

Tipo de artículo: Artículo original
Temática: Tecnologías de bases de datos
Recibido: 25/03/2016 | Aceptado: 30/06/2016

Selección de Base de Datos No SQL para almacenamiento de Históricos en Sistemas de Supervisión

No SQL Database selection for historical storage in supervision and control systems

Aylin Rodríguez Pérez^{1*}, Dairon Rodríguez Hernández¹, Elizabeth Díaz Martínez¹

¹ Universidad de las Ciencias Informáticas, Cuba, {arodriguezp, daironrh, edmartinez}@uci.cu

*Autor para correspondencia: arodriguezp@uci.cu

Resumen

El almacenamiento y procesamiento de históricos para sistemas de supervisión y control en escenarios industriales muestran problemáticas ante el manejo de grandes volúmenes de datos gestionados con el modelo relacional. La presente investigación tiene como objetivos analizar el estado actual de estos sistemas de supervisión y control, sus ventajas y desventajas, y realizar un estudio del arte para la selección de una base de datos No SQL para el registro de series de tiempo. El desarrollo del trabajo se centró en la comparación de los gestores de almacenamiento Orientados a Columnas que son: *HBase*, *Cassandra* y *Big Table*. Este último fue eliminado de la comparación detallada pues su licencia es propietaria. Luego de diseñar e implementar dos casos de pruebas para comparar el comportamiento de ambos gestores ante los escenarios de Lecturas Intensivas y Lectura/Escritura, se concluyó que ambos poseen excelentes prestaciones para el almacenamiento de históricos. Se sugiere, el análisis y desarrollo de pruebas específicas acorde al dominio de aplicación de estas dos tecnologías.

Palabras clave: base de datos, históricos, no SQL, supervisión y control.

Abstract

The storage and processing for historical supervisory and control systems in industrial settings to show problems handling large volumes of data managed by the relational model. This research aims to analyze the current state of these supervisory and control systems, their advantages and disadvantages and make an art studio for a lesson is-not

sql data base for recording time series. The development work focused on comparing storage managers oriented columns are: Hbase, Cassandra and Big Table. The latter was removed from the detailed comparison because it's proprietary license. After design and implement two test cases to compare the performance of both managers to the stage and intensive reading / writing reading was concluded that both have excellent facilities for the storage of historical. It is suggested, analysis and development according to specific application domain evidence of these two technologies.

Keywords: *databases, historical, no SQL, supervisory and control.*

Introducción

El almacenamiento de la información histórica consiste en el registro de grandes concentraciones de datos, a partir de muestras de información emitidas por determinado sistema. Las bases de datos históricas juegan un papel clave en los sistemas de control industrial, quienes almacenan, procesan y proveen la información necesaria para las tareas de supervisión, registro de eventos, gestión y mantenimiento en el sector industrial. (Bitincka, 2010)

Entre los principales retos de las bases de datos históricas se destaca el manejo de grandes volúmenes de información, de modo que permita realizar funciones como el registro, lectura, procesamiento y compresión de la información con adecuados niveles de escalabilidad. La escalabilidad es la propiedad de los sistemas informáticos que estudia su comportamiento durante su crecimiento continuo. (Hyuck, 2011)

Para los sistemas de control esta cualidad es inherente al proceso de mantenimiento y actualización de la plataforma acorde a los requerimientos del negocio. Las labores de incorporación de nuevos dispositivos de control, así como las reestructuraciones de infraestructura y administrativas en las áreas operacionales, convierten a las industrias en un escenario complejo para la evolución de los sistemas informáticos. (Ombergen, 2014)

El almacenamiento de históricos se ha realizado desde los años 80 fundamentalmente con tecnologías de bases de datos relacionales, sin embargo, el modelo relacional introduce limitantes derivadas de su propia concepción que coadyuvan a un comportamiento ineficiente cuando los sistemas crecen, que se manifiesta con la reducción significativa del rendimiento y la disponibilidad para la gestión de series de tiempo (Strauch, 2011).

