

Caso de Estudio: Implementación de una Base de Datos Distribuida en Cassandra para la Plataforma de Pedidos "FastEat"

Descripción del Escenario Empresarial

"FastEat" es una cadena de restaurantes con presencia en varias ciudades de Latinoamérica. Para incrementar sus ventas y mejorar la experiencia de sus clientes, la empresa ha lanzado una plataforma digital de gestión de pedidos y entregas a domicilio, la cual permite a los usuarios realizar órdenes en línea, personalizar sus platos, rastrear el estado de sus pedidos en tiempo real y calificar el servicio recibido.

La expansión acelerada de la cadena y el éxito de la plataforma han hecho que el volumen de transacciones crezca rápidamente, especialmente en horarios pico y durante promociones especiales. Además, la empresa planea implementar un sistema de recomendaciones personalizadas para sugerir platillos a los clientes según su historial y ubicación, así como análisis en tiempo real de las preferencias de los usuarios.

La plataforma debe operar de manera continua (24/7), soportar miles de operaciones simultáneas y garantizar que la información de pedidos y entregas sea precisa, rápida y esté siempre disponible, incluso en caso de fallos o caídas de algún servidor.

Problema a Resolver

El sistema de base de datos relacional utilizado inicialmente por FastEat ha comenzado a presentar graves problemas de rendimiento: demoras en la consulta de pedidos durante horas pico, dificultades para escalar la infraestructura ante el crecimiento de la demanda y limitaciones para integrar nuevos módulos como la recomendación personalizada o el análisis de datos en tiempo real.

Por este motivo, la dirección tecnológica de FastEat ha decidido migrar la plataforma a una base de datos distribuida basada en Cassandra, con el objetivo de asegurar escalabilidad, alta disponibilidad y tolerancia a fallos, además de facilitar el manejo eficiente de grandes volúmenes de datos y transacciones concurrentes.

Objetivos del Proyecto

1. Analizar los requerimientos funcionales del sistema de pedidos y entregas de FastEat, identificando los tipos de datos principales, relaciones y operaciones críticas de consulta y escritura.
2. Diseñar un modelo de datos eficiente para Cassandra, aplicando las mejores prácticas de bases de datos NoSQL, optimizando consultas y garantizando la integridad de la información.

3. Definir la estructura de tablas, claves de partición y clustering keys más adecuadas para responder a los retos de la plataforma (consultas por usuario, seguimiento de pedidos, historial de entregas, etc.).
4. Proponer una estrategia de despliegue en contenedores Docker para la infraestructura de Cassandra, asegurando escalabilidad horizontal y alta disponibilidad.
5. Justificar las decisiones de diseño en función de los escenarios de uso y los retos operativos descritos.
6. Los datos deben ser leídos (desde python con jupyter) de un archivo externo tipo json para que sean insertados en la tabla
7. Desarrollar y ejecutar un conjunto de consultas esenciales sobre la base de datos implementada, utilizando Python en Jupyter Notebook para la conexión y análisis de los datos. Las consultas deben demostrar la recuperación de pedidos, inserción de nuevos pedidos, actualización del estado de entregas y generación de reportes simples.

Indicaciones para el Estudiante

Como desarrolladores responsables del proyecto, deberán analizar el escenario presentado y diseñar una solución de base de datos distribuida en Cassandra que cumpla con los requisitos de FastEat. Se espera un documento técnico donde describan el modelo de datos propuesto, justifiquen la elección de las tablas y claves, y detallen la estrategia de despliegue en Docker.

Además, deben preparar y ejecutar, en Jupyter Notebook, consultas relevantes sobre la base de datos usando Python (por ejemplo, empleando la librería `cassandra-driver`). Deberán incluir ejemplos de inserción de pedidos, consultas de estado, actualización de entregas y visualizaciones de datos relevantes.

El enfoque del trabajo debe estar en la eficiencia, escalabilidad y disponibilidad del sistema, demostrando dominio de la integración entre bases de datos distribuidas y herramientas modernas de análisis de datos.