Gzip

Se implementó en la ruta indicada y paso de 140kb a 8.1bkb

Vamos a trabajar sobre la ruta '/info', en modo fork, agregando ó extrayendo un console.log de la información colectada antes de devolverla al cliente. Además desactivaremos el child_process de la ruta '/randoms'

Para ambas condiciones (con o sin console.log) en la ruta '/info' OBTENER:

1) El perfilamiento del servidor, realizando el test con --prof de node.js. Analizar los resultados obtenidos luego de procesarlos con --prof-process.

Utilizaremos como test de carga Artillery en línea de comandos, emulando 50 conexiones concurrentes con 20 request por cada una. Extraer un reporte con los resultados en archivo de texto.

- Archivo solo Artillery Fork
- Archivo solo Artillery Cluster
- Archivo solo Profiling + Artillery Cluster

Luego utilizaremos Autocannon en línea de comandos, emulando 100 conexiones concurrentes realizadas en un tiempo de 20 segundos. Extraer un reporte con los resultados

2) El perfilamiento del servidor con el modo inspector de node.js --inspect. Revisar el tiempo de los procesos menos performantes sobre el archivo fuente de inspección.

- Archivo node inspector

3) El diagrama de flama con 0x, emulando la carga con Autocannon con los mismos parámetros anteriores.

- Archivo Autocannon
- Archvio 0x Flama

Concluciones: en los datos revisados seguimos confirmando como los servidores clusters son más rápidos al realizar los request desde el usuario con una optimización notable

Así mismo con las demás pruebas y más con el diagrama de flama pudimos notar nuestros procesos “óptimos” basados en el color de la grafica

by: Simon Daniel Meraz Sariñana