Trabajo Final - Captura y Almacenamiento de la Información:

Análisis de tiempo de consulta entre MongoDB y MySQL para un conjunto de datos meteorológicos.

Gelpi Gabriel, Hurtado Santiago, Sagarra Consuelo
consuelosagarra@gmail.com
grgelpi@gmail.com
santiagoh719@gmail.com

16 de septiembre del 2022

1 Introducción y objetivo

Con el avance de la tecnología y el consumo masivo de todo tipo de productos por parte de la sociedad, los datos adquiridos por distintas entidades en el mundo solo crecen y crecen. Con estos datos, una vez analizados es posible tomar decisiones y/o entender qué es lo que pasa con nuestra sociedad o los consumidores. Un paso previo a estos análisis es el almacenamiento de la información. Esta etapa es muy importante en todo el proceso referido al mundo de los datos ya que facilitará cualquier tipo de necesidad por las personas que consumen dichos datos. Una de las formas de almacenar los datos es a través de una base de datos.

Una base de datos es una forma de almacenar información que permite incorporar de buena manera nuevos datos y tenerlos de manera disponible para su consumo cuando sea que se los necesite. A partir de una rápida mirada en internet o bibliografía relacionada a bases de datos puede verse que se habla de base de datos relacionales y no relacionales. Los tipo relacionales son comúnmente conocidas como SQL en referencia a *Structured Query Language*. Son de las más utilizadas en el software profesional, educativo y comercial. Sin embargo en los últimos años las no relacionales o NoSQL han ganado popularidad al aparecer nuevos usos de la tecnología como IoT (*Internet of the Things*), donde los requerimientos para el guardado de los datos requiere una mayor flexibilidad en cuanto a la estructura de los datos y mayor velocidad de almacenaje en una manera óptima.

Desde un punto de vista técnico, una base de datos SQL es aquella que dispone de una relación predefinida entre sus elementos, donde cada registro pueda ser identificado de forma inequívoca. Se componen por un conjunto de tablas en las que los datos están clasificados por categorías. Cada columna de estas tablas corresponde y comprende una cierta cantidad de datos de dicha categoría. Estas tablas respetan generalmente el mismo esquema fijo (cantidad de columnas, tipos de datos y cantidad de datos por fila). Dentro de las bases de datos SQL más utilizadas se encuentran Oracle, SQL Server, MySql, María DB, entre otras. Por otro lado, una base de datos NoSQL o *Not Only SQL* es una base de datos que no cuenta con un identificador que relacione un conjunto de datos con otro. Estas bases de datos no necesitan un esquema fijo y son fácilmente modulares. Existen diferentes tipos de bases de datos NoSQL que permiten adaptarse a múltiples formatos de datos como pueden ser los documentos, videos, gráficos o incluso formatos de claves-valor. El objetivo siempre es recuperar la

información de un mismo lugar sin necesidad de pasar por las relaciones como con las tablas. Algunas de las bases de datos NoSQL más utilizadas son MongoDB, ElasticSearch, Cassandra y HBase.

Una tarea nada sencilla es la elección de la base de datos para un problema dado. Si bien muchas veces queda supeditado a una cuestión de presupuesto y/o seguridad, también pueden intervenir otros factores. Este trabajo tiene como objetivo presentar los tiempos de carga y una comparación estadística de la velocidad de las consultas entre una base de datos relacional y una no relacional utilizando datos meteorológicos. Se utiliza MongoDB como representante de las NoSQL y MySQL por parte de SQL. La comparación se lleva a cabo mediante una serie de consultas a las bases de datos siendo la conexión con el usuario mediante el lenguaje de programación Python. En la metodología se comparte el enlace al Github con los archivos jupyter notebook, datos y archivos de docker.

