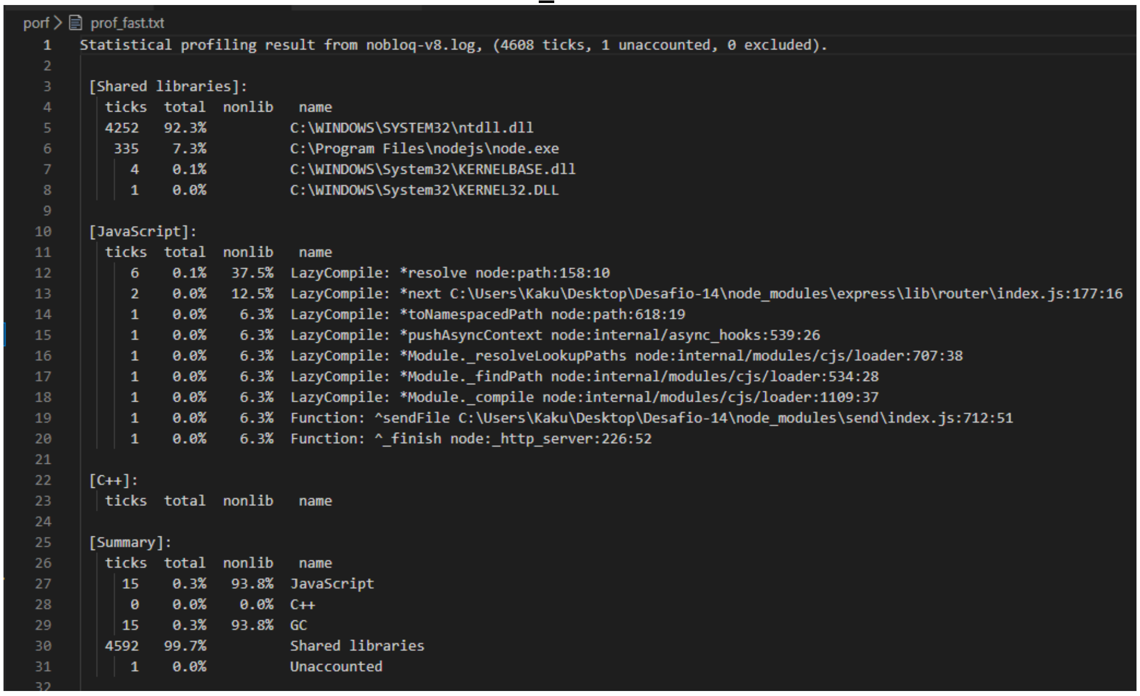
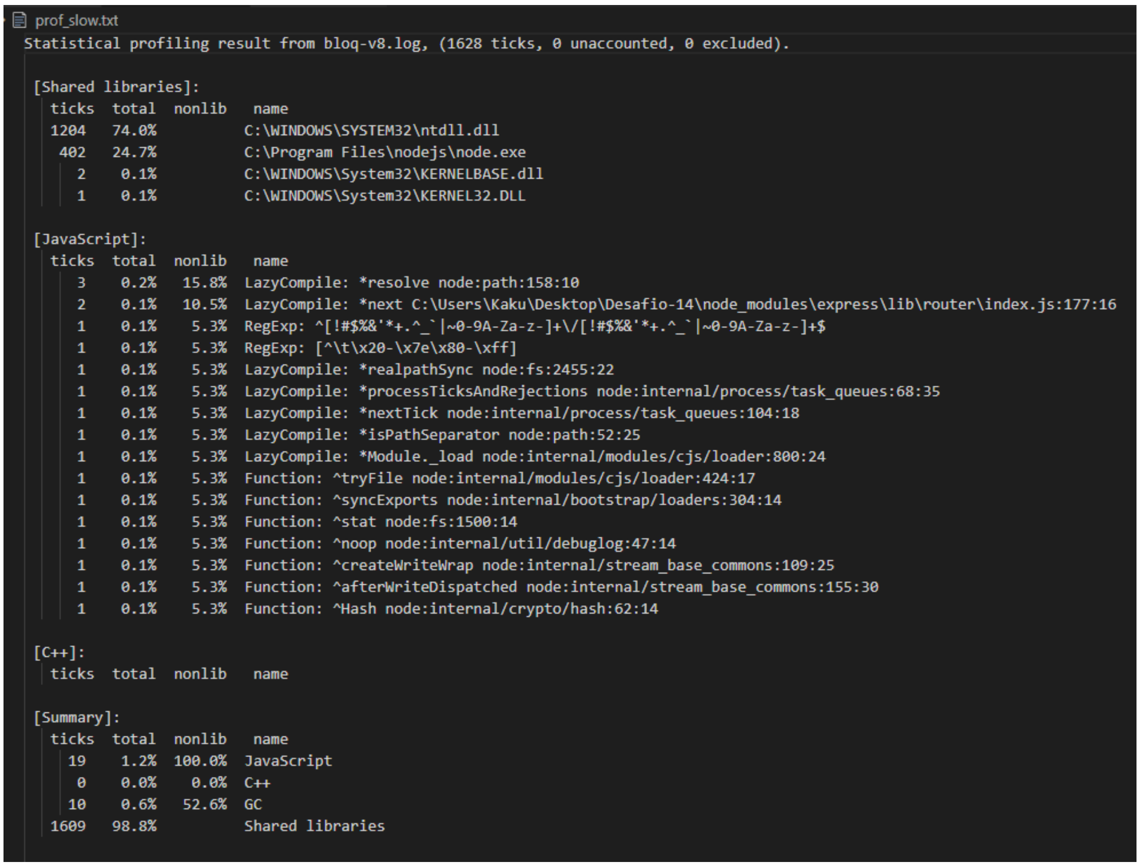
Prof_fast.txt



