

# Big data y NoSQL: su rol en la revolución del *cloud computing* y sus retos hacia la estandarización



# **Big data y NoSQL: su rol en la revolución del *cloud computing* y sus retos hacia la estandarización<sup>1</sup>.**

Big data and NoSQL: their role on the cloud computing revolution and their challenges to standardization.

Javier Enrique De la Hoz Freyle<sup>2</sup>.

Universidad Industrial de Santander (UIS) Bucaramanga, Colombia.

Artículo recibido en julio de 2015; artículo aceptado en agosto de 2015.

Citación del artículo: De la Hoz, J. (2015). *Big data y NoSQL: su rol en la revolución del cloud computing y sus retos hacia la estandarización. I+D Revista de Investigaciones*, 6(2), 111-124.

---

## **Resumen**

El auge en la utilización de las tecnologías de información (TI) orientadas a *cloud computing* ha generado grandes cantidades y variedades de datos, o *Big Data*, en los centros de datos de las organizaciones proveedoras de estos servicios, generando retos a los desarrolladores y administradores de las TI para el manejo eficiente de esos recursos. Frente a este fenómeno, las bases de datos relacionales son consideradas por los profesionales de las TI como una obstrucción para ofrecer las características esenciales del *cloud computing*. Esto ha impulsado una serie de nuevos enfoques y tecnologías para la gestión de datos, conocidos como bases de datos no relacionales o NoSQL, diseñadas para proveer rapidez, escalabilidad, alta disponibilidad y elasticidad, facilitando la administración de

datos en las soluciones tecnológicas en *cloud computing*. En este artículo se exponen los conceptos de *Big Data*, sus principales herramientas tecnológicas para la gestión de datos, el movimiento NoSQL, el rol de estos en la revolución del *cloud computing* y los retos de estas tecnologías hacia la estandarización.

**Palabras clave:** bases de datos, *Big Data*, *cloud computing*, NoSQL

## **Abstract**

The access and usage boom of cloud computing oriented information technologies (IT) has generated huge amounts, and variety of data, or Big Data into the datacenters of the organizations than provide these kind of services,

---

<sup>1</sup>Artículo de investigación de enfoque cualitativo, Fecha de inicio: junio de 2014, fecha de terminación: enero de 2015.

<sup>2</sup>Ingeniero de Sistemas, Universidad Industrial de Santander. Magíster en Ingeniería de Sistemas e Informática, Universidad Industrial de Santander. Ingeniero de Mkit Digital. Dirección: Calle 65 Sur No. 42B-14, Sabaneta, Antioquia (Colombia). Correo electrónico institucional: javier@mkitdigital.com

originating in that way, some challenges for IT managers and developers on handling efficiently those resources. In this phenomenon, relational databases are seen by IT developers as an obstruction to achieve the essential features of cloud computing. This has driven a set of new approaches and technologies for data management, known as no-relational databases or NoSQL, they are designed to provide speed, scalability, and elasticity, making easy the management of cloud computing technological solutions. In this papers are exposed the Big Data and NoSQL concepts, their main technological tools to data management, their role on the cloud computing revolution, and the challenges of this technologies to their standardization.

**Keywords:** Databases, Big Data, cloud computing, NoSQL.

### Introducción

En los últimos años, los negocios con base en TI han experimentado cambios, generando la aparición de nuevos enfoques y paradigmas en la entrega de recursos informáticos. Uno de ellos es *cloud computing*, que proporciona a los usuarios acceso a recursos de TI ya sean en forma de *hardware* o *software* a través de internet (Mell & Grance, 2008; Armbrust, y otros, 2010).

Las principales características que hacen del *cloud computing* un paradigma atractivo para el despliegue y consumo de aplicaciones son la alta disponibilidad, rápida escalabilidad, elasticidad, ubicuidad, y el acceso sobre demanda de los recursos informáticos que se ofrecen (Mell & Grance, 2008; Buyya, Yeo, Broberg, & Brandic,

2009; Godavari, 2010; Bowers, Juels, & Oprea, 2009; Vouk, 2008). Dicho atractivo ha impulsado iniciativas de varias empresas de *software* como Microsoft, Google Inc., IBM, y Amazon, entre otras, para ofrecer a los usuarios todo tipo de servicios de TI a través de internet (Marks & Lozano, 2010), (Chang, Bacigalupo, Wills, & De Roure, 2010), (Choo, 2010), (Armbrust, y otros, 2009), (Pearson, Shen, & Mowbray, 2009).