2 Datos y Bases de datos utilizadas

Los datos utilizados para estos experimentos son datos del reanálisis ERA5 del "Centro Europeo de Previsiones Meteorológicas a Plazo Medio" (ECMWF), obtenidos a través de un script de python de la página Copernicus. Los datos utilizados son presión a nivel medio del mar, temperatura en superficie y precipitación, todos en una retícula de 1^o por 1^o de latitud-longitud en Sudamérica. De esto se desprenden las variables tipo FLOAT: presión a nivel medio del mar, temperatura en superficie, precipitación, latitud y longitud; y la variable tipo DATETIME: Fecha y hora. Los datos de prueba son datos cada 4 horas durante el año 1979 y se cuenta con 177.228 registros para este trabajo. Estos datos se descargan en formato NetCDF V4 que utilizando código R se convirtieron a formato CSV.

Las bases de datos utilizadas para las pruebas son MySQL y MongoDB. La primera es un sistema de gestión de bases de datos (DBMS) relacional *open-source* desarrollado por Oracle. Es considerada como una de las bases de datos de código abierto más populares del mundo junto con Oracle y SQL Server de Microsoft. Al igual que otras bases de datos relacionales, MySQL almacena los datos en tablas conformadas por filas y columnas. Los usuarios pueden definir, manipular, controlar y consultar los datos utilizando el lenguaje SQL. Por su parte, MongoDB es un base de datos de código abierto orientada a documentos. Esta tiene como elemento base documentos BSON (similares a los JSON), que podrían pensarse como una fila en una tabla en un DBMS SQL. La gran diferencia que presenta MongoDB es que estos documentos no tienen que tener los mismos atributos entre sí, dándole una gran versatilidad. Como estos documentos no están relacionados entre sí, toda su información debe estar en cada documento. Esto implica, en general, una repetición de atributos iguales. Por ejemplo, si cada documento es una medición de instrumentos dispersos en una región, cada registro llevará las coordenadas de dicho instrumento repitiendo esos valores. Por otro lado, estos documentos se organizan en colecciones, que se pueden asemejar a las tablas en SQL.

3 Metodología

Para este trabajo se utilizó Python como lenguaje de programación que conecta a las bases de datos MySQL y MongoDB, y permite cargar los datos meteorológicos y realizar las consultas que se requieran. Los códigos utilizados se encuentran en Github. En dicha repositorio puede verse las notebooks por separado para la carga de datos a cada base de datos y la notebook donde se hacen las consultas y se toma el tiempo para el análisis de la velocidad de respuesta de dichas *queries*. También se encuentra el script de las figuras que muestran los resultados obtenidos. Se implementaron tres consultas basadas en las necesidades de consultas de datos meteorológicos, estas son:

1. Obtener todos los datos entre las latitudes $26^oS - 31^oS$ y las longitudes $58^oO - 63^oO$.

Métrica	MySQL	MongoDB
Min	$709392\mu s$	$0\mu s$
Percentil 25	$947676\mu s$	$0\mu s$
Mediana	$992456\mu s$	$0\mu s$
Media	$1014433\mu s$	$27\mu s$
Percentil 75	$1045340\mu s$	$0\mu s$
Max	$2539288\mu s$	$1123\mu s$

Tabla 1: *Métricas de los tiempos de la consulta 1 para MySQL y MongoDB.*

2. Obtener todos los datos entre el 15 y 27 de Febrero de 1979.

3. Obtener el acumulado de precipitación para cada punto de retícula.

En estas tres consultas se consideran las operaciones básicas de cualquier consulta meteorológica, filtrar por una región (latitud-longitud), buscar información necesaria por fechas (filtrado de los datos por fechas), hacer operaciones de agregación y agrupamiento, como es estimar la precipitación acumulada que se registra en cada estación.

Para tener un número representativo del tiempo que lleva cada consulta se ejecutó cada una mil veces, donde se tomaron los respectivos intervalos de tiempo. De esta manera se obtuvieron resultados estadísticos para evaluar la velocidad de las consultas, tales como el tiempo medio, máximo, mínimo, percentil 25, percentil 50 y percentil 75. A partir de esto se realizaron box-plot del tiempo de cada consulta para resumir los resultados.

4 Resultados

En primer instancia se calculo el tiempo de carga de los datos en cada base de datos. La velocidad de inserción para MySQL fue de $1313886980\mu s$ y para MongoDB fue de $4430736\mu s$. Estos valores donde el tiempo de carga es mayor en la base de datos relacional también se observa en Antaño et al. [2]. A medida que el volúmen de datos crece también aumenta la diferencia de los tiempos en las cargas.

