UNIVERSIDAD TÉCNOLOGICA NACIONAL
FACULTAD REGIONAL GENERAL PACHECO

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN

Base de Datos II

Introducción al Paradigma NoSQL

ING. ARIEL HERRERA

Contenido

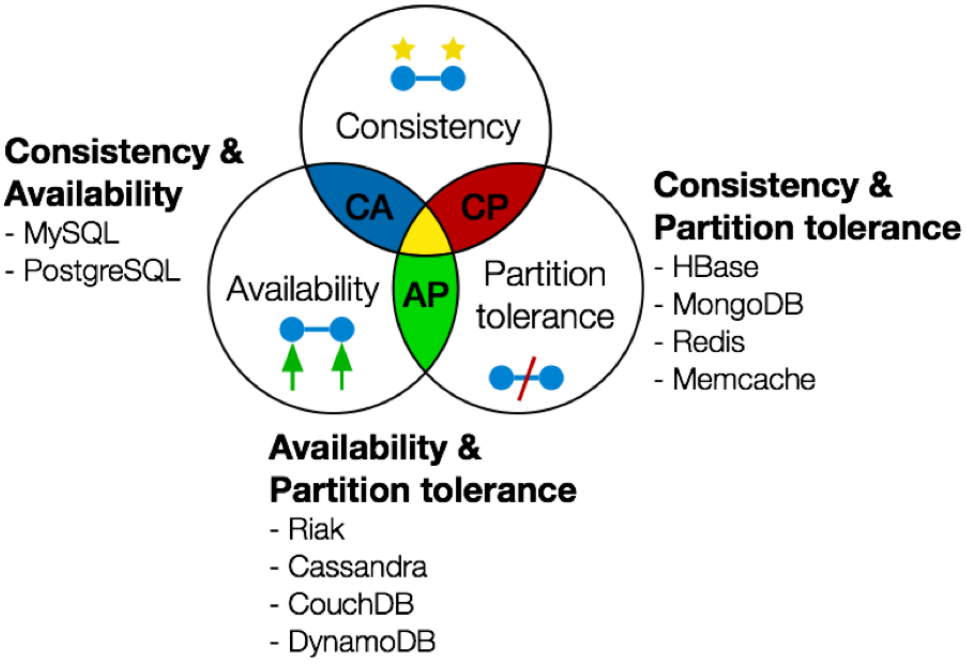
NoSQL

1. El teorema de CAP

Según el teorema de Brewer, es imposible para un sistema computacional distribuido, como puede ser el caso de una base de datos, ofrecer simultáneamente las siguientes tres garantías:

1. Consistencia (consistency): todos los usuarios ven los mismos datos al mismo tiempo.
2. Disponibilidad (availability): garantiza que cada petición de acceso o escritura de datos recibe una respuesta en un tiempo limitado.
3. Tolerancia a la partición (partitioning): la base de datos continúa funcionando aunque se pierdan parte de los datos.

El teorema CAP apunta a que hay que elegir dos de estas opciones, o sea que no es posible contar con las 3 cualidades de manera simultánea. Por lo tanto, se da lugar al siguiente cuadro de decisión:



1) Disponible y resistente a la partición: base de datos no relacionales.
2) Disponible y consistente: bases de datos relacionales.
3) consistente y resistente a la partición: bases de datos no relacionales.

2. Tipos de bases de datos no relacionales

2.1. Arquitectura de las bases de datos NoSQL

Las bases de datos no relacionales suelen ofrecer garantías de consistencia débiles al emplear una arquitectura distribuida.

En este tipo de bases, los datos pueden incluso almacenarse físicamente en servidores distintos usando espacios de memoria disjuntos.

Por este motivo, suelen basarse en estructuras de datos sencillas, tales como arrays asociativos o almacenes de pares de clave-valor.

2.2. Una posible clasificación

según la manera de organizar y almacenar los datos, las bases de datos no relacionales pueden ser de varios tipos

1) Bases de datos como almacenes de clave-valor.
2) Bases de datos como familia de columnas.
3) Bases de datos como almacenes de documentos.
4) Bases de datos como almacenes de grafos.

A continuación, detallamos brevemente cada 1 de estos tipos de bases de datos no relacionales.

Bases de datos clave y multivalor

El precursor de este tipo de base de datos fue Amazon dynamo, basado en DHT (distributed hash tables).

En este caso los datos se almacenan como **pares clave-valor.**

La clave es un valor que identifica a cada registro, y los valores son vistos como cajas negras por la base de datos.