Esto ha permitido la aparición de redes sociales como Facebook, Twitter, Instagram, Pinterest, entre otras, que en conjunto, con varias de las soluciones *cloud*, además de brindar beneficios a los usuarios, han traído consigo retos a los profesionales de TI en cuanto al manejo de los volúmenes de datos que se almacenan en los centros de cómputo (Abu-libdeh & Weatherspoon, 2010; Hu, Yang, & Matthews, 2010; Calder y otros, 2011). Dichos retos han motivado el desarrollo de una serie de soluciones tecnológicas centradas en el manejo y análisis ágil del *Big Data*, denominado así por tratarse de “grandes” y variadas cantidades de datos (Trelles, Prins, Snir, & Jansen, 2011; Borkar, Carey, & Li, 2012; A. ElAbbadi, 2010).

A su vez, aunque las bases de datos relacionales proveen a los desarrolladores de herramientas que les permiten tener atomicidad, consistencia, aislamiento y durabilidad de datos (ACID) (Haerder & Reuter, 1983), al entrar en el manejo de “grandes” cantidades de datos, las bases de datos relacionales presentan disminuciones considerables en el rendimiento de las transacciones (Zhou, States, Grimmer, King, & Science, 2013). A su vez, se plantean retos en cuanto a la escalabilidad, dificultando el aumento de los recursos computacionales para

preservar o mejorar el desempeño de las bases de datos en el uso de *Big Data* (Trelles, Prins, Snir, & Jansen, 2011).

Siendo así, las compañías de *software* han encontrado la solución a estos inconvenientes en la inclusión de sistemas de gestión de archivos distribuidos, en los cuales los datos pueden estar almacenados en distintos servidores en diferentes partes del mundo, distribuyendo la carga de las transacciones (Across & Hardware, 2007). Asimismo, para acrecentar el desempeño de las operaciones, facilitar la escalabilidad, elasticidad, y alta disponibilidad de los datos, las compañías están cambiando en cuanto a la utilización de bases de datos relacionales, reemplazándolas por gestores de datos, que permitan escalabilidad y rapidez en las transacciones. Dicha estimación de cambio ha permitido el surgimiento del movimiento de las bases de datos no relacionales o NoSQL, que representan una alternativa para el desarrollo de proyectos de *cloud computing* vinculando herramientas para el soporte de *Big Data* (Strauch, 2011), (Bobrowski, 2011).

En el presente artículo se ilustra el rol de las soluciones tecnológicas de *Big Data* y NoSQL en la revolución del *cloud computing*, así como los retos que los investigadores y desarrolladores de herramientas deben asumir hacia la estandarización, para aumentar su adopción en el mercado de las TI.

La estructura del artículo es la siguiente: en el segundo aparte se exponen las características esenciales de *cloud computing*; seguidamente, se presenta una descripción del *Big Data* y las principales tecnologías para su gestión, así como

su rol en el *cloud computing*; luego, se detalla el movimiento NoSQL; después, se plasman los retos de estos enfoques tecnológicos hacia la estandarización; y por último, se expresan las conclusiones.

### Cloud computing y sus características

Aunque en la literatura existen diversas definiciones de *cloud computing*, no hay un consenso en los profesionales de TI con respecto a cuál debe ser la definición formal de este término (Buyya, Yeo, Broberg, & Brandic, 2009), (Marks & Lozano, 2010). Sin embargo, en la comunidad de las TI, una de las definiciones más adoptada es la propuesta por Gartner (2013): “*Cloud computing* es un estilo de computación en el cual las capacidades de TI escalables y elásticas son entregadas como servicios utilizando tecnologías de internet”. En la filosofía que cobija al paradigma *cloud computing* se espera que esos servicios, ya sean *hardware* o *software*, posean ciertas características, entre ellas: autoservicio sobre demanda, escalabilidad, rápida elasticidad, y alta disponibilidad (Mell & Grance, 2008), (Godavari, 2010), (Armbrust, y otros, 2009).

Autoservicio sobre demanda se refiere a la capacidad que tienen las soluciones *cloud* de ofrecer acceso a los recursos que el usuario solicita sin la intermediación del proveedor de servicios (Mell & Grance, 2008). Amparándose en el modelo de negocios *Pay as you go*, los usuarios tienen la posibilidad de pagar únicamente por los recursos *hardware* o *software* que consuman, y no un cargo fijo (Chang, Bacigalupo, Wills, & De Roure, 2010), (Armbrust, y otros, 2009), (IDC, 2013), (Corporation, 2013), (Wang, Yang, Dong, Li, &

Li, 2010), (Bhardwaj, Jain, & Jain, 2010), (Foster, Zhao, & Raicu, 2008). La escalabilidad hace referencia a la habilidad de los servicios en *cloud computing* de aumentar o disminuir sus capacidades de TI conforme los usuarios lo requieran. Mientras, la rápida elasticidad es la capacidad de proveer dicha escalabilidad de manera instantánea (Mell & Grance, 2008), (Marks & Lozano, 2010), (Nurmi, y otros, 2009). La alta disponibilidad describe la capacidad de las soluciones de proveer acceso a los datos en cualquier momento, lugar, y dispositivo con conexión de internet; a esto también se le conoce como ubicuidad de los datos (Bowers, Juels, & Oprea, 2009), (Armbrust, y otros, 2009), (Hu, Yang, & Matthews, 2010), (Bhardwaj, Jain, & Jain, 2010). A su vez, se espera que los servicios *cloud* sean entregados siguiendo el modelo *multi-tenancy*, que permite que los recursos de un proveedor de servicios sean compartidos por todos sus usuarios, y que una sola aplicación sea entregada y consumida por diferentes usuarios (*tenants*), haciendo uso de un solo contenedor de datos (Vouk, 2008), (Bobrowski, 2011), (Mietzner, Unger, Titze, & Leymann, 2009).

