

Análisis de performance

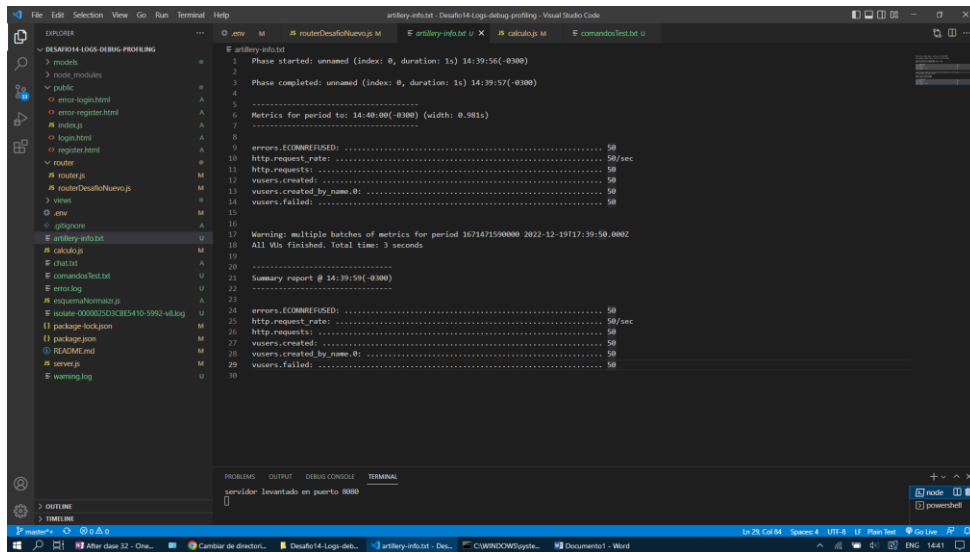
Iniciando el servidor en modo profiling

node --prof server.js

Iniciando el test de carga con Artillery

Emulando 50 conexiones concurrentes con 20 request por cada una:

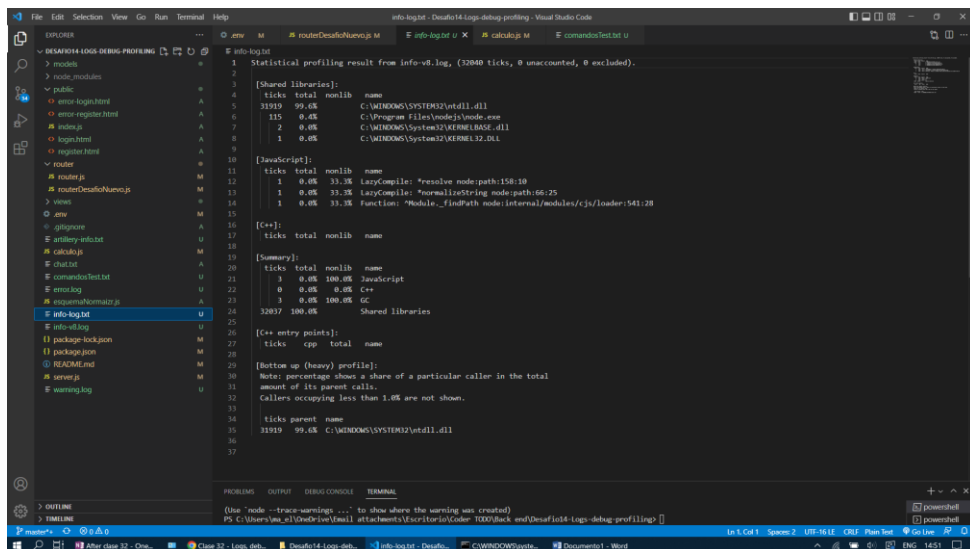
artillery quick -c 50 -n 20 "http://localhost:8081/info" > artillery-info.txt



```
artillery-quick: Default14-logs-debug-profiling - Visual Studio Code
1 Phase started: unnamed (index: 0, duration: 1s) 14:39:55(-0300)
2
3 Phase completed: unnamed (index: 0, duration: 1s) 14:39:57(-0300)
4
5 Metrics for period to: 14:40:00(-0300) (width: 0.981s)
6
7
8 errors.{CONNECTED}: ..... 50
9 http.request.rate: ..... 50/sec
10 http.requests: ..... 50
11 users.created: ..... 50
12 users.failed: ..... 50
13 users.failed_by_name:0: ..... 50
14 users.failed: ..... 50
15
16 Warning: multiple batches of metrics for period 1671471500000 2022-12-19T17:39:50.000Z
17 All VUs finished. Total time: 3 seconds
18
19 Summary report @ 14:39:59(-0300)
20
21
22 errors.{CONNECTED}: ..... 50
23 http.request.rate: ..... 50/sec
24 http.requests: ..... 50
25 users.created: ..... 50
26 users.failed: ..... 50
27 users.failed_by_name:0: ..... 50
28 users.failed: ..... 50
29
30
```

