

Base de datos NoSQL

Caso de estudio:

Postgres como solución NoSQL

Rosmeli Quintero
Escuela de Computación
Licenciatura en Computación
Universidad Central de Venezuela
Caracas, Venezuela
Email: qrosmeli@gmail.com

Jean Carlos Gomes
Escuela de Computación
Licenciatura en Computación
Universidad Central de Venezuela
Caracas, Venezuela
Email: jean.gomes@ucv.ve

Abstract—Las bases de datos relacionales han sido las herramientas por excelencia para el almacenamiento de la información en los sistemas informáticos. No obstante, las bases de datos NoSQL, como tendencia, han venido ganando espacio especialmente por la escalabilidad y velocidad en sus tiempos de respuestas. PostgreSQL ha incorporado algunas características de tipo NoSQL, como el almacenamiento efímero y el manejo de datos JSON; características que pueden aprovecharse para realizar acciones desde el gestor dándole mayor potencia. El objetivo de este artículo es evaluar, mediante toda la documentación encontrada, el comportamiento de las características NoSQL de PostgreSQL frente a un gestor NoSQL, comparándola con MongoDB, respecto a los tiempos de respuestas y dar a conocer las ventajas de uno con respecto al otro.

Palabras Claves: Características NoSQL en PostgreSQL, MongoDB, PostgreSQL

1. Introducción

Las bases de datos relacionales han sido las herramientas por excelencia para el almacenamiento de la información en los sistemas informáticos desde su surgimiento en las décadas del 70 y el 80. No obstante, las bases de datos NoSQL, como tendencia, han venido ganando espacio especialmente por la escalabilidad y velocidad en sus tiempos de respuestas, superiores a los de los sistemas relacionales. Este tipo de bases de datos no garantizan las propiedades ACID (Atomicidad, Consistencia, aislamiento y Durabilidad) en áreas de potenciar el rendimiento y la escalabilidad, basándose en el teorema CAP (Consistencia, Disponibilidad y la tolerancia a particiones), que plantea que un sistema no puede contar con más de dos de estas tres propiedades a la vez.

El movimiento NoSQL no es nuevo, se inició a finales de los 90 cuando las bases de datos XML comenzaron a ganar espacio con excelentes propuestas, sobre todo ofreciendo la posibilidad de consultas sobre datos XML. Característica que posteriormente las bases de datos

relacionales incorporaron. Las bases de datos NoSQL y las relacionales no se solapan, más bien cada una garantiza las características para las cuales fueron desarrolladas. De ahí que muchas empresas las utilicen juntas para diferentes actividades tales como Facebook, Google y otras empresas. PostgreSQL ha incorporado algunas características de tipo NoSQL, como el almacenamiento efímero y el manejo de datos en formato Notación de Objetos JavaScript (JSON); características que pueden aprovecharse para realizar acciones desde el gestor dándole mayor potencia. El objetivo de este artículo es evaluar el comportamiento de las características NoSQL de PostgreSQL frente a un gestor de bases de datos NoSQL como MongoDB, comparando, los tiempos de respuestas, el almacenamiento y dar a conocer las ventajas de uno con respecto al otro.

2. ¿Qué es Postgres NoSQL?

La nueva versión de la base de datos de código abierto PostgreSQL, se ha ido evolucionando y adoptándose al ampliamente usado formato de intercambio de datos JSON, esto apunta al creciente mercado NoSQL de almacenes de datos no relacionales, Postgres NoSQL es la poderosa combinación de tecnologías de bases de datos no estructurados y relacionales en un solo sistema de gestión de bases de dato. Postgres NoSQL proporciona la libertad, la flexibilidad y el rendimiento de la manipulación de datos no estructurados y semi-estructurados, conservando su viabilidad a largo plazo.

Postgres posee todas las características de una base de datos relacional con la capacidad de manejar bases de datos del tipo documento y de clave/valor proporcionan las ventajas de las nuevas tecnologías NoSQL además de la posibilidad de integrarlos en las infraestructuras tradicionales de datos corporativos sin silos de datos, la complejidad operativa, la pérdida de la integridad de los datos, y el riesgo.

3. ¿ACID o BASE?

PostgreSQL es una base de datos 100% ACID, posee control de concurrencia multiversión con un sistema sofisticado que previene a los lectores de bloquear a los escritores y viceversa como es el MVCC (Control de Concurrencia Multi-Versiones), posee un WAL (Write-ahead logging) conocido también como log, tiene soporte completo para claves forneas, uniones, vistas, disparadores y procedimientos almacenados, soporte de tipos de datos documentos.

