

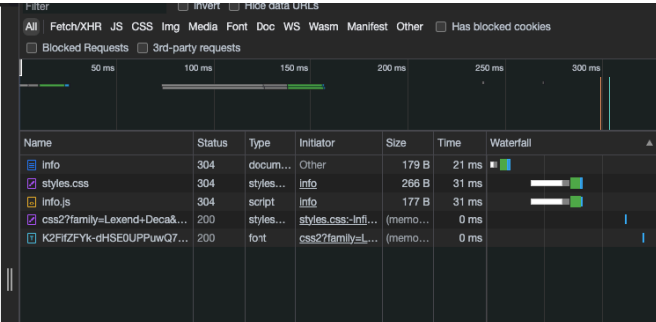
RESULTADOS DE TESTS.

1. Ruta /info con y sin compresión de Gzip:

Sin compresión 179B

Información del proyecto:

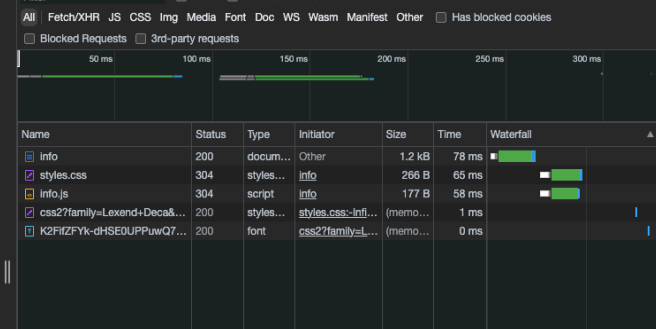
Argumentos:
Plataforma: darwin
Versión de Node JS: v16.17.0
Memoria total reservada: 109506560
Path de ejecución: /usr/local/bin/node
Process ID: 18419
Carpeta del proyecto: /Users/Juli/Desktop/CODER/Backend/desafios/backend-desafios/desafio14
Procesadores: 4



Con compresión 1.2kB

Información del proyecto:

Argumentos:
Plataforma: darwin
Versión de Node JS: v16.17.0
Memoria total reservada: 110682112
Path de ejecución: /usr/local/bin/node
Process ID: 17463
Carpeta del proyecto: /Users/Juli/Desktop/CODER/Backend/desafios/backend-desafios/desafio14
Procesadores: 4



Conclusión:

- Tamaño significativamente menor utilizando compress y menor tiempo de carga.

2. Artillery

Sin console.log:

```
artillery quick \  
  --count 50 \  
  --num 20 \  
  http://localhost:8080/info \  
  --output ./data/artillery.txt
```

Con console.log:

```
artillery quick \  
  --count 50 \  
  --num 20 \  
  http://localhost:8080/info \  
  --output ./data/artilleryConConsole.txt
```

Conclusión:

- Con console.log el tiempo de respuesta y carga del servidor es mayor.

3. Autocannon

Sin console.log:

```
MacBook-Air-de-Julietta:desafio14 Juli$ autocannon http://localhost:8080/info -d 20 -c 100
Running 20s test @ http://localhost:8080/info
100 connections
```

| Stat | 2.5% | 50% | 97.5% | 99% | Avg | Stdev | Max |
|---------|--------|--------|--------|--------|-----------|----------|--------|
| Latency | 139 ms | 195 ms | 337 ms | 362 ms | 209.12 ms | 55.61 ms | 550 ms |

| Stat | 1% | 2.5% | 50% | 97.5% | Avg | Stdev | Min |
|-----------|--------|--------|--------|--------|--------|---------|--------|
| Req/Sec | 365 | 365 | 471 | 578 | 476.05 | 63.07 | 365 |
| Bytes/Sec | 419 kB | 419 kB | 541 kB | 664 kB | 547 kB | 72.4 kB | 419 kB |

Req/Bytes counts sampled once per second.
of samples: 20
10k requests in 20.13s, 10.9 MB read

Con console.log:

```
MacBook-Air-de-Julieta:desafio14 Juli$ autocannon http://localhost:8080/info -d 20 -c 100
Running 20s test @ http://localhost:8080/info
100 connections
```

| Stat | 2.5% | 50% | 97.5% | 99% | Avg | Stdev | Max |
|---------|--------|--------|--------|--------|-----------|----------|--------|
| Latency | 198 ms | 299 ms | 485 ms | 515 ms | 317.31 ms | 79.92 ms | 737 ms |

| Stat | 1% | 2.5% | 50% | 97.5% | Avg | Stdev | Min |
|-----------|--------|--------|--------|--------|--------|---------|--------|
| Req/Sec | 214 | 214 | 315 | 361 | 312.86 | 36.64 | 214 |
| Bytes/Sec | 246 kB | 246 kB | 362 kB | 414 kB | 359 kB | 42.1 kB | 246 kB |

