**Análisis del Archivo textSummary.txt:**

Línea 1-3: ✓ App Transaction Balance response was OK y 97% - ✓ 269891 / x 6759

✓ App Transaction Balance response was OK: Indica que la transacción principal de la aplicación, llamada "Balance", respondió correctamente.

97% - ✓ 269891 / x 6759: Muestra la tasa de éxito de las transacciones.

97%: Es el porcentaje de transacciones exitosas.

✓ 269891: El número de transacciones que se completaron con éxito.

x 6759: El número de transacciones que fallaron.

Esto confirma que la gran mayoría de las transacciones fueron exitosas, pero hubo un pequeño número de fallos.

Línea 4: setup

Indica el inicio de la sección de configuración o los resultados iniciales de la prueba.

Línea 6: ✓ checks........................... : 97.55% ✓ 269891 x 6759

checks: Se refiere a las verificaciones o aserciones que se configuraron en la prueba. Estas verificaciones pueden ser para confirmar que la respuesta de una página contiene un texto específico o que el código de respuesta HTTP es 200 (OK).

97.55% ✓ 269891 x 6759: Al igual que en la línea 3, muestra que el 97.55% de las verificaciones fueron exitosas, con 269,891 verificaciones exitosas y 6,759 fallidas. Esto es consistente con la tasa de éxito de las transacciones.

Línea 7-8: data\_received y data\_sent

data\_received......... : 842 MB 223 kB/s: Muestra la cantidad total de datos recibidos por los usuarios virtuales durante la prueba (842 MB) y la tasa de recepción promedio (223 kB/s).

data\_sent............ : 588 MB 156 kB/s: Muestra la cantidad total de datos enviados por los usuarios virtuales (588 MB) y la tasa de envío promedio (156 kB/s).

Línea 9: failed\_request................... : 2.44% x 6759

failed\_request: Es la tasa de solicitudes (requests) que fallaron.

2.44% x 6759: Coincide con la cantidad de transacciones fallidas (6,759) y representa el porcentaje de fallas. Esto indica que un 2.44% de las solicitudes no se completaron con éxito.

Línea 10-11: Métricas de rendimiento de la red

http\_req\_blocked: Tiempo que una solicitud estuvo bloqueada antes de ser enviada. avg=10.97µs, med=0s, max=35.02ms, p(90)=0s, p(95)=0s. Los valores son bajos, lo que es bueno.

avg: Promedio.

med: Mediana.

max: Máximo.

p(90): Percentil 90. Esto significa que el 90% de las solicitudes tuvieron un tiempo de bloqueo menor o igual a 0 segundos.

p(95): Percentil 95. El 95% de las solicitudes tuvieron un tiempo de bloqueo menor o igual a 0 segundos.

http\_req\_connecting: Tiempo que tomó establecer la conexión TCP. avg=3.3µs, med=0s, max=11.82ms, p(90)=0s, p(95)=0s. Al igual que con el bloqueo, estos tiempos son muy bajos, lo cual es excelente.

En resumen, el textSummary.txt muestra una prueba de carga mayormente exitosa, con un 97% de transacciones completadas. Los tiempos de conexión y bloqueo son muy bajos, lo que sugiere una buena infraestructura de red. Sin embargo, el 2.44% de fallos es un punto a investigar, ya que podría indicar un cuello de botella o un problema en la aplicación que se manifiesta bajo carga.

**Análisis del Diagrama de Monitoreo**

El diagrama muestra la relación entre dos métricas a lo largo del tiempo durante la prueba de carga:

Línea verde (vus): Representa el número de Usuarios Virtuales (Virtual Users) que están activos en un momento dado.

Línea azul (http\_reqs): Representa la tasa de solicitudes HTTP por segundo (requests per second).

Inicio de la Prueba (01:40:00): La línea verde (vus) sube rápidamente hasta alcanzar un nivel constante de 140 VUs. Esto significa que la prueba escaló de 0 a 140 usuarios virtuales activos.

**Período de Carga Constante** (01:40:00 - 01:50:00): La línea verde se mantiene estable en 140 VUs. Durante este tiempo, la línea azul (http\_reqs) fluctúa, pero se mantiene en un rango alto, alrededor de 120-130 solicitudes por segundo, lo que indica que el sistema está respondiendo al tráfico.

Caída de la Carga (01:50:00 - 02:00:00): De repente, tanto la línea verde (vus) como la línea azul (http\_reqs) caen drásticamente.

La línea verde (vus) baja casi a cero. Esto podría ser un fallo en el script de la prueba de carga, un problema en la herramienta de monitoreo, o que los usuarios virtuales no pudieron continuar con sus transacciones y terminaron su ejecución.

La línea azul (http\_reqs) cae en consecuencia, ya que al no haber usuarios virtuales, no se generan solicitudes.

