

Una comparación de rendimiento entre bases de datos NoSQL: MongoDB y ArangoDB

A performance comparison between NoSQL databases: MongoDB and ArangoDB

Marlen Treviño-Villalobos¹, Leonardo Víquez-Acuña²,
Rocío Quirós-Oviedo³, Gaudy Esquivel-Vega⁴

Treviño-Villalobos, M; Víquez-Acuña, L;
Quirós-Oviedo, R; Esquivel-Vega, G. Una comparación
de rendimiento entre bases de datos NoSQL: MongoDB y
ArangoDB. *Tecnología en Marcha*. Vol. 32, Especial. VIII
Encuentro de Investigación y Extensión. Abril 2019.
Pág 5-15.

DOI: <https://doi.org/10.18845/tm.v32i6.4223>



- 1 Instituto Tecnológico de Costa Rica. Carrera de Computación. San Carlos. Costa Rica. Correo electrónico: mtrevino@tec.ac.cr
- 2 Instituto Tecnológico de Costa Rica. Carrera de Computación. San Carlos. Costa Rica. Correo electrónico: lviquez@tec.ac.cr
- 3 Instituto Tecnológico de Costa Rica. Carrera de Computación. San Carlos. Costa Rica. Correo electrónico: rocio.quirós@tec.ac.cr
- 4 Instituto Tecnológico de Costa Rica. Carrera de Computación. San Carlos. Costa Rica. Correo electrónico: gesquivel@tec.ac.cr

Resumen

En la actualidad, se genera gran cantidad de información y datos, es por esto que aparecen soluciones alternativas a lo ya existente, tal es el caso de las bases de datos no relacionales o NoSQL, dichas bases de datos funcionan mejor ante un gran volumen de datos en comparación con las SQL. Sin embargo, se han publicado varias investigaciones en cuanto al rendimiento de las bases de datos NoSQL, pero no existe mucha información acerca del comportamiento que tienen estos Sistemas Gestores de Bases de Datos (SGBD) con datos geográficos, es por esto que el presente artículo plantea una comparación en cuanto al tiempo de respuesta entre los SGBD MongoDB y ArangoDB. Para realizar esta comparación se realizaron pruebas en dos escenarios de prueba, denominados 100% lectura y 95% lectura–5% escritura. Además, para ambos escenarios se ejecutaron 10 diferentes tamaños de operación. Los resultados obtenidos permiten concluir que, en ambos escenarios estudiados, el SGBD que presentó un mejor desempeño fue el de MongoDB.

Palabras clave

MongoDB; ArangoDB; Rendimiento; Bases de datos; JMeter.

Abstract

Currently, a large amount of information and data is generated, which is why alternative solutions appear to what already exists, such as non-relational databases or NoSQL, these databases work better with a large volume of data compared to the SQL. However, several investigations have been published regarding the performance of NoSQL databases, but there is not much information about the behavior of these Database Management Systems (DBMS) with geographic data, that is why this article raises a comparison in terms of the response time between the MongoDB and ArangoDB DBMS. To make this comparison, tests were carried out in two test scenarios, called 100% reading and 95% reading–5% writing. In addition, 10 different operation sizes were executed for both scenarios. The results obtained allow to conclude that, in both scenarios studied, the SGBD that presented a better performance was that of MongoDB.

Keywords

MongoDB; ArangoDB; performance; databases; JMeter.

Introducción

Existen diferentes maneras de organizar la información permitiendo mayor sencillez de almacenamiento y eficiencia en el acceso a los datos, como lo son las bases de datos relacionales y no relacionales. Tradicionalmente, las aplicaciones han sido diseñadas usando sistemas administradores de bases de datos relacionales, los que muestran limitaciones en la escalabilidad horizontal y la elasticidad [1]. Las bases de datos relacionales preceden a los modelos jerárquicos y en red, además se les denomina de esta forma debido a que cada registro de la base de datos contiene información relacionada con un tema y sólo con un tema [2]. Estas funcionan mejor cuando manejan una cantidad limitada de datos, sin embargo ante el incremento de datos generado por el Internet se vuelven menos eficientes [3]. Es por lo anterior, que surge el término NoSQL, principalmente para referirse a almacenamientos distribuidos de datos no relacionales. NoSQL se define como un sistema de base de datos que se distribuye,

puede no requerir esquemas de tablas fijas, generalmente se escala horizontalmente y evita las operaciones de unión, no expone una interfaz SQL y puede ser de código abierto [4].