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

6k requests in 20.12s, 7.18 MB read

Conclusión: sin console.log es más rápida la respuesta (10k requests in 20.13s vs. 6k requests in 20.12s)

4. Node inspect

```
MacBook-Air-de-Julieta:desafio14 Juli$ node --inspect server.js
```

```
Debugger listening on ws://127.0.0.1:9229/3cb31aaf-4125-437e-85ae-2248908089aa
```

```
For help, see: https://nodejs.org/en/docs/inspector
```

```
Servidor http escuchando en el puerto 8080 en modo FORK
```

```
Debugger attached.
```

Sin console.log, procesos menos performantes:

| DevTools - Node.js | | | |
|---------------------------|------------------|---------------------|--|
| Connection | Console | Profiler | Sources Memory |
| Heavy (Bottom Up) ▾ 🔍 ⌕ ⌂ | | | |
| Profiles | Self Time ▾ | Total Time | Function |
| | 32573.3 ms | 32573.3 ms | (idle) |
| CPU PROFILES | 630.4 ms 26.05 % | 760.9 ms 31.44 % | ▶ consoleCall |
| Profile 1 Save | 98.4 ms 4.07 % | 98.4 ms 4.07 % | ▶ stat |
| | 98.1 ms 4.05 % | 98.1 ms 4.05 % | ▶ writev |
| | 90.8 ms 3.75 % | 90.8 ms 3.75 % | (garbage collector) |
| | 46.0 ms 1.90 % | 46.0 ms 1.90 % | ▶ open |
| | 42.7 ms 1.77 % | 148.4 ms 6.13 % | ▶ compile ejs.js:571 |
| | 36.8 ms 1.52 % | 36.8 ms 1.52 % | (program) |
| | 33.6 ms 1.39 % | 33.6 ms 1.39 % | ▶ writeUtf8String |
| | 33.4 ms 1.38 % | 13852.0 ms 572.41 % | ▶ next index.js:177 |
| | 29.7 ms 1.23 % | 49.8 ms 2.06 % | ▶ scanLine ejs.js:815 |
| | 25.8 ms 1.06 % | 451.0 ms 18.64 % | ▶ send response.js:111 |
| | 21.6 ms 0.89 % | 71.1 ms 2.94 % | ▶ (anonymous) ejs.js:742 |
| | 19.5 ms 0.81 % | 37.7 ms 1.56 % | ▶ Hash node:internal/crypto/hash:62 |
| | 19.4 ms 0.80 % | 1819.3 ms 75.18 % | ▶ initialize initialize.js:51 |
| | 18.0 ms 0.74 % | 18.0 ms 0.74 % | ▶ Hash |
| | 17.6 ms 0.73 % | 32.7 ms 1.35 % | ▶ writeHead node: http_server:292 |
| | 17.1 ms 0.71 % | 17.1 ms 0.71 % | ▶ _addOutput ejs.js:788 |
| | 15.9 ms 0.66 % | 52.7 ms 2.18 % | ▶ hash index.js:596 |
| | 15.7 ms 0.65 % | 3079.7 ms 127.26 % | ▶ handle index.js:136 |
| | 15.7 ms 0.65 % | 12797.8 ms 528.84 % | ▶ handle layer.js:86 |
| | 15.6 ms 0.64 % | 17.1 ms 0.71 % | ▶ parse index.js:106 |
| | 14.9 ms 0.61 % | 27.8 ms 1.15 % | ▶ (anonymous) */Users/Julio/De...es/info.ejs*:1 |
| | 13.3 ms 0.55 % | 24.9 ms 1.03 % | ▶ nextTick node:internal/p...ask_queues:104 |
| | 12.6 ms 0.52 % | 14.4 ms 0.59 % | ▶ createWriteWrap node:internal/s...se_commons:109 |
| ⋮ Console | | | |