En la última década han surgido nuevas tendencias y tecnologías de almacenamiento que encaran estas problemáticas,

entre ellas las conocidas Base de Datos NoSQL. Estos proponen una estructura de persistencia más versátil, aunque en algunos casos se pierden ciertas funcionalidades como las transacciones y la incapacidad de hacer uniones de tablas. Entre sus principales atributos se destacan la ausencia de esquema en los registros de datos, escalabilidad horizontal sencilla y velocidad (Strauch, 2011).

El presente artículo realiza una investigación de las tecnologías NoSQL con el objetivo de seleccionar un gestor de base de datos acorde a los requerimientos para el almacenamiento de históricos en sistemas de supervisión y control industrial para entornos críticos donde se requieran sistemas en tiempo real blando (*Soft Real Time Systems*). (Laplante, 2012).

El desarrollo abarca el análisis de las diversas estructuras de las bases de datos No SQL, la definición de los atributos que guiarán el proceso de selección, la comparación y su fundamentación.

Materiales y métodos o Metodología

Métodos Teóricos:

Histórico-lógico: se utiliza para el estudio de los distintos sistemas gestores de base de datos NoSQL

Análisis-Síntesis: se requiere para el estudio de la información sobre NoSQL, que permita su desglose progresivo y el resumen para arribar a ideas generales.

Métodos Empíricos:

Observación: utilizado como medio para la adquisición de conocimientos, en los intercambios realizados con el personal del Centro de Informática Industrial (CEDIN).

Estructuras de las No SQL

Las bases de datos No SQL implementan diferentes estructuras de almacenamiento, las más extendidas son Clave-Valor, Columnas, Documentos y Grafos (López, 2013). En la **Tabla 1** se presenta un resumen de sus principales características, el desarrollo de esta investigación se centrará en las bases de datos orientadas a columnas, las cuales son las más utilizadas para el registro de actividades de monitoreo y series de tiempo.

Tabla 1. Categorías de las bases de datos No SQL

Categoría	Descripción
Clave-Valor	Usando una tabla hash donde cada clave única referenciará un dato particular, esta estructura lo convierte en el modelo más simple y fácil de implementar. Provee rapidez en las búsquedas y sus debilidades son cuando se necesita trabajar con solo una parte del Valor.
Columnas	Creadas para almacenar y procesar grandes cantidades de datos distribuida en varias estaciones, se mantiene la utilización de claves, pero estas referencian múltiples columnas, quienes a su vez se organizan en familias. Provee rapidez en las búsquedas y realiza una buena distribución de la información almacenada.
Documentos	Consiste en documentos versionados que son colecciones de parejas clave-valor, mejora el problema de gestionar solo una parte del valor.
Grafos	Usa un modelo flexible basado en grafos permitiendo caminos más cortos en las búsquedas así como el incremento de las relaciones.

Factores comparativos

Tomando como fundamento los requerimientos de las bases de datos históricas para sistemas de supervisión y control, se identificaron las características que serán el centro del análisis para confeccionar la matriz comparativa que guiará el proceso de selección de la tecnología. En la **Tabla 2** se muestran los principales factores y su descripción.

Tabla 2. Atributos para comparación

Atributo	Descripción
Tipo de almacenamiento	Es la forma en que se guardan los datos en una base de datos. Es una colección de bits que describe tanto lo que son los datos de la base de datos como sus metadatos conexos.
Gestión de series de tiempo	Una serie de tiempo es un conjunto de datos numéricos que se obtienen en períodos regulares a través del tiempo. Los volúmenes potencialmente grandes de datos se pueden gestionar y acceder de una manera de alto rendimiento.
Tipo de licencia	Una licencia de software es un contrato entre el licenciante y el licenciario del programa informático, para utilizar el software cumpliendo una serie de términos y condiciones establecidas dentro de sus cláusulas.
Lenguaje de implementación	Es un idioma artificial diseñado para expresar procesos que pueden ser llevadas a cabo por máquinas como las computadoras.
Almacenamiento de datos	Permite leer o escribir datos en medios o soportes de almacenamiento, y conforma la memoria o almacenamiento secundario de la computadora.
Control de concurrencia	Garantiza la integridad de los datos a pesar de accesos de usuarios concurrentes.