Por otro lado, uno de los campos de datos que generan un gran volumen de información como parte del desarrollo de la sociedad digital es el de los Sistemas de Información Geográfica (SIG), lo anterior debido a que la generación de datos geolocalizados aumenta de forma exponencial a partir de multitud de sensores que registran la actividad humana y de dispositivos con los que los usuarios interactúan [5]. Siendo uno de los mecanismos utilizados para el almacenamiento de dichos datos, las bases de datos espaciales.

En el geoportal denominado IDEHN (Infraestructura de Datos Espaciales de la Región Huetar Norte), se cuenta con una base de datos espacial desarrollada en PostgreSQL [6] que contiene información de cantones, distritos, ríos, hoteles, entre otros y utiliza el servidor de mapas Geoserver [7] para que los usuarios puedan compartir y editar datos geoespaciales mediante la utilización de geoservicios. Sin embargo, durante el desarrollo de dicho geoportal se logró identificar que este servidor de mapas está desaprovechando las ventajas ofrecidas por múltiples paradigmas de bases de datos (como el NoSQL). Por lo que se dispuso realizar una investigación que permita evaluar el rendimiento del geoservicio WMS de Geoserver con una base de datos híbrida y distribuida que utilice los paradigmas SQL y NoSQL. Para identificar la base de datos NoSQL que trabaje con PostgreSQL se debe seleccionar de entre más de 225 bases de datos [8], pero solamente un número reducido soporta operaciones con datos geográficos, dentro de las más destacadas en esta área se encuentran Neo4j, CouchBase, MongoDB [9] y ArangoDB. Para este estudio se decidió analizar el rendimiento y el tiempo de respuesta de los SGBD MongoDB y ArangoDB con el objetivo de poder recomendar el más adecuado para el desarrollo de la base de datos híbrida y distribuida para Geoserver.

La selección de estos dos SGBD se debió fundamentalmente a que MongoDB es un almacén de datos de documentos de código abierto que ofrece un modelo de datos sin esquemas [10]; soporta consultas basadas en cursor, estilo unión, tipos de datos complejos, coincidencia de claves, clasificación, filtros, agregaciones, uniones, disparadores, documentos anidados y referencias. Además, MongoDB es usada en aplicaciones famosas como Foursquare, Sourceforge, Google, Facebook o New York Times [9]. De igual forma, ArangoDB es un sistema de código abierto que implementa un modelo de datos integrando documentos, grafos y clave-valor. Es compatible con el lenguaje de consulta AQL (ArangoDB Query Language), que permite uniones, operaciones en gráficos, iteraciones, filtros, proyecciones, ordenamiento, agrupamiento, agregado funciones, unión e intersección; así como con todas las propiedades ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad, por sus siglas en inglés) [11].

Se identificaron una variedad de trabajos que comparan el rendimiento de bases de datos NoSQL [12] [13] [14] [15]. Sin embargo, la mayoría de estos estudios no considera el aspecto geográfico pues utiliza cargas de datos sintéticas. La presente investigación compara los SGBD MongoDB y ArangoDB utilizando la herramienta de medición JMeter [16] y haciendo uso de datos espaciales correspondientes a la Región Huetar Norte de Costa Rica. Se analizaron dos escenarios de prueba (100% lectura y 95%-5% lectura-escritura) con 10 tamaños diferentes de operación.

Los resultados obtenidos demuestran que en los dos escenarios y con todos los 10 tamaños de operación evaluados, MongoDB tuvo un mejor desempeño. Por lo tanto, se propone la utilización de MongoDB como el SGBD más adecuado para el desarrollo de la base de datos híbrida y distribuida que utilice los paradigmas SQL y NoSQL para la IDEHN.

