

Gestores de base de datos no relacionales

Posi Vargas, Miguel LLanque, Jesus Quenaya y Erlang Vilca

July 2020

1 Resumen

El objetivo de esta investigación es comparar los gestores de base de datos no relacionales, sus características en funcionalidad como en rendimiento a escala, en variedad de modelos de datos como documentos, gráficos, clave-valor, en memoria y búsqueda.

2 Abstract

The objective of this research is to compare non-relational database managers, their characteristics in functionality as well as performance at scale, in a variety of data models such as documents, graphics, key-value, in-memory and search.

3 Introduction

Son muchas las aplicaciones web que utilizan algún tipo de bases de datos para funcionar. Hasta ahora estábamos acostumbrados a utilizar bases de datos SQL como son MySQL, Oracle y entre otras, pero desde hace ya algún tiempo han aparecido otras que reciben el nombre de NoSQL (Not only SQL – No sólo SQL) y que han llegado con la intención de hacer frente a las bases relacionales utilizadas por la mayoría de los usuarios.

4 Desarrollo

4.1 Qué son las bases de datos NoSQL?



Se puede decir que la aparición del término NoSQL aparece con la llegada de la web 2.0 ya que hasta ese momento sólo subían contenido a la red aquellas empresas que tenían un portal, pero con la llegada de aplicaciones como Facebook, Twitter o Youtube, cualquier usuario podía subir contenido, provocando así un crecimiento exponencial de los datos.

Es en este momento cuando empiezan a aparecer los primeros problemas de la gestión de toda esa información almacenada en bases de datos relacionales. En un principio, para solucionar estos problemas de accesibilidad, las empresas optaron por utilizar un mayor número de máquinas pero pronto se dieron cuenta de que esto no solucionaba el problema, además de ser una solución muy cara. La otra solución era la creación de sistemas pensados para un uso específico que con el paso del tiempo han dado lugar a soluciones robustas, apareciendo así el movimiento NoSQL.

4.2 Ventajas de los sistemas NoSQL

Esta forma de almacenar la información ofrece ciertas ventajas sobre los modelos relacionales. Entre las ventajas más significativas podemos destacar:

- Se ejecutan en máquinas con pocos recursos:** Estos sistemas, a diferencia de los sistemas basados en SQL, no requieren de apenas computación, por lo que se pueden montar en máquinas de un coste más reducido.
- Escalabilidad horizontal:** Para mejorar el rendimiento de estos sistemas simplemente se consigue añadiendo más nodos, con la única operación de indicar al sistema cuáles son los nodos que están disponibles.
- Pueden manejar gran cantidad de datos:** Esto es debido a que utiliza una estructura distribuida, en muchos casos mediante tablas Hash.
- No genera cuellos de botella:** El principal problema de los sistemas SQL es que necesitan transcribir cada sentencia para poder ser ejecutada, y cada sentencia compleja requiere además de un nivel de ejecución aún más complejo, lo que constituye un punto de entrada en común, que ante muchas peticiones puede ralentizar el sistema.

4.3 Principales diferencias con las bases de datos SQL?

Algunas de las diferencias más destacables que nos podemos encontrar entre los sistemas NoSQL y los sistemas SQL están:

–**No utilizan SQL como lenguaje de consultas.** La mayoría de las bases de datos NoSQL evitan utilizar este tipo de lenguaje o lo utilizan como un lenguaje de apoyo. Por poner algunos ejemplos, Cassandra utiliza el lenguaje CQL, MongoDB utiliza JSON o BigTable hace uso de GQL. –**No utilizan estructuras fijas como tablas para el almacenamiento de los datos.** Permiten hacer uso de otros tipos de modelos de almacenamiento de información como sistemas de clave-valor, objetos o grafos.

–**No suelen permitir operaciones JOIN.** Al disponer de un volumen de datos tan extremadamente grande suele resultar deseable evitar los JOIN. Esto se debe a que, cuando la operación no es la búsqueda de una clave, la sobrecarga puede llegar a ser muy costosa. Las soluciones más directas consisten en desnormalizar los datos, o bien realizar el JOIN mediante software, en la capa de aplicación.

–**Arquitectura distribuida.** Las bases de datos relacionales suelen estar centralizadas en una única máquina o bien en una estructura máster-esclavo, sin embargo en los casos NoSQL la información.

4.4 diferencias de bases de datos NoSQL

Veamos a continuación algunas diferencias de bases NoSQL más utilizadas actualmente

MongoDB :



Se trata de una base de datos creada por 10gen del tipo orientada a documentos, de esquema libre, es decir, que cada entrada puede tener un esquema de datos diferente que nada tenga que ver con el resto de registros almacenados. Es bastante rápido a la hora de ejecutar sus operaciones ya que está escrito en lenguaje C++.

MongoDB está orientado a ficheros que almacena la información en estructuras BSON con un esquema dinámico que permite su facilidad de integración. Empresas como Google, Facebook, eBay, Cisco o Adobe utilizan MongoDB como Sistema Gestor de Bases de datos.

