

Entrega 3 – Pruebas de carga de stress

Paul A. Calvache Tapia, Fabian O. Ramírez, David A. Vásquez Pachón, Mateo Zapata Lopez

Desarrollo de Soluciones Cloud

Universidad de los Andes, Bogotá, Colombia

{pa.calvache, fo.ramirez50, da.vasquez11, m.zapatal2}@uniandes.edu.co

Fecha de presentación: abril 2 de 2023

Tabla de contenido

1	Objetivo general:	1
2	Objetivos específicos:	1
3	Descripción general:	1
4	Tipos de pruebas a realizar:	2
5	Datos de prueba:	2
6	Métricas:	3
7	Riesgos:	3
8	Resultado escenario 1:	3
9	Resultado escenario 2:	8

1 Objetivo general:

Realizar las pruebas de carga y stress de los escenarios de rutas críticas de la aplicación, para las cuales los servicios cobran valor para los usuarios.

- Creación de 100 tareas en un periodo de 10 segundos, cada tarea se crea en la misma carpeta para el usuario conectado.
- Compresión de 100 archivos en 10 segundos.

2 Objetivos específicos:

Escenario 1 y 2:

- Simular un incremento constante de creación de 100 tareas en 10 segundos, para visualizar el performance del sistema y establecer límites de la aplicación.
- En las pruebas de stress debe llegar al punto en que el tiempo para completar la compresión de un archivo no supere un tiempo de 5 minutos.
- Otro criterio de aceptación se define como, las requisiciones incompletas (errores) no pasen el umbral del 1% durante la prueba.

3 Descripción general:

Realizamos un servicio de compresión de archivos en línea para usuarios que se registren en nuestra página web, este es un servicio que se procesa en batch de manera asíncrona.

En este servicio, los usuarios se registran, después ingresan a la aplicación con usuario y contraseña, esto los direcciona a una página donde pueden visualizar sus tareas, crear tareas, borrar tareas y bajar los documentos que ya han sido comprimidos por la aplicación.

Como parte de los requerimientos de un buen servicio de compresión, diseñamos una ruta crítica las pruebas de los escenarios 1 y 2. Estas se prueban con 100 tareas de un mismo usuario para medir el desempeño de la aplicación.

- Para estas pruebas se utilizará JMeter para simular los tráfico y Compute Engine de Google Cloud para medir el rendimiento de la CPU.
- Los criterios de aceptación son 3:
 - Que el tiempo de respuesta (latencia) no sea superior a 1.500 ms
 - Que no se generen errores por encima del 1%
 - Que el tiempo para iniciar la compresión supere 5 minutos.

4 Tipos de pruebas a realizar:

Los siguientes son los hiper parámetros utilizados en las pruebas de JMeter para los diferentes escenarios.

Escenario 1:

- Pruebas de capacidad: Determinar la capacidad de la aplicación para la creación de tareas, en concreto 100 tareas en 10 segundos.
- Los hiper parámetros de esta prueba son los siguientes:
 - Número de hilos 100
 - Período de aceleración 10
- Pruebas de stress: definir la rata máxima de tareas que se pueden crear en la aplicación en 1 segundo.

Escenario 2:

- Pruebas de capacidad: Determinar el tiempo de compresión de un archivo de 21 kilobytes, 100 veces en 10 segundos.
- Los hiper parámetros de esta prueba son los siguientes:
 - Número de hilos 100
 - Período de aceleración 10
- Pruebas de stress: para definir la carga máxima que puede llegar a comprimir la aplicación, mediremos la cantidad máxima de archivos comprimidos en 1 segundo.

5 Datos de prueba:

Escenario 1:

- Se requiere la generación de nombres de las tareas aleatorias, pues cada tarea debe tener un nombre único.
- Para la función de creación de tareas, se debió configurar en el JMeter el Bearer token.

Escenario 2:

- Se tuvo que crear los 100 archivos con diferentes nombres y subirlos a la instancia NFS, por recomendación del profesor, este proceso se hizo manualmente.
- Para la prueba todos los archivos se comprimieron en el zip.