Dichas características, además de proveer servicios que facilitan el consumo de recursos *hardware* y *software* a los usuarios, generan diversos retos tecnológicos a los desarrolladores de dichos servicios. Uno de esos retos es la selección y uso del motor de las bases de datos, que debe brindar escalabilidad, rápida elasticidad, alto desempeño en las transacciones y manejo distribuido de datos, así como ofrecer las herramientas tecnológicas que faciliten la distribución de la carga de operaciones. Las soluciones de *Big Data* y NoSQL ofrecen una alternativa para dichos retos, y han impulsado el

desarrollo de soluciones *cloud* en años recientes. En el siguiente aparte se describen los términos *Big Data* y NoSQL, así como la forma en que estos han ayudado a proveer las características *cloud computing* a los servicios de TI.

### **Big Data, sus mayores expositores, y su rol en los servicios cloud computing**

La revolución del *cloud computing* condujo a las compañías de TI a incrementar exponencialmente la cantidad de datos que poseían en sus centros de cómputo, por medio de, en la mayoría de los casos, la recolección de datos de usuarios para analizarlos y obtener provecho de ellos (Gartner, 2013), (El Abbadi, 2011). Esto dio origen al concepto de *Big Data*, que según Gartner (2013) “Son los activos de información en grandes volúmenes, alta velocidad, y gran variedad, que demandan formas rentables e innovadoras de procesamiento de información para mejorar la toma de decisiones”.

Ante el reto de administrar y analizar rápida y eficientemente aquellos “grandes” volúmenes de datos estructurados o no, las compañías de TI impulsaron una serie de soluciones tecnológicas, varias de las cuales llegaron en forma de sistemas de gestión de archivos distribuidos, que permiten a los programas distribuir la carga y los volúmenes de datos en diferentes nodos, de manera que se pueda preservar el rendimiento de las operaciones (Across & Hardware, 2007), (Konstantinou, Angelou, Tsoumakos, & Koziris, 2010).

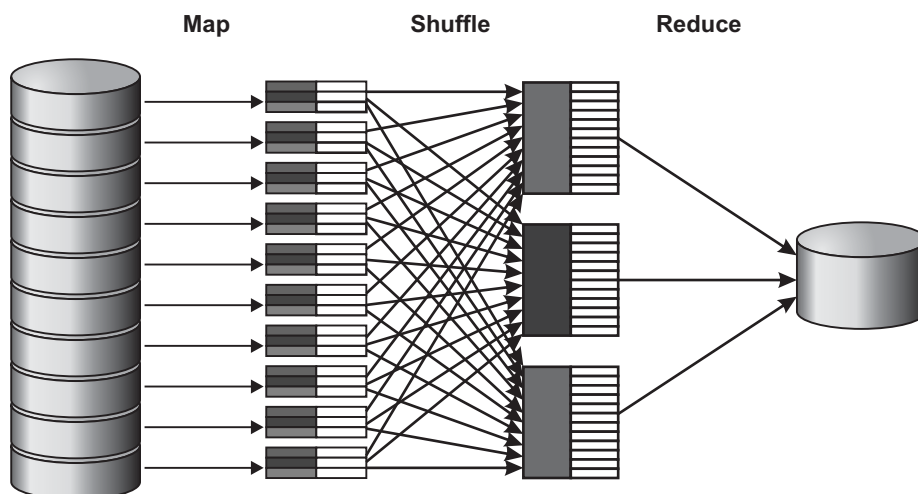
Una de las primeras soluciones fue *Google File System* (GFS), desarrollado por Google Inc. para solventar las necesidades de almacenamiento en su buscador web (Ghemawat,

Gobioff, & Leung, 2003). GFS permite distribuir la carga de almacenamiento de archivos de la compañía en un clúster que cuenta con miles de nodos conectados alrededor del mundo, cada uno con capacidad para albergar cientos de terabytes. A su vez, para procesar los datos en GFS, Google Inc. desarrolló un modelo de computación denominado MapReduce que facilita el procesamiento de datos en paralelo (Dean, 2010).

