QA Rendimiento

Para mejorar el rendimiento de la aplicación se implementó compresión de assets por medio del módulo de node llamado "compression", el cual comprime los cuerpos de respuesta para todas las peticiones. También se implementó cache a través del módulo "node-cache" el cual crea una cache interna, es decir en el servidor que levanta la aplicación, pero actúa como memcached que es directamente la cache en memoria ram.

La cache se implementó de la siguiente manera:

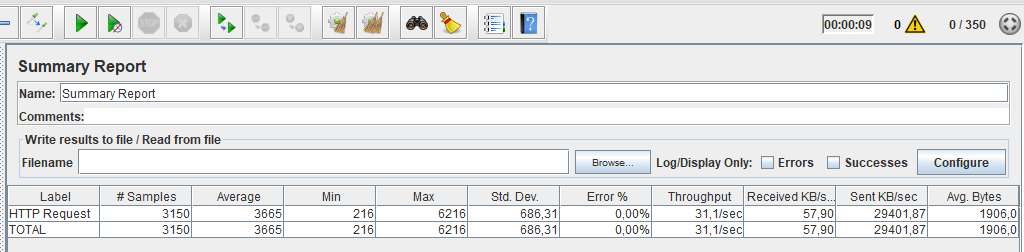
Cuando un usuario consulta sus imágenes, se crea una cache con las imágenes que hay almacenadas para ese usuario en la base de datos. A partir de ese momento, la caché cuenta con un tiempo de vida de 1 minuto. Si el usuario sube más imágenes y la consulta antes de que termine el minuto, no las va a ver; luego de que pase el minuto, cuando el usuario vuelva a consultar sus imágenes, se vuelve a crear la caché con todas las imágenes subidas hasta el momento y puede visualizar las nuevas imágenes. Por otro lado, si el usuario elimina una imagen, la cache se actualiza inmediatamente con las imágenes que se encuentran almacenadas para ese usuario en la base de datos.

Se realizan pruebas por medio de JMeter para 2 escenarios en 3 momentos diferentes:

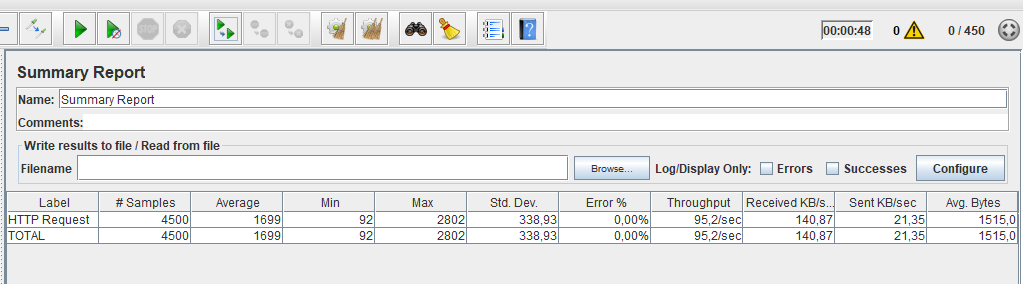
\*Nota: Se considera una imagen mediana, aquella que pesa de 300kb a 1mb. Adicionalmente se tiene en cuenta que en nuestra aplicación utilizamos el módulo “multer” para subir las imágenes, el cuál soporta un archivo de hasta 1mb.

* Sin compresión de assets ni cache

Múltiples usuarios concurrentes subiendo una imagen “mediana”. Aquí se observa 350 usuarios subiendo una imagen de 945kb durante 9 veces. Se obtiene un Throughput de 31,1/seg.

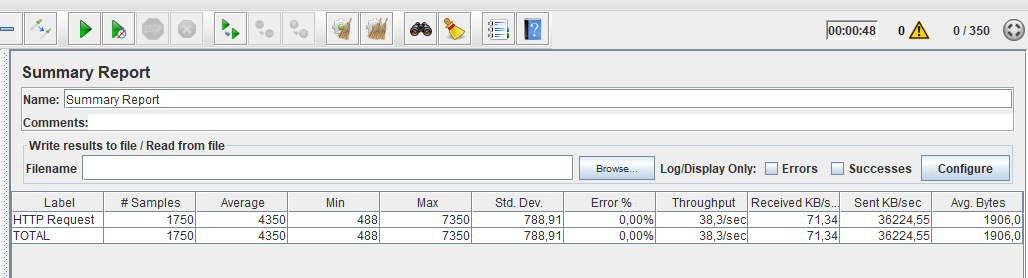


Múltiples usuarios concurrentes buscando múltiples imágenes de tamaño mediano. Se observan 450 usuarios buscando varias imágenes medianas durante 10 veces, donde se observa un Throughput de 95,2/seg y un tiempo promedio de 1699 ms.