6 Métricas:

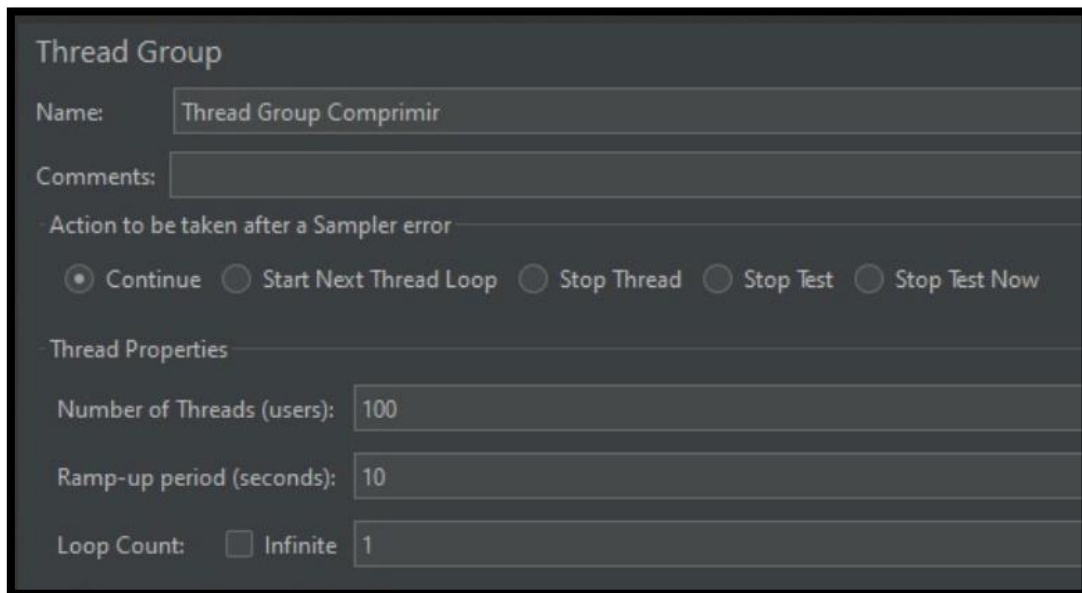
Las métricas que son relevantes para cumplir con los objetivos son:

- Capacidad de procesamiento (Throughput): medir la cantidad de trabajo realizado en un periodo definido de tiempo.
- Capacidad de utilización (Utilization): medir el porcentaje que representa qué tan ocupado está un recurso (CPU, Mem, I/O, etc) en un momento específico.
- Número máximo de usuarios hasta alcanzar tiempos de respuesta de máximo 1.500ms.
- Número de hilos: medir la cantidad máxima de usuarios concurrentes.
- Período de aceleración: 60
- Tiempos mínimos de respuesta de las transacciones: medir el tiempo mínimo que tarda en dar respuesta a las transacciones

7 Riesgos:

- La variabilidad de la latencia del cliente, así como las conexiones de red, pueden generar resultados diferentes en las pruebas.
- Para el ahorro de los créditos estudiantiles se optó por una configuración de red, VM y Cloud SQL más sencillas, esto afecta las pruebas, pues no se realizan escenarios que degraden totalmente la aplicación.

8 Resultado Escenario 1:



The image shows a screenshot of the 'Thread Group' configuration window in Apache JMeter. The window has a dark theme. At the top, the title is 'Thread Group'. Below the title, there are several fields and options:

- Name:** A text field containing 'Thread Group Comprimir'.
- Comments:** An empty text area.
- Action to be taken after a Sampler error:** A section with five radio button options: 'Continue' (selected), 'Start Next Thread Loop', 'Stop Thread', 'Stop Test', and 'Stop Test Now'.
- Thread Properties:** A section with three fields:
 - Number of Threads (users):** A text field containing '100'.
 - Ramp-up period (seconds):** A text field containing '10'.
 - Loop Count:** A section with a checkbox labeled 'Infinite' (which is unchecked) and a text field containing '1'.

Imagen 1. Configuración escenario 2 comprimir.

Web Server

Protocol [http]: Server Name or IP: Port Number:

HTTP Request

POST

☐ Redirect Automatically ☒ Follow Redirects ☒ Use KeepAlive ☐ Use multipart/form-data ☐ Browser-compatible headers

Parameters Body Data Files Upload

[illegible]

View Results in Table

Name:View Results in Table

Comments:

Write results to file / Read from file

Filename

Browse...

Log/Display Only:

☐ Errors

☐ Successes

Configure

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
67	22:51:34.543	Thread Group Com...	HTTP Request Com...	30180		220	485	30180	66
68	22:51:34.643	Thread Group Com...	HTTP Request Com...	30132		220	485	30132	59
69	22:51:34.743	Thread Group Com...	HTTP Request Com...	30149		220	485	30149	60
70	22:51:34.843	Thread Group Com...	HTTP Request Com...	30155		220	485	30155	59
71	22:51:34.943	Thread Group Com...	HTTP Request Com...	30182		220	485	30181	97
72	22:51:35.042	Thread Group Com...	HTTP Request Com...	30153		220	485	30153	60
73	22:51:35.143	Thread Group Com...	HTTP Request Com...	30136		220	485	30136	66
74	22:51:35.243	Thread Group Com...	HTTP Request Com...	30147		220	485	30147	70
75	22:51:35.344	Thread Group Com...	HTTP Request Com...	30170		220	485	30170	64
76	22:51:35.444	Thread Group Com...	HTTP Request Com...	30141		220	485	30141	60
77	22:51:26.444	Thread Group Com...	HTTP Request Com...	39308		203	485	39307	58
78	22:51:35.743	Thread Group Com...	HTTP Request Com...	30127		220	485	30127	59
79	22:51:35.844	Thread Group Com...	HTTP Request Com...	30163		220	485	30163	58
80	22:51:26.343	Thread Group Com...	HTTP Request Com...	39736		203	485	39736	66
81	22:51:35.944	Thread Group Com...	HTTP Request Com...	30146		220	485	30146	70
82	22:51:36.043	Thread Group Com...	HTTP Request Com...	30148		220	485	30148	60
83	22:51:36.143	Thread Group Com...	HTTP Request Com...	30139		220	485	30139	60
84	22:51:26.545	Thread Group Com...	HTTP Request Com...	39825		203	485	39825	62
85	22:51:35.644	Thread Group Com...	HTTP Request Com...	31314		203	485	31314	62
86	22:51:26.744	Thread Group Com...	HTTP Request Com...	40890		203	485	40889	61
87	22:51:33.342	Thread Group Com...	HTTP Request Com...	34513		203	485	34513	15079
88	22:51:36.244	Thread Group Com...	HTTP Request Com...	31726		203	485	31726	64
89	22:51:35.243	Thread Group Com...	HTTP Request Com...	33059		203	485	33058	15093
90	22:51:26.844	Thread Group Com...	HTTP Request Com...	41572		203	485	41572	65
91	22:51:27.043	Thread Group Com...	HTTP Request Com...	41948		203	485	41948	62
92	22:51:27.343	Thread Group Com...	HTTP Request Com...	41962		203	485	41961	79
93	22:51:26.943	Thread Group Com...	HTTP Request Com...	42623		203	485	42623	67
94	22:51:27.443	Thread Group Com...	HTTP Request Com...	42246		203	485	42246	62
95	22:51:27.143	Thread Group Com...	HTTP Request Com...	42936		203	485	42936	62
96	22:51:27.243	Thread Group Com...	HTTP Request Com...	43011		203	485	43010	68
97	22:51:27.643	Thread Group Com...	HTTP Request Com...	42855		203	485	42855	63
98	22:51:27.543	Thread Group Com...	HTTP Request Com...	43112		203	485	43112	63
99	22:51:27.744	Thread Group Com...	HTTP Request Com...	43460		203	485	43460	61
100	22:51:26.644	Thread Group Com...	HTTP Request Com...	46460		203	485	46460	57

Imagen 4. Cantidad de peticiones en estado exitoso y fallido.

