

Comparativa de gestores de base de datos no relacionales

Juan Carlos Mamani, Alejandro Carpio, Yanqui Rodrigo, Fernando Perez,

Jose La Torre 29 de noviembre de 2020

Resumen

Las bases de datos NoSQL utilizan una variedad de modelos de datos para acceder y administrar datos. Estos tipos de bases de datos están optimizados específicamente para aplicaciones que requieren grandes volúmenes de datos, baja latencia y modelos de datos flexibles, lo que se logra mediante la flexibilización de algunas de las restricciones de coherencia de datos en otras bases de datos.

I. INTRODUCCION

En el presente artículo se mostrará sobre tipos de base de datos no relacionales, del mismo modo detallaremos una comparativa entre los gestores que se utilizan en la base de datos NoSQL.

II. ABSTRACT

Las bases de datos NoSQL utilizan una variedad de modelos de datos para acceder y administrar datos. Estos tipos de bases de datos están optimizados específicamente para aplicaciones que requieren grandes volúmenes de datos, baja latencia y modelos de datos flexibles, logrados al relajar algunas de las restricciones de consistencia de datos en otras bases de datos.

III. DESARROLLO

i. Tipos de Bases de Datos NoSQL

Es importante destacar que existe una clasificación que debemos conocer para entender cómo estas BBDD están construidas y nos pueden ser de utilidad. Veamos cuáles son las principales clasificaciones:

- Bases de datos documentales

Una base de datos documental u orientada a documentos es un tipo de BBDD NoSQL que orienta su funcionamiento a datos alojados en documentos o tiendas de documentos. Estas bases de datos se utilizan para almacenar, recuperar y administrar datos semiestructurados.

Las bases de datos documentales almacenan

cada uno de los registros y los datos asociados en un solo documento. Cada documento contiene datos semiestructurados que pueden ser consultados con diferentes herramientas de análisis. Estas bases de datos ofrecen mucha flexibilidad, escritura rápida, y consultas rápidas gracias a su gran capacidad de indexación. Entre las BBDD documentales más reconocidas encontramos la ya conocida por nosotros a MongoDB, de 10gen, y CouchDB, de Apache.

También gozan de un gran reconocimiento en el mundo de las Bases de datos NoSQL la RavenDB, de Hibernating Rhinos, BaseX, djondb, eXist, SimpleDB, de Amazon, IBM Lotus Domino y Terrastore

- Bases de datos orientadas a grafos

Las bases de datos orientadas a grafos son excelentes herramientas para trabajar datos complejos. Estas BBDD nos permite darle representar los datos en estructuras de grafos. Esto es de gran utilidad cuando los datos que vamos a procesar tienen altos niveles de interrelación. Estas versátiles bases de datos, nos permite ejecutar consultas y almacenamiento de datos de cualquier característica sin siquiera preocuparnos por el volumen de los datos.

Gozan de un excepcional rendimiento para responder de forma eficiente al análisis y consulta de volúmenes gigantescos de datos. Ofrece también una diversidad de métodos analíticos y de consulta que la convierten en una

de las opciones más flexibles en NoSQL. Es bastante frecuente conseguir la implementación de este modelo de BBDD en estructuras web de blogs. Twitter es uno de los casos más relevantes donde una BBDD de Grafos está relacionada.

Las Bases de Datos orientadas a grafos también tiene otras funcionalidades que no te puedes perder. Entre las marcas más reconocidas está nuestra ya conocida Neo4j, pero esta no es la única. También están las conocidas HyperGraphDB, InfoGrid, AllegroGraph, InfiniteGraph, Sones y DEX/Sparksee.

- Bases de datos clave/valor

Las bases de datos clave valor son modelos no relacionales que utilizan un método simple de almacenamiento de datos. Este tipo de BBDD toma los datos como un conjunto de pares “clave-valor” en los que las claves cumplen funciones de identificadores únicos.

Las claves o los valores pueden ser cualquier tipo de datos. Pueden ser objetos simples o complejos. Estas BBDD son altamente divisibles y brindan grandes capacidades de escalabilidad horizontal.

Estos modelos de BBDD son uno de los preferidos de los clientes NoSQL, porque son simples en cuanto a funcionalidad y brindan alto rendimiento a la hora de ejecutar las lecturas y escrituras de datos. Ya hemos dado algunos ejemplos de Bases de Datos Clave Valor como Cassandra y Redis, pero es obligatorio expandir la lista cuando tenemos importantes BBDD como BigTable de Google, Dynamo de Amazon, Project Voldemort de LinkedIn, OracleNoSQL y Riak.

- Bases de datos multivalor

Las bases de datos multivalor son sistemas interesantes que incorporan diferentes características multidimensionales y NoSQL para la clasificación y manejo de los datos. Estas BBDD comparten significativas similitudes con los modelos relacionales tradicionales. Ambos esquemas contienen tablas. Pero que esto no te engañe, las BBDD multivalor proporcionan un esquema de trabajo menos rígido.

