



NoSQL, la nueva tendencia en el manejo de datos

NoSQL, the New Trend in Data Management

Harol Andrey Herrera¹ Carlos Rueda Valenzuela²

Para citar este artículo: Herrera, H. A.; Rueda, C. (2015). NoSQL, la nueva tendencia en el manejo de datos, 3(2), [147-150].

Resumen

Debido a la necesidad de gestionar grandes volúmenes de información generada por la alta demanda producida por la evolución de las aplicaciones web, se ha presentado una nueva tendencia en el manejo de bases de datos conocida como NoSQL, que ofrece una mayor libertad en la definición de esquemas, ya que la responsabilidad recae sobre la aplicación y no sobre el motor de bases de datos. Adicionalmente fortalece dos requerimientos importantes en una aplicación como son la escalabilidad y la disponibilidad. Esto se puede evidenciar en aplicaciones como las redes sociales, que presentan gran dinamismo y rendimiento. De acuerdo a su forma de almacenar la información, se clasifican en: llave valor, familias de columnas, almacenes de documentos y de grafos.

Palabras clave: NoSQL, almacén de llave valor, almacén de documentos, familia de columnas, base de datos de grafos.

Abstract

Because of need to manage large volumes of data generated by high demand caused by evolution of web applications, there has been a new trend in management of databases known as NoSQL, which provides greater freedom in the schema definition, since onus is on application and not database engine. It also strengthens two important requirements in an application such as are scalability and availability. This evidences in applications such as social networks, which have great dynamism and performance. According to their way of storing information, they are classified as key-value, column families, documents and graphs stores.

Keywords: NoSQL, key-value store, document store, column family, graph database.

ARTÍCULO CORTO

Fecha de recepción:
22-05-2015

Fecha de aceptación:
17-11-2015

ISSN: 2344-8288

Vol. 4 No. 1

Enero - Junio 2016

Bogotá-Colombia

¹ Ingeniero de Sistemas, Universidad Cooperativa de Colombia. Correo electrónico: herrera.andrey.hh@gmail.com

² Ingeniero de Sistemas, Universidad Cooperativa de Colombia. Correo electrónico: carlosnake.182@gmail.com

INTRODUCCIÓN

NoSQL es una tendencia que surge a partir de las aplicaciones Web 2.0, del dinamismo y la alta demanda que estas requieren. Son aplicaciones que tienen requerimientos que exceden la capacidad de las bases de datos relacionales.

En el pasado las bases de datos SQL eran utilizadas en casi todos los proyectos que exigían persistencia de datos. Sin embargo, los estrictos esquemas relacionales pueden ser una carga para las aplicaciones, como es el caso de los blogs, que cuentan con gran variedad de tipos de atributos como son textos, imágenes, videos, código fuente y otra información que se almacena en múltiples tablas. También el alto volumen de datos en las páginas web se convierte en un problema cuando la prioridad es la disponibilidad, como es el caso de redes sociales Facebook, Twitter, LinkedIn[5] y otros sitios como Amazon y Google, que además de tratar con Tera y Peta Bytes de datos, su lectura y escritura masiva tiene que responder sin que se pueda percibir su latencia, de acuerdo con cualquier requerimiento.

Aunque muchas compañías buscan atender estos requerimientos usando clústeres con miles de máquinas, las bases de datos relacionales y sus características ACID no son adecuadas en estos dominios, ya que se basan en la consistencia y no en la disponibilidad. Su incansable búsqueda por garantizar la integridad y su sinnúmero de cerraduras tienen una influencia negativa en sistemas distribuidos.

Con la aparición de NoSQL se logró mitigar este inconveniente de disponibilidad y se dio un avance en la agilidad de las aplicaciones web, además de la flexibilidad que ofrecen sus esquemas.

