«Comparación de sistemas y modelos gestión de bases de datos NoSQL» por Mark Drake (versión previa O. S. Tezer)

- **En castellano:** Este artículo es una traducción del inglés al castellano, publicado bajo licencia (en idioma inglés) «Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) » escrito por written by Mark Drake (previous version by O. S. Tezer @Ostezer), publicado en línea por la empresa de alojamiento de máquinas virtuales DigitalOcean. El título original en idioma inglés es «***A Comparison of NoSQL Database Management Systems and Models***» y hemos creado una copia en Wayback Machine para prevenir un posible enlace roto a futuro.

Introducción

Cuando la mayoría de las personas piensa en una base de datos, a menudo imaginan el modelo tradicional de base de datos relacional que involucra tablas compuestas de filas y columnas. Si bien los sistemas de administración de bases de datos relacionales aún manejan la mayor parte de los datos en Internet, los modelos de datos alternativos se han vuelto más comunes en los últimos años a medida que los desarrolladores han buscado soluciones a las limitaciones del modelo relacional. Estos modelos de bases de datos no relacionales, cada uno con sus propias ventajas, desventajas y singularidades en su utilización, han sido categorizados como *bases de datos NoSQL*.

Este artículo le presentará algunos de los modelos de bases de datos NoSQL más utilizados. Sopesará algunas de sus fortalezas y desventajas, y proporcionará algunos ejemplos de sistemas de gestión de bases de datos y los potenciales usos para cada uno de los casos.

NoSQL (Wikipedia https://commons.wikimedia.org/wiki/File:Nosql.gif)

Bases de datos relacionales y sus limitaciones

Las bases de datos son racimos de información modelados lógicamente o, simplemente, *datos*. Un *sistema de gestión de bases de datos* (SGBD o, en idioma inglés DBMS), por su parte, es un programa informático que interactúa con una base de datos. Un SGBD le permite controlar el acceso a una base de datos, escribir datos, ejecutar consultas y realizar cualquier otra tarea relacionada con la administración de la base de datos. Aunque los sistemas de gestión de bases de datos a menudo se denominan simplemente "bases de datos", los dos términos no son exactamente intercambiables. Una base de datos puede ser cualquier colección de datos, una o más guardadas en una computadora, mientras que un SGBD es el software específico que le permite interactuar con una base de datos.

Todos los sistemas de gestión de bases de datos tienen un modelo subyacente que estructura cómo se almacenan y acceden los datos. Un *sistema de gestión de bases de datos relacionales* (SGBDR) es un SGBD que emplea el modelo de datos relacionales. En este modelo, los datos se organizan en tablas, las cuales formalmente se denominan relaciones en el contexto de los SGBDR. Los sistemas de gestión de bases de datos relacionales suelen emplear el *Lenguaje de Consulta Estructurado* o LCE (en idioma inglés, *SQL* el cual emplearemos frecuentemente) para gestionar y acceder a los datos contenidos en la base de datos.

El modelo relacional, históricamente, ha sido el enfoque más utilizado para administrar datos, y hasta el día de hoy muchos de los sistemas de administración de bases de datos más populares implementan el modelo relacional. Sin embargo, el modelo relacional presenta varias limitaciones que pueden ser problemáticas en ciertos casos.

Por ejemplo, puede ser difícil potenciar una base de datos relacional horizontalmente. La *potenciación horizontal* o potenciado horizontal (frecuentemente traducido como "escalado horizontal" N. del T.), es la práctica de agregar más máquinas a una batería de servidores existente para distribuir la carga y permitir más tráfico y un procesamiento más rápido. Esto a menudo se contrasta con el potenciado vertical que implica la actualización del hardware de un servidor existente, generalmente agregando más RAM o CPU.

La razón por la que es difícil potenciar una base de datos relacional horizontalmente tiene que ver con el hecho de que el modelo relacional está diseñado para garantizar la *coherencia*, lo que significa que los clientes que consulten la misma base de datos siempre verán los últimos datos, con los mismos resultados. Si tuviera que potenciar una base de datos relacional de manera horizontal en varias máquinas, sería difícil garantizar la coherencia, ya que los clientes pueden escribir datos en un nodo y no en los demás, y es probable que haya un retraso entre la escritura inicial y el momento en que los otros nodos actualizado para reflejar los cambios.

