Bases de datos no convencionales

Bases de datos noSQL

Alejandro Pérez Barreiro
Juan Manuel Pacheco Bravo

Master de Data Science

Contenido

Figuras

Resumen

En la siguiente memoria se abordan cada uno de los puntos propuestos en el enunciado de la práctica de la asignatura.

Para la realización, además del despliegue en máquinas virtuales de MongoDB y Neo4j, se han desplegado ambas aplicaciones en un nodo ECS de AWS, de forma que ambos alumnos pudiéramos acceder a la práctica en paralelo.

Código fuente

Se muestran en la siguiente tabla cada uno de los scripts utilizados para la práctica. Cada uno de ellos pueden encontrarse dentro de la carpeta src adjunta en el paquete entregado, se ha probado su funcionamiento en Python 3.6

Script	Descripción
Xml-json.py	Conversión de la base de datos DBLP a formato json para su carga en MongoDB
Xml-csv.py	Conversión de la base de datos DBLP a formato csv para su carga en Neo4J
practica_mongo.ipynb	Notebook incluyendo las consultas para los ejercicios propuestos sobre mongo con pymongo.

Figura 1. Tabla de scripts

MongoDB

Procesado y carga

Puesto que MongoDB no admite ficheros en formato xml se ha hecho una conversión de los datos a formato json. El fichero de partida "dblp.xml" ha sido obtenido del siguiente enlace http://dblp.uni-trier.de/xml/dblp.xml.gz. Por facilitar el trabajo a la hora de la importación y el análisis de los datos hemos reducido el tamaño de los mismos pasando a formato json únicamente 2 millones de publicaciones de los tres tipos que nos interesan, que son "article", inproceedings" e "incollection", en lugar de cerca de las 4 millones las que se encuentran en el fichero dblp.xml. En cuanto a las características de la publicación, para descargar datos innecesarios nos hemos quedado únicamente con el autor o autores, el año de publicación, el título de la misma y el título del libro al que pertenece esa publicación en caso de que pertenezca a alguno.

El comando ejecutado para la carga del json en MongoDB:

```
mongoimport --db bdnc --collection dblp –-file dblp.json
```

Análisis

Para realizar las consultas en mongo se han utilizado dos aproximaciones:

- Consola de MongoDB
- Pymongo

Abajo, se pueden encontrar cada una de las consultas que se han introducido en la Shell de MongoDB. En el notebook adjunto, las mismas consultas pero utilizando la API de Python pymongo.

1. Listado de todas las publicaciones de un autor determinado.

- Query MongoDB

```
> db.dblp.find({"Authors": "E. F. Codd"}).pretty()
```

- Resultado (se muestran los tres primeros resultados)


```
{
    "_id" : ObjectId("58f4f9e5f6b3f41f2cacca2e"),
    "year" : 1974,
    "title" : "Interactive Support for Non-Programmers: The Relational and Network Approaches.",
    "type" : "article",
    "Authors" : [
        "E. F. Codd",
        "C. J. Date"
    ]
}
{
    "_id" : ObjectId("58f4f9e5f6b3f41f2cacca30"),
    "year" : 1981,
    "title" : "The Capabilities of Relational Database Management Systems.",
    "type" : "article",
    "Authors" : [
        "E. F. Codd"
    ]
}
{
    "_id" : ObjectId("58f4f9e5f6b3f41f2cacca32"),
    "year" : 1972,
    "title" : "Relational Completeness of Data Base Sublanguages.",
    "type" : "article",
    "Authors" : [
        "E. F. Codd"
    ]
}
```

2. Número de publicaciones de un autor determinado.

- Query MongoDB

```
> db.dblp.find({"Authors": "E. F. Codd"}).count()
```

- Resultado

```
34
```

3. Número de artículos en revista para el año 2016.

- Query MongoDB

```
db.dblp.find({$and:[{"year":2016},{"type":"article"}]}).count()
```

- Resultado

```
127165
```

4. Número de autores ocasionales, es decir, que tengan menos de 5 publicaciones en total.

- Query MongoDB:

```
> db.dblp.aggregate([ { $unwind: "$Authors" }, { $sortByCount: "$Authors"}, { $match: {count:{$lt:5} } }, { $count: "AutoresOcasionales"} ], {allowDiskUse:true})
```

- Resultado

```
{ "AutoresOcasionales" : 1152869 }
```

5. Número de artículos de revista (*article*) y número de artículos en congresos (*inproceedings*) de los diez autores con más publicaciones totales.

