

Informe Testing de Performance

Alan Bermudez

Dominio

- Plataforma de comercio electrónico simplificada que permite gestionar usuarios, productos y pedidos a través de servicios REST.

Motivación del proyecto

- Ofrecer un entorno controlado para practicar ingeniería de rendimiento y Site Reliability Engineering (SRE).
- Permitir experimentar con diferentes tipos de pruebas (carga, estrés, resistencia, picos, volumen y escalabilidad).
- Enseñar a interpretar métricas de rendimiento y a traducir hallazgos en acuerdos SLI/SLO/SLA.

Problemas detectados

#Observación: al momento de crear estos reportes Create order, Update order, Create product y Update product funcionaban correctamente pero al parecer en el transcurso del tiempo estas funciones fueron actualizadas y ya no funcionan. Por falta de tiempo no se pudo estudiar correctamente la afectación que este cambio ha podido producir en las pruebas asimismo como no se pudo comprobar si otras opciones no fueron actualizadas y ya no funcionan de la misma forma que lo hacían al inicio.

Se puede apreciar que las siguientes funciones representan problemas para el rendimiento general del servicio:

-Update User

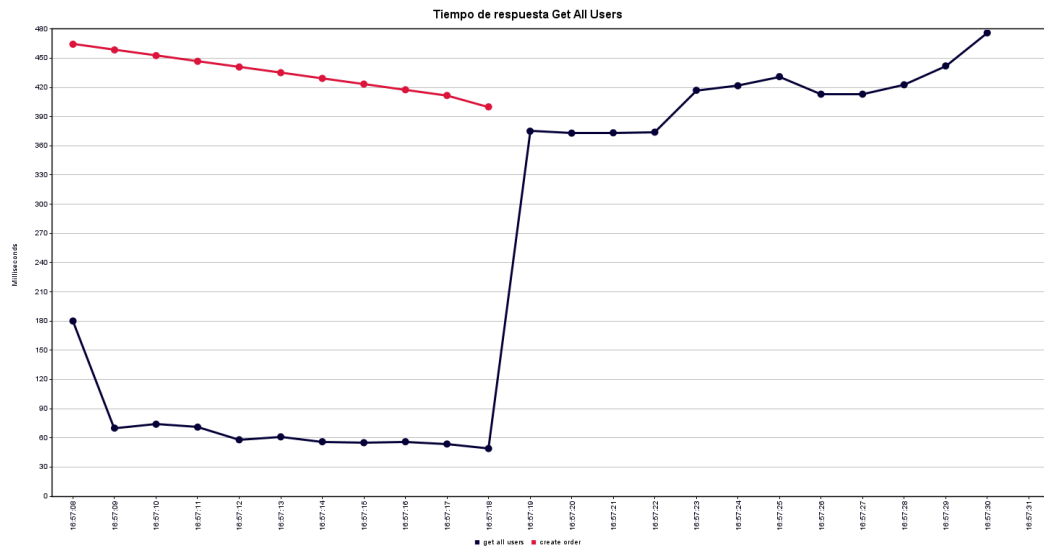
-Create Order

-Update Order

#Get all users

Se puede observar que el uso de "Update user" deteriora el tiempo de respuesta de esta función, generando un incremento de aproximadamente entre 20 y 30 milisegundos. Asimismo, la función "Create order" también afecta el rendimiento de "Get all users", provocando un aumento en el tiempo de respuesta de entre 60 y 70 milisegundos. Del mismo modo, "Update order" ocasiona un deterioro adicional, incrementando su tiempo de respuesta en alrededor de 40 milisegundos.