Sitios como Amazon, por su tipo de negocio, deben garantizar una disponibilidad de no menos del 99.9% durante un año, razón por la cual es necesario que las bases de datos sean fácilmente replicadas y cuenten con un mecanismo de conmutación por error para hacer frente a los

fallos en los centros de datos. También deben ser capaces de balancear las peticiones de lectura con sus esclavos para hacer frente a los picos de consumo en los servidores[2].

Modelo de datos

La mayoría de las bases de datos NoSQL cuentan con modelos que difieren bastante de los relacionales. Estos modelos podemos clasificarlos en 4 grupos principales que son:

- Modelo llave-valor
- Modelo de documentos
- Modelo de familia de columnas
- Modelo de grafos

Modelo llave-valor

Son similares a los objetos Dictionary[6] que se manejan en muchos lenguajes de programación, ya que funcionan como colecciones formadas por llaves únicas y valores. Para este modelo, los valores son matrices de bytes no interpretados, completamente desconocidos para el sistema y las claves son la única manera de recuperar los datos almacenados. Los valores se aíslan e independizan, de manera que las relaciones se manejan en la lógica de aplicación. Debido a esta estructura de datos simple, las bases de datos que implementan este modelo son completamente libres de esquemas. Nuevos valores de cualquier tipo pueden ser añadidos en tiempo de ejecución sin entrar en conflicto con otros datos almacenados y sin afectar la disponibilidad del sistema.

Los almacenes de llaves valor son útiles para operaciones sencillas, que se basan solo en los atributos clave. Su objetivo es facilitar el acceso a datos. Algunos de los motores de base de datos NoSQL que implementan este modelo son: Cassandra, Voldemort, Redis y Membase, entre otros.

Modelo de documentos

Estos encapsulan pares de claves y valores, y los agrupa en documentos con formato JSON. No usan SQL como lenguaje de consulta, por lo que recurren a otros lenguajes como UnSQL[7] o incluso JavaScript[8] y de esta manera extraer o manipular los datos dentro de los documentos.

Cada documento contiene una llave especial o Id, que debe ser única dentro de la colección de documentos, ya que es la que identifica a un documento.

En contraste con los modelos de llave-valor, los documentos permiten almacenar agrupaciones de estructuras, llaves únicas e índices compuestos que hacen de este modelo uno de los más cercanos al relacional[9].

Tanto el esquema de un documento como los tipos de datos que va a contener son definidos por la aplicación que persiste, por lo tanto esta debe garantizar la fiabilidad de la información ingresada[4].

Modelo de familia de columnas

También se conocen como modelo orientado a columnas. Están inspirados en Bigtable, “que es un sistema de gestión de base de datos creado por Google con las características de ser: distribuido, de alta eficiencia y propietario. Esta construido sobre GFS (Google File System)”[10].

Bigtable se utiliza en muchos proyectos de Google que varían en los requisitos de alto rendimiento y los datos sensibles a la latencia de las porciones. Este modelo maneja un número de pares de claves y valores almacenados dentro de las filas.

Este paradigma de bases de datos orientadas a columnas es usado en aplicaciones con un índice bajo de escrituras pero muchas lecturas. Típicamente en data warehouse y sistemas de inteligencia de negocio, donde además resultan ideales para calcular datos agregados.

Modelo orientado a grafos

A diferencia de los modelos mencionados anteriormente, este es especialista en la gestión eficiente de los datos fuertemente vinculados, como lo son las redes sociales. Permite cambiar operaciones de búsquedas intensivas por recorridos eficientes, redundando en la disponibilidad y eficiencia de los sistemas que las implementan.

Los nodos y bordes consisten en objetos con pares de valores clave incrustados. La gama de claves y valores se puede definir en un esquema, en el que la expresión de restricciones más complejas se puede describir fácilmente. Por lo tanto es posible definir que un borde específico solo es aplicable entre unos ciertos tipos de nodos.

Los casos de uso de las bases de datos de grafos son servicios basados en la localización, la representación del conocimiento, la trayectoria, encontrar problemas planteados en los sistemas de navegación, los sistemas de recomendación y las redes sociales, entre otros. Algunos de los motores de base de datos NoSQL que implementan este modelo son: Ne04j y GraphDB.