- Query MongoDB:

```
> db.dblp.aggregate([ {$match: { $or: [ {type:"inproceedings"}, {type:"article"} ] } }, { $unwind: "$Authors" }  ,{ $group: {_id:{ Author: "$Authors"}, publicaciones: {$sum:1}, "type": {"$push":"$type"} }}, {$sort: {"publicaciones":-1}}, {$limit:10} ,{ $unwind: "$type" }, { $group: {_id: {type: "$type", author: "$_id.Author"}, count: {$sum:1} } }, {$sort: {"_id.author":1}} ], {allowDiskUse:true})
```

- Resultado

```
{ "_id" : { "type" : "article", "author" : "Chin-Chen Chang" }, "count" : 584 }
{ "_id" : { "type" : "inproceedings", "author" : "Chin-Chen Chang" }, "count" : 29 }
{ "_id" : { "type" : "inproceedings", "author" : "H. Vincent Poor" }, "count" : 68 }
{ "_id" : { "type" : "article", "author" : "H. Vincent Poor" }, "count" : 915 }
{ "_id" : { "type" : "inproceedings", "author" : "Jing Li" }, "count" : 166 }
{ "_id" : { "type" : "article", "author" : "Jing Li" }, "count" : 406 }
{ "_id" : { "type" : "article", "author" : "Jun Liu" }, "count" : 440 }
{ "_id" : { "type" : "inproceedings", "author" : "Jun Liu" }, "count" : 122 }
{ "_id" : { "type" : "inproceedings", "author" : "Lajos Hanzo" }, "count" : 63 }
```

{ "_id" : { "type" : "article", "author" : "Lajos Hanzo" }, "count" : 558 }

{ "_id" : { "type" : "inproceedings", "author" : "Mohamed-Slim Alouini" }, "count" : 71 }

{ "_id" : { "type" : "article", "author" : "Mohamed-Slim Alouini" }, "count" : 520 }

{ "_id" : { "type" : "inproceedings", "author" : "Wei Zhang" }, "count" : 187 }

{ "_id" : { "type" : "article", "author" : "Wei Zhang" }, "count" : 463 }

{ "_id" : { "type" : "inproceedings", "author" : "Witold Pedrycz" }, "count" : 62 }

{ "_id" : { "type" : "article", "author" : "Witold Pedrycz" }, "count" : 602 }

{ "_id" : { "type" : "inproceedings", "author" : "Xiaodong Wang" }, "count" : 75 }

{ "_id" : { "type" : "article", "author" : "Xiaodong Wang" }, "count" : 482 }

{ "_id" : { "type" : "inproceedings", "author" : "Yang Liu" }, "count" : 164 }

{ "_id" : { "type" : "article", "author" : "Yang Liu" }, "count" : 395 }

6. Número medio de autores de todas las publicaciones que tenga en su conjunto de datos.

- Query MongoDB:

```
> db.dblp.aggregate([ { $project: {"_id": 0, "numero de autores": { $size: "$Authors" } } }, {$group: { _id:null, totalAutores: {$sum: "$numero de autores"} }}, {$project:{_id:0, AutoresPorPublicacion:{ $divide: ["$totalAutores", db.dblp.count()]}}} ])
```

- Resultado

{ "AutoresPorPublicacion" : 2.791140179038313 }

7. Listado de coautores de un autor (Se denomina coautor a cualquier persona que haya firmado una publicación).

- Query MongoDB

```
> db.dblp.aggregate([ {$match: {"Authors":"E. F. Codd"} }, {$unwind: "$Authors"}, {$group: {_id:"$Authors", count: {$sum:1}}}, {$match: {_id: {$ne: "E. F. Codd"}}}, {$project: {_id:1}}, {allowDiskUse:true} ])
```

- Resultado

{ "_id" : "E. S. Lowry" }

{ "_id" : "E. McDonough" }

{ "_id" : "Casper A. Scalzi" }

{ "_id" : "C. J. Date" }

8. Edad de los 5 autores con un periodo de publicaciones más largo (Se considera la Edad de un autor al número de años transcurridos desde la fecha de su primera publicación hasta la última registrada).

