Autor: Poggio Demian

Analisis de Performance

# Índice

[1 OSSNEO-1 - Ingestar en Neo Cluster Test nuevo 2](#_Toc67404399)

[2 OSSNEO-27 - Inclusión de Puertos Placas y relación Resources en NEOLoader 3](#_Toc67404400)

[3 OSSNEO-38 - Crear filtro por tipo de servicio para importación de Cu de Lisy 6](#_Toc67404401)

[4 OSSNEO-28 - Probar el end-2-end del proceso de FTTH/HFC/Cu en Outage 9](#_Toc67404402)

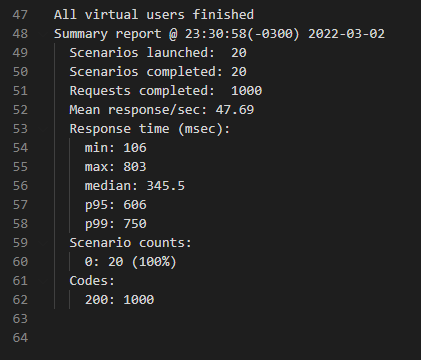
[5 OSSNEO-34 - Diego: tipos de elementos y roles de topología 12](#_Toc67404403)

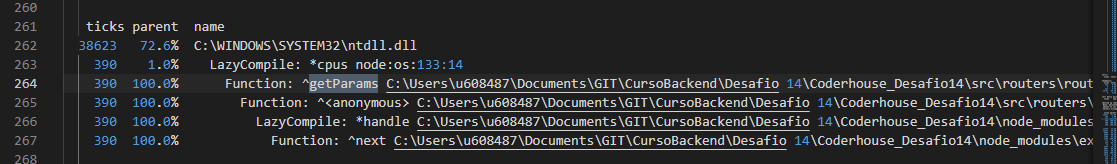
[6 OSSNEO-39 - Ingesta de archivos de sitios móviles 13](#_Toc67404404)

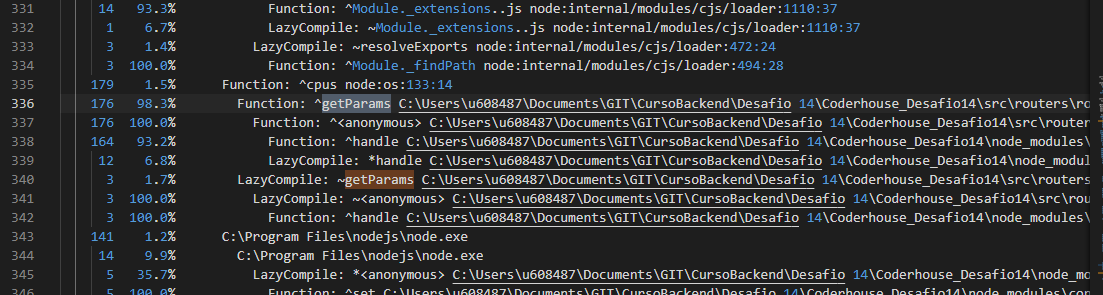
# Perfilamiento usando –prof y Artillery

Los archivos con el detalle se encuentran en la ruta: /reportes/nodeprof/artillery

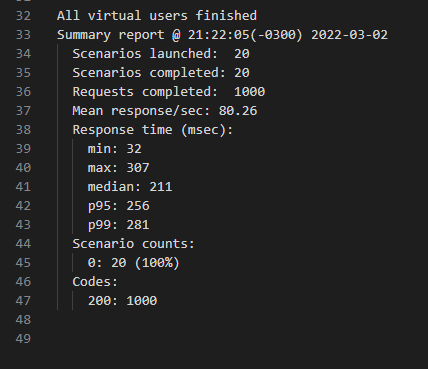
* Resultados Con console.log:

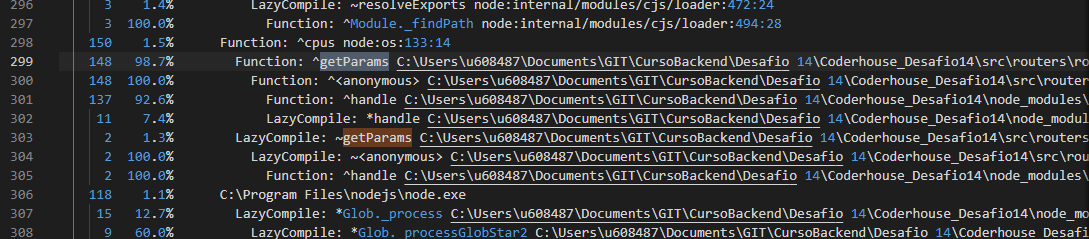






* Resultados Sin console.log:

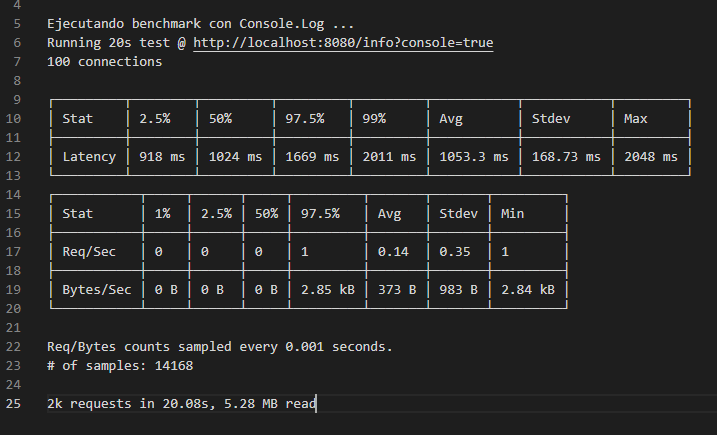


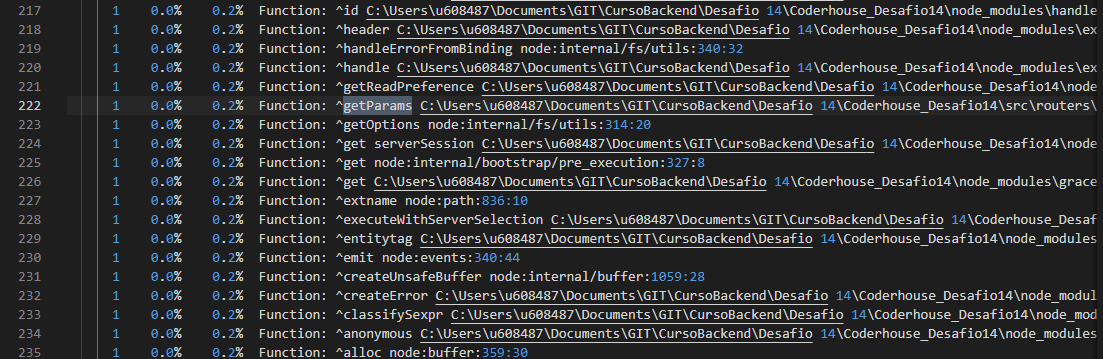


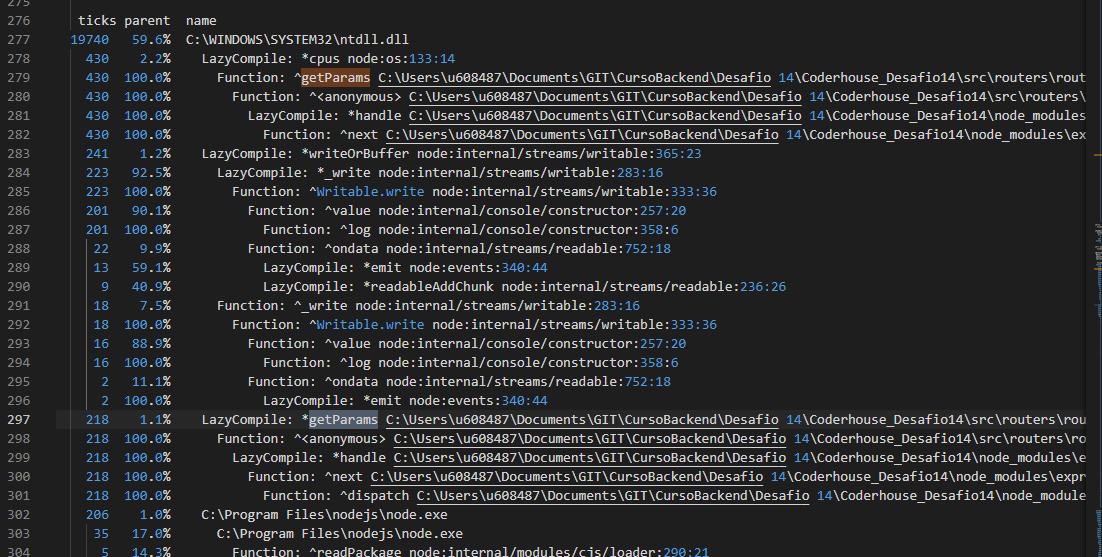
# Perfilamiento usando –prof y Autocannon

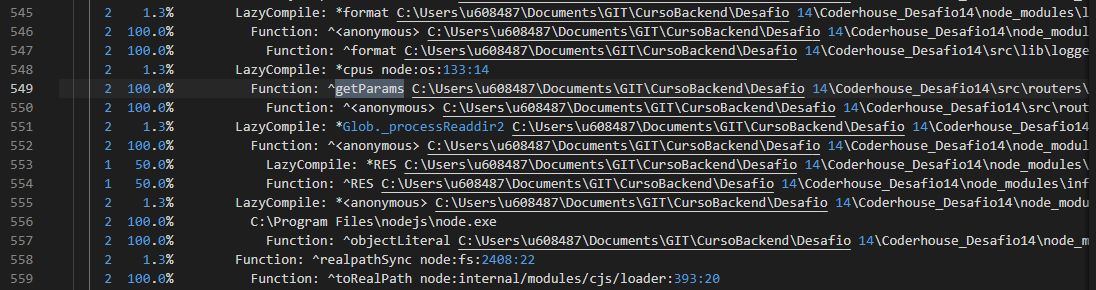
Los archivos con el detalle se encuentran en la ruta: /reportes/nodeprof/autocannon

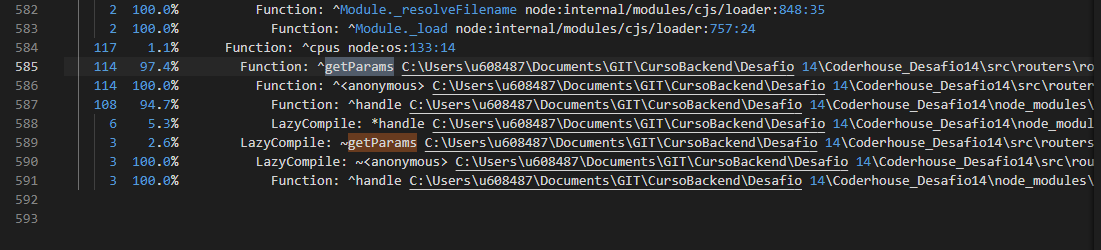
* Resultados Con console.log:



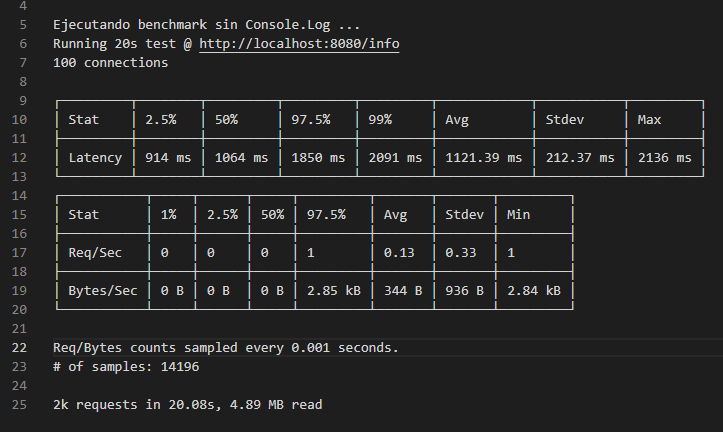


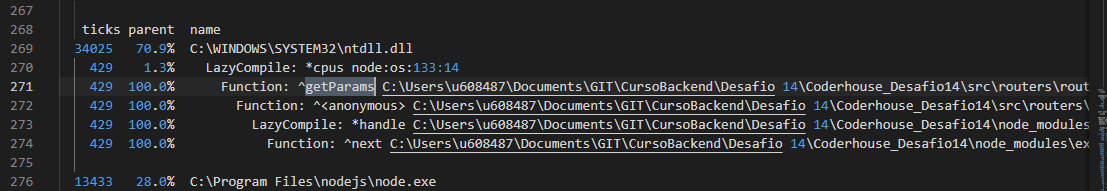


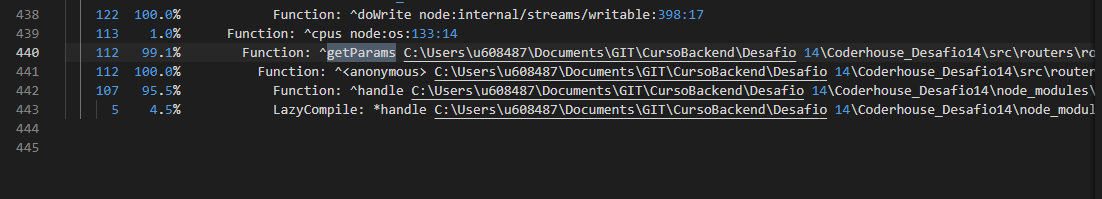




* Resultados Sin console.log:



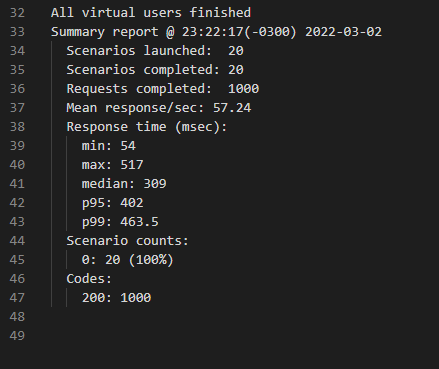


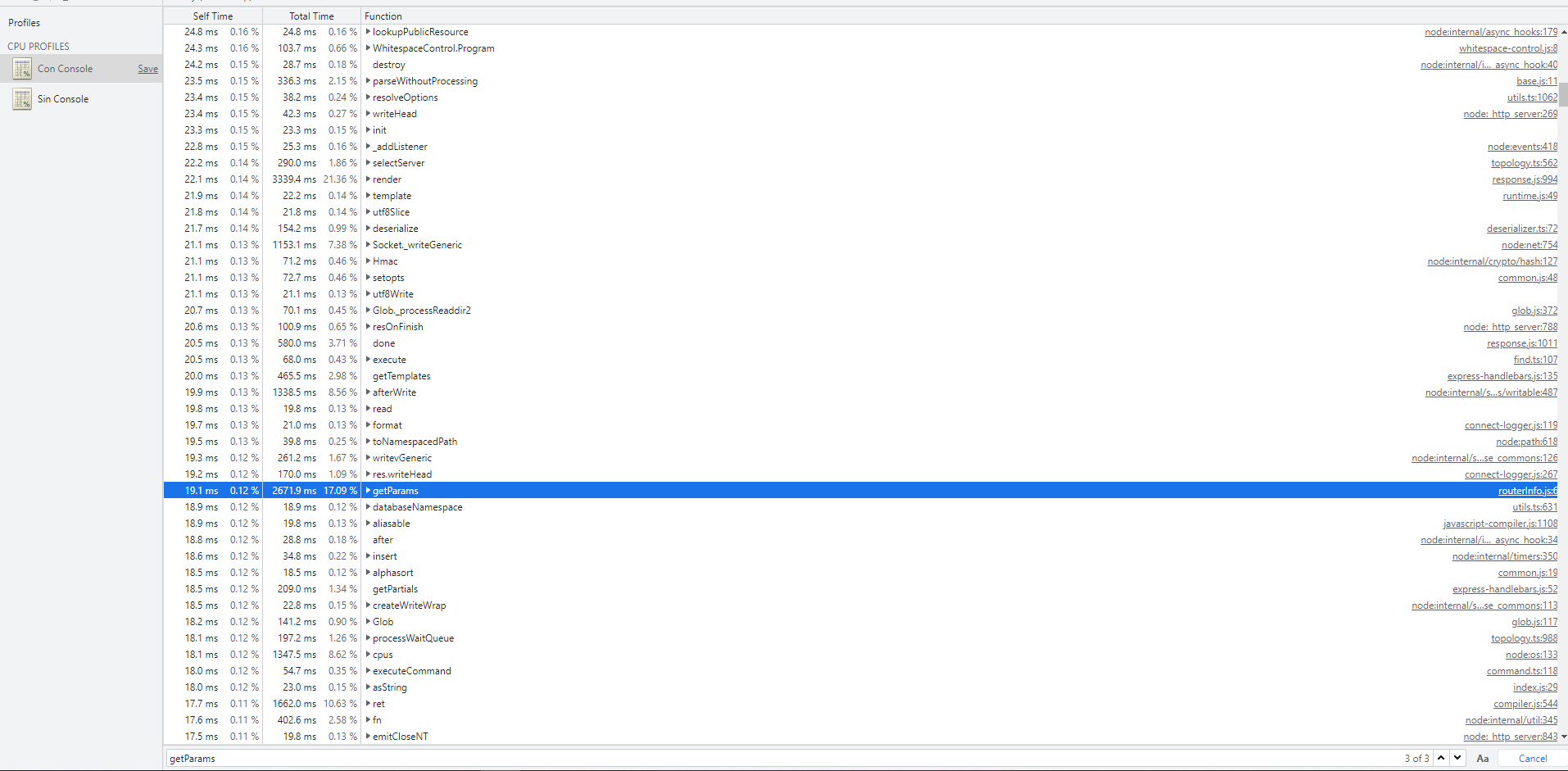


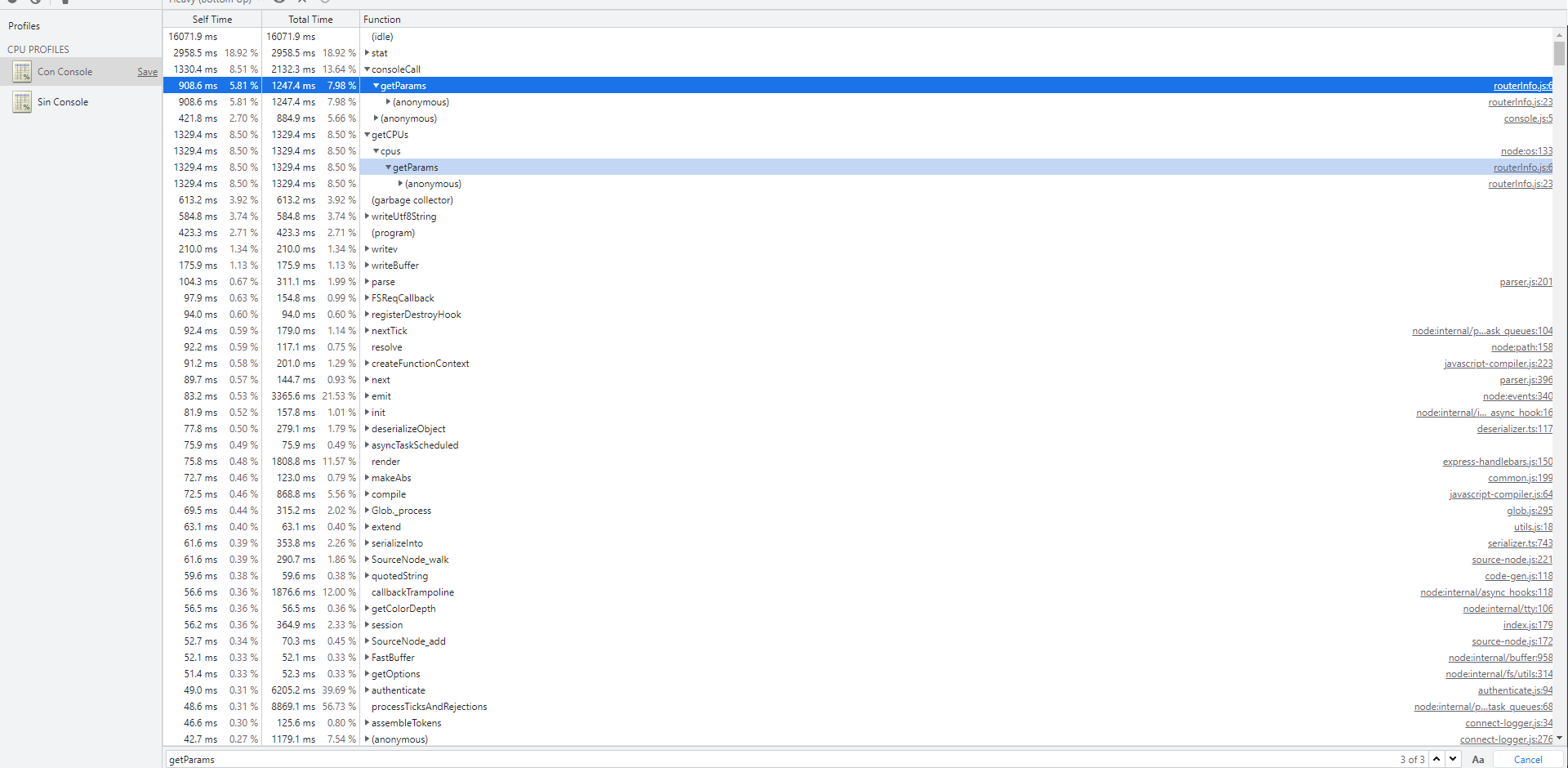
# Perfilamiento usando –inspect y Artillery

Los archivos con el detalle se encuentran en la ruta: /reportes/inspect/artillery

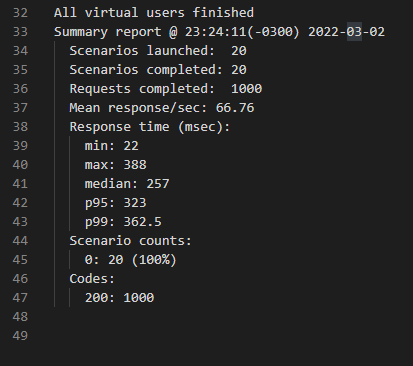
* Resultados Con console.log:

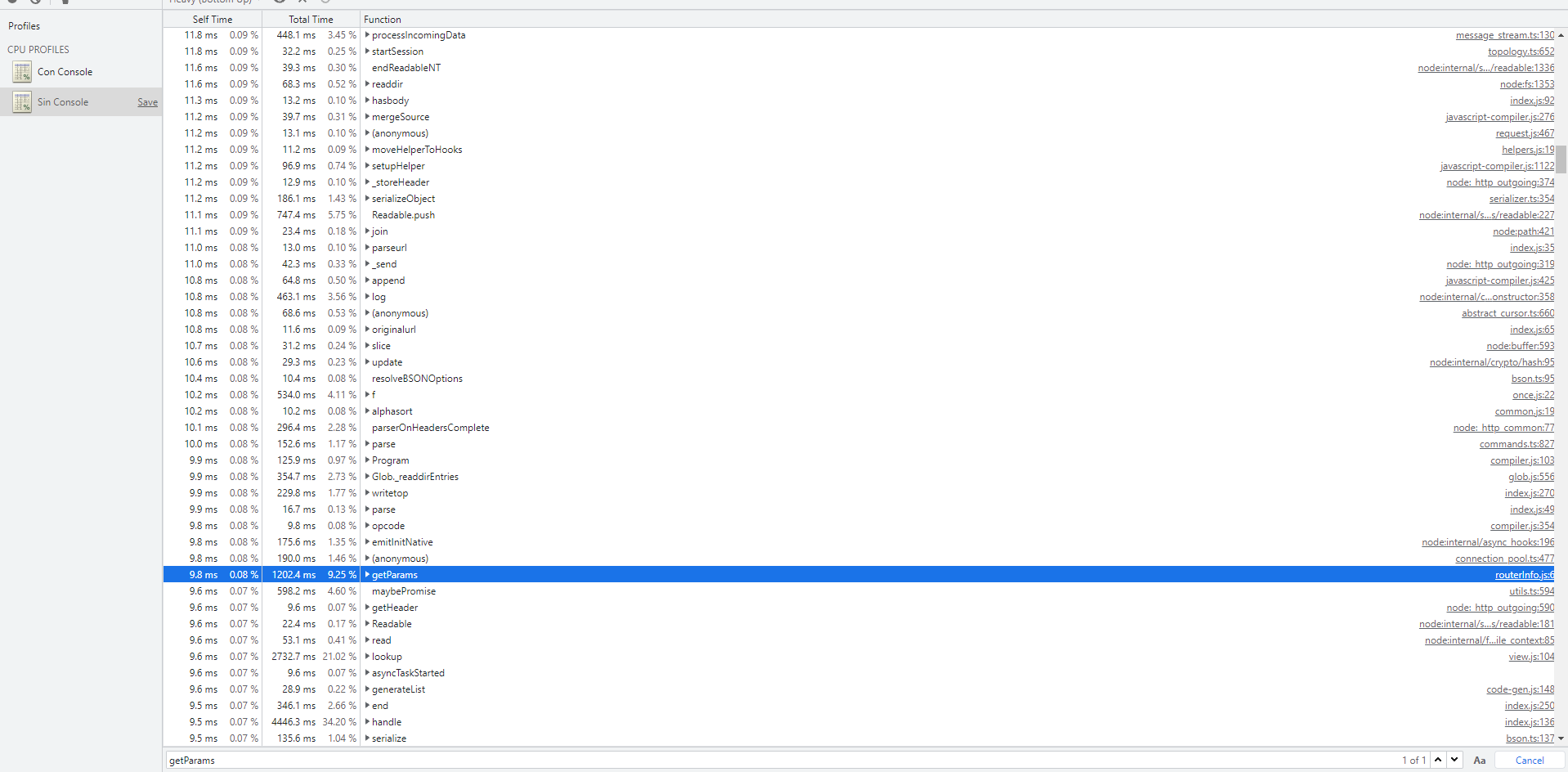






* Resultados Sin console.log:

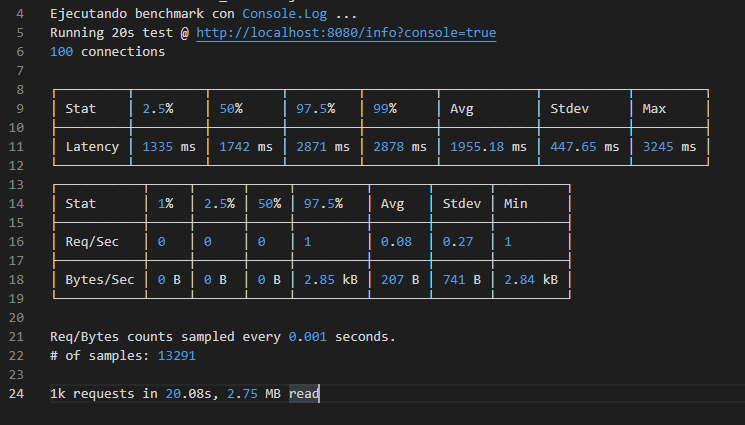


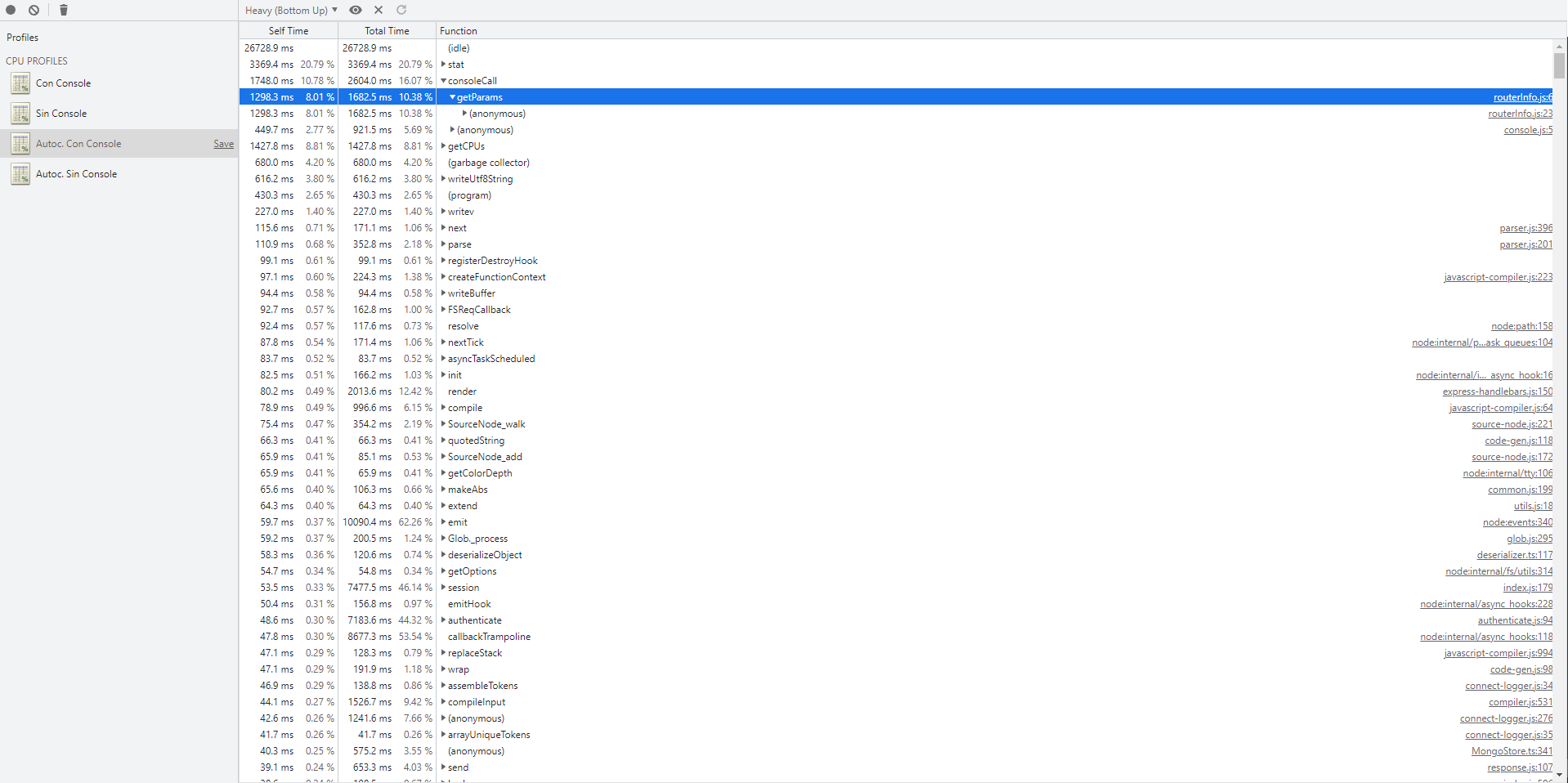


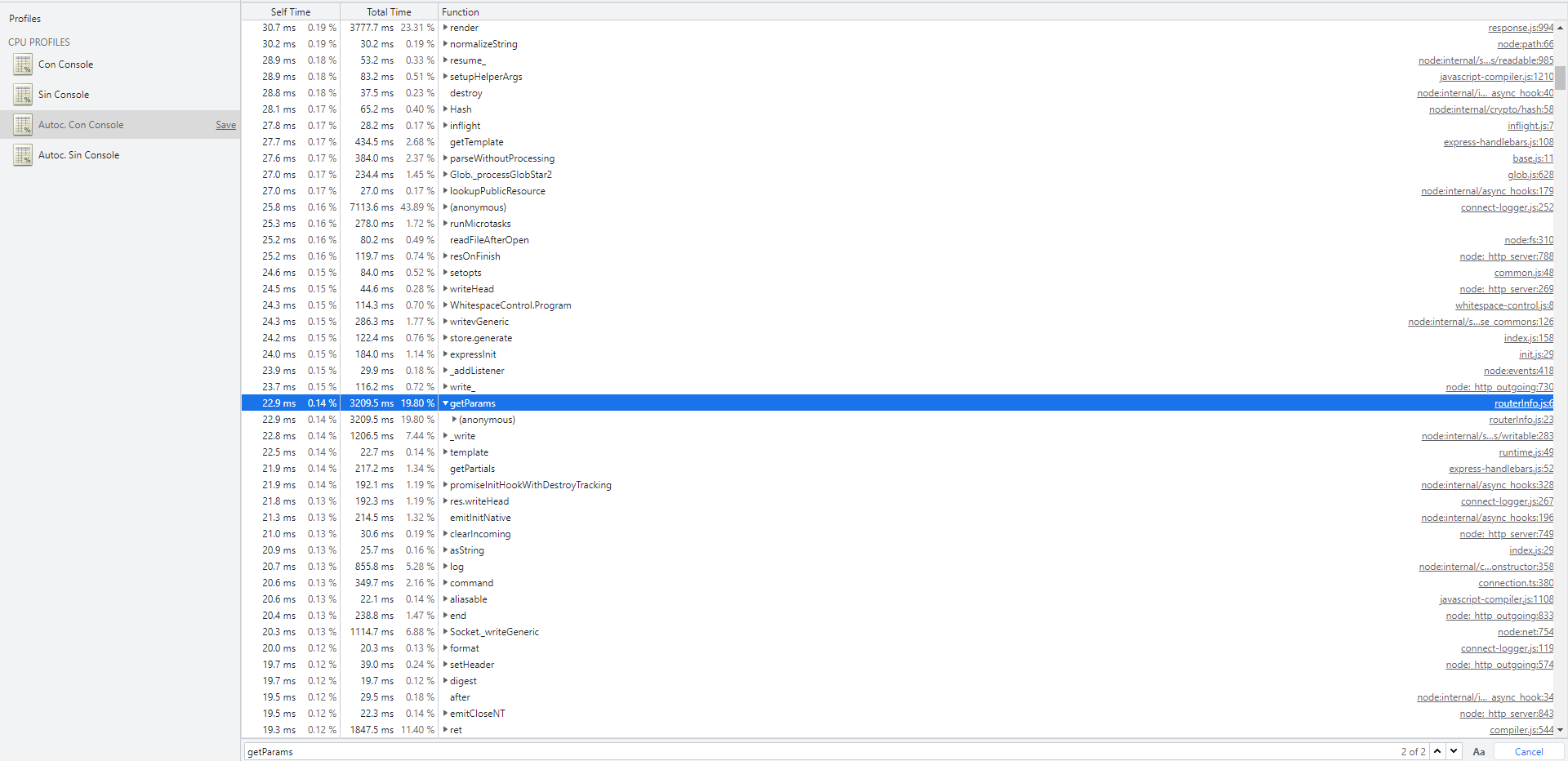
# Perfilamiento usando –inspect y autocannon

