function
optimized unoptimized



(they're the same picture)

Conclusion time:

console.log es, a fin de cuentas, un write al standard_output. write es una operación sincrónica. entre todas las request que le hacemos a /info, promediamos 375 ms de espera hasta que se envía la información al usuario, comparado con 200 ms si salteamos ese logueo. diff 87,5%