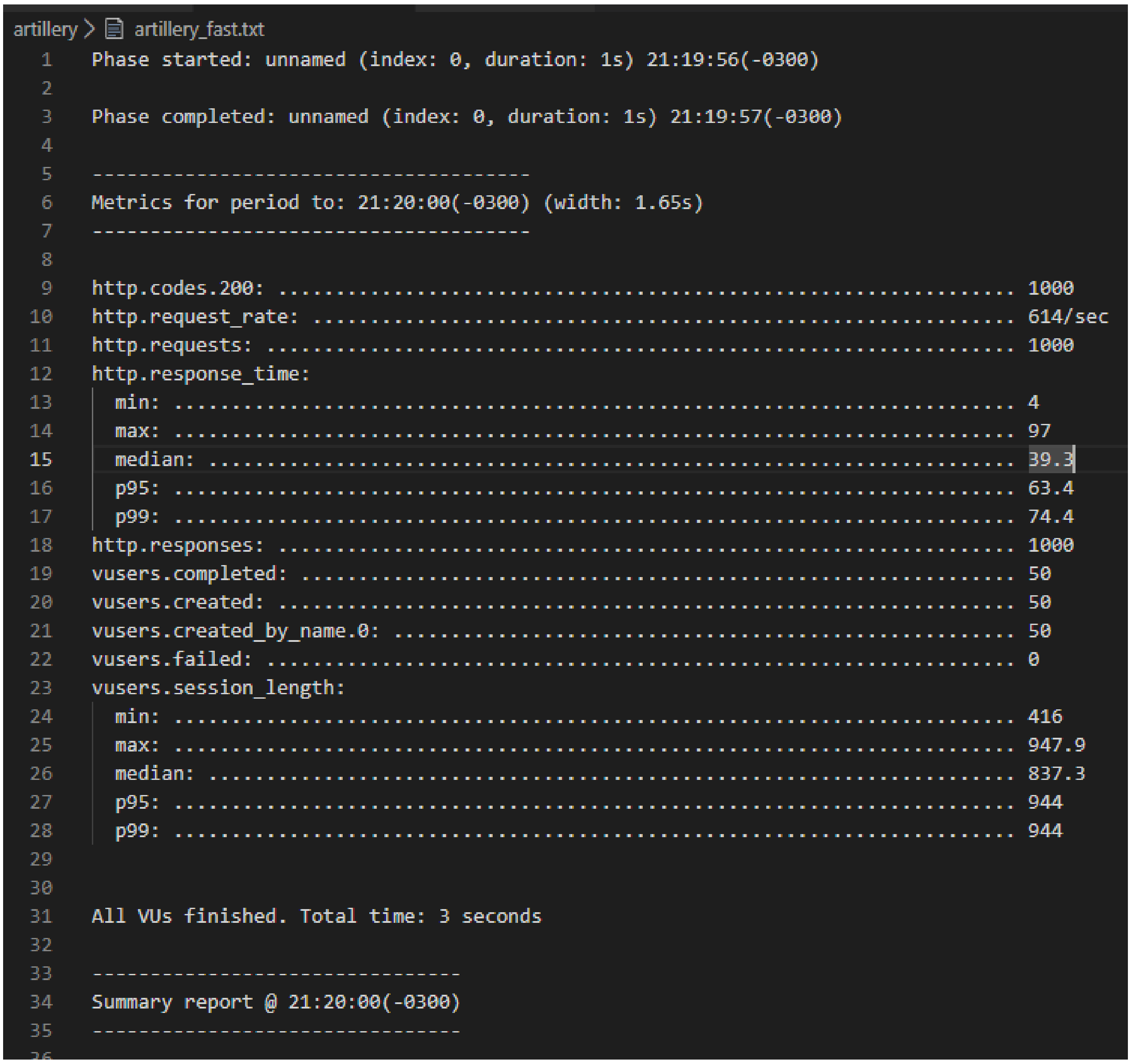
```
Statistical profiling result from nobloq-v8.log, (4608 ticks, 1 unaccounted, 0 excluded).

 [Shared libraries]:
   ticks  total  nonlib   name
   4252   92.3%          C:\WINDOWS\SYSTEM32\ntdll.dll
    335    7.3%          C:\Program Files\nodejs\node.exe
      4    0.1%          C:\WINDOWS\System32\KERNELBASE.dll
      1    0.0%          C:\WINDOWS\System32\KERNEL32.DLL

 [JavaScript]:
   ticks  total  nonlib   name
      6    0.1%   37.5%  LazyCompile: *resolve node:path:158:10
      2    0.0%   12.5%  LazyCompile: *next C:\Users\Kaku\Desktop\Desafio-14\node_modules\express\lib\router\index.js:177:16
      1    0.0%    6.3%  LazyCompile: *toNamespacedPath node:path:618:19
      1    0.0%    6.3%  LazyCompile: *pushAsyncContext node:internal/async_hooks:539:26
      1    0.0%    6.3%  LazyCompile: *Module._resolveLookupPaths node:internal/modules/cjs/loader:707:38
      1    0.0%    6.3%  LazyCompile: *Module._findPath node:internal/modules/cjs/loader:534:28
      1    0.0%    6.3%  LazyCompile: *Module._compile node:internal/modules/cjs/loader:1109:37
      1    0.0%    6.3%  Function: ^sendFile C:\Users\Kaku\Desktop\Desafio-14\node_modules\send\index.js:712:51
      1    0.0%    6.3%  Function: ^_finish node:_http_server:226:52

 [C++]:
   ticks  total  nonlib   name

 [Summary]:
   ticks  total  nonlib   name
     15    0.3%   93.8%  JavaScript
      0    0.0%    0.0%  C++
     15    0.3%   93.8%  GC
   4592   99.7%          Shared libraries
      1    0.0%          Unaccounted
```