Ejemplo al usar create order:



P95% antes de Create order=77

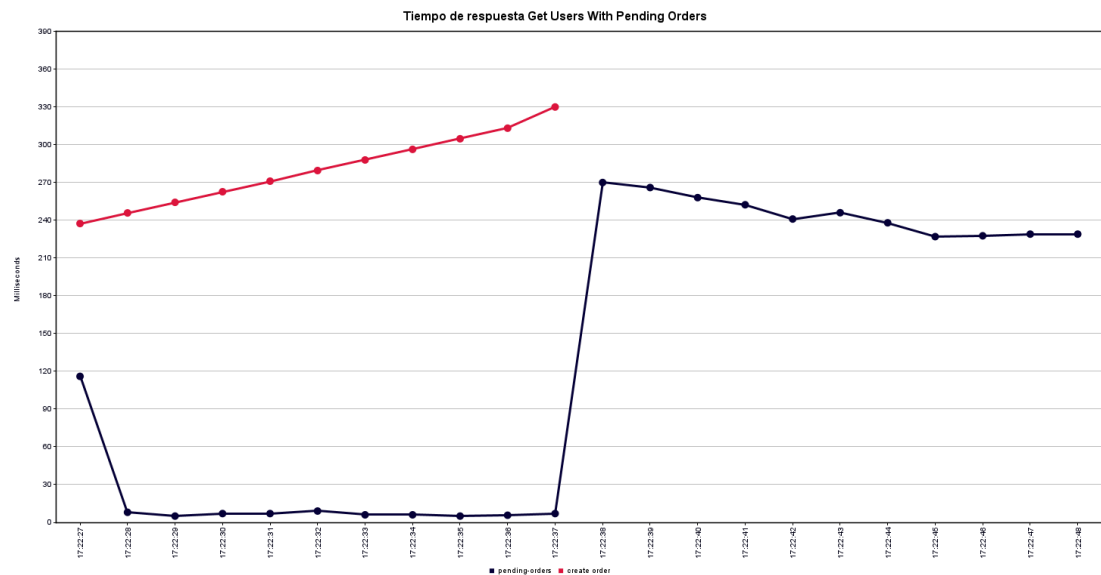
P95% después de Create order= 470

#Get user by id

Se ve afectada de la misma forma que Get all users

#Get users with pending orders

La funcionalidad Get users with pending orders se ve afectada por “Update user”, la cual deteriora su tiempo de respuesta, incrementándolo en aproximadamente entre 20 y 30 milisegundos. Del mismo modo, “Create order” también influye negativamente en su desempeño, provocando un aumento en el tiempo de respuesta de entre 50 y 60 milisegundos.



