



Bases de Datos No SQL

1. Introducción a las BD NoSQL



Universidad
Rey Juan Carlos

Bibliografía

- Dan Sullivan (2015). NoSQL for Mere Mortals. Addison-Wesley Professional.
- Guy Harrison (2016). Next Generation Databases: NoSQL, NewSQL, and Big Data. Apress.
- Xun Wu et al (2018). Seven NoSQL Databases in a Week. Pack Publishing.
- Dan McCreary & Ann Kelly (2014). Making Sense of NoSQL. A guide for managers and the rest of us. Manning Publications.
- Gaurav Vaish (2013). Getting Started with NoSQL. Your guide to the world and technology of NoSQL. PACKT Publishing.
- Joe Celko (2014). Joe Celko's Complete guide to NoSQL. What every SQL professional needs to know about nonrelational databases. Morgan Kauffmann.

Índice

1. Introducción a las Bases de Datos NoSQL

1.1 Limitaciones de las Bases de Datos Relacionales

1.2 Clasificación de las Bases de Datos NoSQL

1.3 Elección de una Base de Datos NoSQL

2. BD Clave-Valor

3. BD orientadas a Documentos

4. BD orientadas a Grafos

5. BD Familia de Columnas

Introducción a las BD NoSQL

Limitaciones de las BD Relacionales

- Limitaciones de las Bases de Datos Relacionales
 - Han sido las BD dominantes de las últimas décadas.
 - Soportar cierta cantidad de usuarios en **negocios tradicionales** es diferente a soportar una enorme cantidad de usuarios web (Google, LinkedIn, Yahoo!, Amazon) en **aplicaciones con una gran cantidad de datos**:
 - Grandes cantidades de operaciones de lectura y escritura
 - Tiempos de respuesta con baja latencia
 - Alta disponibilidad
 - Estos requisitos son **difíciles (y caros)** de cumplir usando BD Relacionales:
 - Más CPU, memoria, usar dispositivos más rápidos... hasta un límite
 - Desnormalizar la BD + rendimiento y + riesgo de anomalías de datos
 - Usar varios servidores: en una BD relacional es complejo (complejidad de implementar transacciones, aumenta el coste a medida que el número de servidores aumenta...)
 - Pese a ello, algunas empresas usan BD Relacionales para algunas operaciones (p.e. Facebook, Google, Twitter usan MySQL).
 - Si no se tienen tantos recursos, una opción es usar bases de datos NoSQL.

Introducción a las BD NoSQL

Limitaciones de las BD Relacionales

- Motivaciones para el uso de Bases de Datos Not Only SQL (NoSQL)
 - La motivación de las BD NoSQL fueron los **problemas del mundo real**.
 - Aplicaciones Web con decenas de miles de usuarios eran difíciles de implementar con BD Relacionales.
 - Gestión de grandes volúmenes de datos requiere de las siguientes **características**:
 - Escalabilidad
 - Coste
 - Flexibilidad
 - Disponibilidad
 - En función de la aplicación, algunas de estas características son más importantes que otras.

Introducción a las BD NoSQL

Limitaciones de las BD Relacionales

- **Escalabilidad**
 - Capacidad de satisfacer las necesidades de **cargas de trabajo variantes eficientemente**.
 - Bases de datos NoSQL diseñadas para utilizar los **servidores disponibles** con **poca intervención** de los administradores.
 - Cuando se añaden o eliminan servidores, las bases de datos NoSQL **se ajustan al nuevo conjunto de servidores** disponibles.
- **Coste**
- **Flexibilidad**
- **Disponibilidad**

Introducción a las BD NoSQL

Limitaciones de las BD Relacionales

- Escalabilidad
- **Coste**
 - Es una consideración obvia para cualquier organización.
 - Los **proveedores de SGBD comerciales** disponen de una amplia gama de licencias (tamaño del servidor, **accesos concurrentes, número de usuarios...**).
 - La mayoría de BD NoSQL están disponibles como **open source**.
 - Existen empresas que proporcionan **soporte** para productos open source.
- Flexibilidad
- Disponibilidad