Prof_slow.txt



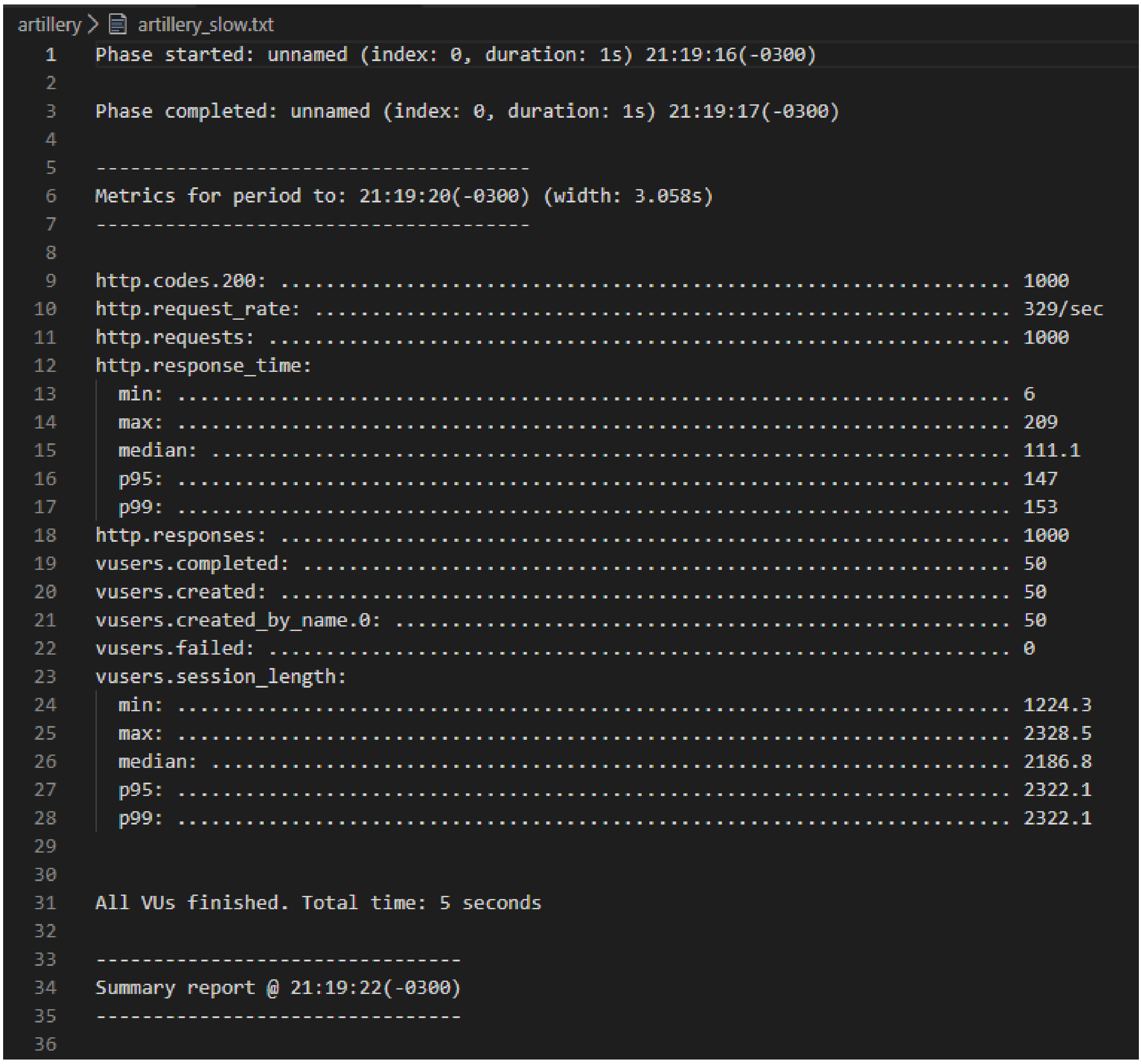
```
Statistical profiling result from bloq-v8.log, (1628 ticks, 0 unaccounted, 0 excluded).

 [Shared libraries]:
   ticks  total  nonlib   name
   1204   74.0%          C:\WINDOWS\SYSTEM32\ntdll.dll
    402   24.7%          C:\Program Files\nodejs\node.exe
      2    0.1%          C:\WINDOWS\System32\KERNELBASE.dll
      1    0.1%          C:\WINDOWS\System32\KERNEL32.DLL

 [JavaScript]:
   ticks  total  nonlib   name
      3    0.2%   15.8%  LazyCompile: *resolve node:path:158:10
      2    0.1%   10.5%  LazyCompile: *next C:\Users\Kaku\Desktop\Desafio-14\node_modules\express\lib\router\index.js:177:16
      1    0.1%    5.3%  RegExp: ^[!#$%&'*+.^_`|~0-9A-Za-z-]+\/[!#$%&'*+.^_`|~0-9A-Za-z-]+$
      1    0.1%    5.3%  RegExp: [^\t\x20-\x7e\x80-\xff]
      1    0.1%    5.3%  LazyCompile: *realpathSync node:fs:2455:22
      1    0.1%    5.3%  LazyCompile: *processTicksAndRejections node:internal/process/task_queues:68:35
      1    0.1%    5.3%  LazyCompile: *nextTick node:internal/process/task_queues:104:18
      1    0.1%    5.3%  LazyCompile: *isPathSeparator node:path:52:25
      1    0.1%    5.3%  LazyCompile: *Module._load node:internal/modules/cjs/loader:800:24
      1    0.1%    5.3%  Function: ^tryFile node:internal/modules/cjs/loader:424:17
      1    0.1%    5.3%  Function: ^syncExports node:internal/bootstrap/loaders:304:14
      1    0.1%    5.3%  Function: ^stat node:fs:1500:14
      1    0.1%    5.3%  Function: ^noop node:internal/util/debuglog:47:14
      1    0.1%    5.3%  Function: ^createWriteWrap node:internal/stream_base_commons:109:25
      1    0.1%    5.3%  Function: ^afterWriteDispatched node:internal/stream_base_commons:155:30
      1    0.1%    5.3%  Function: ^Hash node:internal/crypto/hash:62:14

 [C++]:
   ticks  total  nonlib   name

 [Summary]:
   ticks  total  nonlib   name
     19    1.2%  100.0%  JavaScript
      0    0.0%    0.0%  C++
     10    0.6%   52.6%  GC
   1609   98.8%          Shared libraries
```

