

Análisis de Caso: Amazon DynamoDB

1. Resumen Ejecutivo

La empresa de comercio electrónico enfrenta problemas de rendimiento y escalabilidad debido al crecimiento acelerado de clientes y pedidos. El uso de una base de datos relacional tradicional ha generado latencias cuando se incrementan los pedidos simultáneos. Como solución, se propone migrar a Amazon DynamoDB, una base de datos NoSQL altamente escalable y sin servidor, que permite gestionar millones de transacciones por segundo con bajo mantenimiento y alta disponibilidad.

2. Análisis del Problema

- La base relacional presenta latencias bajo alta concurrencia.
- Dificultades para escalar verticalmente sin incurrir en altos costos.
- Esquema rígido, poco flexible frente a nuevas necesidades del negocio.
- Limitaciones en el manejo de grandes volúmenes de transacciones distribuidas.
- Alta carga administrativa en mantenimiento y ajustes de rendimiento.

3. Diseño de la Solución en DynamoDB

Se propone una tabla principal denominada 'Pedidos' con:

- Clave de partición: ID_Cliente
- Clave de ordenación: ID_Pedido

Además:

- GSI (Índice Secundario Global): Fecha_Pedido + Estado para reportes y filtros.
- LSI (Índice Secundario Local): Tipo_Pago para análisis dentro del cliente.
- Modelo de acceso diseñado según los patrones de consulta del sistema.

4. Evaluación de la Solución

Ventajas:

- Escalabilidad horizontal automática.
- Modelo flexible y orientado a rendimiento.
- Baja latencia en lecturas y escrituras.
- Integración nativa con servicios AWS.

Desventajas:

- Requiere rediseñar el esquema de datos (modelado orientado al acceso).
- Costos variables según el uso.
- La consistencia eventual puede no ser adecuada para todos los casos.

5. Conclusiones y Recomendaciones Finales

Se recomienda avanzar con la migración a Amazon DynamoDB por su capacidad para manejar cargas altas con eficiencia y flexibilidad.

La empresa debe:

- Implementar pruebas piloto.
- Formar al equipo en modelado NoSQL.
- Habilitar autoescalado de capacidad.
- Integrar con AWS Lambda para eventos y API Gateway para exponer servicios.
- Asegurar consistencia cuando sea necesario, utilizando modos de lectura fuerte.