



Neo4j

Bases de datos NoSQL

Realizado por:

Andrés Herranz González

Iván Alba

Luis Muñoz

ÍNDICE

1.- Introducción.....	3
2.- Definición	3,4
3.- Ventajas y desventajas	4, 5, 6
3.1.- Ventajas.....	5
3.2.- Desventajas.....	5,6
4.- Cypher Querys.....	6, 7
5.- Conclusión.....	7
6.- Bibliografía	8

1.- Introducción

Neo4j fue desarrollado por Neo Technology. La primera versión fue lanzada en febrero de 2010.

Diferentes empresas como eBay, Walmart, Telenor, UBS, Cisco, Hewlett-Packard o Lufthansa han confiado en las cualidades de Neo4j para mejorar sus servicios por la capacidad que tiene de analizar muchos datos de una manera muy eficiente, También ha sido empleada para sacar información de los “Papeles de panamá” los cuales tenían 2.6 terabytes, y 11,5 millones de documentos.

2.- Definición

Neo4j es un software libre de Base de datos orientada a grafos, el cual ha sido implementado en Java y pertenece a este tipo de base de datos NOSQL que

últimamente están tan en auge. Además, es completamente transaccional, que almacena datos estructurados en grafos en lugar de tablas. Esta base de datos orientada a grafos (BDOG) nos ayuda a encontrar relaciones y dar sentido al “puzzle” completo.

Los grafos que emplea son:

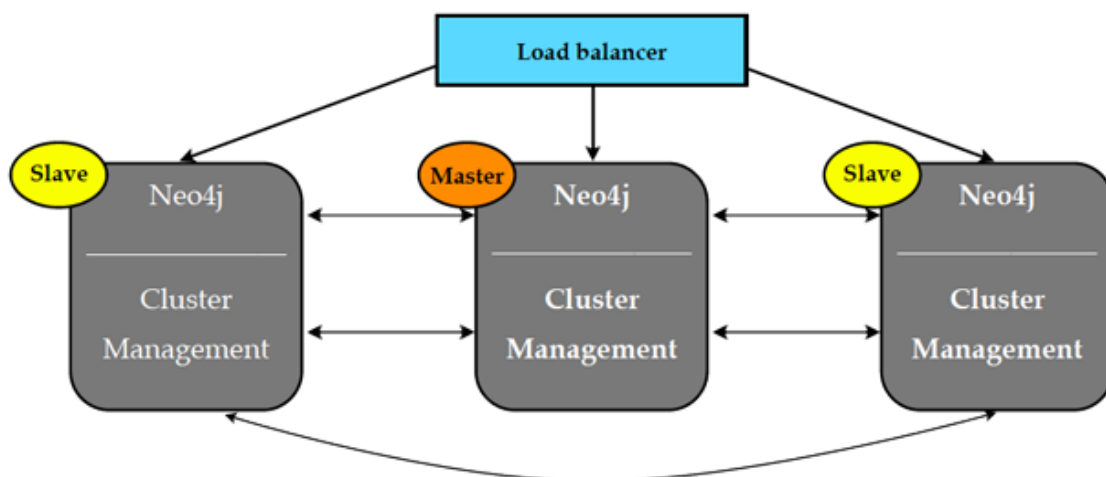
- **Grafos de propiedad:** es un grafo con peso, con etiquetas y donde podemos asignar propiedades tanto a nodos como relaciones (por ejemplo, cuestiones como nombre, edad, país de residencia, nacimiento). Es el más complejo.

Neo4j tiene un lenguaje propio para realizar consultas llamado Cypher que es un lenguaje bastante intuitivo y fácil de entender y es la forma declarativa de realizar consultas, porque luego está la forma nativa que es utilizando la API de java de Neo4j.

En Neo4j no se define ningún esquema, cada nodo o relación puede tener una estructura de datos diferente, lo que favorece la escalabilidad.

3.- Ventajas y desventajas

Justo antes de empezar a hablar de las ventajas y desventajas de Neo4j, queremos especificar que los clústers en Neo4j, permiten distribuir el gráfico en la memoria, al mismo tiempo que garantiza que cada instancia pueda acceder a cualquier nodo o relación en cualquier momento utilizando su propia copia local. La arquitectura de Neo4j le permite mantener la velocidad a medida que crece su gráfico, al tiempo que proporciona una alta disponibilidad a través de un protocolo maestro-esclavo.



