

Informe de performance

Prof--Process

Route: <http://localhost:8080/infoBLOQ>

Statistical profiling result from blog-v8.log, (4453 ticks, 0 unaccounted, 0 excluded).

[Shared libraries]:

ticks	total	<u>nonlib</u>	name
4271	95.9%		C:\WINDOWS\SYSTEM32\ntdll.dll
174	3.9%		C:\Program Files\nodejs\node.exe
1	0.0%		C:\WINDOWS\System32\KERNELBASE.dll

[JavaScript]:

ticks	total	<u>nonlib</u>	name
2	0.0%	28.6%	LazyCompile: *resolve node:path:158:10
1	0.0%	14.3%	RegExp: ^((?:@[^\%]+\% \%)[^\%]*)(\%.*)?\$
1	0.0%	14.3%	LazyCompile: *normalizeString node:path:66:25
1	0.0%	14.3%	Function: ^nextTick node:internal/process/task_queues:103:18
1	0.0%	14.3%	Function: ^fromString C:\Users\Camila Capornio\Desktop\DesafioNº14\node_modules\mongodb\lib\utils.js:315:22
1	0.0%	14.3%	Function: ^FastBuffer node:internal/buffer:960:14

Route: <http://localhost:8080/info>

Statistical profiling result from noblog-v8.log, (1059 ticks, 0 unaccounted, 0 excluded).

[Shared libraries]:

ticks	total	<u>nonlib</u>	name
912	86.1%		C:\WINDOWS\SYSTEM32\ntdll.dll
143	13.5%		C:\Program Files\nodejs\node.exe

[JavaScript]:

ticks	total	<u>nonlib</u>	name
1	0.1%	25.0%	LazyCompile: *resolve node:path:158:10
1	0.1%	25.0%	LazyCompile: *isPathSeparator node:path:52:25
1	0.1%	25.0%	Function: ^push node:internal/fixed_queue:99:7
1	0.1%	25.0%	Function: ^Module node:internal/modules/cjs/loader:206:16

Artillery

Artillery_Slow:

Phase started: unnamed (index: 0, duration: 1s) 17:14:25(-0300)

Phase completed: unnamed (index: 0, duration: 1s) 17:14:26(-0300)

Metrics for period to: 17:14:30(-0300) (width: 1.562s)

http.codes.200: 1000

http.request_rate: 646/sec

http.requests: 1000

http.response_time:

min: 1

max: 116

median: 32.1

p95: 68.7

p99: 74.4

http.responses: 1000

vusers.completed: 50

vusers.created: 50

vusers.created_by_name.0: 50

vusers.failed: 0

vusers.session_length:

min: 257.2

max: 842.1

median: 713.5

p95: 837.3

p99: 837.3

All VUs finished. Total time: 3 seconds

Summary report @ 17:14:29(-0300)

http.codes.200: 1000

http.request_rate: 646/sec

http.requests: 1000

http.response_time:

 min: 1

 max: 116

 median: 32.1

 p95: 68.7

 p99: 74.4

http.responses: 1000

vusers.completed: 50

vusers.created: 50

vusers.created_by_name.0: 50

vusers.failed: 0

vusers.session_length:

 min: 257.2

 max: 842.1

 median: 713.5

 p95: 837.3

 p99: 837.3

Artillery_fast:

Phase started: unnamed (index: 0, duration: 1s) 09:44:40(-0300)

Phase completed: unnamed (index: 0, duration: 1s) 09:44:41(-0300)

Metrics for period to: 09:44:50(-0300) (width: 1.263s)

http.codes.200: 1000

http.request_rate: 799/sec

http.requests: 1000

http.response_time:

min: 0

max: 60

median: 15

p95: 41.7

p99: 48.9

http.responses: 1000

vusers.completed: 50

vusers.created: 50

vusers.created_by_name.0: 50

vusers.failed: 0

vusers.session_length:

min: 83.6

max: 477.7

median: 368.8

p95: 478.3

p99: 478.3

All VUs finished. Total time: 3 seconds

Summary report @ 09:44:44(-0300)

http.codes.200: 1000

http.request_rate: 799/sec

http.requests: 1000

http.response_time:

min: 0

max: 60

median: 15

p95: 41.7

p99: 48.9

http.responses: 1000

vusers.completed: 50

vusers.created: 50

vusers.created_by_name.0: 50

vusers.failed: 0

vusers.session_length:

min: 83.6

max: 477.7

median: 368.8

p95: 478.3

p99: 478.3

Autocannon:

Route: <http://localhost:8080/infoBLOQ>

```
$ npm test

> envandprocess@1.0.0 test
> node benchmark.js

Corriendo el benchmark...
Running 20s test @ http://localhost:8080/infoBLOQ
100 connections
```

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	33 ms	42 ms	75 ms	311 ms	51.57 ms	68.89 ms	982 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	100	100	2251	2357	1924.7	688.24	100
Bytes/Sec	57.4 kB	57.4 kB	1.29 MB	1.35 MB	1.1 MB	395 kB	57.4 kB

