Bases de datos de grafos con manejo de datos espaciales. Un análisis comparativo

Federico Martinez[1], Ariel Aizemberg[2]

[1] Universidad Nacional de Quilmes, Buenos Aires, Argentina,
federico.martinez@unq.edu.ar
[2] Depto. de Ingeniería Informática, ITBA, Buenos Aires, Argentina,
aaizemberg@itba.edu.ar

Resumen Se realiza un estudio comparativo entre bases de datos de grafos con capacidades espaciales, evaluando tiempos de procesamiento, poder expresivo, usabilidad y necesidades de almacenamiento

Keywords: spatial DB, graph DB, benchmarking

1. Descripción del problema

Dentro de las BD NoSQL, las Bases de Datos de Grafos (BDG) han cobrado popularidad en los últimos tiempos, debido al auge del análisis de las redes sociales. La gran mayoría de de los sistemas de BDG manejan datos típicamente alfanuméricos, sin embargo los objetos representados en estos grafos poseen, casi siempre, características espaciales o se encuentran georeferenciados. Por lo tanto, la posibilidad de manipular datos espaciales agrega valor a las BDG, permitiendo consultas como "personas que viven a menos de 100 km una de otra, se conocen entre sí, y tienen gustos similares. Asimismo, la posibilidad de manipular este tipo de datos permite la integración, en forma natural entre los sistemas de información geográfica y las BDG. Actualmente existe una amplia variedad de BDG, tanto de código abierto como propietario y muchas de ellas pueden manejar, en mayor o menor medida, datos espaciales. Entre ellas podemos citar a Neo4j[1], ArangoDB[2], OrientDB, Titan, etc.

En el presente trabajo comparamos la capacidad de dos BDG representativas y de amplia difusión, como Neo4j y ArangoDB, por manipular datos espaciales, centrándonos en la eficiencia para responder consultas, y en la capacidad y flexibilidad para representar y almacenar datos espaciales.

2. Metodología

Para realizar nuestro estudio utilizamos un conjuto de datos provenientes del sitio openflights[3]. El mismo presenta información sobre aerolíneas, aeropuertos y rutas entre ellos.

[3] http://openflights.org/data.html

A continuación y dado que éste es un estudio preliminar, elegimos un conjunto de consultas que consideramos representativas del uso de BDG extendidas con la posibilidad de manipular datos espaciales. En nuestro caso, analizamos las siguientes consultas:

1. Aeropuerto a menos de 100 kilómetros de la Ciudad de Buenos Aires.
2. Aeropuertos a 100 kilómetros de la Ciudad de Buenos Aires que permiten viajar a Barajas.
3. Itinerarios que parten de un aeropuerto a menos de 400 kilómetros de la Ciudad de Buenos Aires, que tengan una escala y terminan en Barajas.
4. Itinerarios que parten de un aeropuerto a menos de 400 km de la Ciudad de Buenos Aires, tienen una escala y terminan a menos de 400 km de Chipre.

Para cada consulta medimos el tiempo de ejecución. La carga de datos y los experimentos se realizaron sobre una máquina virtual de 3 GB de RAM. La máquina física es un Intel Core i5 4690 de 3,5 GHz con 16 Gb de RAM.

Los resultados obtenidos se detallan a continuación:

Experimento	Neo4J	ArangoDB
Carga	20 minutos 27 segundos	25 minutos 37 segundos
Espacio en disco	165.9 MB	437.3 MB
Consulta 1	0.063 segundos	0.012 segundos
Consulta 2	0.016 segundos	0.052 segundos
Consulta 3	0.227 segundos	3.697 segundos
Consulta 4	2.319 segundos	4.901 segundos

3. Discusión de los resultados

Si bien ambas bases de datos proveen soporte para el manejo de datos geográficos y las dos funcionan muy bien para responder consultas sencillas, Neo4j aparece como una mejor alternativa cuando las consultas requieren recorrer el grafo, o de funcionalidades más complejas como importar mapas.

Las principales falencias que encontramos en ArangoDB son: Su lenguaje de consulta no parece estar pensado para recorrer grafos. Esto hace que escribir consultas que combinen información espacial y recorrer el grafo sea ineficiente además de complejo. Si bien es muy bueno que la base brinde soporte geográfico nativo, el mismo es muy básico. Por esa razón es imposible escribir consultas que requieren de capacidades más complejas.

Neo4j por su parte brinda un soporte mucho más avanzado para el manejo de datos geográficos, 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 ej. Shapefile y OpenStreetMap.

El informe completo, las tablas, gráficos y el código fuente del presente estudio, se puede consultar en la siguiente URL: http://goo.gl/HjbqXI

Referencias

1. Neo4j Spatial (contrib) https://github.com/neo4j-contrib/spatial
2. ArangoDB, Geo Indexes. https://docs.arangodb.com/IndexHandling/Geo.html