

Compresión GZIP

Con GZIP:

Estado	Método	Dominio	Archivo	Iniciador	Tipo	Transferido	Tamaño
302	POST	localhost:8080	login	document	html	1,32 KB	3,30 KB
200	GET	localhost:8080	/	document	html	1,46 KB	3,30 KB
200	GET	localhost:8080	style.css	stylesheet	css	1,02 KB	537 B
200	GET	localhost:8080	socket.io.js	script	js	25,42 KB	111,72 KB
200	GET	localhost:8080	app.js	script	js	1,73 KB	5,17 KB
200	GET	cdn.jsdelivrn...	handlebars.j	script	js	40,98 KB	175,10 KB
200	GET	cdn.jsdelivrn...	bootstrap.m	stylesheet	css	29,28 KB	188,99 KB
200	GET	cdn.jsdelivrn...	normalizr.br	script	js	3,41 KB	6,86 KB
200	GET	localhost:8080	chat.hbs	app.js:1 (f...	x-handle...	919 B	398 B
404	GET	localhost:8080	favicon.	FaviconL...	html	610 B	150 B
200	GET	localhost:8080	productList.l	app.js:6 (f...	x-handle...	1,15 KB	660 B
200	GET	localhost:8080	/socket.io/?E	socket.io....	plain	283 B	118 B
200	GET	localhost:8080	user	app.js:32 ...	json	468 B	45 B
200	POST	localhost:8080	/socket.io/?E	socket.io....	html	149 B	2 B
101	GET	localhost:8080	/socket.io/?E	socket.io....	plain	129 B	0 B
200	GET	localhost:8080	/socket.io/?E	socket.io....	plain	196 B	32 B
200	GET	localhost:8080	/socket.io/?E	socket.io....	plain	164 B	1 B

Sin GZIP:

Estado	Método	Dominio	Archivo	Iniciador	Tipo	Transferido	Tamaño
302	POST	localhost:8080	login	document	html	3,68 KB	3,30 KB
200	GET	localhost:8080	/	document	html	3,78 KB	3,30 KB
200	GET	localhost:8080	style.css	stylesheet	css	1,00 KB	537 B
200	GET	localhost:8080	socket.io.js	script	js	25,42 KB	111,72 KB
200	GET	localhost:8080	app.js	script	js	5,66 KB	5,17 KB
200	GET	cdn.jsdelivrn...	handlebars.j	script	js	41,00 KB	175,10 KB
200	GET	cdn.jsdelivrn...	bootstrap.m	stylesheet	css	29,30 KB	188,99 KB
200	GET	cdn.jsdelivrn...	normalizr.br	script	js	3,42 KB	6,86 KB
200	GET	localhost:8080	chat.hbs	app.js:1 (f...	x-handle...	900 B	398 B
404	GET	localhost:8080	favicon.	FaviconL...	html	591 B	150 B
200	GET	localhost:8080	productList.l	app.js:6 (f...	x-handle...	1,13 KB	660 B
200	GET	localhost:8080	/socket.io/?E	socket.io....	plain	283 B	118 B
200	GET	localhost:8080	user	app.js:32 ...	json	449 B	45 B
200	POST	localhost:8080	/socket.io/?E	socket.io....	html	149 B	2 B
200	GET	localhost:8080	/socket.io/?E	socket.io....	plain	196 B	32 B
101	GET	localhost:8080	/socket.io/?E	socket.io....	plain	129 B	0 B
200	GET	localhost:8080	/socket.io/?E	socket.io....	plain	164 B	1 B

Profiling con y sin console.log

Sin Console Log:

```
[Summary]:
  ticks  total  nonlib   name
    13    0.7%  100.0%  JavaScript
     0    0.0%   0.0%   C++
     7    0.4%  53.8%   GC
  1796   99.3%           Shared libraries
```

```
http.codes.200: ..... 1000
http.request_rate: ..... 291/sec
http.requests: ..... 1000
http.response_time:
  min: ..... 4
  max: ..... 98
  median: ..... 43.4
  p95: ..... 76
  p99: ..... 94.6
http.responses: ..... 1000
vusers.completed: ..... 50
vusers.created: ..... 50
vusers.created_by_name.0: ..... 50
vusers.failed: ..... 0
vusers.session_length:
  min: ..... 502.3
  max: ..... 1066
  median: ..... 925.4
  p95: ..... 1064.4
  p99: ..... 1064.4
```