Transacciones	En un Sistema de Gestión de Bases de Datos (SGBD), es un conjunto de órdenes que se ejecutan formando una unidad de trabajo, es decir, en forma indivisible o atómica
Disponibilidad	Capacidad del sistema de funcionar ininterrumpidamente. Garantizar un reducido tiempo entre fallas.
Tolerancia a Fallas	Capacidad del sistema de reaccionar y reponerse ante fallas o situaciones excepcionales.
Consistencia	Garantiza la consistencia de los datos y transacciones.
Complejidad	Analiza la curva de familiarización con la tecnología para la instalación, configuración y administración.
Documentación y Comunidad	Calidad de la documentación, actualizaciones y atención de la comunidad.

Tecnologías a comparar

Una vez definidos los atributos que guiarán la comparación se realizó un estudio del arte de los principales gestores existentes orientados a columnas (Strauch, 2011), (Gómez, 2012), resultando *HBase*, *Cassandra* y *Big Table* los seleccionados. A continuación, se realizará una descripción de sus principales características para conformar la tabla comparativa de las tecnologías.

HBase

El almacenamiento de datos es orientado a columnas y comprende un conjunto de tablas que contienen filas y columnas. Cada tabla debe tener una clave principal, una columna representa un atributo de un objeto. *HBase* permite muchos atributos para ser agrupados juntos en lo que se conoce como familias de columna, los elementos de una familia columna se almacenan todos juntos. Con *HBase* se debe definir previamente el esquema de la tabla y especificar las familias de columnas (Strauch, 2011), (Gómez, 2012).

HBase es como un clon de *Big Table* desarrollado en *Java* como parte de *Apache-MapReduce* marco *Hadoop*, proporcionando un "modo de alta disponibilidad de almacenar grandes cantidades de datos dispersos" (Grehan, 2014).

HBase depende de un sistema de archivos distribuido (HDFS). En este sistema existe la necesidad de compactaciones, compresión de datos, así como filtros para la reducción del acceso al disco. *HBase* puede ser una fuente o un destino para los trabajos *MapReduce* ejecutado a través de *Hadoop*. El uso notable de *HBase* es el sistema de mensajería en

tiempo real de *Facebook* construido sobre *HBase* desde 2010 (Gómez, 2012).

Big Table

Es un sistema de almacenamiento distribuido para gestionar datos estructurados y puede escalar a un tamaño muy grande como: *petabytes* de datos a través de miles de servidores de productos básicos. Google a partir de 2006 lo emplea en sus proyectos (Birgen, 2014).

Ha obtenido hasta el momento relevantes resultados como aplicabilidad amplia, escalabilidad, alto rendimiento y alta disponibilidad. Para *Google* la implementación de *Big Table* tiene demostrado ser ventajosa, pues es impredecible la flexibilidad con la que se ha desarrollado (Strauch, 2011).

Tiene otras características como: distribuido, de alta eficiencia y propietario. Está construido sobre *GFS (Google File System)*, *Chubby Lock Service*, y algunos otros servicios y programas de Google (Strauch, 2011).

Comenzó a partir de 2004. Este gestor almacena la información en tablas multidimensionales. Las celdas disponen de versiones temporales de sus valores, por lo que se puede hacer un seguimiento de los valores históricos (Birgen, 2014).

Cassandra

Como gestor de base de datos está determinado por su escalabilidad y alta disponibilidad sin comprometer el rendimiento. Tiene además gran escalabilidad lineal y probada tolerancia a fallos en el hardware. También brinda apoyo para replicar a través de múltiples centros de datos y es el mejor en su clase, proporcionando una menor latencia para los usuarios y la tranquilidad de saber que usted puede sobrevivir cortes regionales.

Cassandra tiene características como índices de columna con el desempeño de estructuradas actualizaciones, soporte firme para las vistas materializadas y potente capacidad de almacenamiento en caché (Lakshman, 2010)

Permite el uso eficiente para muchas aplicaciones más allá de simple clave / valor. La lectura y escritura tiene gran rendimiento tanto en forma lineal a medida que se añaden nuevas máquinas, sin ningún tiempo de inactividad o interrupción de las aplicaciones. Además, es adecuado para aplicaciones que no pueden permitirse el lujo de perder los datos, incluso cuando un centro de datos deja de funcionar (Gómez, 2012).