Artillery_fast.txt



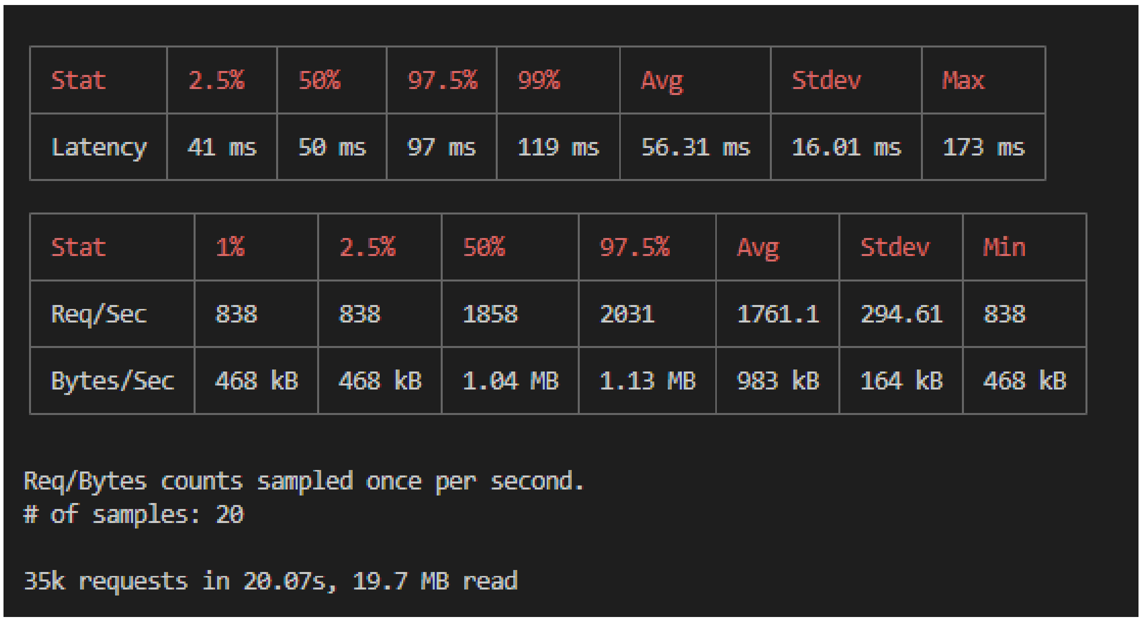
```
Phase started: unnamed (index: 0, duration: 1s) 21:19:56(-0300)

Phase completed: unnamed (index: 0, duration: 1s) 21:19:57(-0300)

--------------------------------------
Metrics for period to: 21:20:00(-0300) (width: 1.65s)
--------------------------------------

http.codes.200: ................................................................ 1000
http.request_rate: ............................................................. 614/sec
http.requests: ................................................................. 1000
http.response_time:
  min: ......................................................................... 4
  max: ......................................................................... 97
  median: ...................................................................... 39.3
  p95: ......................................................................... 63.4
  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: ......................................................................... 416
  max: ......................................................................... 947.9
  median: ...................................................................... 837.3
  p95: ......................................................................... 944
  p99: ......................................................................... 944


All VUs finished. Total time: 3 seconds

--------------------------------
Summary report @ 21:20:00(-0300)
--------------------------------
```