Metodología

Esta investigación tiene un enfoque cuantitativo, pues se valoró el tiempo de respuesta y rendimiento de los SGBD MongoDB y ArangoDB. Además, es un estudio experimental, ya que se hicieron pruebas con la herramienta de medición del comportamiento de los SGBD conocida como JMeter; que es un instrumento electrónico que sirvió como método de recolección de los datos [16]. La selección de esta herramienta con respecto a otras similares, se basó principalmente en que posee una interfaz gráfica de usuario sencilla, ofrece una gran capacidad de generación de carga, es de código abierto e implementada en Java [17]. Asimismo, es un ambiente que permita controlar las variables. Es decir, las operaciones son diseñadas y administradas por el equipo de pruebas y la base de datos usada corresponde a una muestra de la base de datos real del proyecto. Para poder evaluar los resultados se utilizó el componente provisto por la herramienta, que se denomina Summary Report y permite visualizar los resultados de la prueba realizada en una tabla. Los datos que presenta dicho componente son [18]:

- Label: etiqueta de la muestra.
- #Muestras: cantidad de hilos utilizados.
- Media: tiempo de respuesta promedio en milisegundos para un conjunto de resultados.
- Min: tiempo mínimo que demora un hilo en acceder a una página.
- Max: tiempo máximo que demora un hilo en acceder a una página.
- Rendimiento: rendimiento medido en los requerimientos por segundo / minuto / hora.
- Kb/sec: rendimiento medido en Kilobytes por segundo.
- Media en bytes: tamaño medio de respuesta del servidor (en bytes).

Seguidamente se describirán los elementos de la metodología:

A. Datos de prueba

Se optó por utilizar un conjunto de datos correspondiente a la Región Huetar Norte de Costa Rica e incluye una variedad de tipos de representación que van desde estructuras geométricas basadas en puntos, líneas o polígonos. Se eligieron 3 archivos en formato shapefile [19] del sitio web www.idehn.tec.ac.cr, estos son: Ríos de la Región Huetar Norte de Costa Rica que se representan como líneas, Poblados de la Región Huetar Norte de Costa Rica que corresponde a puntos y Distritos de la Región Huetar Norte de Costa Rica que se muestra como polígonos. Posteriormente, dichas capas de datos geográficos se convirtieron a formato GeoJSON [20] por medio de la herramienta QGIS [21]. Para ambos casos, las capas de datos se utilizaron con el sistema de referencia de coordenadas WGS84 [22] asociado al SRID (Spatial Reference System Identifier) número 4326.

B. Carga de datos

La carga de datos se realizó de forma manual por lo que se debió leer, procesar y almacenar en disco los datos. Como se mencionó anteriormente, se utilizaron tres conjuntos de datos vectoriales. La capa de ríos contiene 4150 registros geográficos, la de poblados 658 y la de distritos 36. Tanto MongoDB como ArangoDB tienen su propia herramienta de importación. Para este caso, Mongoimport y Arangoimp respectivamente tradujeron los archivos GeoJSON.

Escenarios de pruebas

Se identificaron en otras publicaciones una serie de escenarios de prueba [23], estos son:

- Escenario A: principalmente actualización: 50/50 lectura y actualización.
- Escenario B: Lectura principalmente: 95/5 lectura/actualización.
- Escenario C: Solamente lectura: 100% lectura.
- Escenario D: Leer sobre la última carga de trabajo: lee los registros recientemente insertados.
- Escenario E: Rangos cortos: la consulta se hace sobre rangos cortos de registros, en lugar de registros individuales.
- Escenario F: lectura-modificación-escritura, para esta carga de trabajo, el cliente lee un registro, lo modifica y vuelve a escribir los cambios.

Debido a que la mayoría de operaciones que se realizan sobre la base de datos de IDEHN son de consulta, se seleccionaron los escenarios B y C para esta investigación. En el cuadro 1, se detallan los dos escenarios de pruebas con la cantidad de operaciones ejecutadas. Se definió este último punto, con el fin de observar el comportamiento de los SGBD ante el incremento del número de operaciones.