Introducción a las BD NoSQL

Limitaciones de las BD Relacionales

- Escalabilidad
- Coste
- **Flexibilidad**
 - Existe un conjunto de **modelos y negocios** en los que los SGBD Relacionales se ajustan muy bien, y su **flexibilidad es suficiente** (p.e. bancos, ventas, energía, salud...)
 - conocemos las tablas y columnas al comienzo del proyecto
 - la mayoría de las columnas tienen sentido para la mayoría de las filas
 - En otros casos, necesitamos **mucha más flexibilidad**. P.e. aplicación de comercio electrónico que vende muchos tipos de productos: ¿una tabla para todos los productos o una tabla por cada producto?
 - Una BD NoSQL **no** requiere una **estructura fija**
 - Por ejemplo, en una BD orientada a documentos podríamos añadir nuevos atributos a medida que se necesiten sin hacer un ALTER TABLE, **basta con añadir un nuevo documento con los nuevos atributos**.
- Disponibilidad

Introducción a las BD NoSQL

Clasificación de Bases de Datos NoSQL

- Las Bases de Datos NoSQL más extendidas son:
 - Clave-Valor
 - Orientadas a Documentos
 - Orientadas a Grafos
 - Familia de Columnas
- Aunque las Bases de Datos NoSQL no tienen por qué estar implementadas en sistemas distribuidos, algunas de sus características más interesantes destacan en una **implementación distribuida**.
 - Tiene sentido implementar una BD NoSQL en **múltiples servidores** cuando la **disponibilidad** y la **escalabilidad** son aspectos básicos.
 - Algunas características importantes en distribución no lo son en implementaciones en un único servidor.
 - En distribución, se necesita un **balance** entre disponibilidad (A), consistencia (C), partition-protection (P) y durabilidad, aspectos fundamentales en BD NoSQL.

Introducción a las BD NoSQL.

Elección de una BD NoSQL

- La llegada de los SGBD Relacionales prácticamente acabó con los gestores prerrelacionales (p.e basados en ficheros, BD en Red y BD Jerárquicas).
- La aparición y expansión de **sistemas web actuales** hizo que los SGBDR **no dieran respuesta** a las necesidades.
- Esto provocó el "**renacimiento**" de **BD no relacionales** (p.e. Yahoo! -> Hadoop, Google -> BigTable, Amazon -> DynamoDB).
- Muchos de los productos son libres, lo cual ha propiciado su expansión.

Introducción a las BD NoSQL.

Elección de una BD NoSQL

- En un **SGBD Relacional** lo fundamental es la **estructura** de las entidades y cómo se **relacionan**.
- En **BD NoSQL**, aunque también modelamos estos aspectos, el **rendimiento** es más importante que preservar el modelo.
- ⇒ El **Modelo Relacional** respondió a la **necesidad** de resolver las **anomalías en los datos** y a la dificultad de **reutilizar BD existentes** para nuevas aplicaciones.
- ⇒ Las **BD NoSQL** también surgen ante **una necesidad**: las dificultades para **escalar** y dar respuesta a las crecientes demandas **de altos volúmenes de operaciones de lectura y escritura**.
Como consecuencia, **perdemos consistencia inmediata y transacciones ACID** (en algunos casos).
- Algunos otros aspectos que influyen y a tener en cuenta a la hora **de elegir una BD NoSQL**:
 - Tipos de consultas que se realizan
 - Volumen de lecturas y escrituras
 - Tolerancia ante datos inconsistentes en las réplicas
 - La naturaleza de las relaciones entre entidades
 - Disponibilidad y requisitos de recuperación ante pérdidas
 - Flexibilidad en los modelos de datos
 - Requisitos de latencia

Introducción a las BD NoSQL.