Artillery_slow.txt



```
Phase started: unnamed (index: 0, duration: 1s) 21:19:16(-0300)

Phase completed: unnamed (index: 0, duration: 1s) 21:19:17(-0300)

--------------------------------------
Metrics for period to: 21:19:20(-0300) (width: 3.058s)
--------------------------------------

http.codes.200: ................................................................ 1000
http.request_rate: ............................................................. 329/sec
http.requests: ................................................................. 1000
http.response_time:
  min: ......................................................................... 6
  max: ......................................................................... 209
  median: ...................................................................... 111.1
  p95: ......................................................................... 147
  p99: ......................................................................... 153
http.responses: ................................................................ 1000
vusers.completed: .............................................................. 50
vusers.created: ................................................................ 50
vusers.created_by_name.0: ...................................................... 50
vusers.failed: ................................................................. 0
vusers.session_length:
  min: ......................................................................... 1224.3
  max: ......................................................................... 2328.5
  median: ...................................................................... 2186.8
  p95: ......................................................................... 2322.1
  p99: ......................................................................... 2322.1


All VUs finished. Total time: 5 seconds

--------------------------------
Summary report @ 21:19:22(-0300)
--------------------------------
```

Autocannon NO Bloqueante

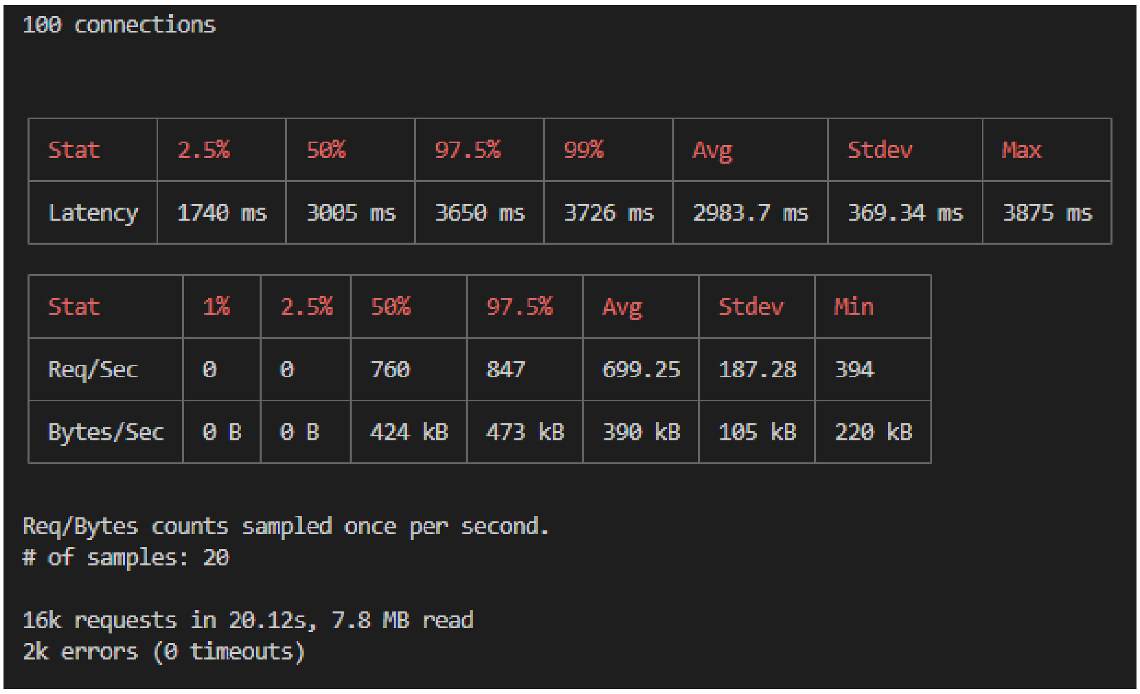


Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	41 ms	50 ms	97 ms	119 ms	56.31 ms	16.01 ms	173 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	838	838	1858	2031	1761.1	294.61	838
Bytes/Sec	468 kB	468 kB	1.04 MB	1.13 MB	983 kB	164 kB	468 kB