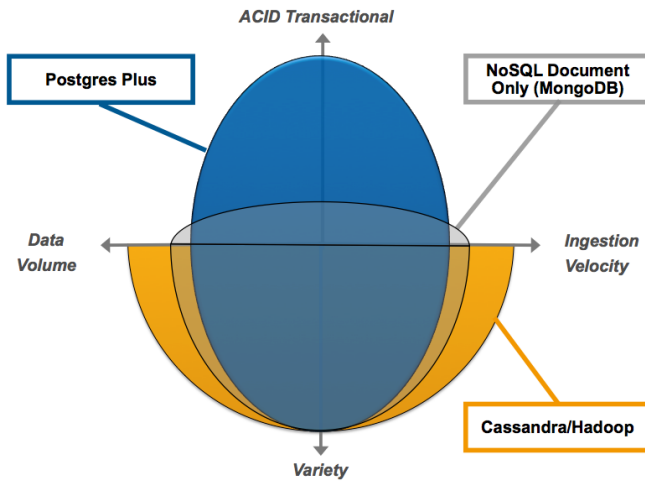


Figure 1. Capacidades de Postgres para cargas de trabajo NoSQL [1].

4. Clave/valor - HStore

El módulo HStore, que permite almacenar pares clave / valor dentro de una sola columna, permite a los usuarios crear un almacén de claves/valor sin esquema. Pero a diferencia de las soluciones NoSQL puros, un almacén de claves-valor creado en Postgres es compatible con ACID.

HStore es una herramienta particularmente útil para los desarrolladores web o alguien que construye una aplicación que requiere la propiedad ACID, Hstore anterior a muchos avances NoSQL. Fue introducido en v8.2 de Postgres en el año 2006, antes de muchos avances de otros gestores NoSQL. Su popularidad aumentó en los últimos años con nuevas demandas para trabajar con datos semi-estructurados.

Hstore no es jerárquica, pero el tipo de datos HStore ofrece avanzada soporte de indexación, lo que hace que sea la solución de elección para muchas aplicaciones. Es particularmente útil para los datos con baja densidad por ejemplo, esto es muy útil para el almacenamiento de productos con múltiples descripciones en una sola tabla, donde cada producto, comparte algunas atributos, como el nombre, precio y peso, pero tienen muchos diferentes atributos basados en el tipo de producto, tales como el tamaño, la presentación, entre otros.

HSTORE provee la funcionalidad de indexar en varios tipos

Operadores	GIN	GIST
Creación	Lento	Rápido
Consultas	Rápido	Lento
Actualización	Lento	Rápido
Memoria	2x más	Poca

Otros tipos de índices son:

- BTREE
- HASH

Ejemplo del módulo HStore en PostgreSQL:

Primer Paso: Antes de trabajar con el tipo de datos hstore, necesita habilitar la extensión hstore, que carga el módulo para habilitar la extensión hstore PostgreSQL

```
1 CREATE EXTENSION hstore;
```

Segundo paso: crear una tabla con una columna del tipo de dato HStore. Para este ejemplo, Creamos una tabla llamada libros que tiene tres columnas:

- id es la clave principal que identifica el libro.
- ttulo es el ttulo de las tiendas de productos
- attr atributos del libro como el ISBN, el peso, el papel de devolucion, etc. El tipo de datos de la columna de attr es hstore.

```
1 CREATE TABLE books (
2   id serial primary key,
3   title VARCHAR (255),
4   attr hstore
5 );
```

Tercer paso: Insertar datos, Nosotros usamos la instrucción INSERT para insertar datos en la columna de la hstore de la siguiente manera, pero usted puede usar otro comando: NOTA: Los datos que insertamos en la columna de la hstore es una lista separadas por comas usando los pares de claves => valor. Tanto las claves y los valores están expresados usando comillas ("").

```
1 INSERT INTO books (title, attr)
2 VALUES
3 (
4   'PostgreSQL Tutorial',
5   "paperback" => "243",
6   "publisher" => "postgresqtutorial.com",
7   "language" => "English",
8   "ISBN-13" => "978-1449370000",
9   "weight" => "11.2 ounces"
10 );
```

Cuarto Paso: Realizar una consulta

```

1 SELECT
2   attr -> 'ISBN-13' AS isbn
3 FROM
4   books;

```

```

isbn
▶ 978-1449370000

```

5. Base de datos Orientada a Documentos-JSON/JSONB

Las capacidades de base de datos orientada a documentos en Postgres avanzaron significativamente cuando se introdujo soporte para el tipo de datos JSON en la versión 9.0. JSONB, un formato de almacenamiento binario que proporciona un mejor rendimiento fue introducido en la versión 9.4 de Postgres.