Cassandra tiene integración *Hadoop*, *MapReduce* con apoyo. Está conformada por familias de columna que

contienen filas y columnas y a su vez cada fila se identifica por una clave. Cada fila tiene varias columnas, cada una de ellas tiene un nombre, valor y marca de tiempo (Gómez, 2012).

Para lograr una alta escalabilidad y la durabilidad de un clúster de *Cassandra*, los datos se replican en un número de nodos que se pueden definir como un factor de replicación por ejemplo *Cassandra*. La replicación está gestionada por un nodo coordinador de la tecla en particular se está modificando, el nodo coordinador para cualquier clave es el primer nodo en el anillo consistente hash que es visitado al caminar desde la posición de la llave en el anillo en la dirección de las agujas del reloj (Cattell, 2011).

En comparación con *Big Table* y *HBase*, *Cassandra* persiste sus datos a archivos locales en lugar de un distribuido sistema de archivos. Sin embargo, la representación de los datos en la memoria y en el disco, así como el procesamiento de lectura y operaciones de escritura es tomada de *Big Table*. Las operaciones de escritura primero van a un registro de confirmación persistente y luego a una estructura de datos en memoria. A continuación, se muestra en la **Tabla 3** un resumen de los factores y su aplicación en cada gestor.

Tabla 3. *Cassandra vs Big Table vs HBase*.

Características	<i>Cassandra</i>	<i>Big Table</i>	<i>HBase</i>
Tipo de licencia	Apache Open Source	Propietaria	Apache Open Source
Lenguaje de implementación	Java	C C++	Java
Ejemplos de uso	Banca, las finanzas	Gmail, Búsqueda de libros de Google	Mensajería, Sistemas de Monitoreo
Mejores usos	Escribir con frecuencia, leer menos	Diseñado para escalar a través de cientos o miles de máquinas	Lectura aleatoria al escribir en bases de datos grandes
Control de concurrencia	Multiversión de concurrencia y control	Logueado	Logueado
Transacciones	Local	Local	Local
Almacenamiento de datos	Disco	GFS	Hadoop

Gestión de Series de tiempo	Es un excelente complemento para los datos de series de tiempo, y es ampliamente utilizado para almacenar las métricas de rendimiento, seguimiento de flota, datos de sensores, registros, datos financieros.		Las consultas sobre esta serie de tiempo podrían recuperar un intervalo de versiones. Varias aplicaciones de monitoreo se han desarrollado sobre esta plataforma como OpenTSDB.
Disponibilidad	Alta disponibilidad	Alta disponibilidad	La disponibilidad es aceptable, han priorizado elevar los niveles de consistencia.
Tolerancia a Fallas	Elevada	Elevada	Elevada
Consistencia	Consistencia aceptable, priorizaron elevar la disponibilidad.	Alta consistencia	Alta consistencia.
Complejidad	Fácil instalación y configuración. Estructura familiar.	De compleja instalación, configuración y administración	De compleja instalación, configuración y administración. Debe dominarse los sistemas Hadoop y HDFS.
Documentación y Comunidad	Muy buena documentación y comunidad de desarrollo activa	Excelente documentación.	Excelente documentación y comunidad de desarrollo.

Luego de realizar un primer análisis partiendo de las características descritas en el apartado 2.3, descartamos *Big Table* debido a su licencia propietaria. La comparación de los factores para *HBase* y *Cassandra* arroja resultados similares sin grandes diferencias significativas para decidir la selección. Por lo tanto, se procede a realizar un análisis más profundo del comportamiento de estos gestores antes determinados escenarios de uso.

Comparativas (Bechmark)

En esta sección se realizará un análisis más detallado de las tecnologías *Hbase* y *Cassandra*. Dos de los casos de uso más utilizados para la gestión de datos históricos son la Escritura y Escritura/Lectura. A continuación, se describe en detalle las pruebas de rendimiento aplicadas y sus resultados.