Otra limitación presentada por los SGBDR es que el modelo relacional fue diseñado para

administrar datos estructurados, o datos que se alinean con un tipo de datos predefinido o que al menos están organizados de alguna manera predeterminada, lo que la hace fácilmente clasificable y por ende facilitando su búsqueda. Sin embargo, con la expansión de la informática personal y el auge de Internet a principios de la década de 1990, los datos no estructurados (mensajes de correo electrónico, fotos, videos, etcétera) se hicieron más comunes.

A medida que estas limitaciones se hicieron más restrictivas, los desarrolladores comenzaron a buscar alternativas al modelo tradicional de datos relacionales, lo que llevó al crecimiento en popularidad de las bases de datos NoSQL.

Acerca de NoSQL

La etiqueta NoSQL tiene una definición bastante difusa. "NoSQL" fue acuñado en 1998 por Carlo Strozzi como el nombre de su nueva base de datos NoSQL, elegida simplemente porque no usa SQL para administrar datos.

El término adquirió un nuevo significado después de 2009 cuando Johan Oskarsson organizó una reunión para desarrolladores para discutir la difusión de "bases de datos de código abierto, distribuidas y no relacionales" como Cassandra y Voldemort. Oskarsson nombró la reunión "NOSQL" y desde entonces el término se ha utilizado como un término general para cualquier base de datos que no emplea el modelo relacional. Curiosamente, la base de datos NoSQL de Strozzi de hecho emplea el modelo relacional, lo que significa que la base de datos NoSQL original no se ajusta a la definición contemporánea de NoSQL.

Debido a que "NoSQL" generalmente se refiere a cualquier SGBD que no emplea el modelo relacional, existen varios modelos de datos operativos asociados con el concepto NoSQL. La siguiente tabla incluye varios de estos modelos de datos, pero tenga en cuenta que esta no es una lista completa:

Modelo operacional BD	Ejemplo
Orientado a clave-valor	Redis, MemcacheDB
Orientado a columnas	Cassandra, Apache HBase
Orientado a documentos	MongoDB, Couchbase
Orientado a grafos	OrientDB, Neo4j

A pesar de estos diferentes modelos de datos subyacentes, la mayoría de las bases de datos NoSQL comparten varias características. Por un lado, las bases de datos NoSQL generalmente están diseñadas para maximizar la disponibilidad a expensas de la coherencia. En este sentido, la

coherencia se refiere a la idea de que cualquier operación de lectura devolverá los datos más recientes escritos en la base de datos. En una base de datos distribuida diseñada para una fuerte consistencia, cualquier dato escrito en un nodo estará inmediatamente disponible en todos los demás nodos; de lo contrario, se producirá un error.

Por el contrario, las bases de datos NoSQL a menudo apuntan a una consistencia eventual. Esto significa que los datos recién escritos están disponibles en otros nodos en la base de datos eventualmente (generalmente en cuestión de unos pocos milisegundos), aunque no necesariamente de manera inmediata. Esto tiene el beneficio de mejorar la disponibilidad de los datos: aunque no vea los últimos datos escritos, puede ver una versión anterior en lugar de recibir un error.

Las bases de datos relacionales están diseñadas para manejar datos normalizados que se ajustan perfectamente a un esquema predefinido. En el contexto de un DBMS, los *datos normalizados* son datos que se han organizado de manera de eliminar redundancias, lo que significa que la base de datos ocupa el menor espacio de almacenamiento posible, mientras que un *esquema* es un resumen de cómo se estructuran los datos en la base de datos.

Si bien las bases de datos NoSQL están equipadas para manejar datos normalizados y pueden clasificar datos dentro de un esquema predefinido, sus respectivos modelos de datos generalmente permiten una flexibilidad mucho mayor que la estructura rígida impuesta por las bases de datos relacionales. Debido a esto, las bases de datos NoSQL tienen la reputación de ser una mejor opción para almacenar datos semiestructurados y no estructurados. Sin embargo, teniendo esto en cuenta, dado que las bases de datos NoSQL no vienen con un esquema predefinido que a menudo significa que depende del administrador de la base de datos definir cómo se deben organizar y acceder a los datos de la manera que tenga más sentido para su aplicación.