JSON (notación de objetos JavaScript) es una de la mayoría de los formatos de datos intercambiados populares en la web. Es apoyado por prácticamente cualquier lenguaje de programación en uso hoy en día, y sigue aumentando la atracción por el uso de JSON por la introducción de nuevas tecnologías de apoyo tales como el motor JavaScript V8, también apoyado en Postgres a través de PL / V8, Node.js y algunos sistemas NoSQL, como MongoDB y CouchDB.

Postgres ofrece soporte robusto para JSON. Postgres tiene un conjunto de datos de tipo JSON, lo que valida y almacena el texto JSON y proporciona funciones para extraer elementos de los valores de JSON y ofrece la posibilidad de codificar fácilmente conjuntos de resultados de la consultas que se pueden utilizarse. Esta última pieza de funcionalidad es particularmente importante, ya que significa que las aplicaciones que prefieren trabajar de forma nativa con JSON puede obtener fácilmente sus datos de Postgres en JSON. Además del tipo de datos nativo JSON, Postgres v9.3 añade un JSON analizador y una variedad de funciones JSON. Esto significa aplicación web, los desarrolladores no necesitan capas de traducción en el código entre las base de datos y el marco web que utiliza JSON, los datos pueden ser enviados directamente a la base de datos PostgreSQL, donde no sólo almacenar los datos, pero adecuadamente validarlos también. Con funciones JSON, Postgres pueden leer datos relacionales de una tabla y devolverlo a la aplicación como válidos cadenas de formato JSON. Y, los datos relacionales pueden ser devueltos como JSON, ya sea para un único valor o un registro completo.

Las bases de datos de tipo documento ms populares son CouchDB y MongoDB segun el sitio db-engines, el

JSON	JSONB
Re-interpretación de diccionario	No necesita re-interpretar
Operaciones toman ms tiempo	Más espacio en disco
Preserva el orden	No preserva el orden
Sin índices	Soporta índices

cual se dedica a mostrar un ranking de las bases de datos ms utilizadas en el mundo. Segun un estudio realizado, MongoDB brinda mejores tiempos de respuesta en la insercin de informacin. Por tanto, para este trabajo, se elige para realizar la comparacin de los tiempos de respuestas de las caractersticas NoSQL de PostgreSQL. Primer Paso: Crear una tabla con el tipo de dato JSON o JSONB, según preferiera, en este ejemplo se usara json. para este ejemplo se creo la tabla de pedidos (orders) que posee dos columnas:

La columna id es la columna de clave principal que identifica la orden. La columna de informacin almacena los datos en formato de JSON.

```

1 CREATE TABLE orders (
2   ID serial NOT NULL PRIMARY KEY,
3   info json NOT NULL
4 );

```

Segundo Paso: Insetar datos JSON.

```

1 INSERT INTO orders (info)
2 VALUES
3 (
4   '{ "customer": "John Doe", "items": {"product": "Beer", "qty": 6}}'
5 );

```

```

1 INSERT INTO orders (info)
2 VALUES
3 (
4   '{ "customer": "Lily Bush", "items": {"product": "Diaper", "qty": 24}}'
5 ),
6 (
7   '{ "customer": "Josh William", "items": {"product": "Toy Car", "qty": 1}}'
8 ),
9 (
10  '{ "customer": "Mary Clark", "items": {"product": "Toy Train", "qty": 2}}'
11 );

```

Tercer paso: Realizar una consulta

```

1 SELECT
2   info
3 FROM
4   orders;

```

```

info
▶ { "customer": "John Doe", "items": {"product": "Beer", "qty": 6}}
{ "customer": "Lily Bush", "items": {"product": "Diaper", "qty": 24}}
{ "customer": "Josh William", "items": {"product": "Toy Car", "qty": 1}}
{ "customer": "Mary Clark", "items": {"product": "Toy Train", "qty": 2}}

```

6. Postgres vs MongoDB - Pruebas de desempeño

Antes de entrar en detalle sobre la prueba de desempeño es importante conocer las características de ambas bases de datos.

