Resultados de pruebas de performance

Artillery - No bloqueante

Node Profiler - No Bloqueante

[Summary]:		
ticks	total	nonlib	name
20	0.1%	90.9%	JavaScript
0	0.0%	0.0%	C++
14	0.1%	63.6%	GC
14237	99.8%		Shared libraries
2	0.0%		Unaccounted

Artillery - Bloqueante

All VUs finished. Total time: 57 seconds	
Summary report @ 18:07:25(-0300)	
vusers.created by name.1:	24
vusers.created.total:	50
<pre>vusers.created_by_name.Get info:</pre>	26
vusers.completed:	50
vusers.session_length:	
	0.1
max:	
	0.6
P	11274.1
p99:	
+	9/sec
	480
	480
http.responses: http.response time:	400
min:	9
	1120
median:	
p95:	
p99:	1022.7

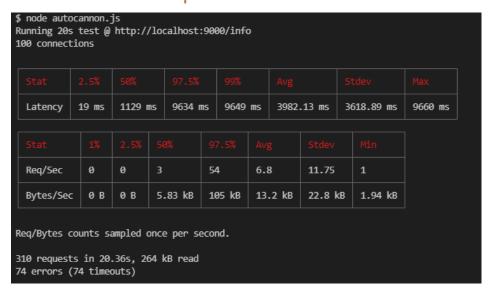
Node Profiler - Bloqueante

[Summary]:		
ticks	total	nonlib	name
29	0.2%	100.0%	JavaScript
0	0.0%	0.0%	C++
13	0.1%	44.8%	GC
11963	99.8%		Shared libraries

Autocannon - No Bloqueante

Latency	13 ms	4820 ms	9244 п	ıs 9246	ms	4150.	.11 ms	37	52.95 ms	9250 ms
Req/Sec	9	0	2	56	7.25	5	12.06		1	
Bytes/Sec	: 0В	0 B	3.89 kB	109 kB	14.1	L kB	23.4 k	:В	1.94 kB	
eg/Bytes counts sampled once per second.										

Autocannon - Bloqueante



Conclusión

En el reporte de artillery para el proceso no bloqueante se ve que el tiempo de respuesta promedio fue de 179.5 milisegundos, mientras que para el bloqueante fue de 190.6, unos 11 milisegundos más de demora. En el reporte del profiler de node se ve que para javascript se registraron 20 ticks consumidos para el proceso no bloqueante y 29 para el bloqueante.

En el reporte de autocannon se ve que el proceso no bloqueante tuvo una latencia máxima de 9250ms, menor a los 9660ms en el proceso bloqueante. Además, se ve que el primero demoró 20.21s en realizar 310 peticiones, mientras que el segundo demoró 20.36s para lo mismo.

En conclusión, si bien en este caso no se ven diferencias demasiado significativas, si es evidente que la inclusión de un sólo console.log impacta la performance del servidor.