Ahora que usted ha entrado en contexto sobre qué son las bases de datos NoSQL y qué las hace diferentes de las bases de datos relacionales, echemos un vistazo más de cerca a algunos de los modelos de bases de datos NoSQL más ampliamente implementados.

Bases de datos orientadas a clave-valor

Las *bases de datos de clave-valor*, también conocidas como *almacenes de valores clave*, funcionan almacenando y gestionando matrices asociativas. Una matriz asociativa, también conocida como diccionario o tabla *hash*, consiste en una colección de pares clave-valor en los que una clave sirve como un identificador único para recuperar un valor asociado. Los valores pueden ser desde objetos simples, como enteros o cadenas, hasta objetos más complejos, como estructuras JSON.

A diferencia de las bases de datos relacionales, que definen una estructura de datos compuesta por tablas de filas y columnas con tipos de datos predefinidos, las bases de datos de valores clave almacenan datos como una sola colección sin ninguna estructura o relación. Después de conectarse al servidor de la base de datos, una aplicación puede definir una clave (por ejemplo, el_significado_de_la_vida) y proporcionar un valor coincidente (por ejemplo, 42) que luego se puede recuperar de la misma manera al proporcionar la clave. Una base de datos de valores clave trata los datos que contiene como un **Fichero de Basura Binaria** *(*en idioma inglés *Opaque binary blob* u OBB); depende de la aplicación entender cómo está estructurada.

Las bases de datos de claves-valores a menudo se describen como de alto rendimiento, eficientes y fácil de potenciar. Los casos de uso comunes para las bases de datos de valores clave son el almacenamiento en caché, la cola de mensajes y la administración de sesiones.

Algunos almacenes de datos clave-valor de código abierto populares son:

Base de datos	Descripción
Redis	Un almacén de datos en memoria utilizado como base de datos, caché o intermediario de mensajes. Redis admite una variedad de estructuras de datos, que van desde cadenas hasta mapas de bits, flujos e índices espaciales.
Memcached	Un sistema de almacenamiento en caché de objetos de memoria de uso general utilizado con frecuencia para acelerar sitios web y aplicaciones basados en datos almacenando en caché datos y objetos en la memoria.
Riak	Una base de datos distribuida de clave-valor con réplica local avanzada y multi-racimo.

Bases de datos orientadas a columnas

Las bases de datos en columnas, a veces llamadas bases de datos orientadas a columnas, son sistemas de bases de datos que almacenan los datos en forma de columnas. Esto puede parecer similar a las bases de datos relacionales tradicionales, pero en lugar de agrupar las columnas en tablas, cada columna se almacena en un archivo o región por separado en el almacenamiento del sistema.

Los datos almacenados en una base de datos de columnas aparecen en el mismo orden de registro, lo que significa que la primera entrada en una columna está relacionada con la primera entrada en otras columnas. Este diseño permite que las consultas solo lean las columnas que necesitan, en lugar de tener que leer cada fila de una tabla y descartar los datos innecesarios después de que se hayan almacenado en la memoria.

Debido a que los datos en cada columna son del mismo tipo, permite varias estrategias de optimización de almacenamiento y lectura. En particular, muchos administradores de bases de datos en columnas implementan una estrategia de compresión como la codificación de longitud de ejecución para minimizar la cantidad de espacio ocupado por una sola columna. Esto puede tener la ventaja de acelerar las lecturas, ya que las consultas deben pasar por menos filas. Sin embargo, un inconveniente con las bases de datos en columnas es que el rendimiento de la carga tiende a ser lento ya que cada columna debe escribirse por separado y los datos a menudo se mantienen comprimidos. Las cargas incrementales en particular, así como las lecturas de registros individuales, pueden ser costosas en términos de rendimiento.

Las bases de datos orientadas a columnas existen desde la década de 1960. Sin embargo, desde mediados de la década de 2000, las bases de datos en columnas se han vuelto más utilizadas para el análisis de datos, ya que el modelo de datos en columna se presta bien para el procesamiento rápido de consultas. También se consideran ventajosos en los casos en que una aplicación necesita realizar funciones agregadas con frecuencia, como encontrar el promedio o la suma total de datos en una columna. Algunos sistemas de gestión de bases de datos en columnas son incluso capaces de usar consultas SQL.