En este tipo de base de datos, las consultas se llevan a cabo por la clave.

Ejemplos de este tipo de bases de datos son Dynamic coma a Voldemort o Tokyo.

Bases de datos Orientadas a columnas

El precursor de este tipo de base de datos es Google BigTable.
Se trata de un conjunto de columnas con distinta información. Cada entrada en la "tabla" puede contener información de columnas diferentes. De algún modo, podemos decir que cada "clave" está asociada con varios "valores" o "atributos" que se corresponden a diferentes columnas en la estructura.
Estas bases de datos son idóneas para:

1) Almacenar grandes cantidades de datos.
2) Permitir cargas masivas de datos.
3) Garantizar alta disponibilidad.

Algunos ejemplos de este tipo de bases de datos son HBase y Hypertable Cassandra o Riak.

Bases de datos Documentales

El precursor de este tipo de base de datos fue Lotus Notes.
En este caso, los datos son colecciones de documentos que contienen colecciones de pares de clave-valor. En general, el concepto de "documento" es utilizado para encapsular y codificar datos o información siguiendo algún formato estándar, como por ejemplo XML y JSON. El formato JSON (javascript object notation) es un estándar muy popular hoy día; se trata de un estándar abierto, basado en texto diseñado para intercambio de datos legible por humanos, que permite representar estructuras de datos simples y listas asociativas.
Por este motivo, se suele decir que este tipo de bases de datos guardan datos semiestructurados.
Los documentos en una base de datos orientada a documentos son similares a registros, pero no requieren un esquema estándar con las mismas secciones, partes, claves, etc.
Los documentos suelen ser direccionable por una clave que los representa de manera única.
Además de la búsqueda por clave de documento, estas bases de datos suelen ofrecer un lenguaje de consultas que permite recuperar documentos a partir de sus contenidos.
Este tipo de bases de datos gozan de las siguientes ventajas:

1) Son un modelo de datos muy natural.

2) Son amigables para el programador.
3) Permiten desarrollo de soluciones muy rápidas.
4) Están muy orientadas a la integración con la web.

Ejemplo de este tipo de bases de datos son CouchDB o la popular MongoDB.

Bases de datos basadas en grafos

Este tipo de bases de datos están inspiradas en la teoría de grafos de Euler. Estas bases de datos organizan la información basándose en la estructura de un grafo, es decir, como un conjunto de nodos y las relaciones entre pares de nodos. Estas bases de datos permiten emplear la teoría de grafos para recorrer la base de datos y hacer consultas sobre los datos.
Ejemplos de este tipo de base de datos son AllegroGraoph, VertexBD y Neo4J.

3. Ejemplos

Por ejemplo, **Instagram** utiliza una base de datos **clave-valor** conocida como **Redis**, para guardar las sesiones de usuario y para almacenar a los usuarios que han subido cada foto. Redis es una base de datos apoyada por VMWare. Se trata de una base de datos tipo clave-valor que se puede imaginar como un bloque de memoria reservada para almacenar datos, datos encadenados, cadenas encriptadas de datos o listas, por ejemplo.
Dentro de las bases de datos NoSQL orientadas a columnas destacan Hbase y Cassandra.

HBase es un sistema gestor de bases de datos agregado en columnas que ofrece alta disponibilidad, se ejecuta sobre HDFS (Hadoop Distributed File System), diseñado para almacenar y gestionar grandes conjuntos de datos y está totalmente integrado con Hadoop.

Cassandra también ofrece alta disponibilidad rápido acceso a cantidades de datos y una alta tolerancia a fallos. Se trata de una solución multiplataforma escrita en Java, que goza de popularidad por haber sido inicialmente adoptada por Facebook. Las últimas versiones incluyen un propio lenguaje de consulta

llamado **CQL** (Cassandra query language), que posee una sintaxis similar a SQL, aunque con menos funcionalidades. Los dos sistemas permiten el almacenamiento de tablas de gran tamaño, es decir, tablas de miles de millones de filas por millones de columnas, en un entorno distribuido.
Aplicaciones como Facebook o Twitter usan **Cassandra** como almacén de datos, aunque en el caso de Facebook, por ejemplo, su estructura de datos es más compleja, ya que combina diferentes situaciones de datos para distintas funcionalidades de la aplicación de red social y mensajería.