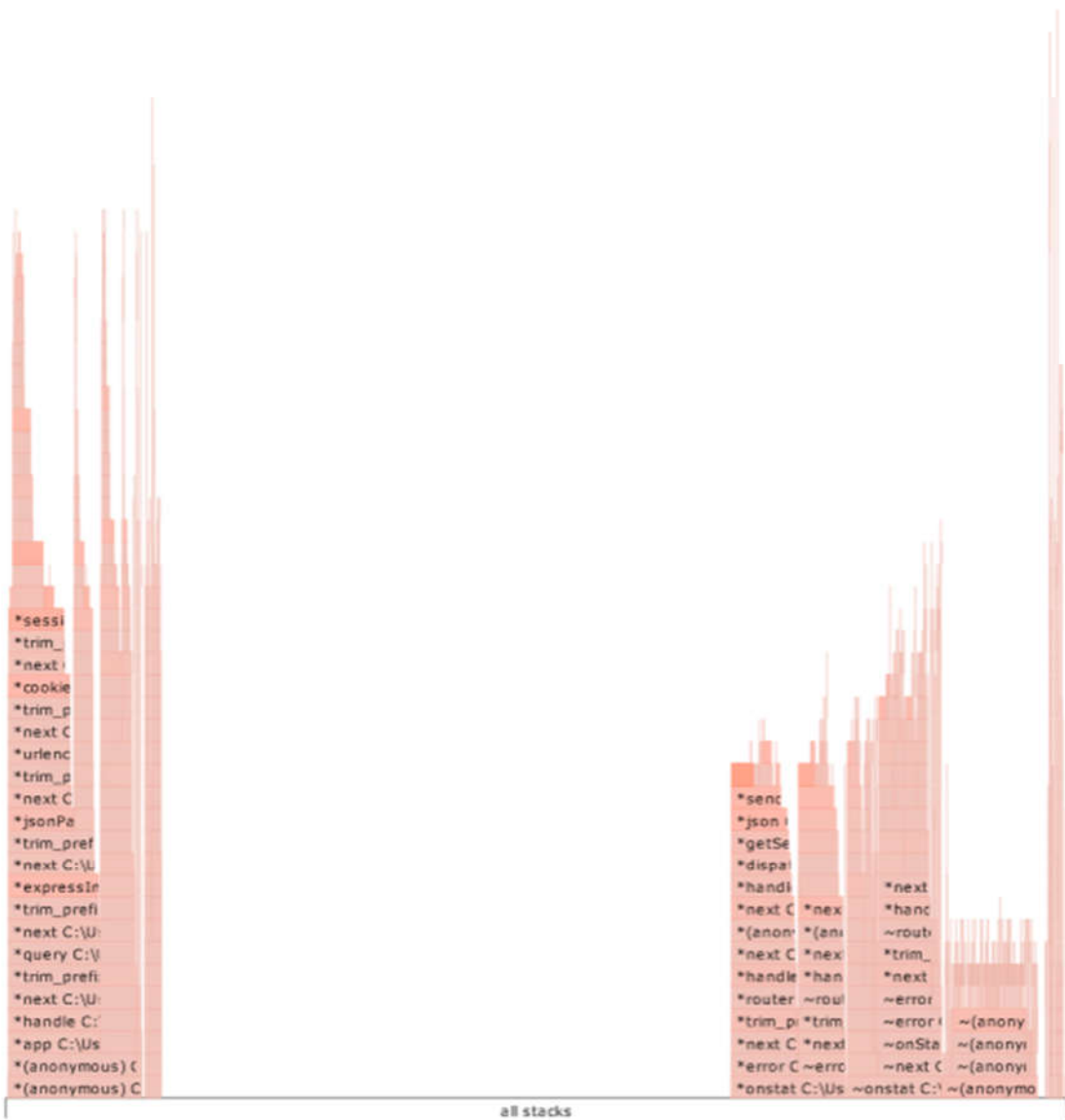
```
Running tests...
Running 20s test @ http://localhost:8080/info
100 connections
```

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	48 ms	59 ms	104 ms	117 ms	63.85 ms	15.3 ms	183 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	898	898	1581	1801	1554.6	228.07	898
Bytes/Sec	592 kB	592 kB	1.04 MB	1.19 MB	1.02 MB	150 kB	592 kB

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

```
31k requests in 20.09s, 20.5 MB read
```



Con Console Log:

```
[Summary]:
ticks  total  nonlib  name
   21    0.3%  100.0%  JavaScript
    0    0.0%    0.0%  C++
   12    0.2%   57.1%  GC
 7259   99.7%           Shared libraries
```

```
http.codes.200: ..... 1000
http.request_rate: ..... 104/sec
http.requests: ..... 1000
http.response_time:
  min: ..... 21
  max: ..... 396
  median: ..... 223.7
  p95: ..... 257.3
  p99: ..... 278.7
http.responses: ..... 1000
vusers.completed: ..... 50
vusers.created: ..... 50
vusers.created_by_name.0: ..... 50
vusers.failed: ..... 0
vusers.session_length:
  min: ..... 3052.5
  max: ..... 4179.4
  median: ..... 4065.2
  p95: ..... 4147.4
  p99: ..... 4147.4
```

