

1)

Analisis con console.log

```
Statistical profiling result from con_console-v8.log, (5417 ticks, 1 unaccounted, 0 excluded).

[Shared libraries]:
  ticks  total  nonlib   name
   4230   78.1%         C:\Windows\SYSTEM32\ntdll.dll
   1102   20.3%         C:\Program Files\nodejs\node.exe
     8    0.1%         C:\Windows\System32\KERNELBASE.dll
     1    0.0%         C:\Windows\system32\mswsock.dll
     1    0.0%         C:\Windows\System32\KERNEL32.DLL
```

Análisis sin console.log

```
Statistical profiling result from sin_console-v8.log, (9455 ticks, 2 unaccounted, 0 excluded).

[Shared libraries]:
  ticks  total  nonlib   name
   8391   88.7%         C:\Windows\SYSTEM32\ntdll.dll
    995   10.5%         C:\Program Files\nodejs\node.exe
     2    0.0%         C:\Windows\System32\KERNELBASE.dll
```

3)

Resultados de análisis de autocannon

Analisis con console.log

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	446 ms	554 ms	1085 ms	1173 ms	586.06 ms	136.52 ms	1325 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	22	22	182	205	169.2	41.03	22
Bytes/Sec	39.8 kB	39.8 kB	329 kB	371 kB	306 kB	74.2 kB	39.8 kB

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

3k requests in 20.28s, 6.12 MB read

Analisis sin console.log

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	358 ms	420 ms	595 ms	827 ms	443.06 ms	79.65 ms	959 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	100	100	223	287	224.5	41.69	100
Bytes/Sec	181 kB	181 kB	404 kB	520 kB	406 kB	75.5 kB	181 kB

Req/Bytes counts sampled once per second.

of samples: 20

5k requests in 20.36s, 8.13 MB read

Comparando los resultados de los cuadros se puede ver cómo sin console.log trabaja más rápido, pudiendo atender muchas más requests por segundo.

Diagramas de flama

Con console



Sin console

