

Bases de Datos NoSql en Cloud Computing

Adriana Martín¹, Susana Chávez², Nelson Rodríguez³, Adriana Valenzuela⁴, María Murazzo⁵

Departamento e Instituto de Informática - F.C.E.F. y N. - U.N.S.J.

Complejo Islas Malvinas. Cereceto y Meglioli. 5400. Rivadavia. San Juan

¹arianamartinsj@gmail.com ²schavez@iinfo.unsj.edu.ar ³nelson@iinfo.unsj.edu.ar

⁴franciscaadriana.valenzuela@gmail.com ⁵marite@unsj-cuim.edu.ar

Resumen

Las bases de datos NoSQL son sistemas de almacenamiento de información que **no cumplen con el esquema entidad-relación**. Mientras que las tradicionales bases de datos relacionales basan su funcionamiento en tablas, joins y transacciones. Las bases de datos NoSQL **no imponen una estructura de datos en forma de tablas y relaciones entre ellas sino que proveen un esquema mucho más flexible**.

Las bases NoSQL son adecuadas para una escalabilidad realmente enorme, y tienden a utilizar modelos de consistencia relajados, no garantizando la consistencia de los datos, con el fin de lograr una mayor performance y disponibilidad. A esto se agrega el inconveniente de que no tienen un lenguaje de consulta declarativo, por lo que requiere de mayor programación para la manipulación de los datos.

En general se pueden mencionar Sistemas NoSQL clasificados en cuatro categorías:

- Framework **Map-Reduce** (usado por aplicaciones que hacen procesamiento analítico online - OLAP), Por ejemplo Hadoop.
- Almacenamiento **Clave-Valor** (sistemas que tienden al procesamiento de transacciones online - OLTP), Por ejemplo: Google

BigTable, Amazon Dynamo, Cassandra, Voldemort, HBase.

- Almacenamiento de **Documentos**
Por ejemplo: CouchDB, MongoDDB, SimpleDB
- Sistemas de base de datos **Gráficas**. Por ejemplo: Neo4j, FlockDB, Pregel.

Con respecto al almacenamiento en **Columnas** que en general son tratados como Sistemas NoSQL, no son más que una forma de organización de un sistema de base de datos relacional. Sin embargo por la alta performance para cierto tipo de aplicaciones son considerados como del tipo almacenamiento Clave-Valor.

En resumen, para **manipular enormes cantidades de información de manera muy rápida** los Sistemas NoSQL trabajan mejor que los sistemas de base de datos tradicionales, sin embargo para muchísimas aplicaciones la solución está en las bases de datos tradicionales.

Palabras clave: NoSql, Map- Reduce, Cloud Computing.

Contexto

El presente trabajo se encuadra dentro del área de Bases de Datos y Minería de Datos, y se enmarca dentro del proyecto de investigación Implantación de un ambiente de Cloud Computing para integración de recursos, el cual tiene como unidades ejecutoras al Departamento e Instituto de Informática

de la FCEfyN de la UNSJ. Los trabajos iniciados en el citado proyecto tienden al desarrollo de aplicaciones sobre Cloud Computing, y almacenamiento de grandes cantidades de información y rápido acceso sobre el mismo

Introducción

Las bases de datos NoSQL surgen como una solución a los constantes requerimientos de procesamiento y análisis a gran escala de enormes cantidades de datos, y para los cuales los sistemas tradicionales de base de datos son insuficientes.

El termino **NoSQL** ha evolucionado no para significar *no lenguaje sql*, sino para referirse a sistemas que *no son DMBS* (database management system) *tradicionales*.

El framework MapReduce es una gran alternativa a los sistemas tradicionales. En base a este framework Google ha desarrollado **Hadoop**, un sistema open source usado ampliamente. Se caracteriza por:

- No existe un modelo de datos, los datos se almacenan en archivos HDFS (Hadoop Distributed File System).
- Los usuarios proveen las funciones específicas los datos usando el Framework MapReduce. Las funciones disponibles son: map(), reduce(), reader(), writer() y combine().
- El sistema provee procesamiento de datos "glue" escalable y con tolerancia a fallos (Glue procesa los datos a través de las funciones garantizando la tolerancia a fallos).

El almacenamiento Clave-Valor, está diseñado para el procesamiento de transacciones online (OLTP). Este tipo de aplicaciones son operaciones pequeñas sobre algunos datos de una base de datos masiva. Por esto, son mas simples

Para ello suelen almacenar toda la información que puedan en memoria y

están preparadas para escalar horizontalmente sin perder rendimiento.

Si pretendemos desarrollar una aplicación que requiera la lectura/escritura de cantidades de datos y pueda dar servicio a millones de usuarios sin perder rendimiento, entonces debemos plantearnos el uso de una base de datos NoSQL. Las grandes redes sociales como [facebook](#) y [twitter](#) o el propio [Google](#) las utilizan como medio fundamental de almacenamiento de información.

Se puede utilizar una base de datos NoSQL para almacenar toda la información de una aplicación para aquellas funcionalidades que requieren millones de consultas en tiempo real.

