

ANÁLISIS DE PERFORMANCE DE SERVIDOR

INSTRUCCIONES

Sobre la ruta '/info', en modo fork, se agrega y extrae un console.log de la información colectada antes de devolverla al cliente. El child_process de la ruta '/randoms' estará desactivado.

Obtener:

- 1) El perfilamiento del servidor, realizando el test con --prof de node.js. Análisis de resultados después de procesarlos con --prof-process.
- 2) Extraer reporte con los resultados de utilizar como test de carga Artillery en línea de comandos emulando 50 conexiones concurrentes con 20 request por cada una en archivo de texto.
- 3) Emulando 100 conexiones concurrentes en 20 segundos con Autocannon en línea de comandos, extraer un reporte con los resultados.
- 4) 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.
- 5) El diagrama de flama con Ox, emulando la carga con Autocannon con los mismos parámetros anteriores.

Incluir en este informe sobre incluyendo los resultados de todas las pruebas realizadas y la conclusión obtenida a partir del análisis de los datos.

El perfilamiento del servidor, realizando el test con `--prof` de `node.js`. Análisis de resultados después de procesarlos con `--prof-process`.

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

Emulando 100 conexiones concurrentes en 20 segundos con Autocannon en línea de comandos, extraer un reporte con los resultados.

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.

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

CONCLUSIÓN