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

39k requests in 20.1s, 22.1 MB read

Route: <http://localhost:8080/info>

```
$ npm test
```

```
> envandprocess@1.0.0 test
> node benchmark.js
```

Corriendo el benchmark...
Running 20s test @ <http://localhost:8080/info>
100 connections

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	32 ms	39 ms	65 ms	75 ms	42.32 ms	9.17 ms	108 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	1462	1462	2409	2531	2337.31	245.46	1462
Bytes/Sec	840 kB	840 kB	1.38 MB	1.45 MB	1.34 MB	141 kB	839 kB

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

47k requests in 20.07s, 26.8 MB read

Inspect:

Route: <http://localhost:8080/infoBLOQ>

```

70 app.use(torMiddleware);
71 app.use(userRouter);
72
73 app.get('/info', compression(), (req, res) => {
74   4.2 ms   res.json({
75     1.2 ms   argumentos_de_entrada: process.argv.slice(2).join(' '),
76     0.1 ms   sistema_operativo: process.platform,
77     0.1 ms   version_de_Node: process.version,
78     0.3 ms   memoria_total_reservada: process.memoryUsage().rss,
79     0.5 ms   path_de_ejecucion: process.argv[0],
80     0.1 ms   process_id: process.pid,
81     0.5 ms   carpeta_del_proyecto: process.cwd(),
82     numero_de_procesadores: numCPUs,
83     1.8 ms   puerto: puerto,
84   });
85   0.7 ms   routeLogger(req, 'info');
86 });
87
88 console.log('ruta bloqueante')
89 res.json({
90   argumentos_de_entrada: process.argv.slice(2).join(' '),

```

proceso no bloqueante

Route: <http://localhost:8080/info>

```

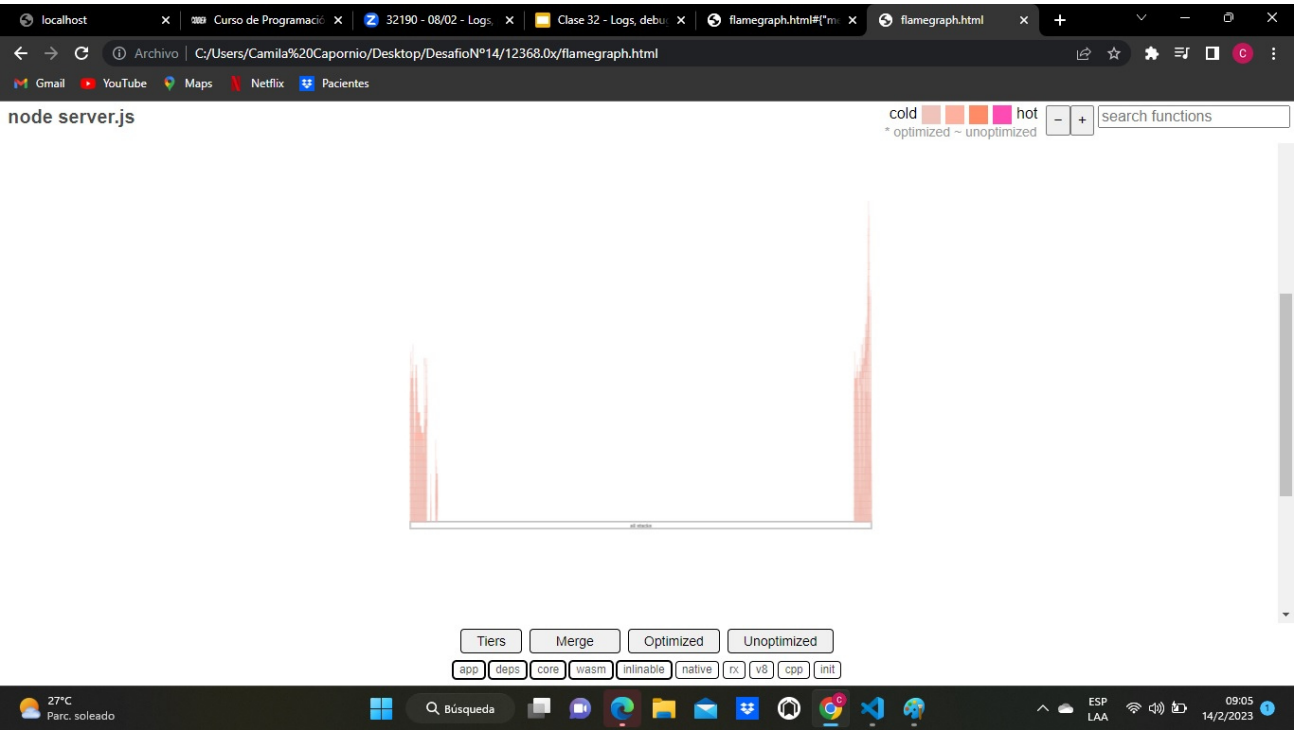
81   carpeta_del_proyecto: process.cwd(),
82   numero_de_procesadores: numCPUs,
83   puerto: puerto,
84 });
85 routeLogger(req, 'info');
86 });
87 app.get('/infoBLOQ', compression(), (req, res) => {
88   1.3 ms   console.log('ruta bloqueante')
89   7.2 ms   res.json({
90     1.1 ms   argumentos_de_entrada: process.argv.slice(2).join(' '),
91     sistema_operativo: process.platform,
92     version_de_Node: process.version,
93     0.3 ms   memoria_total_reservada: process.memoryUsage().rss,
94     0.1 ms   path_de_ejecucion: process.argv[0],
95     process_id: process.pid,
96     0.1 ms   carpeta_del_proyecto: process.cwd(),
97     0.1 ms   numero_de_procesadores: numCPUs,
98     puerto: puerto,
99   });
100   0.5 ms   routeLogger(req, 'info');
101 });
102
103
104