Existe una gran variedad de bases de datos basadas en documentos NoSQL para clasificar la información en un formato estructurado, mientras afrontamos la estructura flexible de los puntos de datos individuales. Muchos entornos NoSQL también brindan apoyo para consultas Map-Reduce exhaustivas y para el procesamiento, lo que las hace ideales para el procesamiento de una gran cantidad de datos en un formato resumido.

Líneas de investigación y desarrollo

Afortunadamente existen varias bases de datos NoSql conocidas. Mientras todas ellas comparten muchas características, también hay algunas diferencias significativas

Base de Datos	Características	Aplicabilidad
Mongo DB	<ul style="list-style-type: none"> • Escrito en C++Algunas características SQL (Query, index) • Protocolo binario • Replicación maestro-esclavo 	<p>Para queries dinámicasNo para Map/Reduce</p> <p>Cuando necesitas CouchDB con muchos cambios</p> <p>Para muchas</p>

	<ul style="list-style-type: none"> • Sharding • Permite ejecutar Javascript • Rendimiento sobre características • En 32bits sólo 2.5 Gb • GridFS para almacenar BigData • Indexing geoespacial 	cosas que ahora se hacen con MySQL		<ul style="list-style-type: none"> • en memoria con backup en disco • Sin disk-swap • Replicación maestro-esclavo • Clave-Valor, pero valores pueden ser List, Hashes, Sets, • Transacciones 	tamaño)Bases de datos de tiempo real
Riak	<ul style="list-style-type: none"> • Escrito en Erlang y CTolerancia a fallos • Protocolo binario o HTTP/REST • Pre y postcommits en JS y Erlang • Map/Reduce en JS o Erlang • Soporte objetos grandes • Versión opensource y Enterprise • Búsqueda Full text, indexing,... con Riak Search Server • Multireplicación 	Estilo Cassandra pero sin su complejidad		<ul style="list-style-type: none"> • Escrito en JavaBillones de filas x millones de columnas • Protocolo HTTP/REST y Thrift • Basada en Google Big Table • Map/Reduce con Hadoop • Optimizaciones para queries en tiempo real • Gateway Thrift de alto rendimiento • HTTP soporta XML, protobuf y binario • Módulos para Cascading, Hive y Pig • Shell basada en JRuby • Rendimiento random-acces como MySQL 	Analíticas Stock prices Comunicación en tiempo real
CouchDB	<ul style="list-style-type: none"> • Escrito en ErlangProtocolo HTTP/REST • Replicación bidireccional con detección de conflictos • MVCC • Versiones previas de documentos disponibles • Necesita compactación de vez en cuando • Soporta attachment • Incluye librería JQuery 	Para acumular datos que sólo cambian ocasionalmente con queries predefinidasPara sistemas que necesitan versionado		<ul style="list-style-type: none"> • Escrito en JavaBase de datos de grafos • Protocolo HTTP/REST o Java • Funcionamiento standalone o embebido en Java • Full ACID • Lenguaje de queries pattern-maching • Web de administración incluida • Path-finding algoritmos 	Mejor opción para Map/ReduceAlmacenamiento y análisis ficheros de log
Redis	<ul style="list-style-type: none"> • Escrito en C++Muy rápida • Protocolo estilo Telnet • Bases de datos 	Para datos cambiantes almacenados en memoria (no de gran			Para datos ricos interconectados estilo grafosPara redes sociales, topologías de red

	<ul style="list-style-type: none"> • Indexado de claves y relaciones • Optimizado para lecturas • Transacciones en API Java • Scripts en Groovy • Backup online, monitorización y alta disponibilidad en versión comercial AGPL 	
Cassandra	<ul style="list-style-type: none"> • Escrito en JavaLo mejor de BigTable y Dinamo • Protocolo binario (Thrift) • Tuneable para distribución o replicación • Búsqueda por columnas o rango de claves • Características BigTable • Índices secundarios • Escrituras más rápidas que lecturas • Map/Reduce con Hadoop 	<p>Mas escritura que lectura (logging) Cuando todos los componentes son Java</p> <p>Análisis tiempo real</p>
Membase	<ul style="list-style-type: none"> • Escrito en Erlang y CCompatible Memcached pero con persistencia y clustering • Protocolo memcached • Acceso muy rápido por clave • Persistencia en disco • GUI para gestión del cluster • Actualizaciones de SW sin parar la BD 	<p>Aplicaciones con acceso de muy baja latenciaAplicaciones con alta concurrencia y alta disponibilidad</p> <p>Online gaming (Zinga)</p>

En general, en el área de la computación distribuida Map se utiliza para fraccionar una operación compleja entre varios nodos y Fold/Reduce para recoger los resultados y unificarlos.

Por su parte, los [frameworks MapReduce](#) toman la base de las operaciones mencionadas anteriormente para crear una operación genérica y más compleja, cuyo funcionamiento es realmente útil para las bases de datos NoSQL: en vez de usarse sobre listas de valores unidimensionales, ésta toma como parámetros entrantes una lista de tuplas de tipo (*clave, valor*) y devuelve una lista de valores. Entre las operaciones [map](#) (distribuida) y [reduce](#) (normalmente localizada) se genera una lista de tuplas (*clave, valor*) con valores temporales, de las que [reduce](#) filtra solamente las que tengan una determinada clave.