3.1 Ventajas

Neo4j cuenta con las siguientes ventajas:

- Una de las ventajas de Neo4J es su capacidad de gestionar grandes números de datos de forma eficiente. Su capacidad se acerca a los 34.000 millones de nodos (datos), 34.000 millones de relaciones entre esos datos y 68.000 millones de propiedades.
- Las bases de datos orientadas a grafos como Neo4j tienen mejor rendimiento que las relacionales (SQL) y las no relacionales (NoSQL). La clave es que, aunque las consultas de datos aumenten exponencialmente, el rendimiento de Neo4j no desciende, frente a lo que sí sucede con las BBDD relacionales como MySQL. Las BDOG responden a las consultas actualizando el nodo y las relaciones de esa búsqueda y no todo el grafo completo. Eso optimiza mucho el proceso.
- Volker Pacher, desarrollador de eBay y cliente de Neo4j, explica con datos lo que supuso el cambio de MySQL a esta BDOG en el rendimiento de Shutl, la plataforma que coordina la entregas entre tiendas, mensajerías y compradores en eBay Now: “Nuestra solución Neo4j es literalmente mil veces más rápida que la solución anterior MySQL, con búsquedas que requieren entre 10 y 100 veces menos código”.

3.2 Desventajas

Neo4j tiene los siguientes inconvenientes:

- Al ser el almacenamiento de datos lo que más ocupa, si se almacena demasiada información en los nodos puede hacer que las consultas sean ineficientes (si no se ha modelado la información).

- Otra desventaja es que, al igual que MongoDB y otras BDD, quiere que tengas todo el grafo en memoria. Esto es bastante ineficiente ya que habrá partes del grafo que no usas.

4.- cypher Querys

Los 3 tipos de creates, cada uno con sus atributos:

- Para crear películas
`CREATE (TheMatrix:Movie {title:'The Matrix', released:1999, tagline:'Welcome to the Real World'})`
- Para crear Personas
`CREATE (Keanu:Person {name:'Keanu Reeves', born:1964})`
- Para crear relaciones:
`CREATE`
`(Keanu)-[:ACTED_IN {roles:['Neo']}]>(TheMatrix),`
`(Carrie)-[:ACTED_IN {roles:['Trinity']}]>(TheMatrix),`
`(Laurence)-[:ACTED_IN {roles:['Morpheus']}]>(TheMatrix),`
`(Hugo)-[:ACTED_IN {roles:['Agent Smith']}]>(TheMatrix),`
`(LillyW)-[:DIRECTED]>(TheMatrix),`
`(LanaW)-[:DIRECTED]>(TheMatrix),`
`(JoelS)-[:PRODUCED]→(TheMatrix)`

Algunos ejemplos de Querys con MATCH:

- Find movies released in the 1990s...
`MATCH (nineties:Movie) WHERE nineties.released > 1990 AND nineties.released < 2000 RETURN nineties.title`

- List all Tom Hanks movies...
MATCH (tom:Person {name: "Tom Hanks"})-[:ACTED_IN]->(tomHanksMovies) RETURN tom,tomHanksMovies
- How people are related to "Cloud Atlas" ...
MATCH (people:Person)-[:relatedTo]-(:Movie {title: "Cloud Atlas"})
RETURN people.name, Type(relatedTo), relatedTo
- Bacon path, the shortest path of any relationships to Meg Ryan
MATCH p=shortestPath((bacon:Person {name:"Kevin Bacon"})-[*]-
(meg:Person {name:"Meg Ryan"})) RETURN p

5.- Conclusión

Como conclusión obtenemos que se trata de una base de datos muy potente, utilizada por grandes empresas debido a las facilidades que aporta. Además, las funcionalidades especiales que tiene como, por ejemplo, ver qué relación existe entre dos objetos, hacen de esta base de datos una muy buena opción.

Por último, Cypher es un lenguaje entretenido, sencillo y amigable, con un gran potencial.

6.- Bibliografía

- https://neo4j.com/online_training/graphdatabases/?alild=IC8%3D
- <http://www.tutorialspoint.com/neo4j/AI08> Auditoría producto
- <http://www.w3ii.com/es/neo4j/default.html>
- Wikipedia