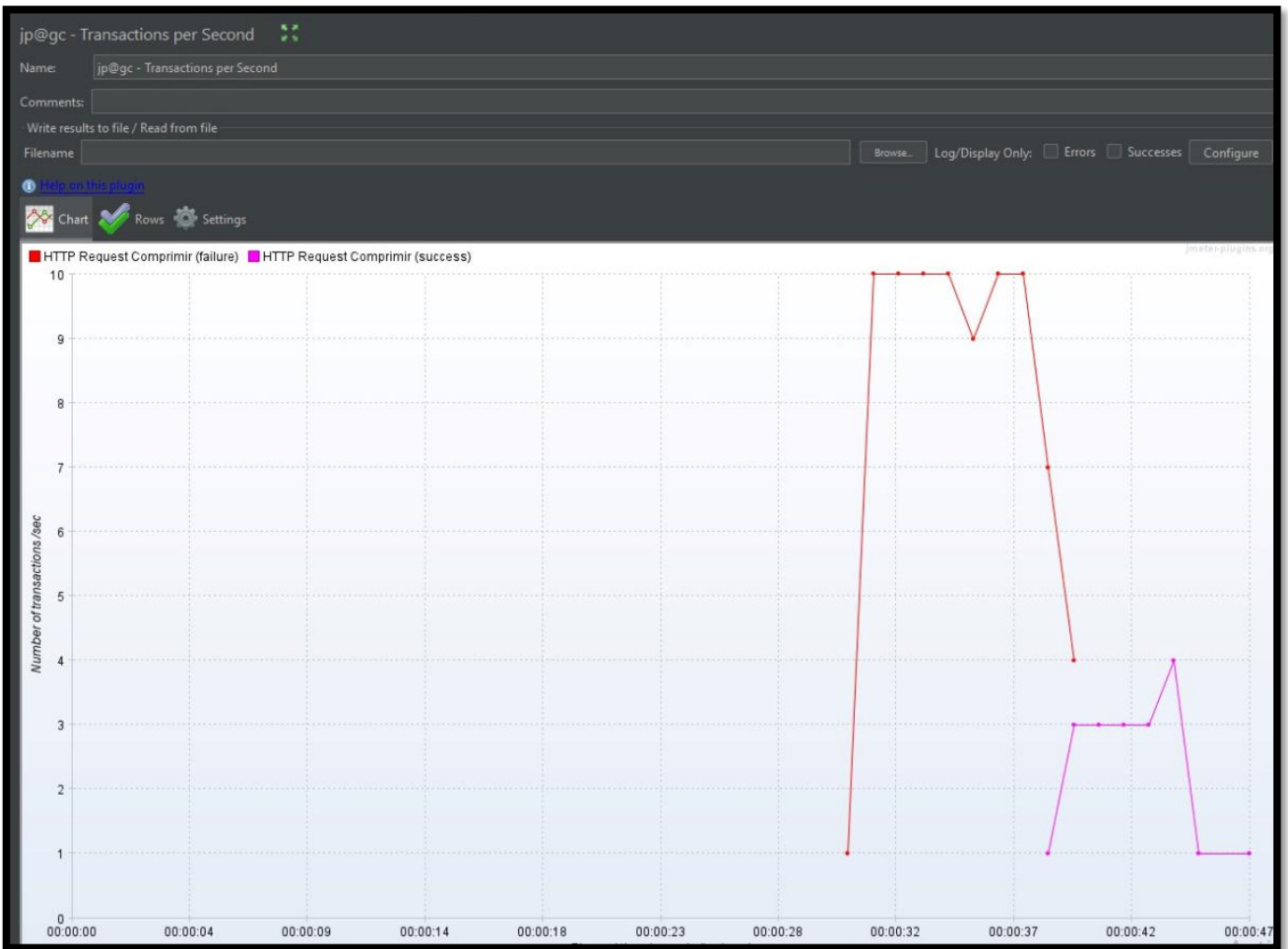


Imagen 5. Transacciones por segundo del escenario comprimir.

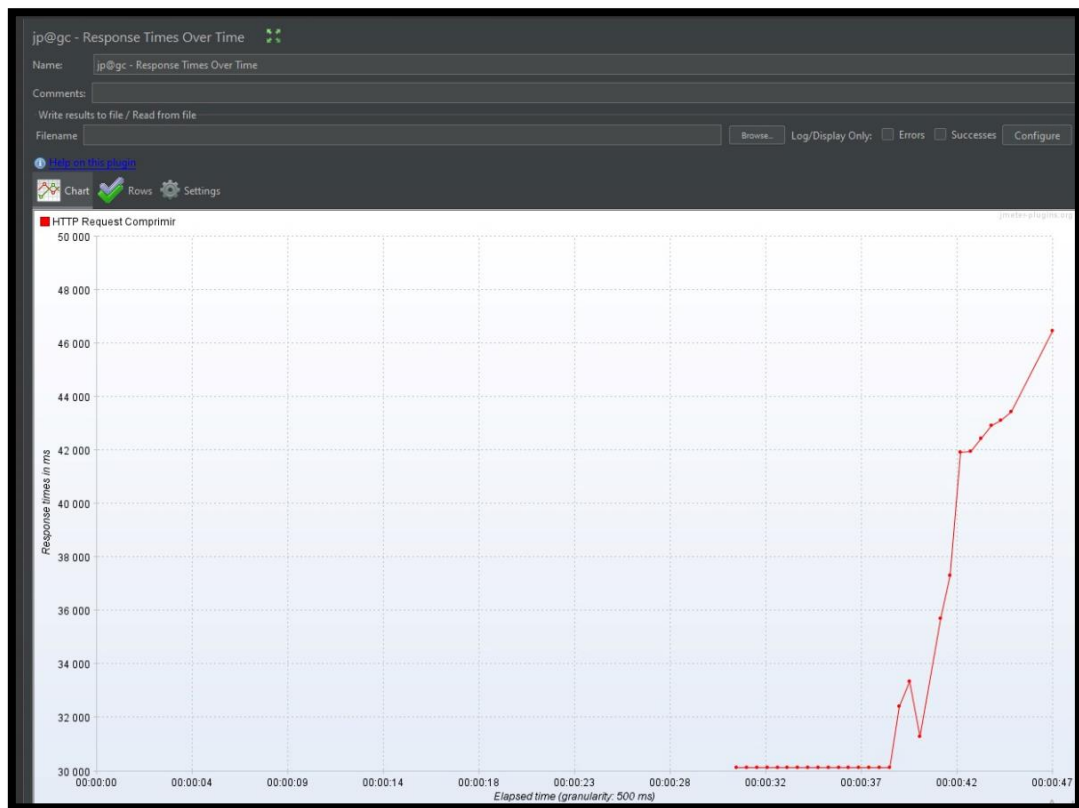


Imagen 6. Respuesta sobre el tiempo escenario comprimir.