Lectura

Este escenario tiene como objetivo probar el comportamiento de los dos gestores ante la lectura intensiva. Para ello se implementa un caso de prueba que realizará el 95% de lecturas y 5% de escrituras durante 10 minutos, se utilizarán 10 millones de registros para cada nodo, tomando como referencia la investigación (Gómez Villamor, et al., 2012). Se limitarán las pruebas a 1 y 3 estaciones que son los el mínimo y máximo de nodos que usualmente se usan en el sector para las tareas de supervisión y control. En la **Tabla 4** se muestran los resultados obtenidos.

Tabla 4. Comparativas para lectura intensiva

Nodos	Factor	<i>HBase</i>	<i>Cassandra</i>
1	Tasa de transferencia (operaciones/segundo)	2.5 K	25 K
3	Tasa de transferencia (operaciones/segundo)	4.5 K	50 K
1	Latencia en la lectura (milisegundos)	50	5
3	Latencia en la lectura (milisegundos)	90	8
1	Latencia en la escritura (milisegundos)	0.2	4
3	Latencia en la escritura (milisegundos)	0.2	6

Analizando los resultados obtenidos se puede concluir parcialmente que *Cassandra* brinda mejores prestaciones que *HBase* para la lectura intensiva. En el caso de la *Tasa de Transferencia* y *Lectura* muestra valores superiores alrededor de 10 veces, sin embargo, en la *Escritura* ocurre lo contrario, aunque las muestras de *HBase* son excelentes en este rubro las de *Cassandra* son adecuadas para el dominio de aplicación de nuestro interés, que se limita a aplicaciones de tiempo real blando (Laplante, 2012).

Lectura / Escritura

Este escenario tiene como objetivo probar el comportamiento de los dos gestores ante el balance de lectura y escritura. Para ello se implementa un caso de prueba que realizará el 50% de lecturas y 50% de escrituras. En la **Tabla 5**, se muestran los resultados obtenidos.

Tabla 5. Comparativas para Lectura/Escritura

Nodos	Factor	<i>HBase</i>	<i>Cassandra</i>
1	Tasa de transferencia (operaciones/segundo)	8 k	30 K
3	Tasa de transferencia (operaciones/segundo)	10 K	60 K

1	Latencia en la lectura (milisegundos)	90	8
3	Latencia en la lectura (milisegundos)	98	9
1	Latencia en la escritura (milisegundos)	0.08	6
3	Latencia en la escritura (milisegundos)	0.08	8

Nuevamente ante las pruebas de Lectura/Escritura se obtienen resultados similares a la prueba precedente. *Cassandra* supera ampliamente en la Tasa de Transferencia y Lectura quedando por debajo en la Latencia durante la Escritura, aunque con niveles muy buenos acordes a los requerimientos del dominio de aplicación.

Resultados y discusión

Luego del desarrollo de los puntos 2,3 y 4 se puede argumentar que *HBase* y *Cassandra* son excelentes tecnologías para gestionar grandes volúmenes de datos y el desarrollo de aplicaciones de monitoreo y control.

Ambos presentan magníficas prestaciones e implementan funcionalidades claves para su actividad como los altos niveles de consistencia y disponibilidad respectivamente. De igual modo destacan sus capacidades para la replicación, tolerancia a fallas y sus niveles de escalabilidad horizontal.

Independientemente de sus semejantes prestaciones. *Cassandra* resaltó en las pruebas de lectura intensiva y lectura escritura, además se comprobó la facilidad de asimilar su estructura y esquema de trabajo, ello propició la agilización del proceso de instalación y configuración de las pruebas, incluyendo productivas experiencias para extender la plataforma e incorporar nuevas funcionalidades. Igualmente destacó su escalabilidad en clústeres pequeños.

Conclusiones

Los sistemas de bases de datos han dado un giro significativo en la última década. Una muestra de ello ha sido el desarrollo de nuevos modelos de almacenamiento como las bases de datos No SQL, que se han consolidado exitosamente en determinados dominios como las redes sociales y otras aplicaciones donde los datos crecen con gran velocidad.