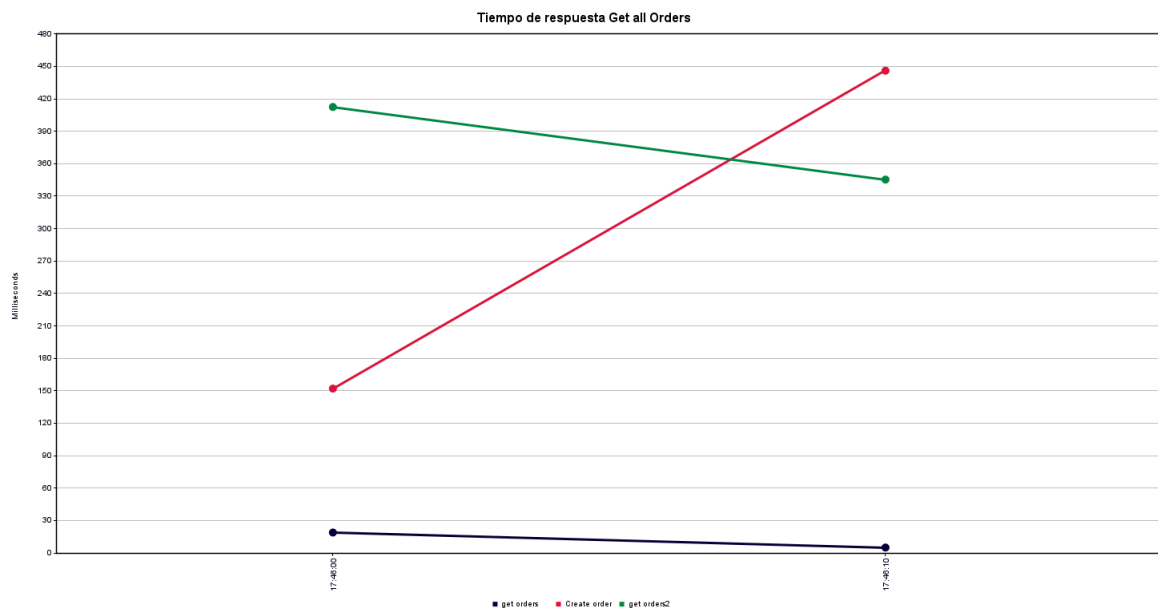
P95% antes de update user: 9

P95% después de update user: 232

#Get all orders

La funcionalidad Get all orders presenta un deterioro en su tiempo de respuesta al verse afectada por “Create order”, que incrementa el tiempo de ejecución en aproximadamente entre 40 y 50 milisegundos. De igual manera, “Update order” también impacta negativamente en su rendimiento,

provocando un aumento similar en el tiempo de respuesta, de entre 40 y 50 milisegundos adicionales.



P95% antes de create order: 12

P95% después de create order: 401

Se detectó que Get all orders se rompe tras usar "create order" OutOfMemoryError - Fuga de memoria

#Get order by id

-Esta función solo se vió afectada por "Update order" la cual empeora el tiempo de respuesta en 40-50ms

#Get orders by user id

La funcionalidad Get orders by user id presenta un deterioro en su tiempo de respuesta debido a la influencia de "Create order", que incrementa el tiempo de ejecución en aproximadamente entre 40 y 50 milisegundos. De igual forma, "Update order" también afecta negativamente su rendimiento, ocasionando un aumento similar en el tiempo de respuesta de entre 40 y 50 milisegundos adicionales.

-Se detectó que Get orders by user id se rompe tras usar la funcionalidad diez veces- OutOfMemoryError - fuga de memoria

#Get orders by status

-Afectada por "create order", aumenta el tiempo de respuesta a 57ms

#Get orders by min amount

La funcionalidad Get orders by min amount muestra un deterioro en su tiempo de respuesta debido a la influencia de "Create order", que incrementa el tiempo de ejecución en aproximadamente entre 40 y 50 milisegundos. Asimismo, "Update order" también impacta negativamente en su rendimiento, generando un aumento similar en el tiempo de respuesta de entre 40 y 50 milisegundos adicionales.

#Get orders with notes

La funcionalidad Get orders with notes presenta un deterioro en su tiempo de respuesta como consecuencia de la influencia de "Create order", que incrementa el tiempo de ejecución en aproximadamente entre 40 y 50 milisegundos. Del mismo modo, "Update order" también afecta su rendimiento, provocando un aumento similar en el tiempo de respuesta de entre 40 y 50 milisegundos adicionales.

Observamos que también existen funciones rotas las cuales no otorgan una respuesta satisfactoria al momento de interactuar con ellas y son las siguientes:

-Create product: error 500

-Update product: error 404

-Create order with items: error 400

SLI:

En el contexto de este proyecto, un SLI (Indicador de Nivel de Servicio) representa la métrica que cuantifica el rendimiento real de los servicios REST de la plataforma de comercio electrónico.

Para este caso, los SLIs se centraron principalmente en el tiempo de respuesta (percentil P95) de las distintas funcionalidades, tales como Get all users, Create order, Update order o Get orders by user id, ya que esta métrica permite medir cómo responde el sistema bajo diferentes condiciones de carga y concurrencia.

Por ejemplo, se observaron variaciones significativas en los tiempos de respuesta, como el aumento del P95 de 77 ms a 470 ms al ejecutar la función Create order, lo que refleja una degradación del servicio.

En resumen, los SLIs son los valores medidos que permiten evaluar objetivamente el rendimiento y estabilidad del sistema.