Las características principales son :

- Indexación y replicación
- Balanceo de carga
- Almacenamiento en ficheros
- Consultas ad hoc
- Escalabilidad horizontal
- Open Source

Como desventaja principal, MongoDB no es un SGBD adecuado para realizar transacciones complejas.

Redis :



Se trata de una base de datos creada por Salvatore Sanfilippo y Pieter Noordhuis y está apoyado por VMware. Se trata de una base de datos del tipo clave-valor. Se puede imaginar como un array gigante en memoria para almacenar datos, datos que pueden ser cadenas, hashes, conjuntos de datos o listas.

Las características principales son :

- Atomicidad y persistencia
- Gran velocidad
- Simplicidad
- Multiplataforma

Cassandra :





Se trata de una base de datos creada por Apache del tipo clave-valor. Dispone de un lenguaje propio para realizar consultas CQL (Cassandra Query Language). Cassandra es una aplicación Java por lo que puede correr en cualquier plataforma que cuente con la JVM.

Las características principales son :

- Arquitectura escalable: gracias a un diseño masterless, en el que todos los nodos son iguales, lo que ofrece simplicidad operativa y fácil escalabilidad horizontal.
- Detección de fallos y recuperación transparente: para nodos que no pueden ser fácilmente restaurados o reemplazados.
- Compresión de datos: garantiza que los datos se comprimirán hasta un 80 por ciento sin que ello suponga un gasto de recursos.
- CQL (Lenguaje de Consulta Cassandra): un lenguaje similar a SQL que consigue que la transición desde una base de datos relacional sea muy sencilla.

4.5 Comparativas entre Cassandra y MongoDB

La comparativa y necesidad de uso depende mucho de la necesidad de nuestra aplicación, ay muchas variables que debes analizar, por ejemplo, cantidad de escrituras vs lecturas en la base de datos, almacenamiento constante de información, escalabilidad, velocidad de respuesta, incrementos exponenciales en el consumo de información, datos en constante actualización o bajo porcentaje de modificación, entre otras cosas.

	 cassandra	 mongoDB
Quiénes lo usan	Instagram (se almacenen 80 millones de fotos todos los días) Facebook, Twitter, Ibm, Spotify	Google Compute Engine, Forbes, Adobe, Facebook
Cómo se almacena la información	«Columns Families»	En documentos (JSON)
Sintaxis	CONSULTAR REGISTROS SELECT * FROM products INSERTAR REGISTROS INSERT INTO contacts (contact_id,name,last_name) VALUES ('c00001','Gonzalo','Chacaltana Buleje') ACTUALIZAR REGISTROS UPDATE products SET stock=2000 WHERE product_name='Milk' MongoDB:	CONSULTAR REGISTROS db.products.find() INSERTAR REGISTROS db.contacts.insert({contact_id:'c00001', name:'Gonzalo', last_name:'Chacaltana Buleje'}) ACTUALIZAR REGISTROS db.products.update({product_name:'Milk'}, {\$set:{stock:2000}})

¿Qué base de datos es adecuada para mi aplicación?

	Cassandra	MongoDB
Casos de Uso	Aplicaciones en tiempo real, comercio electrónico, sistemas de detección de fraude, Internet de las cosas, streaming.	Analítica en tiempo real, catálogo de productos, Content Management System, Mobile, IoT, almacenamiento de logs, sistema de inventarios.
Almacenamiento de datos	Columnas con ancho flexible. La información se almacena en objetos de base de datos similares a las tablas de una base de datos relacional.	Todo se almacena en documentos.
Replicación	Replicación incorporada, fácil de configurar.	Replicación incorporada, pero necesitas algún soporte para configurarlo.
Sintaxis de consulta	Si usas SQL, CQL es fácil de aprender.	Requiere aprender un nuevo lenguaje de consulta.
Escalabilidad	Ofrece escalabilidad y alta disponibilidad con un mínimo de administración.	Es una gran alternativa si tienes data no estructurada o estructurada sin una clara definición.

5 Conclusiones

- La selección de la tecnología de almacenamiento adecuada involucra la consideración de numerosos aspectos. Aunque el rendimiento suele ser el factor más importante.
- Cada vez con más frecuencia estamos viendo cómo las tecnologías NoSQL forman parte de la solución en proyectos empresariales, gracias a beneficios como la mejora en la productividad de los equipos de desarrollo, y la posibilidad de llegar antes al mercado.
- Con la reciente explosión de arquitecturas basadas en microservicios, cada vez más veremos cómo cada servicio encapsula su propia solución de gestión de datos, haciendo uso en la mayoría de las ocasiones de alguna de las tecnologías NoSQL disponibles.

6 Bibliografía

- Bases de datos NoSQL. - <https://www.acens.com/wp-content/images/2014/02/bbdd-nosql-wp-acens.pdf>
- Cassandra - <https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/cassandra-base-de-datos-agilidad-y-rendimiento-a-prueba-de-fallos>
- Gestión de Bases de Datos - <https://gestionbasesdatos.readthedocs.io/es/latest/Tema1/Teoria.html>