Centrado en los gestores de almacenamiento Orientados a Columnas fueron seleccionados *HBase*, *Cassandra* y *Big Table*. Este último fue eliminado de la comparación detallada pues su licencia es propietaria. Luego de diseñar e

implementar dos casos de pruebas para comparar el comportamiento de ambos gestores ante los escenarios de Lecturas Intensivas y Lectura/Escritura se concluyó que ambos poseen excelentes prestaciones para el almacenamiento de históricos. Se sugiere, el análisis y desarrollo de pruebas específicas acorde al dominio de aplicación de estas dos tecnologías.

A pesar de ello *Cassandra* presentó un predominio en factores como la Tasa de Transferencia y Latencia en la lectura, además posee una estructura de fácil asimilación para su instalación y administración, así como una gran comunidad, documentación y capacidades de extensión que le hacen más atractivo y práctico de usar.

Referencias

BIRGEN, CANSU. *SQL vs. NoSQL*. Documento. Advanced Simulation, 2014. [Consultado el: 20 de enero de 2016]. Disponible en:

[http://folk.ntnu.no/preisig/HAP_Specials/AdvancedSimulation_files/2014/AdvSim-2014__Birgen_Cansu_Databases.pdf.]

LAKSHMAN, AVINASH Y MALIK, PRASHANT *Cassandra: a decentralized structured storage system..* ACM SIGOPS Operating Systems Review, 2010, Vol. 44 (2): p. 35-40. ISSN: 0163-5980.

GREHAN, RICK . 2014. *Apache HBase offers extreme scalability, reliability, and flexibility, but at the cost of many moving parts*. Documento. INFOWORLD. [En línea] 31 de Marzo de 2014. [Consultado el: 20 Febrero del 2016]. Disponible en: [<http://www.infoworld.com/article/2610709/database/review--hbase-is-massively-scalable-and-hugely-complex.html>.]

HYUCK HAN, SANG Y KUK KIM, YOUNG. 2011. *An Architecture of Real-Time, Historical Database System for Industrial Process Control and Monitoring*. IEEE, Computer Society. [En línea] Mayo de 2011. Disponible en: [<https://www.computer.org/csdl/proceedings/cnsi/2011/4417/00/4417a031.pdf>.] ISBN: 978-0-7695-4417-5.

LÓPEZ IPIÑA, DIEGO . 2013. *Conceptos básicos de NoSQL. Introducción a Cassandra, CouchDB,*

Editorial “Ediciones Futuro”
Universidad de las Ciencias Informáticas. La Habana, Cuba
rcci@uci.cu

MongoDB y Neo4j. Santiago de Compostela, Universidade de Santiago de Compostela, España : s.n., 17 de Julio de 2013.

OMBERGEN, STEVEN VAN. *A Comparison of Five Document-Store Query Languages*. 2014. [En línea, Consultado el: 25 de Febrero del 2016]. Disponible en: [[http://dare.uva.nl/cgi/arno/show.cgi?fid=524062.](http://dare.uva.nl/cgi/arno/show.cgi?fid=524062)]

BITINCKA, LEDION , y otros. *Optimizing data analysis with a semi-structured time series database*. 2010. s.l. : USENIX Association Berkeley, CA, USA, 2010, ACM/Digital Library, págs. 3-37.

LAPLANTE, PHILLIP A. *Real-Time Systems Design and Analysis: An Engineer's Handbook.*. 2012. s.l. : Wiley-IEEE Press, ACM/Digital Library, pág. 384. ISBN:0780334000.

CATTELL, RICK. *Scalable SQL and NoSQL data stores*. 2011. 4, New York : s.n., 6 de Mayo de 2011, SIGMOD ACM Special Interest Group on Management of Data , Vol. 39, págs. 12-27 . ISSN: 0163-5808.

GÓMEZ VILLAMOR, SERGIO , y otros. *Solving big data challenges for enterprise application performance management.*. 2012. s.l. : Publications by Affiliated Organizations , 1 de Agosto de 2012, ACM/Digital Library, Vol. 5 , págs. 1724-1735. ISSN: 2150-8097.

STRAUCH, CHRISTOF . 2011. *Selected Topics on Software-Technology*. [En línea] 2011. [Consultado el: 10 Diciembre del 2015]. Disponible en: [[http://www.odbms.org/2014/03/nosql-databases-2/.](http://www.odbms.org/2014/03/nosql-databases-2/)]