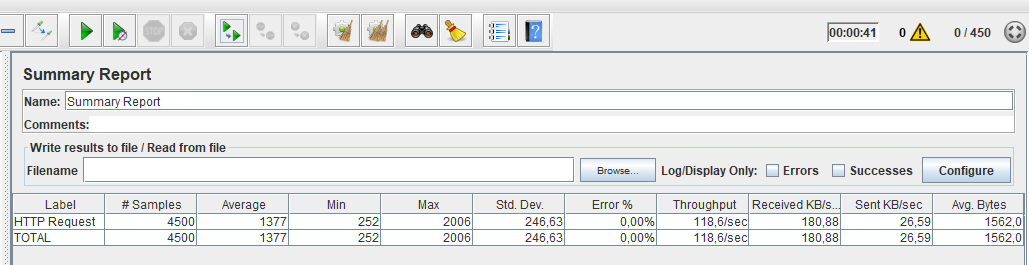


* Con compresión de assets

Para los mismos 350 usuarios subiendo una imagen de 945kb se observa una leve mejora en el Throughput, es cual paso de 31,1/seg a 38,3/seg

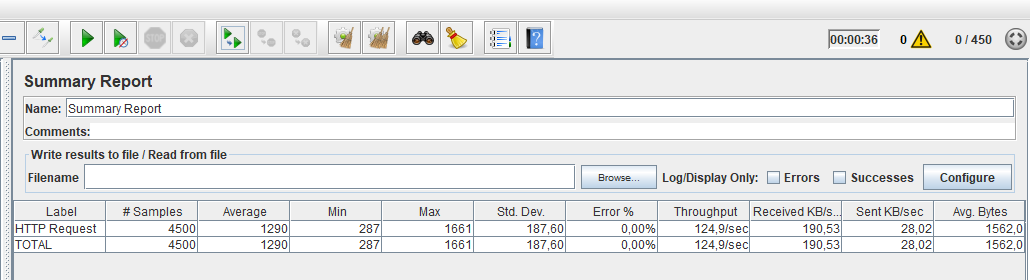


Para los mismos 450 usuarios buscando una imagen se observa una leve mejora en el tiempo promedio, el cual paso de 1699 a 1377. Igualmente en el throughput pasa de 95,2/seg a 118,6/seg.



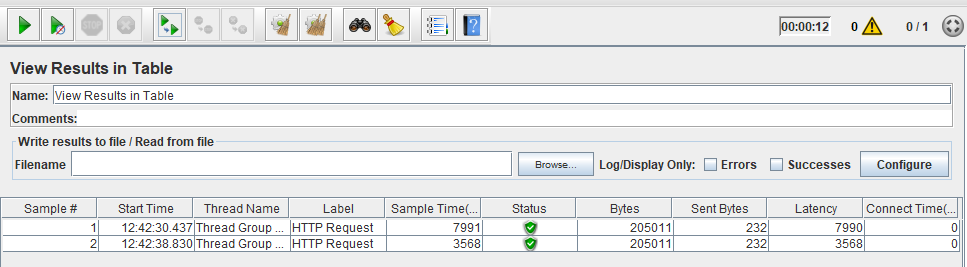
* Con cache

Se puede observar una mejora en el throughput, el cual aumenta a 124,9/seg. Esto es buscando imagen.



Finalmente, se puede observar claramente como el tiempo de respuesta mejora cuando se realiza una búsqueda por medio de cache.

Ej1: La primera vez que un usuario busca sus imágenes obtiene un tiempo de respuesta de 7990 ms; en esta primera búsqueda no está utilizando cache. Una vez realizada la primera búsqueda, se crea la cache, la cual se utiliza cuando el usuario vuelve a realizar una consulta de sus imágenes, en la cual se obtiene un tiempo de respuesta de 3568 ms.



Ej2: En este ejemplo podemos observar igualmente una mejora en el tiempo de respuesta, donde la primera vez que el usuario consulta sus imágenes, obtiene un tiempo de respuesta de 509 ms; a partir de este momento se crea la cache y en las próximas consultas se obtiene una mejora notable debido a que ya se está haciendo uso de la cache.