Además de proporcionar mayor

flexibilidad, los datos almacenados acá pueden contener listas de valores. Esto quiere decir que cualquier dato almacenado puede tener diversos valores asignados.

Estas bases de datos tienen un nivel de complejidad un poco más elevado debido a que incorporan reglas de normalización para su diseño. Entre las BBDD multivalor más conocidas podemos destacar Rocket D3 DBMS, Rocket mvBase DBMS, Rocket U2 Universe, Rocket U2 Unidata, OpenQM, Caché InterSystems, Reality, Jbase, OpenInsight, Extensible storage engine

- Marcas de Bases de Datos NoSQL

Existe una amplia variedad de Bases de Datos NoSQL que podemos utilizar e implementar en nuestros proyectos. Todas ellas cumplen funciones especiales con un excelente rendimiento. Pero debemos analizar nuestros requerimientos y necesidades para saber escoger cual tomaremos en cuenta para nuestro proyecto.

Conozcamos algunas de las BBDD NoSQL más reconocidas en la actualidad:

MongoDB

Construido en C++ tiene un rendimiento increíble que le permite ser muy flexible y potente. Es ideal para proyectos en los que se requiera alto nivel de escalabilidad. Tiene capacidad para recibir miles de lecturas por segundo sin pestañear. Múltiples gestores de contenido y juegos online se benefician de sus propiedades.

Apache Cassandra

Es una base de datos de tipo clave-valor. Cassandra es una BBDD NoSQL está diseñada para almacenar cantidades gigantescas de datos y realizar distribuciones a través de varios nodos. Esto permite que el almacenamiento de datos pueda estar repartido entre diversos servidores sin un solo punto de fallo. La gran mayoría de nosotros tiene una cercanía con esta base de datos ya que es una de las herramientas esenciales de Facebook.

CouchDB

La BBDD CouchDB llega a nosotros con la intención de facilitar la accesibilidad y compatibilidad web con diferentes tipos de dispositivos. Su primera versión estable llegó al público en el año 2010. En esta BBDD los datos se incluyen en formato JSON. La forma en la que los datos son organizados es según pares de valor clave.

Información editorial proporcionada por DB-Engines		
Nombre	Cassandra	CouchDB
Descripción	Tienda de columna ancha basada en ideas de BigTable y DynamoDB	Un JSON nativo: almacén de documentos inspirado en Lotus Notes, escalable desde clústeres de servidores distribuidos globalmente hasta teléfonos móviles.
Modelo de base de datos primaria	Tienda de columna ancha	Almacén de documentos
Ranking de motores DB Gráfico de tendencia	Puntuación 118,75 Rango # 10 En general # 1 Almacenes de columnas anchas	Puntuación 17,25 Rango # 35 En general # 5 Almacenes de documentos
Sitio web	cassandra.apache.org	couchdb.apache.org
Documentación técnica	cassandra.apache.org/doc/latest	docs.couchdb.org/en/stable
Desarrollador	Fundación de software Apache	Fundación de software Apache
Versión inicial	2008	2005
Lanzamiento actual	3.11.8, agosto de 2020	3.0, febrero de 2020
Licencia	Fuente abierta	Fuente abierta
Solo basado en la nube	No	No
Ofertas de DBaaS (enlaces patrocinados)		
Lenguaje de implementación	Java	Erlang
Sistemas operativos de servidor	BSD Linux OS X Windows	Android BSD Linux OS X Solaris Windows

Redis

Otro importante motor de base de datos de tipo calve-valor es Redis. Esta BBDD NoSQL de Código abierto es patrocinada y desarrollada por RedisLabs. Su diseño principal está basado en el almacenamiento de tablas de hashes aunque no es restrictiva sólo hacia este modelo. También tiene la posibilidad de ser utilizada como una BBDD persistente.

Neo4j

Las características principales que ofrece Neo4J a sus desarrolladores es un alto desempeño, si las consultas empiezan a crecer de forma exponencial el rendimiento de esta BBDD no disminuye. Es Ágil, flexible y altamente escalable. De sus usos y posibilidades de implementación te hablaremos con seguridad más adelante.

ESTRUCTURA DE DATOS

Cassandra

Una columna es la estructura de datos más básica en el modelo de datos de Cassandra. Una columna contiene un nombre, un valor y un reloj, el cual puedes tomarlo como una marca horaria por ahora. Aunque estemos familiarizados con el término “columna” desde el mundo relacional, no funciona igual

en Cassandra. Primero que todo, cuando diseñamos una base de datos relacional, primero especificas la estructura de las tablas y el nombre de sus columnas, más adelante escribes los datos sobre la estructura ya definida.

En Cassandra, no tienes que definir las columnas que necesitas. Solamente define las familias de columnas que deseas en el espacio clave, y después escribir los datos sin tener definidas las

columnas. Esto es posible porque en Cassandra, todas las columnas son proporcionadas por el cliente.

Los tipos de dato para los nombres y valores de una columna son arreglos de bytes de java, frecuentemente proporcionados como cadenas. Como el nombre y el valor son de tipo binario, pueden tener cualquier longitud. El tipo de dato definido para el reloj es un IClock, propio de Cassandra.