Luego se procedió a realizar las consultas descriptas en la Metodología. Los resultados de la primera se observan en la Figura1. En dicho gráfico se muestra el boxplot de los tiempos al realizar la consulta 1 con MongoDB y MySQL. Se puede observar que MongoDB tiene tiempos de respuesta mucho menores en comparación con MySQL. El tiempo mínimo obtenido con MySQL es de $709.392\mu s$ mientras que el mayor tiempo obtenido para MongoDB es de $1.123\mu s$ siendo dos órdenes de magnitud menor. De las 1000 consultas realizadas con MongoDB, 965 consultas tardaron $0\mu s$, esto puede deberse a que MongoDB posee una memoria volátil que reduce el tiempo de ejecución [1].

La Figura 2 muestra el boxplot de los tiempos al realizar la consulta 2 con MongoDB y MySQL, esta consulta filtra los datos por fechas. Se puede observar que, como para la consulta 1, MongoDB tiene tiempos de respuesta mucho menores en comparación con MySQL. El tiempo mínimo obtenido con MySQL es de $358.164\mu s$ mientras que el mayor tiempo obtenido para MongoDB es de $1.030\mu s$. Como ocurrió anteriormente, los tiempos para la consulta 2 de MongoDB se realizaron en $0\mu s$ (938 consultas de 1.000). Los tiempos optenidos en las consultas 1 y 2 son consistentes con el análisis realizado por Patil et al. [5], Dipina Damodaran et al. [3] y Narrero et al. [4] que encuentra mejores tiempos para MongoDb respecto de MySQL.