```

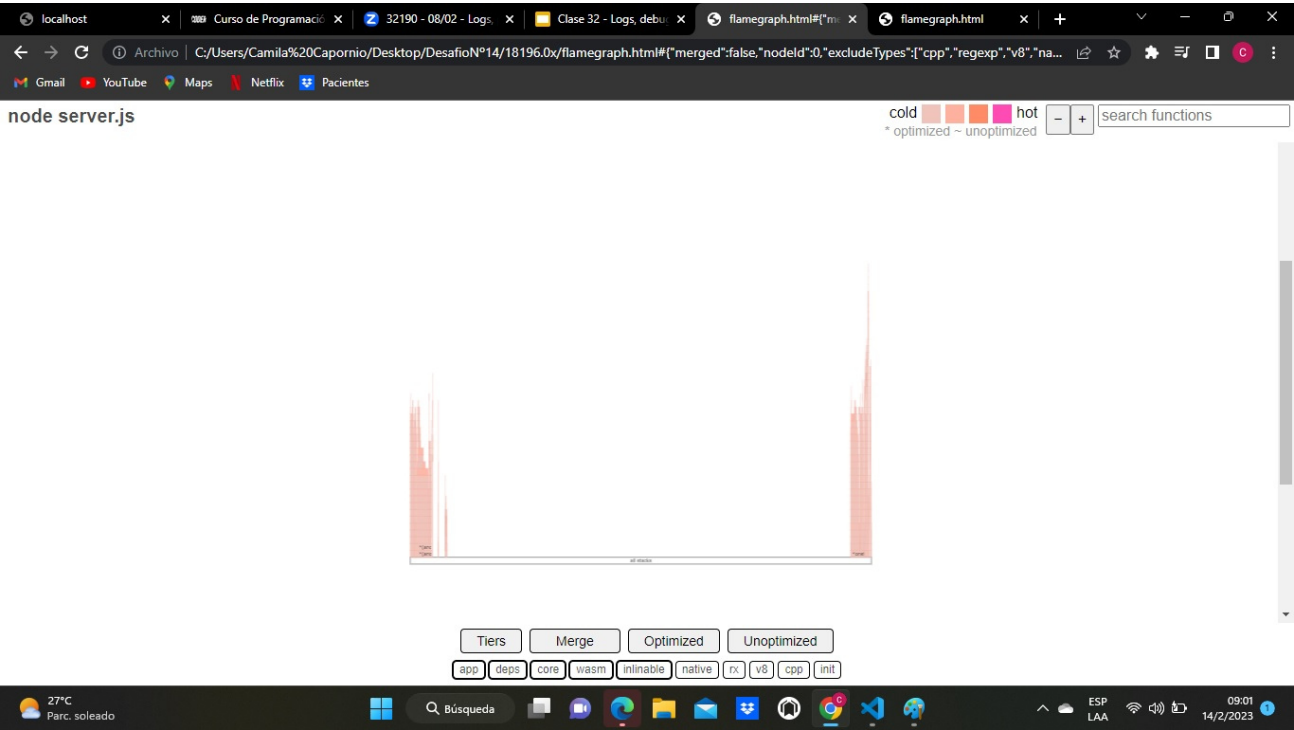
Proceso bloqueante

0x:

Route: <http://localhost:8080/infoBLOQ>



Route: <http://localhost:8080/info>



Conclusión:

En las pruebas realizadas se puede observar una media de tiempo de ejecución menor en cada request, un numero de ticks menor y una mayor cantidad de req/seg en favor de la ruta que no posee el `console.log()`, lo cual resulta esperable debido a que esta es una función sincrona y no permitirá que se ejecute la siguiente linea de código hasta que la misma se haya completado. En el gráfico de flama la deferencia no es tan apreciable a simple vista debido a que la tarea a realizar era relativamente corta y es difícil realizar una captura de pantalla con el detalle necesario.