Column		
name: byte[]	value: byte[]	clock: IClock

















MongoDB

A diferencia de las bases de datos tradicionales, donde tenemos que crear la estructura de nuestra base de datos con las columnas correspondientes y los tipo de datos etc, para posteriormente insertar datos; en mongoDB nada de esto es necesario ya que al ser orientado a documentos mongoDB no necesita crear la estructura, ya que en vez de tener "tablas", mongoDB tiene "colecciones". En lugar de guardar los datos en tablas como se hace en las base de datos relacionales, MongoDB guarda estructuras de datos en documentos similares a JSON con un esquema dinámico (MongoDB utiliza una especificación llamada BSON), haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

Como MongoDB no tiene una estructura de datos definida, cada documento puede tener una estructura diferente es así que por ejemplo en nuestra colección de Libros:

Un libro puede tener ciertos atributos como "autor, título, descripción" y otro puede tener un atributo adicional "autor, título, descripción, categorías"

Esquema de datos	sin esquema y esquema opcional	sin esquema
Mecanografía 	si	parcial 
Soporte XML 		No
Índices secundarios	si 	si 
SQL 	No	No
API y otros métodos de acceso	Protocolo Bolt Lenguaje de consulta cifrado API Java Neo4j-CGM RESTful HTTP API Spring Data Neo4j TinkerPop 	protocolo propietario 
Lenguajes de programación compatibles	.Net Clojure Elixir Go Groovy Haskell Java JavaScript Perl PHP Python Ruby Scala	C C # C ++ Clojure Crystal D Dart Elixir Erlang Fancy Go Haskell Haxe Java JavaScript (Node.js)

Información editorial proporcionada por DB-Engines		
Nombre	Neo4j 	Redis 
Descripción	Base de datos de gráficos de código abierto	Almacén de estructura de datos en memoria, utilizado como base de datos, caché y agente de mensajes 
Modelo de base de datos primaria	Gráfico DBMS	Almacén de valor clave 
Modelos de bases de datos secundarias		Almacén de documentos DBMS de gráficos Motor de búsqueda DBMS de series temporales   
Ranking de motores DB  Gráfico de tendencia 	Puntuación: 52,53 Rango: # 20 En general # 1 Gráfico DBMS	Puntuación: 155,42 Rango: # 7 En general # 1 Tendos de valores clave
Sitio web	neo4j.com	redis.io
Documentación técnica	neo4j.com/docs	redis.io/documentation
Desarrollador	Neo4j, Inc.	Salvatore Sanfilippo 
Versión inicial	2007	2009
Lanzamiento actual	4.1.4, noviembre de 2020	6.0.8, octubre de 2020
Licencia 	Fuente abierta 	Fuente abierta 
Solo basado en la nube 	No	No
Ofertas de DBaaS (unidades patrocinadas) 		ScaleGrid para Redis: alojamiento totalmente administrado para Redis en AWS, Azure y DigitalOcean con alta disponibilidad y acceso SSH en el DBaaS de múltiples nubes número uno.
Lenguaje de implementación	Java, Scala	C
Sistemas operativos de servidor	Linux OS X Solaris Windows 	BSD Linux OS X Windows 

IV. CONCLUSIONES

Las bases de datos NoSQL son ya una opción más en la cartera de alternativas para almacenar los datos de tus aplicaciones. Existen varios tipos de ellas, pero en general su objetivo principal es resolver los problemas de performance y de escalabilidad de las RDBMS.

Muchos desarrolladores al considerar MongoDB vs Redis, no consideran cómo pueden trabajar mano a mano.

MongoDB es una base de datos orientada a documentos y basada en disco optimizada para la simplicidad

operativa, el diseño sin esquemas y volúmenes de datos muy grandes. Redis es un almacén de estructura de datos persistente en memoria que permite a los desarrolladores realizar operaciones comunes con una complejidad mínima y un rendimiento máximo. MongoDB y Redis son lenguajes NoSQL, pero debido a que están

diseñados para diferentes propósitos, las capacidades mejoradas de Redis aumentan significativamente las capacidades de MongoDB.

V. RECOMENDACIONES

- Desarrollar un caso de uso
- Determinar cómo el modelo encaja con las implementaciones de bases de datos existentes
- Reconocemos que será perjudicial
- Comience con algo pequeño
- Identificar los decisores y ejecutores de la organización
- Identificar los procesos que van a inhibir y habilitar la iniciativa
- Determine si tiene los conocimientos adecuados
- Indicador de comprensión de la NoSQL DBMS y lenguajes de consulta
- Establecer un sentido de urgencia
- Apóyese en los que tienen experiencia, incluyendo compañeros y consultores

VI. BIBLIOGRAFÍA

- <https://www.grapheverywhere.com/bases-de-datos-nosql-marcas-tipos-ventajas/>
- <https://db-engines.com/en/system/Neo4j%3BRedis>
- <https://db-engines.com/en/system/Cassandra%3BCouchDB>