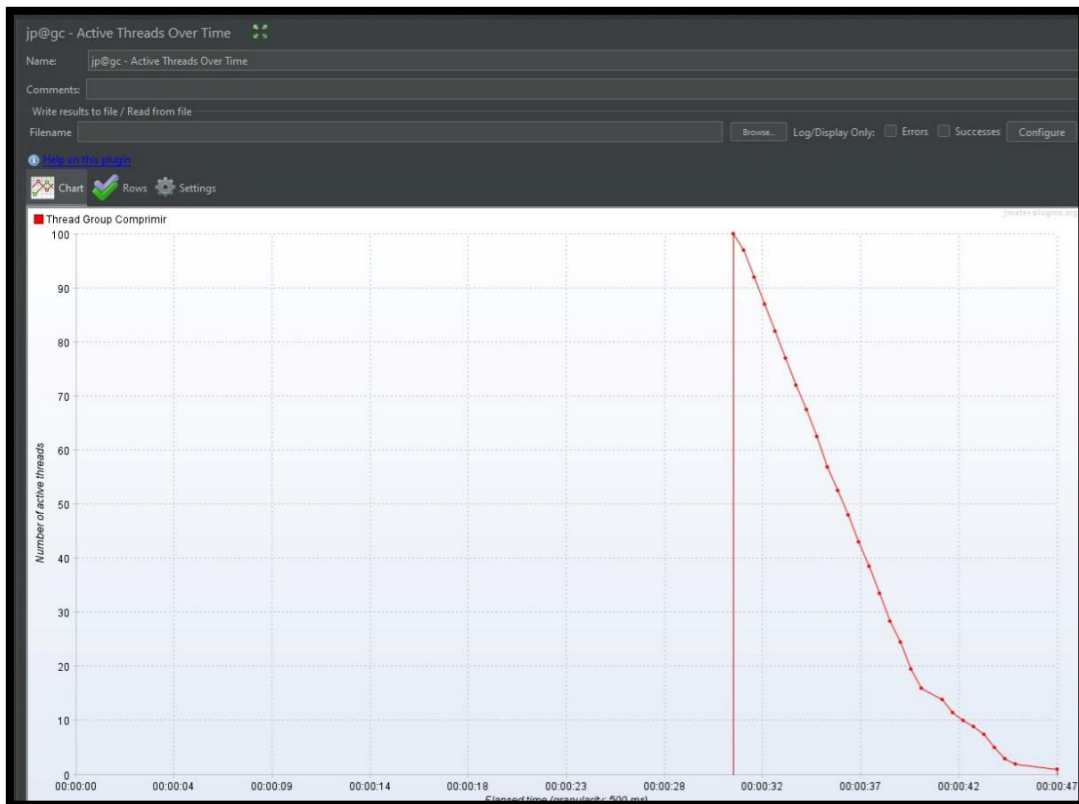


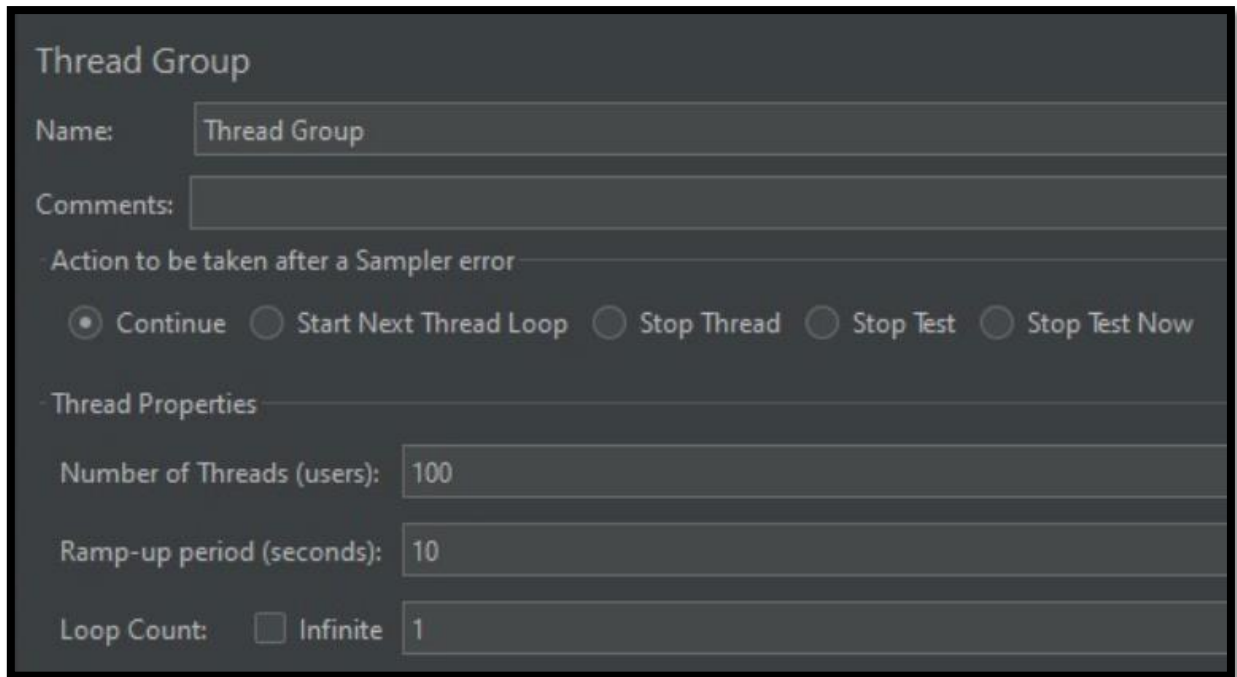
Imagen 7. Cantidad de peticiones sobre el tiempo del escenario comprimir.