Cuadro 1. Escenarios de pruebas

Carga de Trabajo	Operaciones	Cantidad de Operaciones
Solamente Lectura	100% Lectura	1000 Operaciones 2000 Operaciones 3000 Operaciones 4000 Operaciones 5000 Operaciones 6000 Operaciones 7000 Operaciones 8000 Operaciones 9000 Operaciones 10000 Operaciones
Lectura pesada	95% Lectura y 5% Escritura	1000 Operaciones 2000 Operaciones 3000 Operaciones 4000 Operaciones 5000 Operaciones 6000 Operaciones 7000 Operaciones 8000 Operaciones 9000 Operaciones 10000 Operaciones

Ambiente de pruebas

Todas las pruebas se realizaron en una máquina virtual con el sistema operativo Ubuntu 16.04 LTS de 64 bits con 16 GB de memoria RAM. Además, las versiones de los SGBD son: MongoDB Server 3.4.10 [24], y ArangoDB 3.3.4.1 [25]. Finalmente, la versión de la herramienta utilizada para la evaluación y comparación del rendimiento fue Apache JMeter 3.2 [16]. Los drivers utilizados para la conexión con cada SGBD son MongoDB 2.11.3 Java driver y ArangoDB 4.2.4 Java driver.

Resultados

Solamente lectura

Primero, se examinará la ejecución de la carga de trabajo denominada “solamente lectura” con 10 diferentes números de operaciones tanto en MongoDB como en ArangoDB. En el cuadro 2, se aprecia que conforme aumenta el número de operaciones en ambos SGBD se presenta un incremento del rendimiento. No obstante, el rendimiento de ArangoDB varía muy poco en relación al número de operaciones ejecutadas, mientras que en MongoDB las variaciones son más significativas. En la figura 1, se puede apreciar visualmente que en todos los números de operaciones en este escenario de pruebas MongoDB fue superior que ArangoDB. Por ejemplo, con 10000 operaciones MongoDB superó a ArangoDB en un 3292%.

Cuadro 2. Resultados escenario solamente lectura

Número de operaciones	MongoDB		ArangoDB	
	Rendimiento	Tiempo de respuesta	Rendimiento	Tiempo de respuesta
1000	809,06	5,81	45,69	102,51
2000	1060,45	4,49	47,18	101,34
3000	1303,78	3,66	47,69	99,75
4000	1341,83	3,55	48,96	98,56
5000	1405,68	3,43	48,12	100,13
6000	1574,39	3,03	46,32	104,43
7000	1541,85	3,13	47,83	100,55
8000	1500,09	3,22	47,75	100,92
9000	1553,06	3,11	47,53	101,76
10000	1691,76	2,87	49,88	96,94

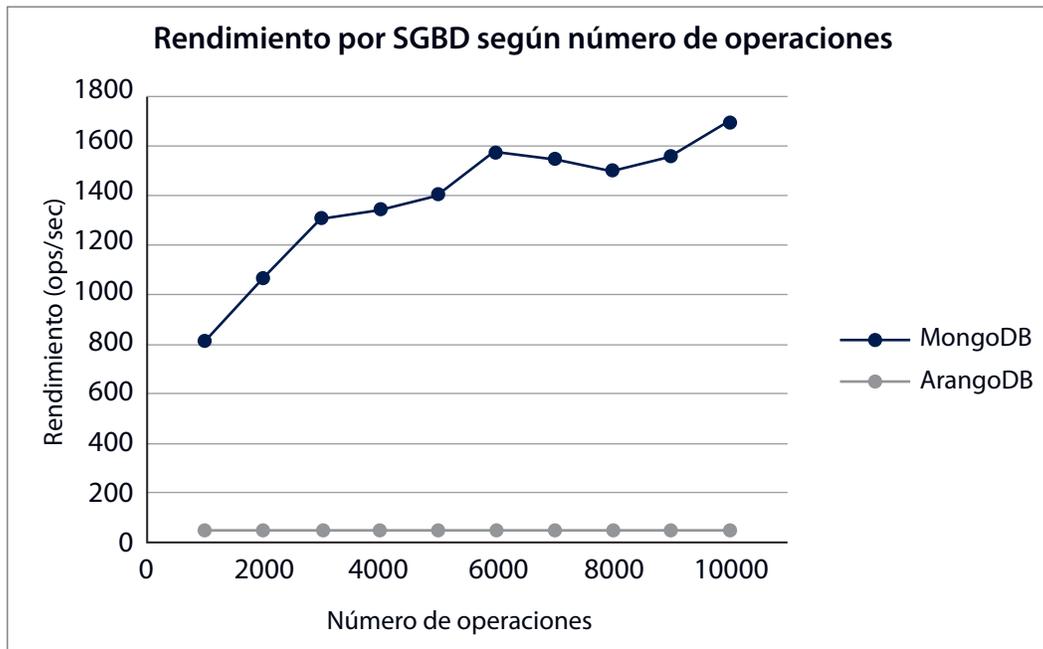


Figura 1. Rendimiento en milisegundos por SGBD según número de operaciones

Es probable, que los rendimientos de MongoDB se deban a que este SGBD utiliza una indexación similar a las bases de datos relacionales, permitiendo de este modo una ejecución eficiente de las consultas [26].