El modelo MapReduce depende de dos funciones, Map y Reduce, para ejecutar los

análisis de datos en múltiples nodos de un clúster de computadores. La función Map procesa entradas de datos en duplas de llaves y valores por cada nodo, o en inglés *Key/Value*, que dependiendo del objetivo del análisis, genera valores intermedios en duplas *Key/Value*. La función *Reduce* se encarga de unir las duplas intermedias que posean la misma llave (*key*) y genera los valores (*value*) dependiendo del objetivo y las funciones determinadas por los desarrolladores (Strauch, 2011). En la Figura 1 se ilustra el modelo MapReduce.

Figura 1. Modelo de MapReduce.



Fuente: Tomado y adaptado de (Google, 2013)

Posteriormente, otras compañías de TI, lideradas por Yahoo, Facebook y Apache, unieron esfuerzos para desarrollar un sistema de archivos distribuidos libre, tomando como inspiración a GFS y el modelo MapReduce de Google Inc., dando origen así a Hadoop (Borkar, Carey, & Li, 2012), (Zhou, States, Grimmer, King, & Science, 2013), (Borthakur & Gray, 2011), (Kambatla, 2009). Este sistema de archivos distribuidos ganó popularidad entre los desarrolladores de soluciones con necesidades de

administrar el *Big Data*, y actualmente es usado por compañías promotoras del movimiento *cloud computing* como Facebook, Twitter, Amazon, Netflix, Instagram, Rackspace, y Yahoo, entre otras (Konstantinou, Angelou, Tsoumakos, & Koziris, 2010), (Kambatla, 2009), (F. Wang, 2009).

Por su parte, Microsoft desarrolló su propia tecnología para el manejo de *Big Data*, denominada Dryad (Borkar, Carey, & Li, 2012),

un modelo de programación en paralelo que permite escalar los recursos de los clústeres en cuanto a *hardware*, para distribuir la carga de operaciones (Microsoft, 2013). Inicialmente, Dryad fue diseñado para computación de alto rendimiento con el sistema operativo Windows Server 2008, pero, posteriormente, la compañía decidió discontinuar el proyecto para desarrollar distribuciones de Windows Azure y Windows Server utilizando Hadoop como sistema de archivos distribuidos (Foley, 2013).

A pesar de las soluciones antes mencionadas, que suministran a las organizaciones proveedoras de servicios *cloud* así como a las compañías con necesidades de gestionar *Big Data*, herramientas tecnológicas capaces de distribuir la carga de las operaciones de datos, existe una falencia debida uso de bases de datos relacionales, que representa un cuello de botella al momento de implementar las características del paradigma *cloud computing* (Bartholomew, 2010), (Cattell, 2010). Este hecho indujo al desarrollo de diferentes enfoques y herramientas para la gestión de datos, lo que originó el movimiento de las bases de datos no relacionales o NoSQL, que se describen a continuación.

### **El movimiento NoSQL y sus implicaciones en *cloud computing***

Las bases de datos no relacionales o NoSQL (*Not Only SQL*, no solo SQL) a diferencia de las bases de datos relacionales, almacenan datos sin mecanismos estructurados para que puedan ser vinculados de un contenedor de a otro, lo que facilita la gestión, tanto de datos estructurados como de aquellos no estructurados (Padhy, Patra, & Satapathy, 2011)

Según Cattell (2010), las bases de datos NoSQL poseen seis características principales: la habilidad para escalar el rendimiento de las operaciones a lo largo de varios servidores; replicación y distribución de datos en varios equipos; protocolos e interfaces simples a nivel de transacciones; modelos de concurrencia menos robustos en comparación con el ACID de las bases de datos relacionales; uso eficiente de memoria e índices distribuidos para el almacenamiento de datos; adición de atributos dinámicamente a los registros de datos.

Al igual que las relacionales, las bases de datos NoSQL se basan en la capacidad de cumplir con dos de las propiedades del teorema CAP de Brewer (Brewer, 2000), en donde C es consistencia, A disponibilidad, y P tolerancia a la partición (Cattell, 2010), (Xiang, Hou, & Zhou, 2010).

Las bases de datos relacionales se enfocan en proveer las capacidades C y A del teorema de Brewer, es decir, consistencia y disponibilidad. Entre las soluciones NoSQL existen algunas orientadas a estas mismas capacidades, mientras otras proveen las capacidades A y P del teorema, es decir, disponibilidad y tolerancia a la partición, y unas más proporcionan C y P, consistencia y tolerancia a la partición (Bartholomew, 2010). La ventaja principal de las soluciones NoSQL sobre las bases de datos relacionales es la habilidad para escalar el rendimiento de las operaciones a lo largo de varios servidores interconectados, permitiendo obtener niveles bajos de latencia en los procesadores de los nodos, inclusive si existe un alto número de transacciones, suministrando rendimientos deseables en las operaciones (Cattell, 2010), (Leavitt, 2010), (Stonebraker,

2011). En la Tabla 1 se pueden observar los motores de bases de datos más relevantes, tanto

SQL como NoSQL, y su enfoque en la provisión de las capacidades del teorema CAP.