Recuperación (alrededor de 02:00:00): El número de VUs vuelve a subir, nuevamente hasta 140. La línea azul (http\_reqs) también se recupera, indicando que las solicitudes por segundo volvieron a un nivel similar al del inicio de la prueba.

Período de Carga Estable (02:00:00 en adelante): La prueba continúa con 140 VUs, y la tasa de solicitudes por segundo (http\_reqs) se mantiene estable, con fluctuaciones normales, alrededor de 82.6 solicitudes por segundo, como se muestra en la etiqueta a las 02:02:00.

**Conclusión del Diagrama:**

El punto más importante a destacar es la caída inesperada de la carga entre las 01:50:00 y las 02:00:00. Esto sugiere un problema significativo durante la prueba. Las posibles causas podrían ser:

**Error en la Aplicación Bajo Prueba (AUT):** Un error crítico en el sistema que causó que los usuarios virtuales no pudieran continuar, provocando que la prueba se detuviera momentáneamente. Esto podría estar relacionado con los 2.44% de fallos que se vieron en el textSummary.txt.

**Problema de Infraestructura**: El servidor de la aplicación, la base de datos o algún componente del entorno de prueba pudo haber colapsado y reiniciado.

La tasa de solicitudes por segundo (http\_reqs) de 82.6/s a las 02:02:00 es una métrica clave. Si esta es la tasa esperada, el rendimiento es bueno. Sin embargo, la caída del servicio es el hallazgo más crítico de este gráfico.

**Hallazgos:**

* **Comportamiento de los Usuarios Virtuales (VUs):** El número de VUs se mantuvo relativamente estable alrededor de 140 durante la mayor parte de la prueba. Sin embargo, se observa una caída significativa de los VUs en un período específico, aproximadamente entre 01:50:00 y 02:00:00, donde la actividad parece haber disminuido drásticamente antes de volver a estabilizarse.
* **Comportamiento de las Peticiones HTTP (http\_reqs):** Las peticiones por segundo (http\_reqs) siguen una pauta similar a la de los VUs. Se mantuvieron alrededor de 82.6 por segundo durante los períodos de carga normal y también experimentaron una caída pronunciada durante el mismo intervalo de tiempo que los VUs, lo que indica una correlación directa entre la cantidad de usuarios y las peticiones generadas.
* **Correlación entre VUs y Peticiones:** El diagrama muestra una relación directa entre el número de usuarios virtuales y la cantidad de peticiones por segundo. Cuando los VUs bajan, las peticiones también bajan, y viceversa. Esto sugiere que el sistema está respondiendo a la carga de manera consistente, al menos en términos de generación de peticiones.

**Conclusiones:**

* **Estabilidad del Sistema:** Fuera del período de caída, el sistema parece mantener un rendimiento relativamente estable. Los valores de VUs y peticiones por segundo se mantienen dentro de un rango consistente, lo que sugiere que el sistema puede manejar esta carga de manera predecible.
* **Evento Anómalo:** La caída abrupta en los VUs y las peticiones entre 01:50:00 y 02:00:00 es el punto más notable de la prueba. Esto podría deberse a un problema del sistema que causó la desconexión o el cese de actividad de los usuarios, o a un problema en el script de la prueba misma. La causa de esta anomalía necesita ser investigada a fondo.
* **Capacidad de Carga:** El sistema demuestra ser capaz de manejar una carga de aproximadamente 140 VUs, lo que se traduce en unas 82 peticiones por segundo, sin mostrar signos evidentes de degradación en el gráfico (como un aumento incontrolado de tiempos de respuesta o una caída de peticiones sin una caída de VUs).

**Recomendaciones:**

* **Investigar el Evento Anómalo:** Es crucial investigar la causa de la caída en los VUs y las peticiones. Se debe revisar los logs del servidor de pruebas, los logs de la aplicación bajo prueba y la configuración del script de la prueba para identificar qué causó este evento. ¿Fue un error de la aplicación, una desconexión de red o un problema en el generador de carga?
* **Realizar Pruebas de Estrés:** Una vez que el sistema se considere estable bajo una carga normal, se recomienda realizar pruebas de estrés para determinar el punto de ruptura del sistema. Aumentar progresivamente los VUs hasta que se observe una degradación significativa en el rendimiento (ej. aumento en los tiempos de respuesta o errores). Esto ayudará a definir los límites de capacidad del sistema.
* **Monitoreo Adicional:** Para futuras pruebas, se recomienda monitorear métricas adicionales como los tiempos de respuesta de la aplicación, el uso de CPU y memoria del servidor, y la tasa de errores. Esto proporcionará una visión más completa del rendimiento y la salud del sistema bajo carga.
* **Optimizar el Sistema:** Si los resultados de las pruebas de estrés indican que el sistema no cumple con los requisitos de rendimiento esperados, se deben considerar optimizaciones a nivel de código, infraestructura o configuración para mejorar su capacidad de respuesta y estabilidad bajo carga alta.