Lectura pesada

Igual que en el escenario anterior, los resultados obtenidos en este escenario de pruebas reflejan que MongoDB tiene un mejor rendimiento que ArangoDB (ver cuadro 3 y figura 2).

Cuadro 3. Resultados escenario lectura pesada

Número de operaciones	MongoDB		ArangoDB	
	Rendimiento	Tiempo de respuesta	Rendimiento	Tiempo de respuesta
1000	784,93	6,9	46,55	102,05
2000	1083,42	4,95	51,07	94,73
3000	1326,26	4,05	48,44	100,05
4000	1402,03	3,78	49,94	96,74
5000	1543,21	3,45	51,11	94,48
6000	1515,53	3,52	50,01	97,13
7000	1375,52	3,75	50,22	96,92
8000	1512,57	3,50	49,68	97,59
9000	1471,55	3,52	49,14	98,86
10000	1568,38	3,37	50,07	96,72

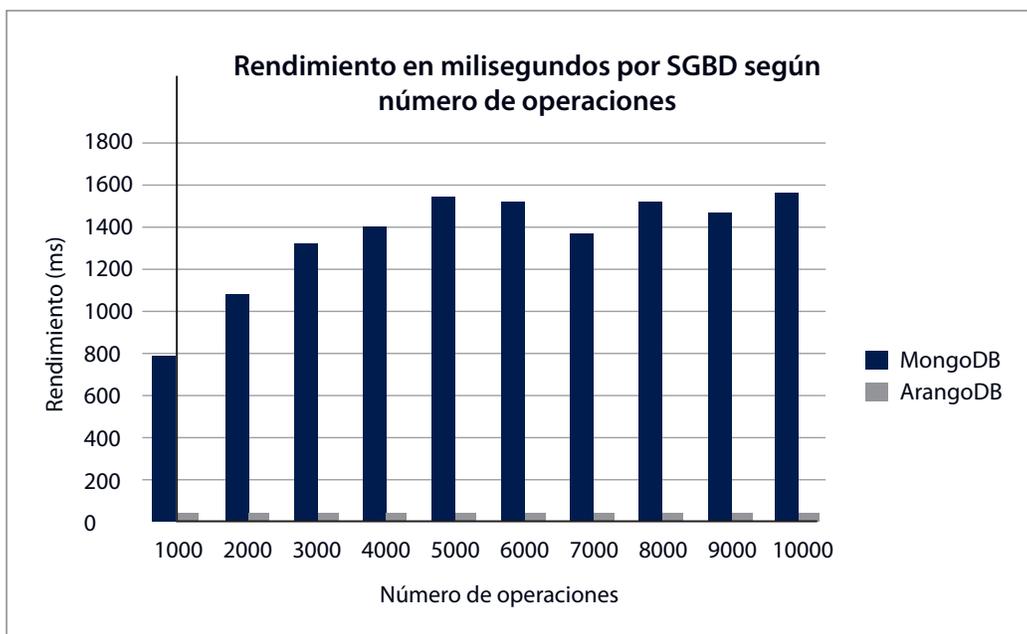
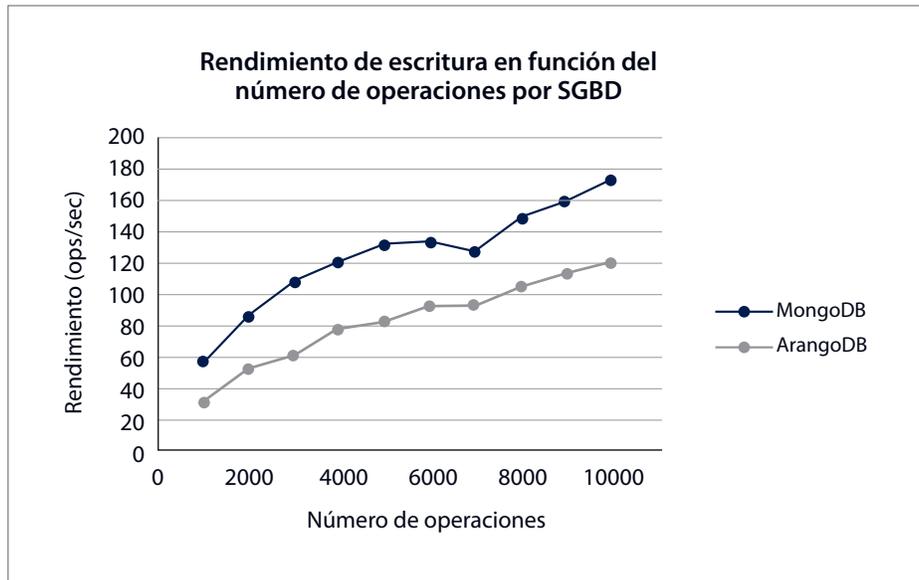