Tabla 1. *Motores de bases de datos y su enfoque al teorema CAP*

Base de datos	NoSQL	Propietario
Big Tables	x	Google Inc.
Cassandra	x	Facebook, Apache
Couch DB	x	Apache
MongoDB	x	10gen Inc.
MySQL		Oracle
Oracle		Oracle
PostgreSQL		PostgreSQL
Simple DB	x	Amazon
SQL Server		Microsoft

Fuente: El autor.

La diferencia de enfoque en cuanto al teorema CAP de algunas bases de datos NoSQL con respecto a las relacionales, así como la inclusión del modelo MapReduce en muchas de ellas, permite que los proveedores de servicios *cloud* puedan optar por este tipo de soluciones para soportar el manejo del *Big Data* en conjunto con los sistemas de archivos distribuidos como Hadoop; de esa manera se proveen de herramientas que facilitan la gestión de datos tanto estructurados como no estructurados, mientras obtienen altos rendimientos en el análisis de grandes cantidades de datos.

Uno de los primeros expositores del movimiento NoSQL para gestionar *Big Data* fue Google Inc., que presentó *Google Big Tables* como su solución para satisfacer las necesidades tanto de su buscador web, como de *Google Earth*, *Google Docs*, y otros servicios de la compañía (Cattell, 2010), (Chang, y otros, 2006). *Big Tables*

es un sistema de bases de datos NoSQL diseñado y desarrollado para manejar datos estructurados y escalarlos en grandes tamaños mientras sus operaciones son distribuidas sobre miles de servidores utilizando GFS y MapReduce.

Posteriormente, Amazon desarrolló SimpleDB, que es un sistema de base de datos NoSQL con alta disponibilidad y escalabilidad (Padhy, Patra, & Satapathy, 2011), (Sciore & Hill, 2007), (Amazon, 2013) que permite distribuir la carga de las operaciones sobre los datos en servidores ubicados en diferentes partes geográficas, a la vez que replica la información de los clientes para asegurar la disponibilidad de los datos frente a fallas en alguno de los equipos (Amazon, 2013). Esta herramienta es utilizada para el soporte en los servicios *cloud* ofrecidos por la compañía como EC2, Amazon Cloud Drive, S3, entre otros.



Por su parte, la compañía Apache ofrece una solución NoSQL a la comunidad de desarrollo de aplicaciones *cloud*, inspirada en el producto SimpleDB, denominado CouchDB (Cattell, 2010), que provee escalabilidad en múltiples nodos así como replicación asincrónica para garantizar la restauración de datos frente a posibles fallos. Las operaciones de los datos en CouchDB se distribuyen paralelamente sobre múltiples nodos para obtener alto rendimiento, a la vez que se producen bajas latencias en los ciclos de los procesadores (Cattell, 2010), (Apache, 2013).

La empresa de la red social Facebook, con miras de solventar sus necesidades de gestión de datos, desarrolló su propio sistema de bases de datos no relacional. Este fue elaborado en conjunto con Apache, y tiene por nombre Cassandra, y permite la interacción de flujos de datos con Hadoop para la distribución de la carga de operaciones. Cassandra tiene un modelo de concurrencia débil, que permite el aumento de la disponibilidad de los datos y facilita la actualización de réplicas asincrónicas (Borthakur & Gray, 2011), (Cattell, 2010). Hacia el 2013, Cassandra funcionaba en más de 400 servidores interconectados que dan soporte a las operaciones de Facebook, mientras que otras compañías proveedoras de servicios *cloud* como Twitter, eBay, NetFlix, y Cisco, entre otras, lo utilizan como motor de bases de datos (Apache, Apache Cassandra, 2013).

Otras compañías desarrolladoras de soluciones de TI se unieron al auge del movimiento NoSQL y desde 2008 han ofrecido diversos motores de bases de datos no relacionales a los desarrolladores de servicios

*cloud computing*; entre los más utilizados se encuentran HBase, MongoDB, HyperTable, Amazon's Dynamo, Voldemort, entre otros, que han permitido la aparición de servicios *cloud* como Foursquare bit.ly, infolinks, LinkedIn, SAP, SourceForge, Craigslist, Shutterfly, GitHub, Pinterest, Groupon, reddit.com, Flipboard, Mendeley, entre otros (Strauch, 2011), (Cattell, 2010), (Padhy, Patra, & Satapathy, 2011), (Leavitt, 2010), (Shao & Wang, 2012).

### **Retos del *Big Data* y NoSQL hacia la estandarización**

A pesar del soporte que brindan las soluciones tecnológicas del *Big Data* y NoSQL para la gestión de grandes volúmenes de datos para los servicios *cloud computing*, también presentan barreras y dificultades que obstaculizan su adopción por los equipos de desarrollo de TI. Una de las barreras que más afecta a estos enfoques tecnológicos es la falta de estandarización de métodos de consulta y transacciones en las bases de datos NoSQL.