SLO:

El SLO (Objetivo de Nivel de Servicio) define el nivel de rendimiento esperado o meta que debe cumplir el sistema según los indicadores establecidos (SLIs).

Dentro de este proyecto, los SLO se utilizaron para definir umbrales aceptables de comportamiento del servicio bajo diferentes tipos de pruebas (carga, estrés, resistencia, etc.).

Por ejemplo:

Se estableció que el P95 del tiempo de respuesta de las operaciones principales (Get all users, Get all orders, etc.) no debía superar los 200 ms bajo carga normal.

También se consideró que la tasa de errores debía mantenerse por debajo del 1 % durante las pruebas de ejecución prolongada.

De esta forma, los SLO permiten determinar cuándo el rendimiento del sistema se encuentra dentro de los márgenes aceptables y cuándo requiere intervención o ajustes.

SLA:

El SLA (Acuerdo de Nivel de Servicio) representa el compromiso formal o interno entre el proveedor del servicio y los usuarios o clientes, basado en los SLO previamente definidos.

En el contexto de este proyecto, el SLA funcionaría como una referencia de confiabilidad esperada del sistema en un entorno real de producción.

Por ejemplo, se podría establecer que:

El tiempo de respuesta promedio no debe exceder los 250 ms en condiciones normales.

En caso de superar dichos umbrales (por ejemplo, caídas de servicio o tiempos de respuesta excesivos), se deben aplicar acciones correctivas o procesos de mejora dentro del ciclo de monitoreo continuo.

En este sentido, los SLA traducen los objetivos técnicos (SLO) en acuerdos operativos, garantizando que el servicio cumpla con los niveles de calidad esperados por los usuarios finales.

Propuestas de optimización:

A partir de los resultados obtenidos durante las pruebas de performance y el análisis de degradación entre servicios, se proponen las siguientes acciones de optimización para mejorar el rendimiento y la confiabilidad de la plataforma:

1. Optimización del manejo de memoria

Se detectaron errores del tipo OutOfMemoryError en funcionalidades como Get all orders y Get orders by user id tras ejecuciones repetidas.

Se recomienda:

Implementar un sistema de liberación y reutilización de recursos (por ejemplo, cerrar conexiones o flujos de datos correctamente).

Revisar el uso de colecciones en memoria y evitar mantener grandes volúmenes de datos cargados sin necesidad.

Configurar parámetros de JVM o heap size adecuados si se usa Java.

Incorporar mecanismos de paginación en endpoints que retornan grandes volúmenes de datos (Get all users, Get all orders, etc.).

2. Optimización de consultas y lógica de negocio

Algunos servicios (Create order, Update user, Update order) impactan negativamente en el tiempo de respuesta de otros endpoints.

Esto sugiere dependencias o bloqueos en operaciones de base de datos o procesamiento interno.

Se propone:

Revisar la eficiencia de las consultas SQL o las operaciones ORM, agregando índices cuando corresponda.

Reducir el número de operaciones encadenadas o transacciones largas.

Aplicar caché local o distribuido para datos de lectura frecuente (por ejemplo, información de usuarios o pedidos ya consultados).

3. Mejorar la concurrencia y el manejo de threads

Durante las pruebas de carga, ciertas operaciones mostraron degradación bajo concurrencia. Para mejorar la respuesta en escenarios de múltiples solicitudes simultáneas:

Implementar pooling de conexiones (base de datos o HTTP).

Revisar y optimizar el número máximo de hilos activos por servicio.

Aplicar técnicas de asincronismo o colas de procesamiento (por ejemplo, RabbitMQ o Kafka para operaciones de creación de pedidos).

Conclusión:

La aplicación de estas medidas permitirá mejorar el cumplimiento de los SLO definidos, reducir los tiempos de respuesta promedio, mitigar errores de memoria y aumentar la disponibilidad general del sistema, garantizando así una mejor experiencia para los usuarios y mayor estabilidad del entorno de pruebas.