9 Resultado Escenario 2:



The screenshot shows the 'Basic' tab of a JMeter configuration window. The 'Web Server' section has 'Protocol [http]:' set to 'http', 'Server Name or IP:' set to '34.139.21.102', and 'Port Number:' set to '5000'. The 'HTTP Request' section has a dropdown menu set to 'POST' and a 'Path:' field set to '/api/tasks'. Below this, there are checkboxes for 'Redirect Automatically' (unchecked), 'Follow Redirects' (unchecked), 'Use KeepAlive' (checked), 'Use multipart/form-data' (unchecked), and 'Browser-compatible headers' (unchecked). At the bottom, there are tabs for 'Parameters', 'Body Data', and 'Files Upload'.

Imagen 8. Endpoint creación de tareas.



The screenshot shows the 'Thread Group' configuration window. The 'Name:' field is set to 'Thread Group'. The 'Comments:' field is empty. Under 'Action to be taken after a Sampler error', the 'Continue' radio button is selected. Under 'Thread Properties', the 'Number of Threads (users):' is set to '100', the 'Ramp-up period (seconds):' is set to '10', and the 'Loop Count:' is set to '1' with the 'Infinite' checkbox unchecked.

Imagen 9. Configuración JMeter para escenario creación tareas.

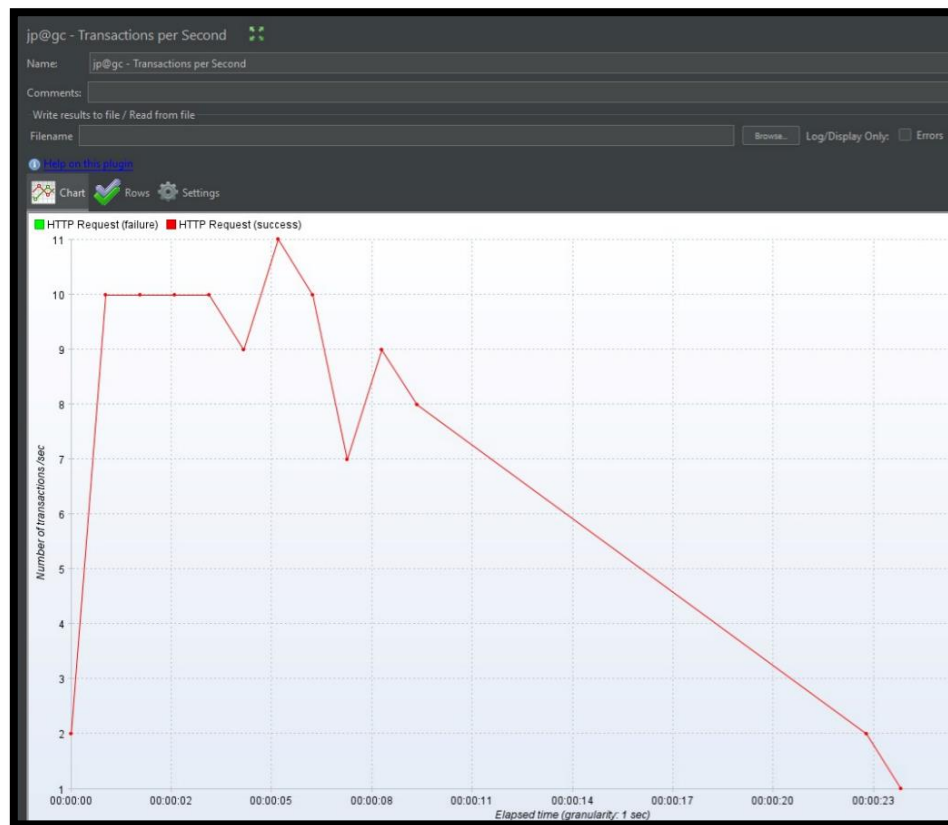


Imagen 10. Transacciones por segundo escenario creación tareas.