A diferencia de las bases de datos relacionales, en las que cada proveedor se acoge al lenguaje estándar de consultas SQL para realizar las operaciones esenciales sobre los datos en los contenedores o tablas, las soluciones NoSQL poseen cada una su propio lenguaje de transacciones y consultas sobre los datos.

Existen diferentes tipos de estructura en los contenedores de datos NoSQL, como los documentos, las parejas *key/value*, la orientación a columnas, entre otros (Cattell, 2010), y se podría asumir que para cada tipo de estructura se necesitaría un lenguaje particular por tratarse de orientaciones diferentes hacia la estructuración

de los datos. Sin embargo, se encuentran diferencias en el lenguaje de las transacciones entre motores de bases de datos NoSQL con la misma orientación. Por ejemplo, si se quisiera realizar una búsqueda de todos los elementos de una colección/tabla de nombre documentos en las bases de datos orientadas a documentos, con MongoDB sería:

```
db.documentos.find();
```

y con CouchDB sería:

```
GET/nombre_base_de_datos/_design/documentos/_view/all HTTP/1.0
```

Asimismo, aunque Hadoop es el sistema de archivos distribuidos más utilizado para la gestión de *Big Data* en las organizaciones que prestan servicios *cloud computing* (Wang, Yang, Dong, Li, & Li, 2010), no es totalmente compatible con todos los sistemas operativos para la administración de servidores del mercado. De hecho, Hadoop solo logra la total funcionalidad de sus características en sistemas operativos Linux y Microsoft Windows. De esa manera, las empresas prestadoras de servicios *cloud*, antes de optar por el uso de Hadoop, deben considerar los sistemas operativos de sus equipos. Igualmente, no todas las versiones de la herramienta permiten el flujo de datos con motores de bases de datos, lo que implicaría gastos de tiempo y dinero en reconfiguración de equipos si una compañía inicia un proyecto con unas necesidades, y a medida que avanza, el modelo de negocios le exige otras, como suele ocurrir en el mundo de las TI.

Siendo así, han surgido problemas, principalmente por la falta de estandarización, que deberían ser asumidos por consorcios de organizaciones desarrolladoras de este tipo de

soluciones tecnológicas, para poder normalizar, los lenguajes de programación de las bases de datos NoSQL, ya sea por orientación a la estructuración de los datos, o por el tipo de capacidades que se ofrecen según el teorema de CAP, para brindar marcos de trabajo tanto a los desarrolladores de soluciones *cloud* como a todo aquel interesado en incorporar los beneficios del NoSQL en sus proyectos, que permitan facilitar el aprendizaje de dichos lenguajes, así como la integración con otros lenguajes de programación, y migración entre motores de bases de datos.

Por su parte, las organizaciones desarrolladoras de la herramienta Hadoop deberían tener en cuenta los diferentes sistemas operativos, así como la compatibilidad entre versiones, para así ofrecer soluciones tecnológicas más robustas a los programadores, e impulsar una mayor adopción del enfoque tecnológico hacia la gestión de *Big Data*.

## Conclusiones

El rol de las soluciones de TI para el manejo de *Big Data* en la revolución del *cloud computing* ha sido fundamental para la administración de grandes volúmenes de datos mientras se mantiene la velocidad en las operaciones y la variedad en la estructura de estos. Sin embargo, en muchas ocasiones requieren de un motor de bases de datos que permita escalabilidad para obtener el mayor provecho de dichas soluciones.

La inclusión de bases de datos no relacionales en el desarrollo de soluciones *cloud* ha facilitado la obtención de las características esenciales de ese paradigma para los proveedores de servicios.

No obstante, aún no existe una norma que permita estandarizar el uso de lenguajes de transacciones sobre los datos en los motores NoSQL, lo que dificulta la adaptación de los programadores a cada una de las soluciones.

### Referencias

A. El Abbadi, “. D. (2010). Big Data y Cloud Computing : New Wine or just New Bottles? The VLDB Endowment VLDB Endowment Hompage archive, 3(1), 1647-1648.

Abu-libdeh, H., & Weatherspoon, H. (2010). RACS: A Case for Cloud Storage Diversity. RACS: A Case for Cloud Storage Diversity, 229240.

Across, P., & Hardware, H. (2007). The Hadoop Distributed File System: Architecture and Design. 26th Symposium on Mass Storage Systems and Technologies (MSST), 1-14. Obtenido de P. Across y H. Hardware, “The Hadoop Distributed File System: Architecture and Design,” 26th Symposium on Mass Storage Systems and Technologies (MSST). 2007. pp. 114.

Amazon. (09 de 04 de 2013). SimpleDB. Obtenido de <http://aws.amazon.com/simpledb> Apache. (09 de 04 de 2013).

Apache Cassandra. Obtenido de <http://cassandra.apache.org> Apache. (09 de 04 de 2013).