En definitiva, MapReduce es fundamental en las bases de datos NoSQL para permitir la utilización de funciones de agregación de datos, ya que al carecer de esquema son mucho más complicadas que en las bases de datos relacionales clásicas [RDBMS](#) (*Relational Database Management System*).

RESULTADOS Y OBJETIVOS

Resultados Obtenidos

Se han publicado siete (7) trabajos de investigación en diferentes Congresos y Jornadas, y tres (3) trabajos de divulgación: un trabajo en el Symposium Internacional. La computadora contra el estigma de la discapacidad. Montevideo. 2009[26], otro trabajo en el Workshop de Investigadores en Ciencias de la Computación 2010 [1], otro en el WICC 2011 [8], dos (2) trabajos en el Congreso Argentino de Ciencias de la Computación 2010 [2] [3], otro en Perú [4] otro en V Seminario Internacional “De legados y Horizontes para el Siglo XXI”, 2010, organizado por RUEDA [4], y otro en el 1º Seminario Argentina Brasil de Tecnologías de la Información y la Computación [7]. El trabajo de divulgación es citado en [6]. También se ha desarrollado una tesina en Windows Azure, otra con Google AppEngine, otra

sobre Eye OS, y tres basadas en aplicaciones para lengua de señas.

Resultados Esperados

El objetivo del grupo de investigación en esta línea, es la comparación en varios aspectos de las distintas Bases de Datos NoSql, y la integración de distintas tecnologías Open Source como las que propone Google.

FORMACION DE RECURSOS HUMANOS

El proyecto marco sobre el que se realizan las investigaciones comenzó 2010, las publicaciones y trabajos de divulgación se han desarrollado en colaboración con becarios y alumnos avanzados, como [1] [2] [3] [4] [5] [6] [7] [8].

Se desarrolló una beca de investigación de alumno avanzado sobre Mobile Cloud Computing y se está desarrollando otra sobre orquestación en servicios Web. Se aprobaron tres (3) tesinas de tecnicatura sobre el tema de lengua de señas, dos de ellas sobre plataforma móvil y la otra implementando servicios Web. Por otro lado también se han aprobado 3 (tres) tesinas de licenciatura sobre Cloud Computing, sobre distintas tecnologías como Windows Azure y Google App Engine, una de las cuales se integra con una plataforma móvil con el OS Android. Además se encuentra en desarrollo y se proyectan dos de licenciatura y otra de tecnicatura, y se espera realizar alguna tesis de maestría y aumentar el número de publicaciones. Por otro lado también se prevé la divulgación de varios temas investigados por medio de cursos de postgrado y actualización o publicaciones de divulgación.

REFERENCIAS

- [1] Murazzo, Rodríguez. "Mobile Cloud Computing". WICC 2010. Calafate. Mayo 2010.
- [2] Murazzo, Millán, Rodríguez, Segura, Villafañe. Desarrollo de aplicaciones para

Cloud Computing. CACIC 2010. Morón. Oct. 2010.

[3] Murazzo, Rodríguez, Millán, Segura y Villafañe. "Plataformas Educativas Implementadas Con Cloud Computing". XVI Congreso Argentino de Ciencias de la Computación – CACIC 2010, Workshop de Tecnologías Informáticas Aplicadas a la Educación. Morón. Oct. 2010.

[4] Murazzo, Rodríguez. "Una propuesta para el desarrollo de aplicaciones para Mobile Cloud Computing". Congreso Internacional de Computación y Telecomunicaciones – COMTEL 2010, Lima, Perú. Oct. 2010.

[5] Millán, Murazzo, Rodríguez. "Plataformas Educativas Implementadas Con Mobile Cloud Computing". V Seminario Internacional "De legados y Horizontes para el Siglo XXI", organizado por RUEDA. Tandil. Sep. 2010.

[6] Rodríguez, Murazzo, Ene. "Cloud Computing". X Workshop de investigadores en Ciencias de la Computación y Sistemas de Información. San Juan. Mayo 2009.

[7] Nelson R. Rodríguez, María A. Murazzo, Cecilia di Sciacio. "Integración de Computación móvil con Cloud Computing". 1º Seminario Argentina Brasil de Tecnologías de la Información y la Computación; bajo el lema "Las TIC como oportunidad de integración". Rosario Nov. 2011.

[8] Rodríguez, Chavez, Martin, Murazzo, Valenzuela. "Interoperabilidad en Cloud Computing". XII Workshop de investigadores en Ciencias de la Computación y Sistemas de Información. Rosario. Mayo 2011.

[9] Rodríguez, Villafañe, Murazzo, Gallardo, Tarrachano. "GAE, una estrategia para complementar SaaS y PaaS a través de la Web". 2do SABTIC. Tres de Maio, Brasil. Agosto 2012.

BIBLIOGRAFIA CONSULTADA

Antonopoulos- Gillan "Cloud Computing Principles, Systems and Applications" Springer 2010- 978-1-84996-241-4

LINK DE INTERES

<http://www.nosql.es/blog/nosql/mapreduce.html>