Elección de una BD NoSQL

- **Clave-Valor**

- Apropriadas para:
 - Lecturas y escrituras pequeñas pero frecuentes
 - Modelo de datos sencillo
 - Dominios sencillos (strings, enteros, booleanos, ...), o estructurados (listas, conjuntos, ...)
- Menor capacidad de consultas que BD orientadas a documentos, orientadas a grafos y de familia de columnas.
- Ejemplos:
 - Caché de BD relacional para mejorar el rendimiento
 - Atributos temporales en aplicaciones web (carrito)
 - Datos de configuración y usuario en aplicaciones móviles
 - Almacenamiento de grandes objetos, como imágenes o audio
- Productos:
 - **Redis**, Riak, SimpleDB, DynamoDB, Oracle BerkeleyDB, ...

Introducción a las BD NoSQL.

Elección de una BD NoSQL

- **Orientadas a Documentos**

- Su principal ventaja es la **flexibilidad**:
 - Distintos tipos de atributos y gran cantidad de información
 - Desnormalización, por medio de documentos embebidos
- Mayor **capacidad de consultas** que las Clave-Valor (filtrado de documentos basado en atributos, uso de índices,...)
- Las BD NoSQL más populares por su flexibilidad, rendimiento y facilidad de uso.
- Ejemplos:
 - Sitios web *con alta carga de lecturas y escrituras*
 - Gestión de tipos de datos con atributos variables
 - Aplicaciones que utilizan estructuras de datos JSON
 - Aplicaciones que sacan partido de la desnormalización de datos (embebiendo documentos)
- Productos:
 - **MongoDB**, CouchDB, CouchBase, RethinkDB, Cloudant

Introducción a las BD NoSQL.

Elección de una BD NoSQL

- **Orientadas a Grafos**

- Dominios en los que aparecen redes de entidades conectadas
 - Ocurrencias de las mismas entidades relacionadas entre sí
 - Ejemplos: autovías que conectan ciudades, proteínas que interactúan, investigadores que colaboran
- Ejemplos:
 - Gestión de redes e infraestructuras
 - Recomendación de productos y servicios
 - Redes sociales (para determinadas funcionalidades)
- Productos:
 - **Neo4j**, Titan

Introducción a las BD NoSQL.

Elección de una BD NoSQL

- **Familia de Columnas**
 - Grandes volúmenes de datos, rendimiento de lectura/escritura, alta disponibilidad.
 - No está pensada para funcionar en un servidor único.
 - Son especialmente indicadas cuando:
 - Grandes volúmenes de datos (cientos de TB)
 - Aplicaciones requieran alta carga de escritura en la BD
 - Aplicaciones distribuidas geográficamente en múltiples centros de datos
 - Aplicaciones que toleren cierta (breve) inconsistencia en las réplicas
 - Productos:
 - **Cassandra** (desarrollado por Facebook), Hbase, BigTable (desarrollado por Google)
- Clave-Valor, Orientadas a Documentos y Familia de Columnas tienen objetivos similares.

Introducción a las BD NoSQL.

Elección de una BD NoSQL

- **NoSQL y Relacionales**
 - **Complementarias**
 - Relacional aporta integridad y reduce el riesgo de anomalías, pero a cambio de penalizar el rendimiento.
 - Los SGBD Relacionales continuarán dando soporte a sistemas transaccionales (OLTP) y aplicaciones de BI.
 - Otras aplicaciones (web, servicios móviles, big data) pueden, o no, funcionar bien con SGBD Relacionales.
 - Distintas tecnologías de BD pueden coexistir para resolver diferentes problemas.
- Al contrario que ocurrió con los SGBD Relacionales, que desplazaron a su antecesores, **las BD NoSQL coexisten entre sí, y con los SGBD Relacionales** por las diferentes necesidades.

Índice

1. *Introducción a las Bases de Datos NoSQL*

1.1 Limitaciones de las Bases de Datos Relacionales

1.2 Clasificación de las Bases de Datos NoSQL

1.3 Elección de una Base de Datos NoSQL

2. **BD Clave-Valor**

3. **BD orientadas a Documentos**

4. **BD orientadas a Grafos**

5. **BD Familia de Columnas**