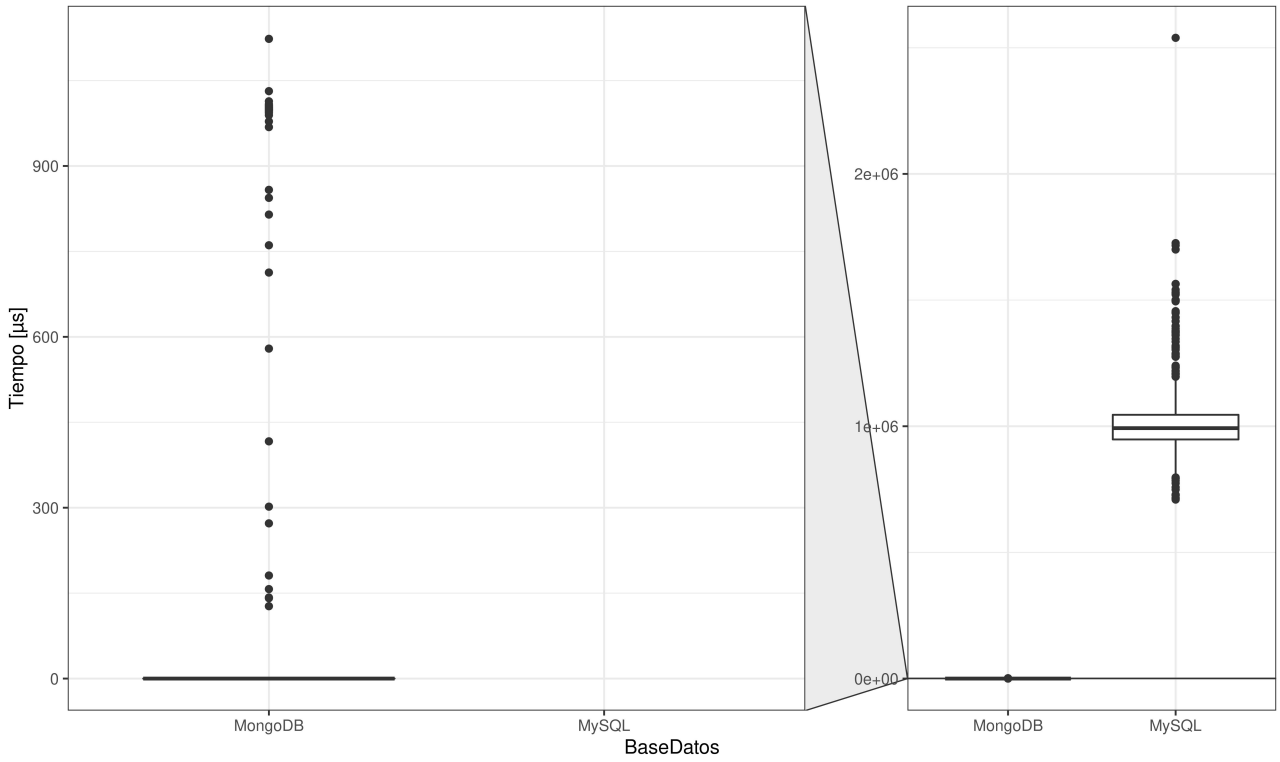

Figura 1: *Boxplot de los tiempos de ejecución de la consulta 1: Obtener todos los datos entre las latitudes* $26^oS - 31^oS$ *y las longitudes* $58^oO - 63^oO$*. En el panel izquierdo se ve un zoom de los datos.*

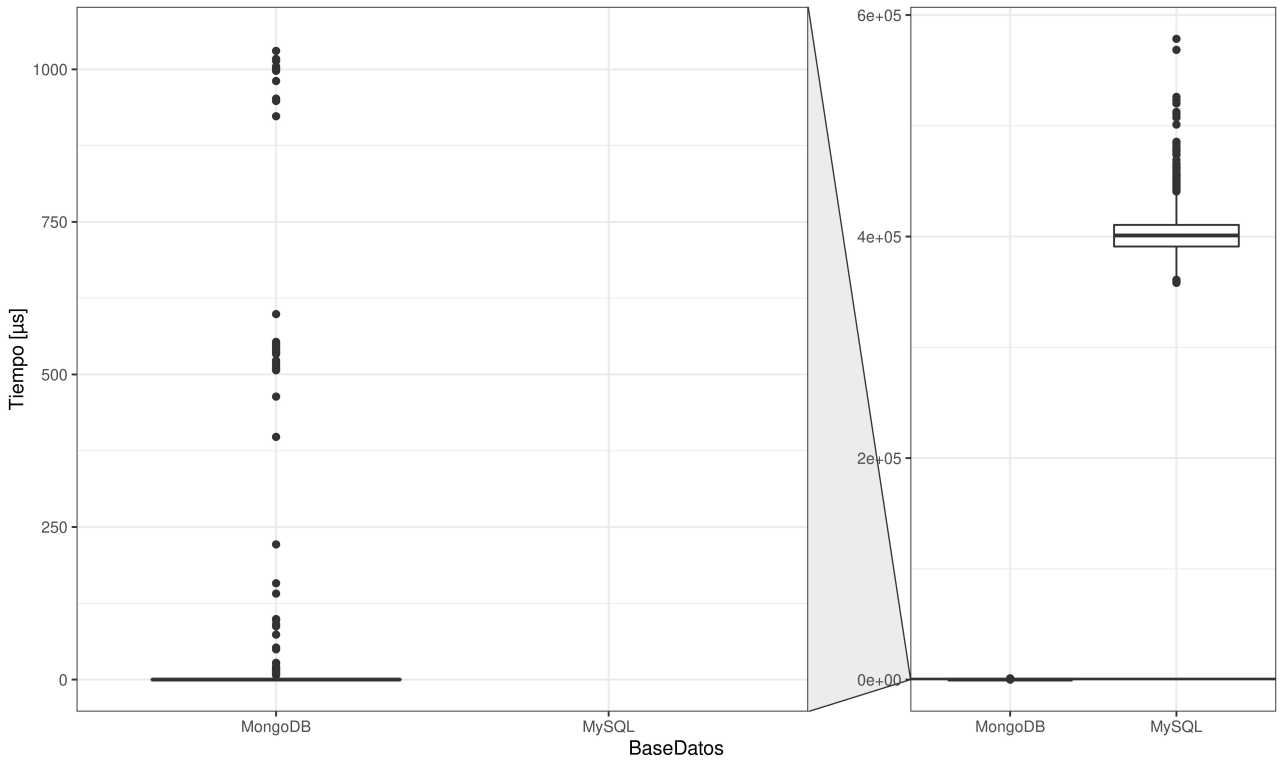

Figura 2: *Boxplot de los tiempos de ejecución de la consulta 2: Obtener todos los datos entre el 15 y 27 de Febrero de 1979. En el panel izquierdo se muestra un zoom de los datos.*