Apache CouchDB. Obtenido de <http://couchdb.apache.org>

Armbrust, B., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., . . . Rabkin, A. (2010). Above the Clouds: A View of Cloud Computing. *Communications of the ACM*, 53(4), 50-58.

Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., . . . Zaharia, M. (2009). Above the Clouds: A Berkeley View of Cloud Computing. *Cloud Computing: An Old Idea Whose Time Has (Finally) Come*. 7-13.

Bartholomew, D. (2010). SQL vs . NoSQL The Case for NoSQL. *Linux Journal*, 2010, 1-8.

Bhardwaj, S., Jain, L., & Jain, S. (2010). An Approach for Investigating Perspective of Cloud Software-as-a-Service (SaaS). *International Journal of Computer Applications.*, 10(2), 44-47.

Bhardwaj, S., Jain, L., & Jain, S. (2010). Cloud computing: A study of infraestructure as a services ( IaaS ). *Int. J. Eng. Inf. Technol.*, 2(1), 60-63.

Bobrowski, S. (2011). Optimal Multitenant Designs for Cloud Apps. 2011 IEEE 4th International Conference on Cloud Computing, 654-659.

Borkar, V., Carey, M., & Li, C. (2012). Inside Big Data Management: Ogres, Onions, or Parfaits? 15th International Conference on Extending Database Technology, 3-14.

Borthakur, D. R., & Gray, J. S. (2011). Apache hadoop goes realtime at Facebook. *Proceedings of the 2011 international conference on Management of data - SIGMOD 11*, 71. Obtenido de Borthakur, D., Rash, S., Schmidt, R., Aiyer,

A.; Gray, J., Sen Sarma, J., Muthukkaruppan, K., Spiegelberg, N., Kuang, H., Ranganathan, K., Molkov, D., Menon, A. "Apache hadoop goes realtime at Facebook," in Proceedings of the 2011 international conferenc.

Bowers, K., Juels, A., & Oprea, A. (2009). HAI : A High-Availability and Integrity Layer for Cloud Storage. The 16th ACM conference on Computer and communications security, 1-12.

Brewer, E. (2000). Towards Robust Distributed Systems. ACM Symposium on Principles of Distributed Computing, 7.

Buyya, R., Yeo, C. V., Broberg, J., & Brandic, I. (2009). "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems, 25(6), 599-616.

Calder, B., Wang, J., Ogus, A., Nilakantan, N., Skjolsvold, A., Mckelvie, S., . . . Dayanand, S. (2011). Windows Azure Storage: A Highly Available Cloud Storage Service with Strong Consistency. 143-157.

Cattell, R. (2010). Scalable SQL and NoSQL Data Stores. 39, 12-27.

Chang, F., Dean, J., Ghemawat, S., Hsieh, W., Wallach, D. B., Chandra, T., . . . Gruber, R. (2006). Bigtable : A Distributed Storage System for Structured Data.

Chang, V., Bacigalupo, D., Wills, G., & De Roure, D. (2010). V. Chang, D. Bacigalupo, G. Wills, y D. De Roure, "A Categorisation of Cloud Computing Business Models," 10th

IEEE. V. Chang, D. Bacigalupo, G. Wills, y D. De Roure, "A Categorisation of Cloud Computing Business Models," ACM International Conference on Cluster, Cloud and Grid Computing, 509-512.

Choo, K. (2010). Cloud computing : Challenges and future directions. Trends & Issues in Crime and Criminal Justice, 1(400), 1-8.

Corporation, E. (11 de 04 de 2013). CIO Top of Mind for 2013. Obtenido de [http://www.idgconnect.com/view\\_abstract/13330/cio-top-mind-2013](http://www.idgconnect.com/view_abstract/13330/cio-top-mind-2013)

Dean, J. A. (2010). MapReduce: A Flexible Data Processing Tool. Commun. ACM, 53(1), 72-77.

El Abbadi, E. (2011). Big Data and Cloud Computing : Current State and Future Opportunities. 14th International Conference on Extending Database Technology, 1-3.

F. Wang, J. Q. (2009). Hadoop high availability through metadata replication . Proceeding of the first international workshop on Cloud data management - CloudDB 09, 37.

Foley, M. (11 de 04 de 2013). Microsoft drops Dryad; puts its big-data bets on Hadoop. Ob t e n i d o d e <http://www.zdnet.com/blog/microsoft/microsoft-drops-dryad-puts-its-big-data-bets-on-hadoop/11226>

Foster, I., Zhao, Y., & Raicu, I. L. (2008). Cloud Computing and Grid Computing 360-Degree Compared. Grid Computing Environments Workshop, 1-10.