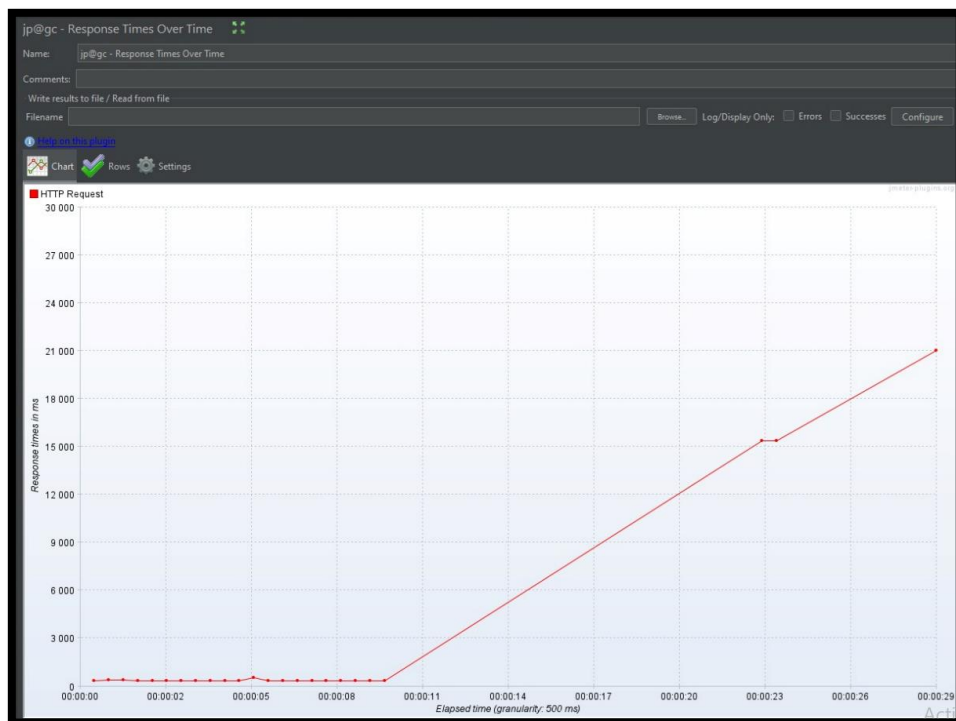


Imagen 11. Respuesta sobre el tiempo de las peticiones en el escenario de creación de tareas.

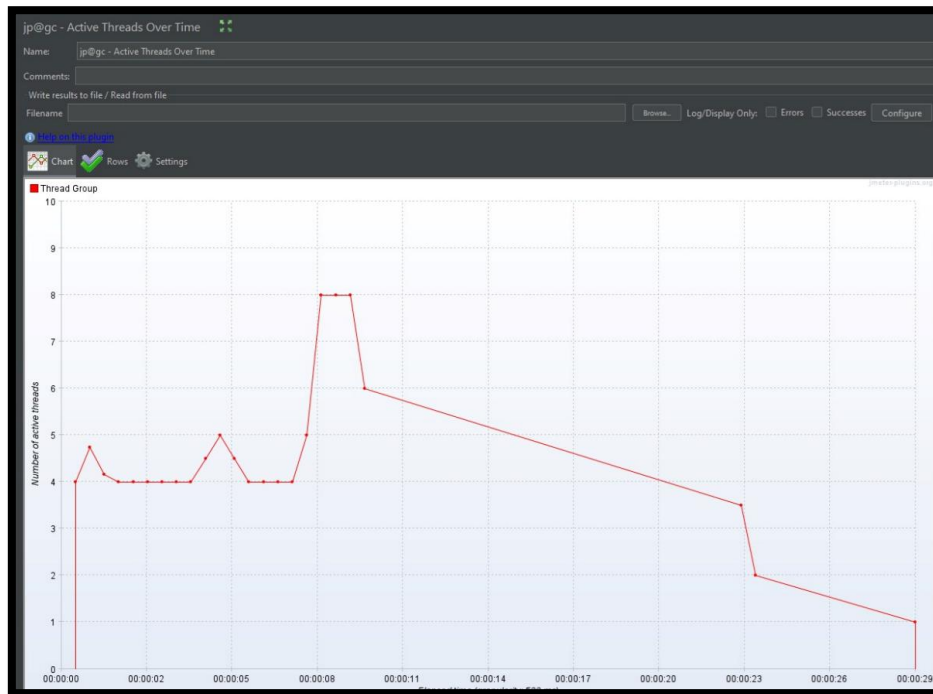


Imagen 12. Peticiones activas sobre el tiempo escenario de creación de tareas.