Figura 2. Rendimiento en milisegundos por SGBD según número de operaciones

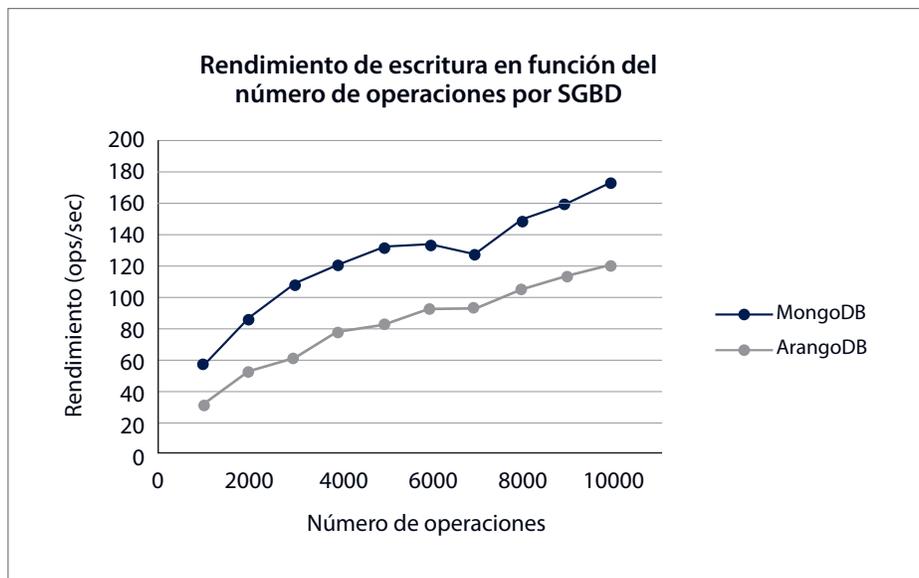
En este escenario, al analizar los procesos de lectura y escritura de manera individual (ver cuadro 4), se puede observar que la operación lectura tiene el mismo comportamiento que en el escenario anterior. Presentando un mejor rendimiento MongoDB, pero en la operación de escritura la diferencia en el rendimiento no es tan marcada como en el caso de la lectura (ver figura 3); pues la diferencia para el caso de 10000 operaciones entre estos dos SGBD es apenas de 43% a favor de MongoDB.

Cuadro 4. Resultados escenario lectura pesada por tipo de operación

Número de operaciones	MongoDB				ArangoDB			
	Lectura		Escritura		Lectura		Escritura	
	Rendimiento	Tiempo de respuesta						
1000	745,68	6,38	57,27	16,86	44,22	105,86	31,57	29,60
2000	1029,25	4,64	87,57	10,90	48,51	98,77	53,53	17,85
3000	1259,95	3,82	108,93	8,48	46,02	104,51	62,11	15,31
4000	1331,93	3,56	120,77	7,80	47,45	101,2	78,25	12,01
5000	1466,05	3,28	132,49	6,58	48,56	98,86	83,58	11,25
6000	1439,76	3,33	134,05	7,08	47,51	101,72	93,08	9,99
7000	1306,74	3,56	127,50	7,50	47,71	101,49	93,76	10,05
8000	1436,94	3,35	148,48	6,38	47,19	102,25	105,9	8,93
9000	1397,97	3,41	159,07	5,66	46,68	103,63	114,21	8,21
10000	1489,96	3,26	173,79	5,39	47,57	101,42	121,51	7,57



(a)



(b)

Figura 3. Rendimiento del tipo de operación en función del número de operaciones por SGBD

Conclusiones

Se evaluaron dos escenarios de prueba, solamente lectura y lectura pesada; con 10 diferentes tamaños de operaciones y haciendo uso de una base de datos geográfica. En los dos escenarios y con todos los tamaños de operación, MongoDB fue el SGBD con mejor desempeño. Sin embargo, en el caso específico de la escritura la diferencia no fue tan marcada.