Gartner. (05 de 05 de 2013). Big Data, IT Glossary 2012. Obtenido de <http://www.gartner.com/it-glossary/big-data/Gartner>. (03 de 05 de 2013). Gartner definition of cloud computing. Obtenido de Gartner definition of cloud computing, 2012: <http://www.gartner.com/it-glossary/cloud-computing/>

Ghemawat, S., Gobioff, H., & Leung, S. (2003). The Google file system. *ACM SIGOPS Operating Systems Review*, 37(5), 29.

Godavari, W. (2010). Reviewing Some Platforms in Cloud. *International Journal of Engineering*, 2(5), 348-353.

Google. (09 de 04 de 2013). MapReduce Overview. Obtenido de <https://developers.google.com/appengine/docs/python/dataprocessing/overview>

Haerder, T., & Reuter, A. (1983). Principles of transaction-oriented database recovery. *ACM Computing Surveys*, 15(4), 287-317.

Hu, W., Yang, T., & Matthews, J. (2010). The good, the bad and the ugly of consumer cloud storage. *ACM SIGOPS Operating Systems Review*, 44(3), 110.

IDC. (10 de 4 de 2013). IDC Latin America Predictions 2013. Obtenido de IDC Latin America Predictions: <http://www.idclatin.com/campaign/predictions/>

Kambatla, K. P. (2009). Towards Optimizing Hadoop Provisioning in the Cloud. *HotCloud 09*, 22.

Konstantinou, I., Angelou, E., Tsoumakos, D., & Koziris, N. (2010). Distributed indexing of web scale datasets for the cloud. *Proceedings of the 2010 Workshop on Massive Data Analytics on the Cloud - MDAC 10.*, 1-6.

Leavitt, N. (2010). Will NoSQL databases live up to their promise? *Computer*, 3(2), 12-14.

Marks, E., & Lozano, B. (2010). *Executives Guide to Cloud Computing*. Nueva York: Wiley & Sons, Inc.

Mell, P., & Grance, T. (2008). *The NIST Definition of Cloud Computing: Recommendations of the National Institute of Standards and Technology*. NIST Special Publication., 145.

Microsoft. (07 de 04 de 2013). Dryad . Obtenido de <http://research.microsoft.com/en-us/projects/dryad>

Mietzner, R., Unger, T., Titze, R., & Leymann, F. (2009). Combining Different Multi-Tenancy Patterns in Service-Oriented Applications. *Enterprise Distributed Object Computing Conference*, 131-140.

Nurmi, D., Wolski, R., Grzegorzczak, C., Obertelli, G., Soman, S., Youseff, L., & Zagorodnov, D. (2009). The Eucalyptus Open-Source Cloud-Computing System. *2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, 124-131.

Padhy, R., Patra, M., & Satapathy, S. (2011). "RDBMS to NoSQL: Reviewing Some Next-Generation Non-Relational Databases.

International Journal of Advanced Engineering Science and Technologies, 11(1), 15-30.

Pearson, S., Shen, Y., & Mowbray, M. (2009). "A Privacy Manager for Cloud Computing. Lecture Notes in Computer Science, 5931, 90-106.

Sciore, E., & Hill, C. (2007). SimpleDB : A Simple Java-Based Multiuser System for Teaching Database Internals. 28th SIGCSE Technical Symposium on Computer Science Education, 561565.

Shao, B., & Wang, H. (2012). Managing and Mining Large Graphs : Systems and Implementations. ACM International Conference on Management of Data (SIGMOD), 1, 589592.

Stonebraker, M. (2011). Stonebraker on NoSQL and enterprises. Communications of the ACM, 54(8), 10-11.

Strauch, C. (2011). NoSQL databases: a step to database scalability in web environment. 13th International Conference on Information Integration and Web-based Applications and Services, C. Strauch, "NoSQL databases: a step to database scalability in web environment." 13th International Co278-283.

Trelles, O., Prins, P., Snir, M., & Jansen, R. (2011). Big data, but are we ready? Nature reviews. Genetics, 12(3), 224.

Vouk, M. (2008). Cloud computing Issues, research and implementations. ITI 2008 - 30th International Conference on Information Technology Interfaces, 31-40.

Wang, F. Q., Yang, J., Dong, B., Li, X., & Li, Y. (2010). Distributed Systems Meet Economics : Pricing in the Cloud. En H. Wang, Q. Jing, R. Chen, B. He, Z. Qian, & L. Zhou, 2nd USENIX conference on Hot topics in cloud computing (pág. 6).

Xiang, P., Hou, R., & Zhou, Z. (2010). Cache and Consistency in NOSQL Consistency hash algorithm. Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on, 2010, 6, 117-120.

Zhou, M., States, U., Grimmer, J., King, G., & Science, Q. (3 de 5 de 2013). The Age of Big Data . O b t e n i d o d e <http://damfoundation.org/2012/02/the-age-of-big-data>

11 M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, y M. Zaharia, "Above the Clouds : A Berkeley View of Cloud Computing Cloud Computing: An Old Idea Whose Time Has ( Finally ) Come,". 2009. pp. 7-13.