Algunas bases de datos orientados a columnas que son populares en la actualidad:

Base de datos	Descripción
Apache Cassandra	Diseñada para potenciar fácilmente la disponibilidad y el desempeño.
Apache HBase	Una base de datos distribuida que apoya el almacenamiento estructurado para grandes cantidades de datos y diseñada para trabajar con las librerías de Hadoop.
ClickHouse	Un SGBD con tolerancia a fallos que apoya la generación de datos analíticos en tiempo real además de consultas SQL.

Bases de datos orientadas a documentos

Las bases de datos orientadas a documentos, o almacenes de documentos, son bases de datos NoSQL que almacenan datos en forma de documentos. Los almacenes de documentos son un tipo de almacén de valores clave: cada documento tiene un identificador único, su clave, y el documento mismo sirve como valor.

La diferencia entre estos dos modelos es que, en una base de datos de valores clave, los datos se tratan como opacos y la base de datos no conoce ni se preocupa por los datos que contiene; depende de la aplicación entender qué datos se almacenan. Sin embargo, en un almacén de documentos, cada documento contiene algún tipo de metadatos que proporcionan un grado de estructura a los datos. Los almacenes de documentos a menudo vienen con una API o lenguaje de consulta que permite a los usuarios recuperar documentos basados en los metadatos que contienen. También permiten estructuras de datos complejas, ya que puede anidar documentos dentro de otros documentos.

A diferencia de las bases de datos relacionales, en las cuales la información de un objeto dado puede extenderse a través de múltiples tablas o bases de datos, una base de datos orientada a documentos puede almacenar todos los datos de un objeto dado en un solo documento. Los almacenes de documentos suelen almacenar datos como documentos JSON, BSON, XML o YAML, y algunos pueden almacenar formatos binarios como documentos PDF. Algunos usan una variante de SQL, búsqueda de texto completo o su propio lenguaje de consulta nativo para la recuperación de datos, y otros presentan más de un método de consulta.

Las bases de datos orientadas a documentos han experimentado un enorme crecimiento en popularidad en los últimos años. Gracias a su esquema flexible, han encontrado un uso regular en plataformas de comercio electrónico, blogs y en plataformas de análisis, así como en sistemas de gestión de contenido. Los almacenes de documentos se consideran altamente escalables, y el fragmentación es una estrategia de escala horizontal común. También son excelentes para mantener grandes cantidades de información compleja y no relacionada que varía en estructura.

Algunas bases de datos populares, de código abierto, orientadas a documentos son:

Base de datos	Descripción
MongoDB	De propósito general, y de almacenamiento de documentos de manera distribuida, MongoDB es la base de datos de mayor uso en este campo al momento de escribir estas líneas.
Couchbase	Conocida originalmente como Membase, está especializad en almacenar JSON pero en modo de *multimodelo*, Couchbase también puede funcionar como orientada

	a clave-valor.
Apache CouchDB	Un proyecto de la Fundación Apache Software, también almacena datos como documentos JSON y utiliza el lenguaje JavaScript para las consultas.

Bases de datos orientadas a grafos

Las bases de datos orientadas a grafos pueden considerarse una subcategoría del modelo de las orientadas a documentos, ya que almacenan datos en documentos y no insisten en que los datos se adhieran a un esquema predefinido. Sin embargo, la diferencia es que las bases de datos de grafos agregan una capa adicional al modelo de documento al resaltar las relaciones entre documentos individuales.

Para comprender mejor el concepto de bases de datos de gráficos, es importante comprender los siguientes términos:

- **Nodo** o **vértice**: Un *vértice* es una representación de una entidad individual rastreada por una base de datos de grafos. Es más o menos equivalente al concepto de un *registro* o *fila* en una base de datos relacional o un *documento* en un almacén de documentos. Por ejemplo, en una base de datos de grafos de artistas de grabación de música, un nodo puede representar un solo intérprete o banda.
- **Propiedad** o **arista**: Una *arista* es información relevante relacionada con vértices individuales. Sobre la base de nuestro ejemplo de artista de grabación, algunas propiedades pueden ser "vocalista", "jazz" o "artista vendedor de platino", según la información que sea relevante para la base de datos.
- **Límite** o **relación binaria**: También conocido como gráfico o relación, una relación binaria es la representación acerca de cómo se relacionan dos vértices, y es un concepto clave de las bases de datos de grafos que los diferencia de los SGBDR y las bases de datos orientadas a documentos. Los *límites* pueden ser dirigidos o no dirigidos.
    - **Indirecto**: en un grafo indirecto, las realciones binarias entre los vértices existen solo para mostrar una conexión entre ellos. En este caso, las relaciones binarias pueden considerarse relaciones "bidireccionales": no existe una diferencia implícita entre cómo un vértice se relaciona con el otro.
    - **Directo**: En un grafo directo, las relaciones binarias pueden tener diferentes significados en función de la dirección desde la que se origina la relación. En este caso, los límites son relaciones "unidireccionales". Por ejemplo, una base de datos de grafos directa podría especificar una relación de Sammy (persona) a "The

Seaweeds" (banda) que muestre que Sammy produjo un álbum para el grupo, pero podría no mostrar una relación equivalente de "The Seaweeds" a Sammy.

Ciertas operaciones son mucho más simples de realizar utilizando bases de datos gráficas debido a cómo vinculan y agrupan piezas de información relacionadas. Estas bases de datos se usan comúnmente en casos en los que es importante poder obtener información de las relaciones entre los puntos de datos o en aplicaciones donde la información disponible para los usuarios finales está determinada por sus conexiones con otros, como en una red social. Han encontrado un uso regular en la detección de fraudes, motores de recomendación y aplicaciones de administración de identidad y acceso.

Algunas bases de datos populares orientadas a grafos son, actualmente:

Base de datos	Descripción
Neo4j	Un SGBD compatible con ACID y con almacenamiento y procesamiento de grafos nativos. Al escribir estas líneas, Neo4j es la base de datos de grafos más popular del mundo.
ArangoDB	No es exclusivamente una base de datos de grafos, ArangoDB es una base de datos multimodelo que une los modelos de datos de grafos, documentos y valores clave en un SGBD. Cuenta con AQL (un lenguaje de consulta similar al SQL nativo), búsqueda de texto completo y un motor de clasificación.
OrientDB	Otra base de datos multimodelo, OrientDB es compatible con los modelos de grafos, documentos, valores-clave y objetos. Admite consultas SQL y transacciones ACID.

Conclusión

En este tutorial, hemos revisado solo algunos de los modelos de datos NoSQL en uso hoy en día. Algunos modelos NoSQL, como las bases de datos orientadas a objetos, han visto niveles variables de uso a lo largo de los años, pero siguen siendo alternativas viables al modelo relacional en algunos casos de uso. Otros, como las bases de datos relacionales de objetos y las bases de datos de series temporales, combinan elementos de modelos de datos relacionales y NoSQL para formar una especie de punto medio entre los dos extremos del espectro.

La categoría de bases de datos NoSQL es extremadamente amplia y continúa evolucionando hasta nuestros días. Si está interesado en aprender más sobre los sistemas y conceptos de gestión de bases de datos NoSQL, le recomendamos que consulte nuestra biblioteca de contenido relacionado con NoSQL (en el sitio web de DigitalOcean).

Escrito por: Mark Drake (versión anterior O. S. Tezer)

Traducido por: Jimmy Olano

», written by Mark Drake (previous version by O. S. Tezer @Ostezer), published on line by the company for leasing virtual machines DigitalOcean. The tittle is «***A Comparison of NoSQL Database Management Systems and Models***» and we created a copy at Wayback Machine for prevent in future a broken link. This work is licensed under the mencioned license but, of course, in castilian language (AKA *spanish*): «Atribución-NoComercial-CompartirIgual 4.0 Internacional (CC BY-NC-SA 4.0) ».

- Esta obra está bajo una Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional.
- **En castellano:** Este artículo es una traducción del inglés al castellano, publicado bajo licencia (en idioma inglés) «Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) *»* escrito por written by Mark Drake (previous version by O. S. Tezer @Ostezer), publicado en línea por la empresa de alojamiento de máquinas virtuales DigitalOcean. El título original en idioma inglés es «***A Comparison of NoSQL Database Management Systems and Models***» y hemos creado una copia en Wayback Machine para prevenir un posible enlace roto a futuro.