Otro aspecto importante de destacar, si bien es muy bueno que ArangoDB brinde soporte geográfico nativo, el mismo es muy básico [27]. Por su parte, MongoDB brinda un soporte más avanzado para el manejo de datos geográficos [28], posibilitando realizar consultas complejas de manera eficiente. No solo cuenta con más operaciones para las consultas de datos espaciales y geométricos, sino que además se integra con los formatos más comunes para este tipo de datos, por ejemplo: OpenStreetMap [29].

Un punto a favor de ArangoDB, es que MongoDB utiliza la sintaxis JSON para consultas y no es compatible con un lenguaje de consulta declarativa. Por el contrario, ArangoDB desarrolló su propio lenguaje de consulta similar a SQL (AQL) para consultas complejas, permitiendo la combinación de patrones de acceso en una sola consulta [25].

Asimismo, se logró identificar que MongoDB presenta una debilidad significativa en el tema geográfico; solamente se pueden crear índices geográficos a datos con la proyección WGS84 [30]. Este es un tema muy sensible, pues actualmente no se tiene soporte para la proyección CRTM05 que es la proyección oficial para Costa Rica [31]. También, la transformación de coordenadas entre sistemas de referencia puede implicar pérdida de precisión de los datos.

Tomando en cuenta los hallazgos anteriores, se recomienda la utilización de MongoDB como el SGBD más adecuado para el desarrollo de la base de datos híbrida y distribuida que utilice los paradigmas SQL y NoSQL.

En la evaluación realizada en el presente trabajo, se utiliza una base de datos geográfica, pese a esto no se hizo énfasis en la consulta realizada. Por lo que sería conveniente en un próximo trabajo efectuar la comparación con consultas espaciales, es decir recuperación de información aplicando procesamiento geográfico. Finalmente, es pertinente realizar un análisis estadístico de los datos obtenidos para extraer conclusiones más precisas y poder establecer relaciones entre las variables de la población estudiada a partir de los datos recopilados.

Referencias

- [1] V. Reniers, D. Van Landuyt y R. Ansar , «On the State of NoSQL Benchmarks,» *ICPE '17 Companion: Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering Companion*, pp. 107-112, abril 2017.
- [2] F. L. Osorio Rivera, Base de datos relacionales. Teoría y Práctica, 1 ed., Medellín: Instituto Tecnológico Metropolitano, 2008.
- [3] S. Khan y V. Mane, «SQL Support over MongoDB using Metadata,» *International Journal of Scientific and Research Publications*, pp. 1-5, October 2013.
- [4] C. Bucur y G. T. Bogdan , «A comparison between several NoSQL databases with comments and notes,» de *RoEduNet International Conference 10th Edition: Networking in Education and Research*, Iasi, Romania, 2011.
- [5] J. Gutiérrez Puebla, J. C. García Palomares y M. H. Salas Olmedo, «Big (Geo) Data en Ciencias Sociales: Retos y Oportunidades,» *Revista de Estudios Andaluces*, vol. 33, n° 1, pp. 1-23, 2016.
- [6] The PostgreSQL Global Development Group, «PostgreSQL 10.3 Released!» The World's Most Advanced Open Source Database,» [En línea]. Available: <https://www.postgresql.org/>. [Último acceso: 7 marzo 2018].
- [7] geoserver.org, Geoserver, 2014.
- [8] NoSQL, «NoSQL,» [En línea]. Available: <http://nosql-database.org>. [Último acceso: 14 marzo 2018].
- [9] H. H. Ramírez Arévalo y J. F. Herrera Cubides, «Un viaje a través de bases de datos espaciales. NoSQL,» *Redes de ingeniería, Univ. Distrital Francisco J de Caldas*, vol. 4, pp. 35-47., 2013.
- [10] R. Suter, «MongoDB An introduction and performance analysis,» de *Conference about Computer Science, HSR Hochschule für Technik Rapperswil*, Sweden, 2012.