La tabla en la figura 2, compara las características más importantes de una base de datos NoSQL Orientada a Documento (como MongoDB) contra las características de una base de datos relacional (como Postgres):

6.1. Características del experimento

Las pruebas de comparación o benchmarking son experimentos que evalúan una o más herramientas para comparar su comportamiento. Para realizar las pruebas se deben definir métricas, que en el marco de este estudio serán el tiempo de carga de datos, el tiempo de inserción o de escritura, el tiempo de consulta y el espacio en disco que ocupara PostgreSQL y MongoDB, cabe acotar que los tiempos serán medidos en segundos, y espacio en disco será medido en Gigabyte.

EnterpriseDB (EDB) empezó a correr evaluaciones comparativas para ayudar a los usuarios a evaluar correctamente las capacidades de NoSQL Postgres. El conjunto inicial de pruebas en comparación MongoDB v2.6 contra Postgres v9.4 beta, esta prueba de rendimiento se realizó en los casos de una máquina individual. Ambos sistemas fueron instalados usando los servicios de Amazon Web Services M3.2XLARGE con 32 GB de memoria.

6.2. ¿Por que escogimos este estudio y no otro?

Hay muchas pruebas encontradas en internet de Postgres vs Mongo, pero la razón por la cual se escogió esta fue:

- EnterpriseDB es una empresa privada estadounidense que proporciona soporte y herramientas para PostgreSQL. Esta empresa la consideramos como una fuente confiable, con muy buenos artículos que publican constantemente y con gran prestigio en cuanto al soporte que brinda para PostgreSQL.
- Tienen disponible en GitHub un script para que uno pueda realizar esta prueba en cualquier máquina.
- Al estar el código publicado, podemos ver a detalle si realmente hubo manipulación de estos datos y así confiar en esta prueba de desempeño.

Features	MongoDB	PostgreSQL
Open Source Development Start	2009	1995
Schemas	Dynamic	Static and Dynamic
Supports Hierarchical Document Data	Yes	Yes (since 2012)
Supports Key-Value Data	Yes	Yes (since 2006)
Supports Relational Data / Normalized Form Storage	No	Yes
Data Constraints	No	Yes
Joining of Data and Foreign Keys	No	Yes
Powerful Query Language	No	Yes
Transaction Support and Multi Version Concurrency Controls	No	Yes
Atomic Transactions	Within a Document	Across the Database
Supported Web Development Languages	JavaScript, Python, Ruby, more...	JavaScript, Python, Ruby, more...
Common Web Data Format Support	JSON (Document), Key-Value, XML	JSON (Document), Key-Value, XML
Geo-spatial Support	Yes	Yes
Easiest path to Scaling	Horizontal Scale Out	Vertical Scale Up
Sharding	Easy	Complex
Server Side Programming	None	Many Procedural Languages like Python, JavaScript, C, C++, Tcl, Perl and many more
Easy Integration with Other Data Stores	No	Foreign Data Wrappers to Oracle, MySQL, MongoDB, CouchDB, Redis, Neo4j, Twitter, LDAP, File, Hadoop and more...

Figure 2. Tabla comparativa de las características de MongoDB y Postgres

6.3. Resultado del experimento

- La carga de grandes volúmenes de datos fue de aproximadamente 2,1 veces más rápido en Postgres MongoDB
- Las inserciones en la base de datos, tomaron aproximadamente 2,1 veces más tiempo en Postgres MongoDB

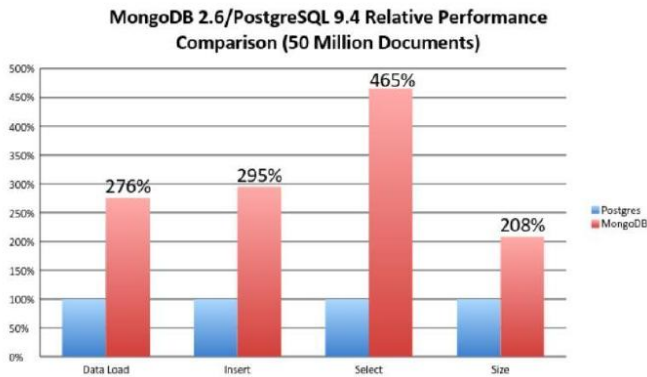


Figure 3. Grafico de los resultados obtenidos

inmadamente 2,5 veces más larga en tiempo en MongoDB que en Postgres

- Las consultas en MongoDB tardó casi 3 veces más en tiempo con respecto a Postgres.
- MongoDB consumen 33% ms en el almacenamiento de los datos en disco.