Métrica	MySQL	MongoDB
Min	$358164\mu s$	$0\mu s$
Percentil 25	$390993\mu s$	$0\mu s$
Mediana	$400962\mu s$	$0\mu s$
Media	$403855\mu s$	$24\mu s$
Percentil 75	$410501\mu s$	$0\mu s$
Max	$578508\mu s$	$1030\mu s$

Tabla 2: *Métricas de los tiempos de la consulta 2 para MySQL y MongoDB.*

La Figura 3 muestra el boxplot de los tiempos al realizar la consulta 3 con MongoDB y MySQL, esta consulta realiza una operación de agregación (suma) y de agrupamiento, diferente a las consultas 1 y 2. Se puede observar que, contrario de las otras dos consultas, MySQL muestra en términos medios mejores tiempos de respuesta que MongoDB. No obstante, estas diferencias están dentro del mismo orden de magnitud. Al observar los valores de la Tabla 3, se desprende que los tiempos de esta consulta en MongoDB son aproximadamente el doble de los tiempos obtenidos con MySQL. Este resultado concuerda con lo visto en Narrero et al. [4] donde también se ve que al realizar una consulta de agregación el tiempo de MongoDB es superior al de las primeras consultas. Sin embargo, no se observa una gran diferencia en comparación con la consulta de MySQL como en dicho trabajo.