Los archivos con el detalle se encuentran en la ruta: /reportes/inspect/autocannon

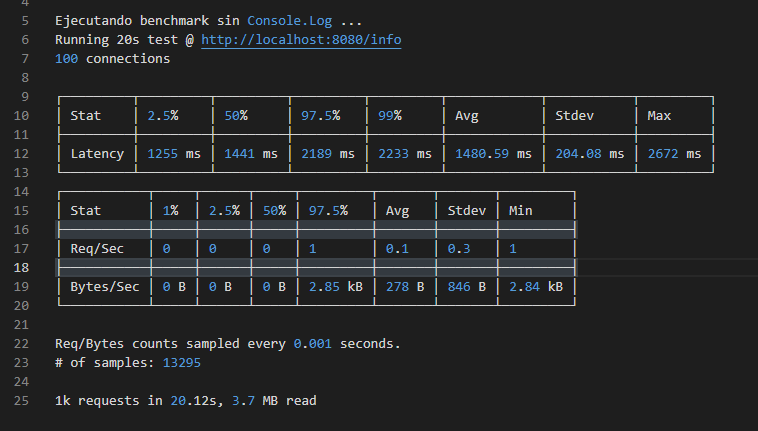
* Resultados Con console.log:

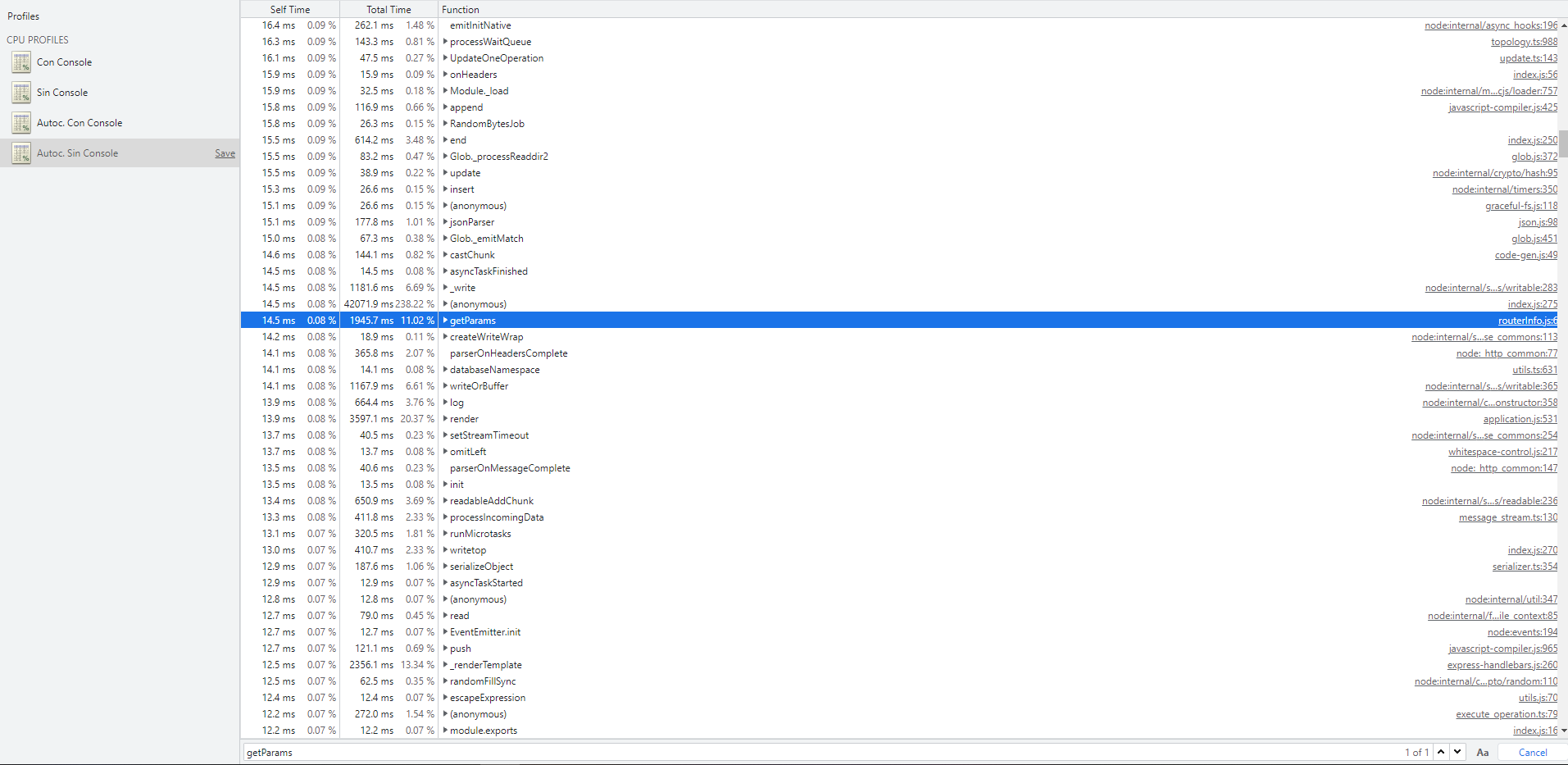






* Resultados Sin console.log:

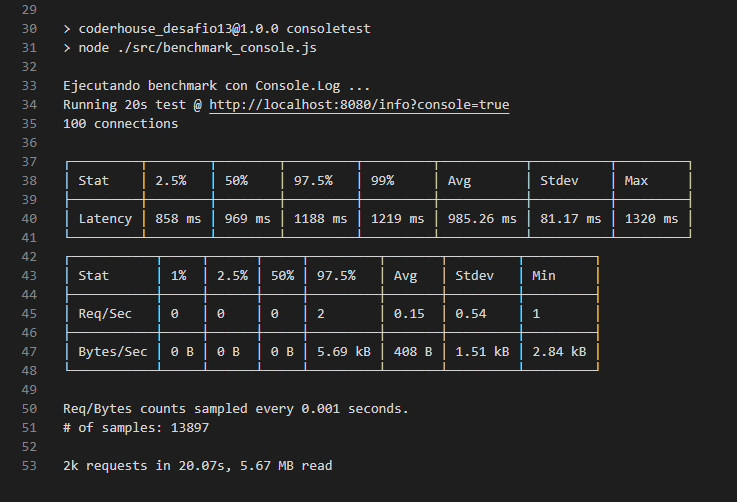




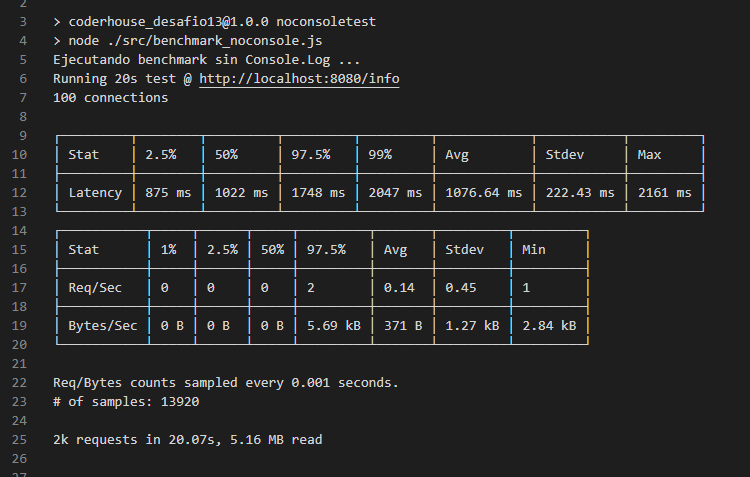
# Perfilamiento usando 0x y autocannon

Los archivos con el detalle se encuentran en la ruta: /reportes/0x

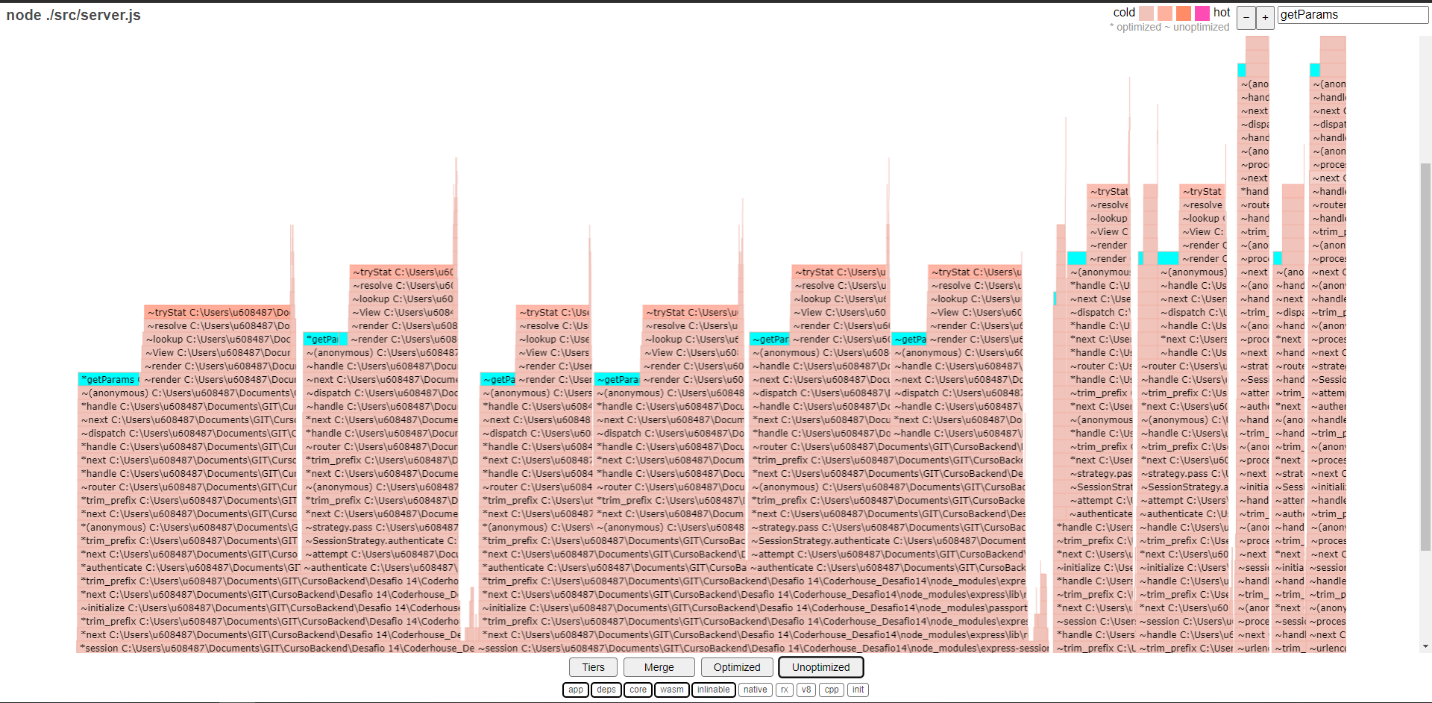
* Resultados Con console.log:



* Resultados Sin console.log:



* Gráfico de Flama: Se ejecutaron ambos test en el siguiente orden
  1. Ejecución sin console.log
  2. Ejecución con console.log



# CONCLUSIONES finales

En las pruebas realizadas con Artillery, en todos los casos se nota una mejora en el tiempo medio de respuesta del caso sin console.log; sin embargo en las pruebas realizadas con Autocannon no se nota una mejora al utilizar console.log (inclusive, en algunos casos la respuesta es mas rápida utilizando console.log).

En cuanto al perfilamiento, en todos los casos se puede ver que la llamada a la función getParams tiene más ticks cuando se utiliza el console.log cuando utiliza la librería “ntdll” del sistema.