¿Cuál elegir?

En la actualidad se puede encontrar una gran diversidad de software que sea capaz de gestionar los datos, cada uno tiene ventajas y desventajas específicas para cierto tipo de casos de uso.

Las bases de datos de llave-valor son sin duda las de mejor rendimiento pero ofrecen una funcionalidad más limitada.

Las bases de datos orientadas a columnas son muy buenas para el trabajo con datos agregados.

Las bases de datos de almacenes de documentos son una gran opción si se quiere tener consultas complejas pero sin perder la velocidad de los almacenes llave-valor[4].

Las bases de datos orientadas a grafos son fuertes en grandes relaciones en cadena y facilitan

recorridos eficientes a través de grandes volúmenes de datos.

Por último, las bases de datos relacionales siguen siendo la mejor opción en cuanto se refieren a temas que requieran mantener la integridad y consistencia de la información[2][3].

CONCLUSIONES

NoSQL es una opción más en la lista de alternativas para el almacenamiento de los datos de una aplicación.

Existen diversos tipos de motores de bases de datos no relacionales, pero todos apuntan a un mismo objetivo que es: resolver los problemas de disponibilidad y escalabilidad de las que carecen los sistemas de bases de datos relacionales.

NoSQL en la actualidad es la tendencia para la gestión y tratamiento de los datos ya que su carencia de esquemas predefinidos ofrece flexibilidad al momento de persistir la información.

NoSQL cuenta con las ventajas del escalamiento horizontal y de bajo costo en la infraestructura que las soporta lo que le ha permitido ofrecer a los desarrollos de soluciones ágiles más oportunidades de crecimiento y evolución incremental, pensando siempre en la presentación de un producto funcional, rápido y que satisfaga las necesidades del cliente.

REFERENCIAS

[1] Nathan Hurst's Blog (2010). *Visual Guide to NoSQL Systems*. Recuperado de: <http://blog.na-hurst.com/visual-guide-to-nosql-systems>

[2] Camacho, E. (2010). NoSQL la evolución de las bases de datos. *SG Software Guru*, 28. Recuperado de: http://sg.com.mx/revista/42/nosql-la-e-volucion-las-bases-datos#.VnnRe_nhDIV

[3] Paramio, C. (2011). *El concepto NoSQL, o cómo almacenar tus datos en una base de datos no relacional*. Genbeta Dev. Recuperado de: <http://www.genbetadev.com/bases-de-datos/el-concepto-nosql-o-como-almacenar-tus-datos-en-una-base-de-datos-no-relacional>

[4] MongoDB. (2015) *NoSQL Databases Explained*. Recuperado de: <http://www.mongodb.com/nos-ql-explained>

[5] Graterol, Y. (2014). *NoSQL como el futuro de las bases de datos*. Maestros del Web. Recuperado de: <http://www.maestrosdelweb.com/nosql-com-o-el-futuro-de-las-bases-de-datos/>

[6] MSDN–Microsoft. (2015). *Dictionary (TKey, Tvalue)*. Recuperado de: [https://msdn.microsoft.com/es-es/library/xfhwa508\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/xfhwa508(v=vs.110).aspx)

[7] Rodríguez, T. (2011). *UnQL, un lenguaje de consulta unificado para todas las bases de datos NoSQL*. Genbeta Dev. Recuperado de: <http://www-w.genbetadev.com/tag/unsq>

[8] Manejando datos. (2013). *Consultas con MongoDB*. Recuperado de: <http://www.manejandodatos.es/2013/12/consultas-con-mongodb/>

[9] Quantum Mode. (2014). *Las bases de datos NoSQL (II): El modelo de datos*. Recuperado de: <http://quantummode.com/las-bases-de-datos-no-sql-ii-el-modelo-de-datos/>

[10] Wikipedia. *BigTable*. Recuperado de: <https://es.wikipedia.org/wiki/BigTable>