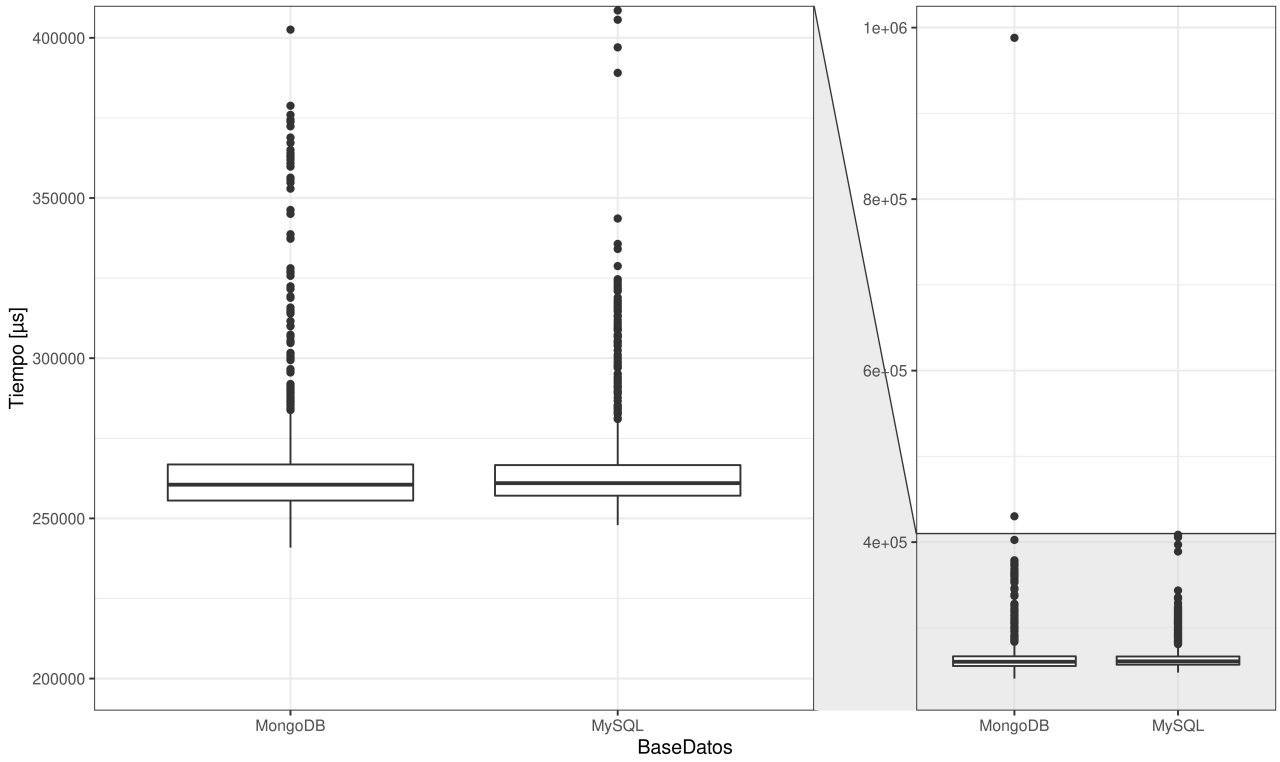

Figura 3: *Boxplot de los tiempos de ejecución de la consulta 3: Obtener el acumulado de precipitación para cada punto de retícula.*

Métrica	MySQL	MongoDB
Min	$247915\mu s$	$240881\mu s$
Percentil 25	$257091\mu s$	$255564\mu s$
Mediana	$260999\mu s$	$260520\mu s$
Media	$265501\mu s$	$266350\mu s$
Percentil 75	$266638\mu s$	$266834\mu s$
Max	$408537\mu s$	$988095\mu s$

Tabla 3: *Métricas de los tiempos de la consulta 3 para MySQL y MongoDB.*

5 Conclusión

En este trabajo se evaluó el tiempo de consultas entre una base de datos relacional (MySQL) y otra no relacional (MongoDB) sobre un conjunto de datos meteorológicos, a fin de determinar la más adecuada para utilizar en esta disciplina en consultas típicas de información meteorológica. Para esto se realizaron tres consultas diferentes mil veces a fin de determinar el comportamiento de los tiempos de consulta para cada base de datos.

De este estudio se desprende que las consultas de filtrado de datos son más rápidas con MongoDB respecto a MySQL, ya que en dos consultas de filtrado se obtuvieron tiempos dos órdenes de magnitud inferiores. No obstante, la velocidad de consulta para una operación de agregación y agrupamiento fueron comparables.

Dado que en las aplicaciones meteorológicas lo que más se utiliza son consultas de filtrado para luego poder hacer un procesamiento de una base de datos más chica con otras herramienta de programación se puede concluir que lo más óptimo es utilizar MongoDB como base de datos por sobre MySQL.

Bibliografía

[1] Abramova, V., Bernardino, J., & Furtado, P. (2014). Experimental evaluation of nosql databases. *International Journal of Database Management Systems (IJDMS)*, 6.

[2] Antaño, A. C. M., Castro, J. M. M., & Valencia, R. E. C. (2014). Migración de bases de datos sql a nosql. *Revista Tlamati(CICOM)*, 3.

[3] Dipina Damodaran, B., Salim, S., & Vargese, S. M. (2016). Performance evaluation of mysql and mongodb databases. *Int. J. Cybern. Inform.(IJCI)*, 5.

[4] Narrero, L., Olsowy, V., Thomas, P., Delia, L., Tesone, F., Sosa, J. F., & Pesado, P. (2019). Un estudio comparativo de bases de datos relacionales y bases de datos nosql. In *XXV Congreso Argentino de Ciencias de la Computación (CACIC)* (pp. 589–3600).: Facultad de Ciencias Exactas, Físico-Química y Naturales, Universidad Nacional de Río Cuarto.

[5] Patil, M. M., Hanni, A., Tejeshwar, C., & Patil, P. (2017). A qualitative analysis of the performance of mongodb vs mysql database based on insertion and retriewal operations using a web/android application to explore load balancing—sharding in mongodb and its advantages. In *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)* (pp. 325–330).: IEEE.