MongoDB es un sistema gestor de bases de datos documental de código abierto. Pretende combinar lo mejor de los almacenes de clave-valor, las bases de datos documentales y las tradicionales bases de datos relacionales. MongoDB usa JSON y tiene un propio lenguaje para hacer consultas. El hecho de estar programado en C++ hace que tenga muy buena acogida por parte de los desarrolladores. Se trata de la base de datos usada por SourceForge, Bit.ly, Foursquare o GitHub entre otros. MongoDB es un sistema de datos NoSQL orientado a documentos; guarda estructuras de datos en documentos de tipo BSON (JSON binario), lo que hace que la integración de datos y el acceso y la edición de los mismos sean realmente muy ágiles y rápidos.

Neo4J es un sistema gestor de bases de datos en grafo de código abierto. Almacena los datos en grafos de propiedades. En estos tipos de grafos, la información se representa mediante nodos, relaciones entre nodos, etiquetas y propiedades. Los nodos permiten representar datos concretos del mundo real; las relaciones hacen posible representar interrelaciones entre nodos; las etiquetas permiten asignar nombres a los nodos y relaciones; y las propiedades son parejas clave valor que pueden asignarse tanto a los nodos como a las relaciones. A diferencia de las otras bases de datos comentadas anteriormente, la distribución de los datos en las bases de datos en grafo es problemática y compleja. Compañías como eBay o Walmart utilizan Neo4J en sus soluciones empresariales. Otras, como por ejemplo Google, han desarrollado una base de datos en grafo propia llamada Pregel, para relacionar páginas web.

Conclusiones

cómo hemos visto, existen ventajas e inconvenientes respecto a las bases de datos relacionales y no relacionales. No hay una solución para todo; en algunos casos es necesario seleccionar entre una opción y otra, y adoptar soluciones de compromiso; en otros casos, es posible que diferentes servicios deban usar esquemas de datos diferentes, combinando distintas soluciones y aplicando a bases de datos relacionales y no relacionales combinadas para dar una solución completa.

En general en la base de datos no es SQL los datos se pueden recuperar más rápido que en las bases de datos RDBMS (Relational Database Management System). Sin embargo, las consultas que se pueden hacer son más limitadas, de modo que se necesita trasladar la complejidad del tratamiento de los datos a un ámbito de aplicación. Es decir, si se desea aplicar técnicas de Big Data o machine learning, por ejemplo, en el caso de las bases de datos no relacionales, Estas técnicas deben aplicarse en un ámbito de aplicación, y no de base de datos. Las bases de datos relacionales, por su parte, permiten aplicar inteligencia en el método en el que se consulta sus datos.

Por lo tanto, el uso de base de datos no es SQL no es recomendable para aplicaciones que generan informes que usan consultas complejas.

Cuando los datos son muy estructurados, por lo tanto, la base de datos relacionales siguen siendo una alternativa muy interesante.

Como conclusión final, podemos decir que, en un ámbito de sistema, de aplicación extremo a extremo, la combinación de SQLY no SQL podría ser la solución ideal.

Resumen

En este material, hemos visto las limitaciones de las bases de datos relacionales. La gran estructuración de los datos en la base de datos relacionales impone limitaciones para muchas de las aplicaciones que serán posibles en un mundo hiper conectado al que nos acercamos, precursor de la industria 4.0.

Como solución, aparecen las bases de datos no relacionales, que ofrecen mucha

más flexibilidad y escalabilidad, algo que, por lo tanto, las hace muy interesantes para uso en las aplicaciones de la industria 4.0. Su uso viene motivado también por la generalización de la computación en la nube y la computación distribuida.
Hemos visto que hay diferentes tipos de bases de datos no relacionales y hemos descrito, en cada caso, las principales ventajas y desventajas de cada tipo de base de datos:

1) Bases de datos clave y multivalor.
2) Bases de datos documentales.
3) Bases de datos basadas en grafos.
4) Bases de datos orientadas a columnas.

Finalmente, hemos descrito algunos ejemplos específicos reales de empresas y productos que se basan en el uso de bases de datos no relacionales, en contraposición con las bases de datos relacionales tradicionales.
A lo largo del material, hemos valorado pros y contras de las bases de datos relacionales y no relacionales. Como conclusión final podemos decir que, en un ámbito de sistema, hora de aplicación extremo a extremo, la combinación de SQL y no es SQL podría ser la solución ideal. Las dos opciones tienen puntos fuertes y debilidades; lo más aconsejable, por tanto, es combinar soluciones para obtener lo mejor de las dos alternativas.

Bases de datos no relacionales, Jesús Alonso-Zárate, Jordi Casas Roma. Universitat Oberta de Catalunya