Informe Tarea 2: Sistema para la gestión de pedidos de un delivery

Jorge Toro Macías
jorge.toro1@mail.udp.cl

Junio, 2023

1 Describa en detalle cómo la arquitectura de Apache Kafka permite brindar tolerancia a fallos en el sistema propuesto.

Una de las principales características que hace a Kafka una buena plataforma con alta tolerancia a fallos es su sistema de particiones y réplicas. Los datos en Kafka se dividen en particiones dentro de un tema (topic). Cada una de estas particiones es una secuencia de registros que siguen un orden, por lo que brinda capacidad de paralelismo. Esto significa una alta capacidad de distribución de carga de trabajo entre varios nodos. Además, estas particiones son capaces de replicarse en varios brokers, de manera que si el broker principal falla, otro toma el lugar, asegurando así disponibilidad en todo momento de los servicios.

2 Considerando el sistema propuesto, ¿cómo se aseguraría de que el sistema es capaz de escalar para manejar un aumento significativo en el número de solicitudes? Describa cualquier ajuste de configuración o estrategia de escalado que implementaría.

Tal como es comentado en el punto anterior, Kafka maneja particiones y réplicas, lo que lo hace escalable de manera que los datos, tales como los mensajes, son capaces de almacenarse y producirse desde distintos puntos o nodos, manteniendo así una disponibilidad continua, y además se mantiene el orden original, por lo que no hay pérdida de datos.

3 Dada la naturaleza variable de la demanda ¿cómo podría optimizar el uso de recursos del sistema durante los períodos de baja demanda, y cómo prepararía el sistema para los picos de demanda? Explique las estrategias de optimización y autoscaling que podría emplear.

En un período de baja demanda, hablando del concepto de autoscaling, se podría reducir la capacidad de recursos individuales de las instancias existentes, tales como GPU, memoria, etc. De esta manera es más rápido ajustar las capacidades, al no haber necesidad de adicionar más instancias. Por otra parte, en cuanto a escalabilidad horizontal, se podrían reducir las instancias existentes al no existir una demanda que requiera una mayor cantidad. Ambas son formas de optimizar los recursos en un momento de baja demanda.
En un período de alta demanda existen técnicas, métodos y configuraciones que se pueden aplicar a los organismos involucrados en el sistema. Por ejemplo, existen políticas de autoscaling basadas en métricas como el uso de memoria, CPU, latencia de solicitudes y tasa de mensajes en Kafka. Se aplica lo mismo para otras plataformas tales como Kubernetes. También existe el uso de colas para manejar los picos que se generen en las cargas. Esto le facilita a los consumidores procesar los mensajes a su propio ritmo, desacoplándose de los productores.