Decodificando la información

node --prof-process info-v8.log > info-log.txt



```
info-log.txt: Default14-logs-debug-profiling - Visual Studio Code
1 Statistical profiling result from info-v8.log, (32040 ticks, 0 unaccounted, 0 excluded).
2
3 [Shared libraries]:
4 ticks total nonlib name
5 31919 99.6% C:\WINDOWS\SYSTEM32\ntdll.dll
6 115 0.4% C:\Program Files\nodejs\node.exe
7 2 0.0% C:\WINDOWS\System32\USER32.dll
8 1 0.0% C:\WINDOWS\System32\USER32.dll
9
10 [JavaScript]:
11 ticks total nonlib name
12 1 0.0% 33.3% LazyCompile: Resolve node:path:150:10
13 1 0.0% 33.3% LazyCompile: normalizeString node:path:66:25
14 1 0.0% 33.3% Function: Module._findPath node:internal/modules/cjs/loader:541:28
15
16 [C++]:
17 ticks total nonlib name
18
19 [Summary]:
20 ticks total nonlib name
21 3 0.0% 100.0% JavaScript
22 0 0.0% 0.0% C++
23 3 0.0% 100.0% GC
24 32037 100.0% Shared libraries
25
26 [C++ entry points]:
27 ticks cpp total name
28
29 [Bottom up (heavy) profile]:
30 Note: percentage shows a share of a particular caller in the total
31 amount of its parent calls.
32 Callers occupying less than 1.0% are not shown.
33
34 ticks parent name
35 31919 99.6% C:\WINDOWS\SYSTEM32\ntdll.dll
36
37
```

AUTOCANNON Y 0X

npm start

npm run test

Autocannon

The screenshot shows the Visual Studio Code interface with the 'TERMINAL' panel active. The terminal displays the output of the command 'npm run test' for the 'desafio14-logs-debug-profiling' project. The output indicates that benchmarks are running in parallel and provides performance metrics for two different endpoints: '/randoms' and '/info'.

Running all benchmarks in parallel ...
Running 20s test @ http://localhost:8080/api/randoms
100 connections

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	4261 ms	4261 ms	8419 ms	8419 ms	6340 ms	2079 ms	8419 ms

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

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	0	0	0	1	0.1	0.31	1
Bytes/Sec	0 B	0 B	0 B	11.6 kB	1.16 kB	3.49 kB	11.6 kB

451 requests in 20.29s, 23.3 kB read
349 errors (197 timeouts)
Running 20s test @ http://localhost:8080/api/info
100 connections

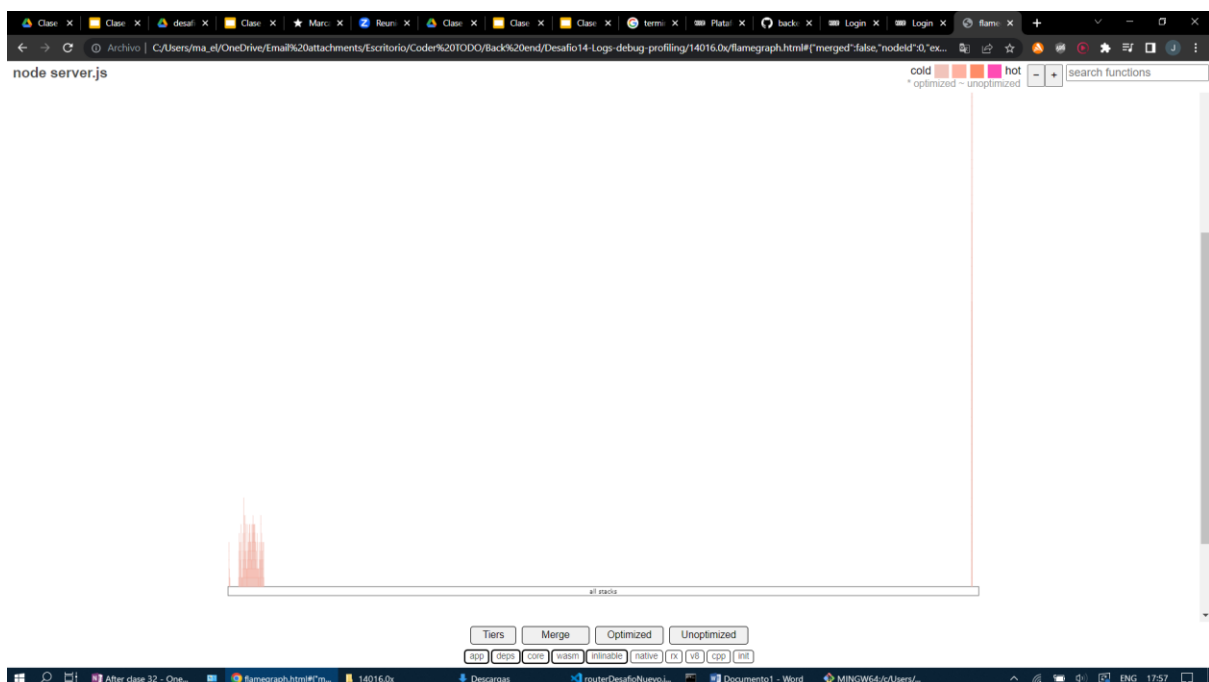
Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	0 ms	0 ms	0 ms	0 ms	0 ms	0 ms	0 ms

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

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	0	0	0	0	0	0	0
Bytes/Sec	0 B	0 B	0 B	0 B	0 B	0 B	0 B

429 requests in 20.31s, 0 B read
329 errors (200 timeouts)
PS C:\Users\ma_e\OneDrive\Email attachments\Escritorio\Coder T000\Back end\Desafio14-logs-debug-profiling>

0x flamegraph



Conclusión

Por lo que se puede apreciar en el flamegraph los procesos que más recursos consumen son los relacionados a los módulos de node.js