Con console.log, procesos menos performantes:

| DevTools - Node.js | | | |
|---------------------------|-------------------|---------------------|---|
| Connection | Console | Profiler | Sources Memory |
| Heavy (Bottom Up) ▾ 🔍 ⌕ ⌂ | | | |
| Profiles | Self Time ▾ | Total Time | Function |
| | 42825.7 ms | 42825.7 ms | (idle) |
| CPU PROFILES | 1583.2 ms 41.08 % | 1864.5 ms 48.37 % | ▶ consoleCall |
| Profile 1 Save | 116.6 ms 3.03 % | 116.6 ms 3.03 % | ▶ writev |
| | 110.1 ms 2.86 % | 110.1 ms 2.86 % | ▶ stat |
| | 91.6 ms 2.38 % | 91.6 ms 2.38 % | (garbage collector) |
| | 56.4 ms 1.46 % | 56.4 ms 1.46 % | (program) |
| | 52.9 ms 1.37 % | 52.9 ms 1.37 % | ▶ getColorDepth node:internal/tty:106 |
| | 44.1 ms 1.14 % | 140.3 ms 3.64 % | ▶ compile ejs.js:571 |
| | 41.1 ms 1.07 % | 41.1 ms 1.07 % | ▶ open |
| | 36.4 ms 0.95 % | 3025.6 ms 78.50 % | ▶ initialize initialize.js:51 |
| | 32.9 ms 0.85 % | 32.9 ms 0.85 % | ▶ writeUtf8String |
| | 32.7 ms 0.85 % | 49.0 ms 1.27 % | ▶ scanLine ejs.js:815 |
| | 27.3 ms 0.71 % | 49.7 ms 1.29 % | ▶ nextTick node:internal/p...ask_queues:104 |
| | 26.9 ms 0.70 % | 25506.5 ms 661.76 % | ▶ next index.js:177 |
| | 25.2 ms 0.65 % | 522.3 ms 13.55 % | ▶ send response.js:111 |
| | 23.5 ms 0.61 % | 69.9 ms 1.81 % | ▶ hash index.js:596 |
| | 23.0 ms 0.60 % | 23.0 ms 0.60 % | ▶ Hash |
| | 22.8 ms 0.59 % | 46.3 ms 1.20 % | ▶ Hash node:internal/crypto/hash:62 |
| | 22.1 ms 0.57 % | 166.5 ms 4.32 % | ▶ value node:internal/c...onstructor:336 |
| | 21.1 ms 0.55 % | 38.5 ms 1.00 % | ▶ writeHead node: http_server:292 |
| | 20.0 ms 0.52 % | 3152.1 ms 81.78 % | ▶ session index.js:179 |
| | 19.6 ms 0.51 % | 281.1 ms 7.29 % | ▶ log node:internal/c...onstructor:375 |
| | 19.4 ms 0.50 % | 53.4 ms 1.39 % | ▶ formatRaw node:internal/util/inspect:828 |
| | 17.0 ms 0.44 % | 66.9 ms 1.74 % | ▶ resOnFinish node: http_server:816 |
| | 16.8 ms 0.44 % | 23340.6 ms 605.56 % | ▶ handle layer.js:86 |
| ⋮ Console | | | |