	Postgres	MongoDB
Data Load (s)	4,732	13,046
Insert (s)	29,236	86,253
Select (s)	594	2,763
Size (GB)	69	145

Figure 4. Valores obtenidos en el estudio.

6.4. ¿Según este resultado es PostgreSQL mejor que MongoDB?

Comparando este resultado con la versiones actuales, PostgreSQL sigue siendo mas rapido que MongoDB, la unica diferencia es en el almacenamiento ya que mongoDB en su version actual es mas eficiente, MongoDB usa un formato de almacenamiento propio de ellos, llamado BSON, que es mas eficiente, ya que es mas comprimido. Ahora bien, respondiendo a la pregunta, de cual es mejor, la respuesta es MongoDB, EnterPriceDB realizo este estudio en una sola maquina, el problema viene cuando tenemos un sistemas distribuidos con muchos usuarios. MongoDB posee replicaset, son muy sencillos de crear y de modificar, en cambio PostgreSQL posee "artificios" para obtenerlos, PostgreSQL utiliza las primitivas de particinamiento para simular esto. Otra diferencia es en la escalabilidad. Postgres maneja grandes cantidades de datos para un gran número de usuarios concurrentes aun as Postgres posee escalabilidad vertical, a diferencia de la mayorías de las bases de datos NoSQL que poseen escalabilidad Horizontal como MongoDB, esta trae como problema en cuanto a mayor números de usuarios accediendo a ella, el nivel de concurrencia se ve afectado. Aunque postgresQL puede manejar la escalabilidad horizontal, esta es muy complicada, aunque en la versiones que

esta por venir de PostgreSQL v9.6 se reseña en el articulo una mejora muy importante en cuanto a la escalabilidad Horizontal.

6.5. Conclusión del experimento

El gestor de bases de datos PostgreSQL ha ido incorporando paulatinamente características NoSQL, destacando los tipos de datos de documentos JSON y el almacenamiento efímero. Dichas características fueron evaluadas con respecto a MongoDB, el gestor NoSQL de mejores tiempos de respuestas. El estudio realizado muestra que PostgreSQL ha mejorado considerablemente los tiempos de respuestas con la incorporación de estas particularidades que tiene las bases de datos NoSQL. Si bien aún no está al nivel de los tiempos de respuesta de MongoDB, por el factor visto de base de datos distribuidos y el nivel de concurrencia que conlleva este, sí constituye un paso de avance en la incorporación de estas características, que les permitirán a los usuarios hacer uso de ellas sin tener que migrar a un nuevo gestor de bases de datos.

7. Conclusión

SQL y NoSQL han sido grandes inventos en el tiempo en el área de gestión de datos y se han utilizado para mantener el almacenamiento y recuperación optimizado de datos. Es todavía difícil de criticar, y hay que saber cuando nos conviene mejor SQL con respecto a NoSQL o cuando usar ambas. No se tiene que ver como una pelea entre SQL y NoSQL. Ambas tecnologías son los mejores en lo que hacen y es hasta que un desarrollador decide probar para poner en uso dependiendo de las situaciones y necesidades empresariales, esta articulo trato de invitar al lector a probar nuevas tecnologías, y no solo escoger una base de datos porque esta de moda, o por que maneja datos semi-estructurados, ya que se menciono que las bases de datos SQL tambien manejan documentos, si una base de datos no tiene un buen rendimiento es necesario enfrentarse y probar con otra base de datos, aunque las bases de datos NoSQL se están convirtiendo en una parte importante, sin embargo, las empresas deben actuar con precaución y ser consciente de las limitaciones asociados a estas bases de datos.

References

[1] S. Redner, "How popular is your paper? an empirical study of the citation distribution," 1998.

AUN FALTA REVISAR LA ORTOGRAFIA, COLOCAR LAS REFERENCIAS DE DONDE SACAMOS TODOS ESTOS PUNTOS, Y COMPLETAR UNA INFORMACION EN EL PUNTO 5 Y 6