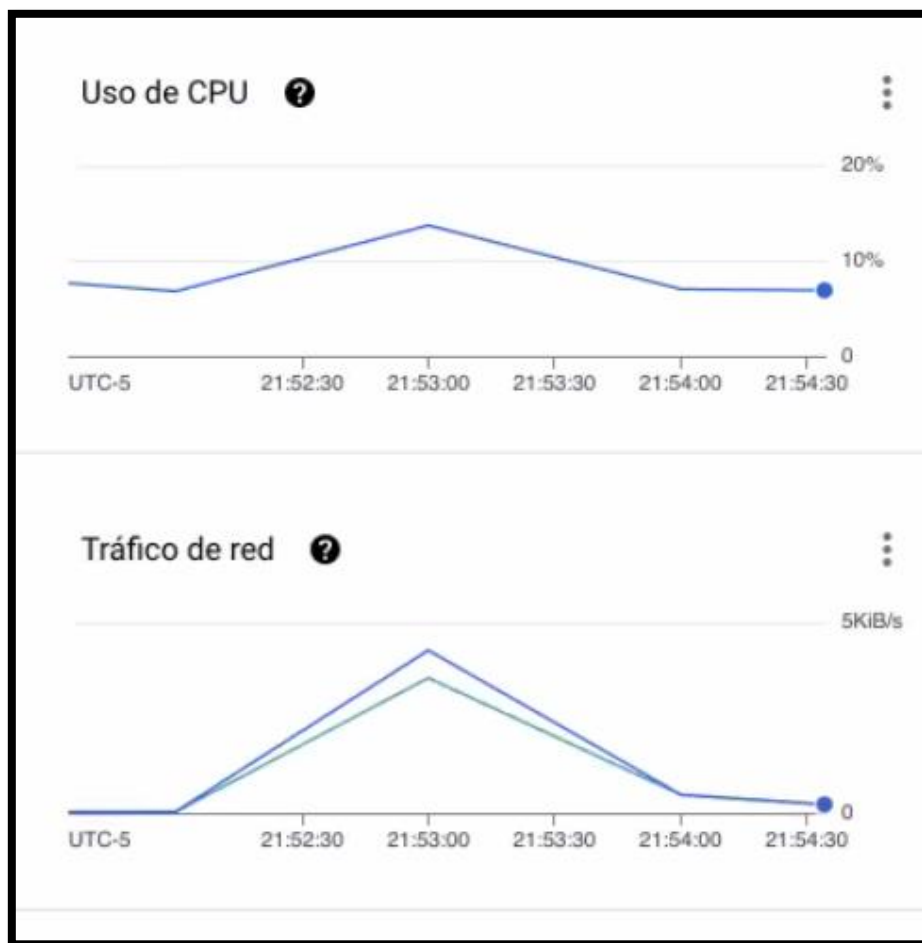


Imagen 13. Uso de CPU y tráfico de red para los servicios del Web Server y worker.



Imagen 14. Performance de los servidores Webserver y worker.

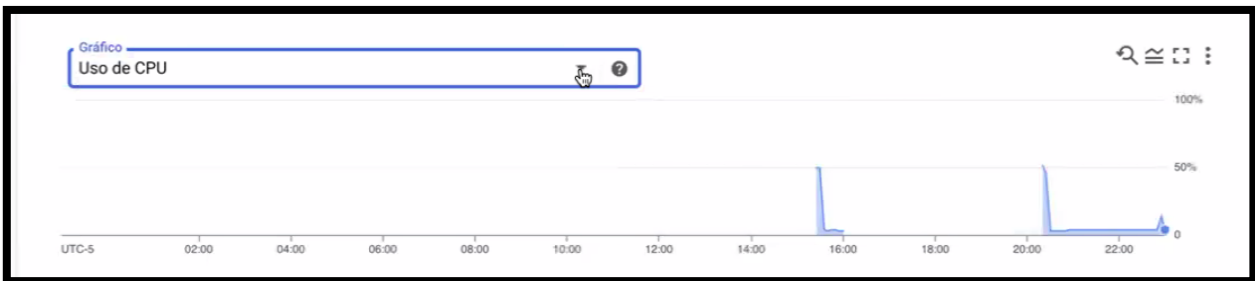


Imagen 15. Uso de CPU del Cloud SQL.

10 Conclusiones:

- Para los próximos escenarios, debemos aumentar los límites de creación de tareas, con el fin de encontrar el punto de quiebre de la aplicación. Esto debido a que el límite actual de 100 fue superado con normalidad en 100 % de efectividad.
- Las pruebas de estrés orientadas al sistema de colas de compresión, a pesar de que desarrollo un 100% de compresión, si presento errores en la ejecución de colas, esto se debe revisar a nivel de desarrollo para encontrar el flujo que recibe y enruta las colas de la aplicación.
- El cpu utilizado por nuestro web Server/worker llegó a un máximo de 50% en el pico de estrés de las pruebas, en consecuencia, de encontrar el punto de quiebre de la aplicación y el aumento del # de peticiones a realizarle a la aplicación web. Se puede pensar en asignar el mismo a esta instancia para mantener los níveles máximos de uso del servidor en un máximo del 90 %
- El servidor de base de datos llegó a un uso de CPU del 90%, esta instancia debe aumentar su recurso para las próximas pruebas debido a que se planea aumentar la carga de estrés con el objetivo de punto de quiebre, se debe aumentar para mantener su capacidad en un 90 %.