- [11] F. R. Oliveira y L. Del Val Cura, «Performance Evaluation of NoSQL Multi-Model Data Stores in Polyglot Persistence Applications,» de *Proceedings of the 20th International Database Engineering & Applications Symposium*, 2016.
- [12] O. Mishra, P. Lodhi y S. Mehta, «Document Oriented NoSQL Databases: An Empirical Study,» *International Conference on Recent Developments in Science, Engineering and Technology*, pp. 126-136, 2017.
- [13] E. Tang y F. Yushun, «Performance Comparison between Five NoSQL Databases,» de *Cloud Computing and Big Data (CCBD), 2016 7th International Conference*, Macau, China, 2016.
- [14] W. Puangsaijai y S. Puntheeranurak, «A Comparative Study of Relational Database and Key-Value Database for Big Data Applications,» Pattaya, , 2017.
- [15] K. S. Kumar, Srividya y S. Mohanavalli, «A performance comparison of document oriented NoSQL databases,» de *Computer, Communication and Signal Processing (ICCCSP), 2017 International Conference*, Chennai, 2017.
- [16] Apache Software Foundation, «Apache JMeter,» [En línea]. Available: <http://jmeter.apache.org/>. [Último acceso: 5 diciembre 2017].
- [17] B. Patel, J. Parikh y R. Shah, «A Review Paper on Comparison of SQL Performance Analyzer Tools: Apache JMeter and HP LoadRunner,» *International Journal of Current Engineering and Technology*, vol. 4, n° 5, pp. 3642-3645, 2014.
- [18] F. J. Díaz, C. M. Banchoff Tzancoff, A. S. Rodríguez y V. Soria, «Usando Jmeter para pruebas de rendimiento,» de *XIV Congreso Argentino de Ciencias de la Computación*, 2008.
- [19] ESRI, «Shapefile technical description,» *ESRI White Paper*, 1998.
- [20] H. Butler, M. Daly, A. Doyle, S. Gillies, T. Schaub y C. Schmidt, «The GeoJSON format specification,» *Rapport technique*, p. 67, 2008.
- [21] QGIS, «Bienvenido Al Proyecto QGIS!,» [En línea]. Available: www.qgis.org/es/site/. [Último acceso: 14 marzo 2018].
- [22] R. W. Smith, «Department of Defense World Geodetic System 1984: its definition and relationships with local geodetic systems,» Defense Mapping Agency, 1987.
- [23] J. Bernardino y V. Abramova, «NoSQL databases: MongoDB vs cassandra,» de *Proceeding C3S2E '13 Proceedings of the International C* Conference on Computer Science and Software Engineering*, Porto, Portugal, 2013.
- [24] MongoDB, «The MongoDB 3.4 Manual,» [En línea]. Available: <https://docs.mongodb.com/v3.4/>. [Último acceso: 6 diciembre 2017].
- [25] ArangoDB, «ArangoDB v3.2.9 Documentation,» [En línea]. Available: <https://docs.arangodb.com/3.2/Manual/>. [Último acceso: 6 diciembre 2017].
- [26] J. M. Rojas Gonzales, «Análisis comparativo de bases de datos relacionales y no relacionales,» 2017.
- [27] F. Martínez y A. Aizemberg, «Bases de datos de grafos con manejo de datos espaciales. Un análisis comparativo,» de *Simposio Argentino de Grandes Datos (AGRANDA 2015)*, Rosario, 2015.
- [28] A. Josefsson, «Comparing the performance of relational and document databases for hierarchical geospatial data,» Stockholm, Sweden, 2018.
- [29] G. Amat, J. Fernandez, A. Arranz y A. Ramos, «Using Open Street Maps data and tools for indoor mapping in a Smart City scenario,» de *AGILE'2014 International Conference on Geographic Information Science*, Castellón, 2014.
- [30] S. Schmid, E. Galicz y W. Reinhardt, «Performance investigation of selected SQL and NoSQL databases,» de *AGILE 2015*, Lisbon, 2015.
- [31] «NTIG_CR04_01.2016: Perfil Oficial de Metadatos Geográficos de Costa Rica,» 2016.