5. Diagrama de Flama 0x en la ruta routes/info:

Sin console.log

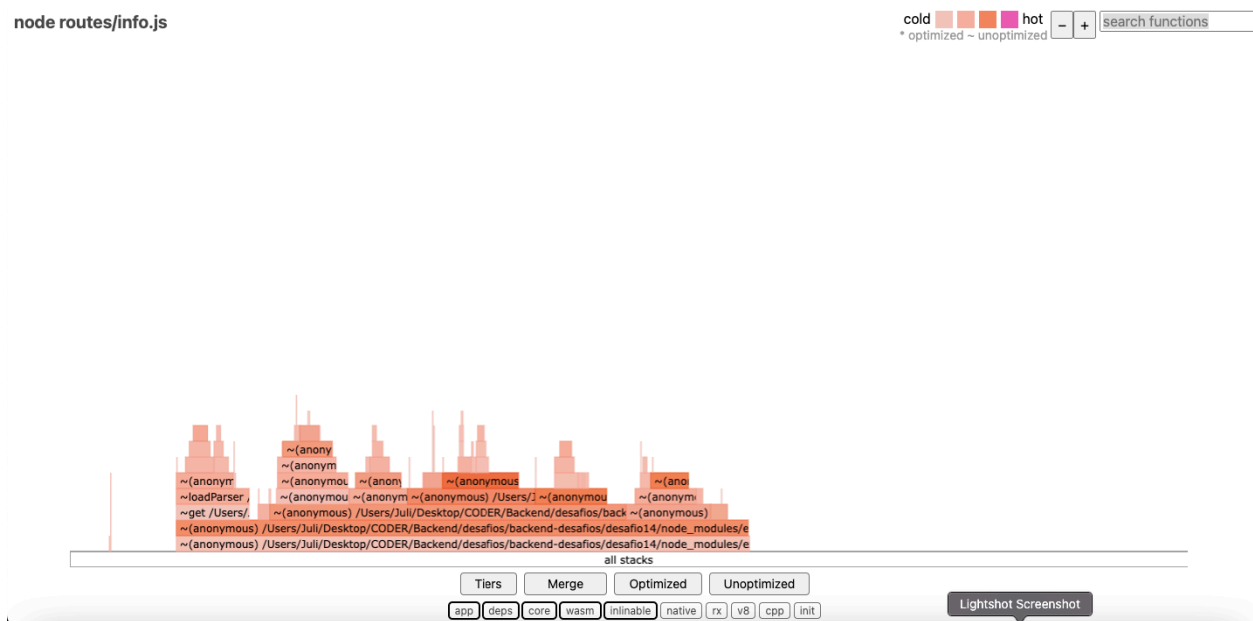
Carpeta 16037.0x dentro de data

MacBook-Air-de-Julieta:desafio14 Juli\$ 0x -o routes/info.js



Flamegraph generated in

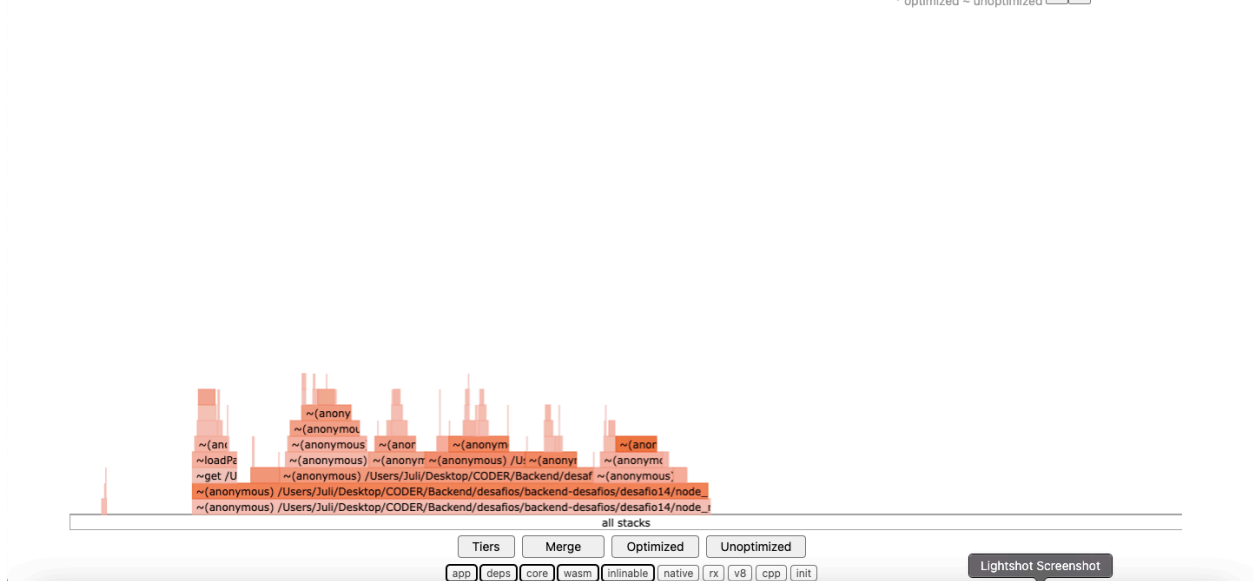
<file:///Users/Juli/Desktop/CODER/Backend/desafios/backend-desafios/desafio14/16037.0x/flamegraph.html>



Con console.log

node routes/info.js

cold hot
* optimized ~ unoptimized - + search functions



Conclusión general:

- Si bien no es mucha la diferencia porque la información no es muy extensa, en todos los casos la respuesta fue más rápida sin el console.log y el esfuerzo del servidor fue menor.