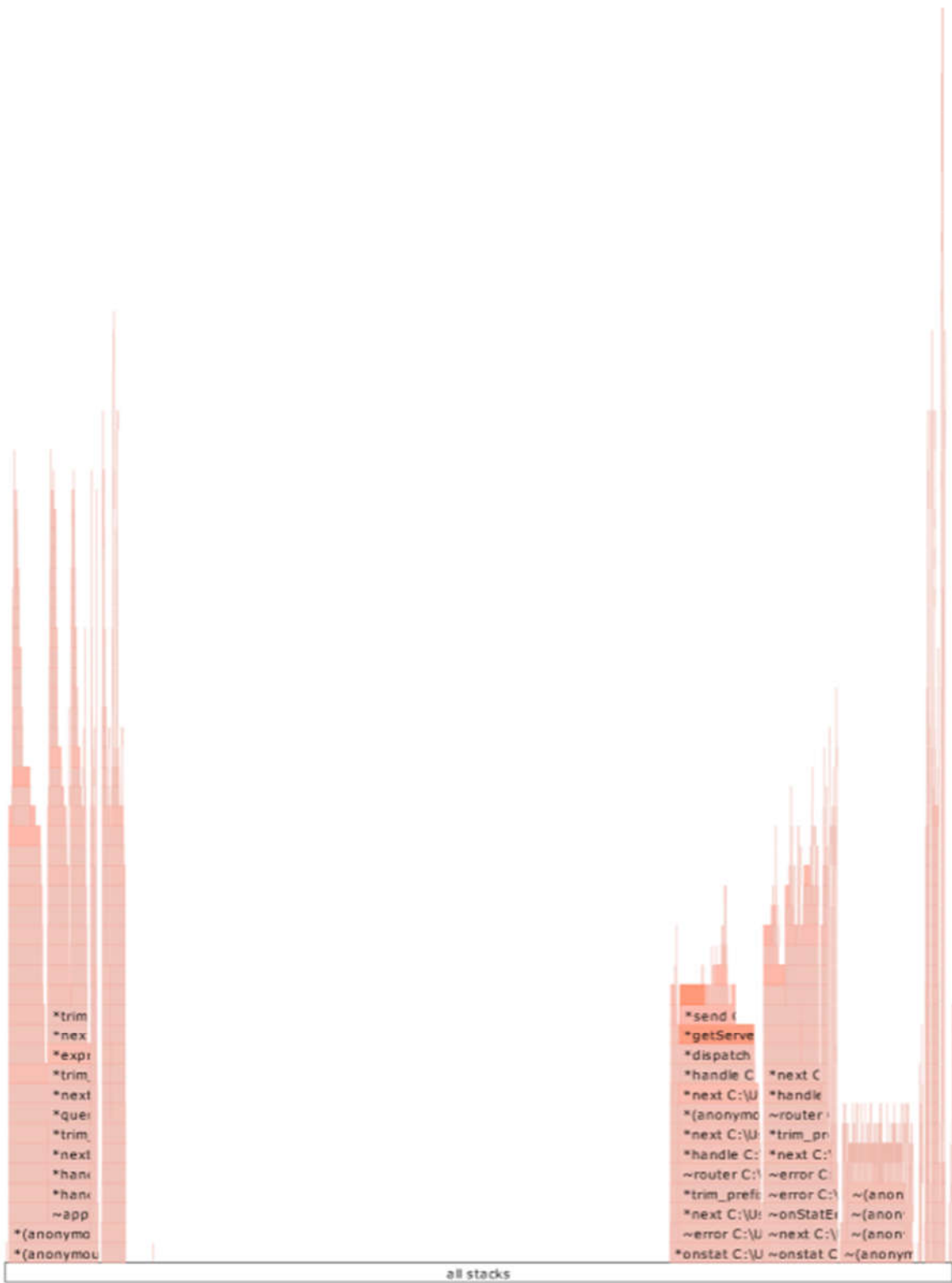
```
Running tests...
Running 20s test @ http://localhost:8080/info
100 connections
```

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	93 ms	112 ms	189 ms	204 ms	121.45 ms	26.95 ms	257 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	498	498	874	978	819	137.76	498
Bytes/Sec	329 kB	329 kB	577 kB	646 kB	541 kB	90.9 kB	329 kB

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

```
16k requests in 20.12s, 10.8 MB read
```



Conclusión

Cuando se loguean por consola los resultados antes de enviar la respuesta, se ve un incremento grande tanto en los ticks como en el tiempo de respuesta y, por consiguiente, una disminución en las respuestas enviadas por segundo.

En cuanto a la compresión con GZIP, la mayor diferencia en la cantidad de bytes se aprecia en scripts y archivos HTML. Por otro lado, noto que no hay mucha variación en archivos estáticos multimedia y hojas de estilo.