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

35k requests in 20.07s, 19.7 MB read
```

Autocannon Bloqueante



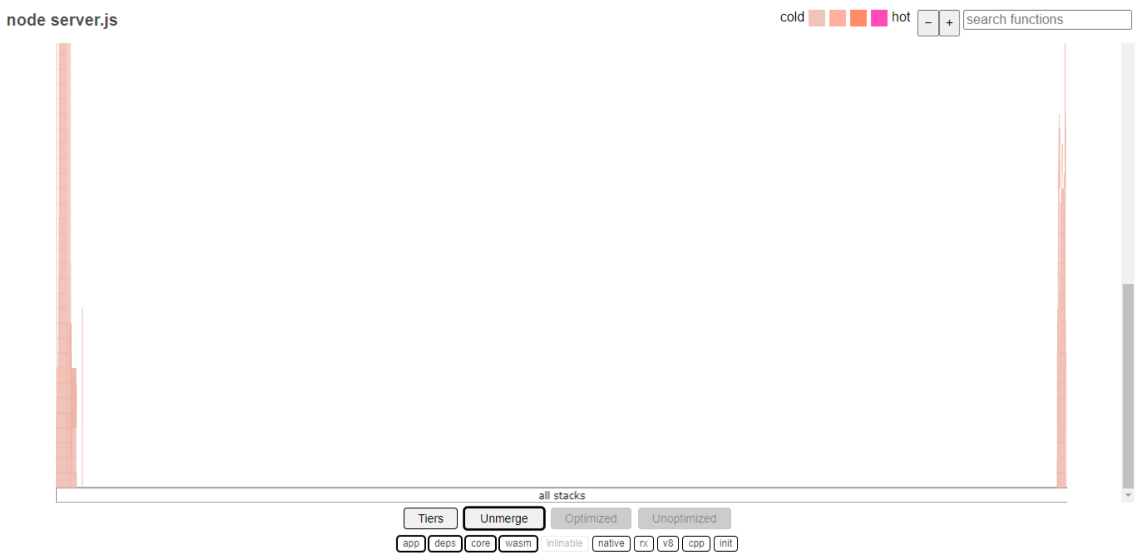
```
100 connections
```

Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	1740 ms	3005 ms	3650 ms	3726 ms	2983.7 ms	369.34 ms	3875 ms

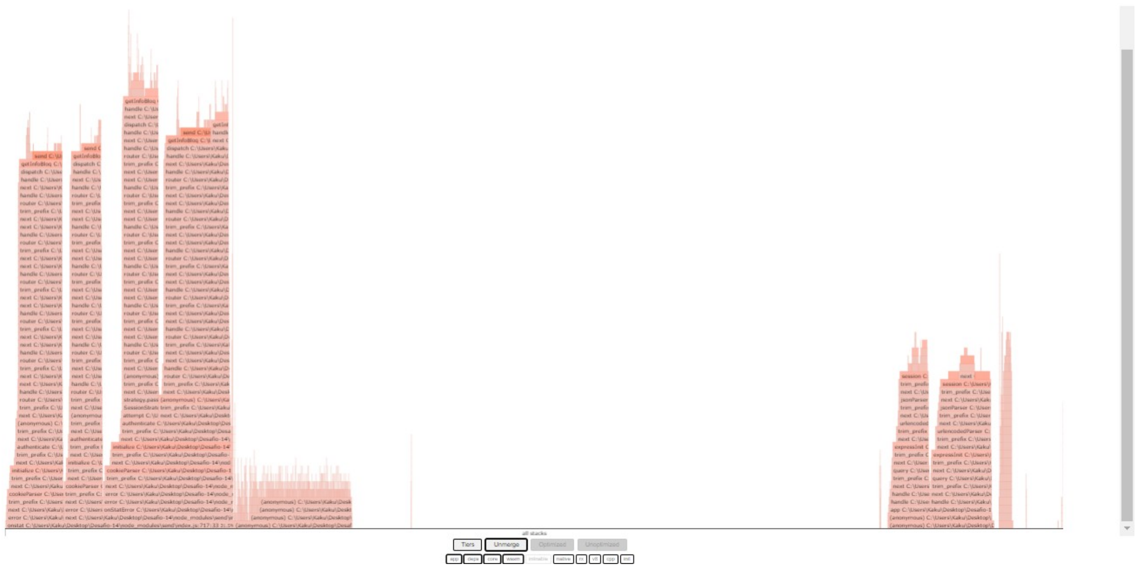
Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	0	0	760	847	699.25	187.28	394
Bytes/Sec	0 B	0 B	424 kB	473 kB	390 kB	105 kB	220 kB

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

16k requests in 20.12s, 7.8 MB read
2k errors (0 timeouts)
```

Flama NO Bloqueante



Flama Bloqueante



Conclusión:

1- Al realizar pruebas con profiling vemos en las imágenes de Prof_fast.txt(No Bloqueante) y Prof_slow.txt(Bloqueante) podemos observar que en las estadísticas de Prof_fast.txt quien no posee console.log en la ruta http://localhost:8080/info, obtiene 4608 ticks, mientras que en las estadísticas de Prof_slow.txt quien posee solamente 1 console.log en la ruta http://localhost:8080/infoBloq, obtiene 1628 ticks.

2- Al realizar pruebas con Artillery 50 conexiones con 20 request por cada una, en modo bloqueante artillery_fast.txt(sin console.log) y no bloqueante artillery_slow(con console.log), observamos que en el modo Bloqueante, el http.request_rate es de 614/sec y una media de 39.3, mientras que en el modo NO bloqueante, el http.request_rate es de 329/sec y una media de 111.1

3- Al realizar pruebas con Autocannon 100 conexiones durante un tiempo de 20 segundos observamos que el No Bloqueante recibe 35k requests en 20 segundos sin fallar ninguna, mientras que el modo Bloqueante recibe 18k requests en 20 segundos de las cuales 2k falla.

4- Al realizar Gráficos con Flama 0x No bloqueante y bloqueante, en el primer caso observamos que los procesos son más finos mientras que en el segundo son más anchos.

Analizando los 4 puntos anteriores, observando las amplias diferencias. Es recomendable escribir un código síncrono (no bloqueante) ya tiene más capacidades y es más eficiente.