- Query MongoDB:

> db.dblp.aggregate([{ $unwind: "$Authors" } , {$group : { _id: "$Authors", anyos: { $push: "$year"} }}, {$project:{ maximo: {$max: "$anyos"}, minimo:{ $min: "$anyos"} }}, {$project: {edad: {$subtract: ["$maximo", "$minimo"] } } }, {$sort: {edad:-1}}, {$limit: 5}], {allowDiskUse:true})

- Resultado

{ "_id" : "Alan M. Turing", "edad" : 75 }

{ "_id" : "Rudolf Carnap", "edad" : 71 }

{ "_id" : "David Nelson", "edad" : 64 }

{ "_id" : "Eric Weiss", "edad" : 64 }

{ "_id" : "George E. Collins", "edad" : 63 }

9. Número de autores novatos, es decir, que tengan una Edad menor de 5 años (Se considera la edad de un autor al número de años transcurridos desde la fecha de su primera publicación hasta la última registrada).

- Query MongoDB:

> db.dblp.aggregate([{ $unwind: "$Authors" } , {$group : { _id: "$Authors", anyos: { $push: "$year"} }}, {$project:{ maximo: {$max: "$anyos"}, minimo:{ $min: "$anyos"} }}, {$project: {edad: {$subtract: ["$maximo", "$minimo"] } } }, {$match: {edad: {$lt:5}}}, {$count: "AutoresNovatos"}], {allowDiskUse:true})

- Resultado

{ "AutoresNovatos" : 1071478 }

10. Porcentaje de publicaciones en revistas con respecto al total de publicaciones.

- Query MongoDB

> db.dblp.aggregate([{ $match: { type: "article"} }, { $count: "PublicacionesRevistas"}, {$project:{_id:0, PorcentajeRevistas:{ $divide: ["$PublicacionesRevistas", db.dblp.count()]}}}])

- Resultado

{ "PorcentajeRevistas" : 0.7319727866171638 }

Neo4j

Procesado y carga

Para realizar la carga y dado que Neo4j no lee de forma nativa en ficheros en formato json, se ha obtado por traducir el fichero xml original (dblp.xml) a formato csv.

El modelo de datos que se ha tenido en cuenta para la carga de los datos se ha obtenido con la herramienta online disponible en http://www.apcjones.com/arrows. El resultado es el siguiente:

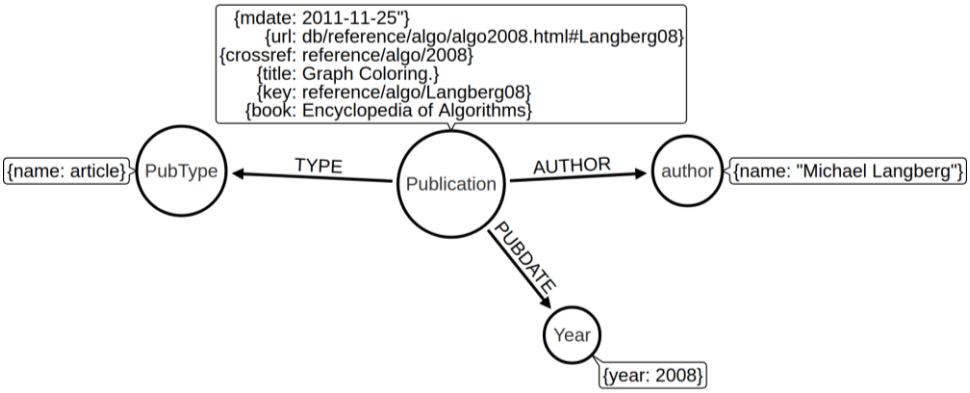

Figura 2. Modelo de Datos para Neo4j

Para la carga, se ha utilizado el siguiente código en Cypher:

```
CREATE CONSTRAINT ON (p:Publication) ASSERT p.key IS UNIQUE;
CREATE CONSTRAINT ON (p:Author) ASSERT p.name IS UNIQUE;
CREATE CONSTRAINT ON (p:Year) ASSERT p.id IS UNIQUE;
CREATE CONSTRAINT ON (p:PubType) ASSERT p.name IS UNIQUE;

USING PERIODIC COMMIT 10000
load csv with headers from "file:/dblp-full-2.csv" AS row
CREATE (pub:Publication {title: row.title, key: row.key, ee: row.ee, mdate:
row.mdate, url: row.url, book: row.booktitle})
merge (year:Year {id: row.year})
merge (pubtype:PubType {name: row.pubtype})
CREATE (pub)-[:PUBDATE]->(year)
CREATE (pub)-[:TYPE]->(pubtype)
with pub,split(row.author, ";") as authors
unwind authors as author
merge (a:Author {name: author})
CREATE (pub)-[:AUTHOR]->(a)
```

Una muestra del resultado de la carga se representa en la siguiente figura:

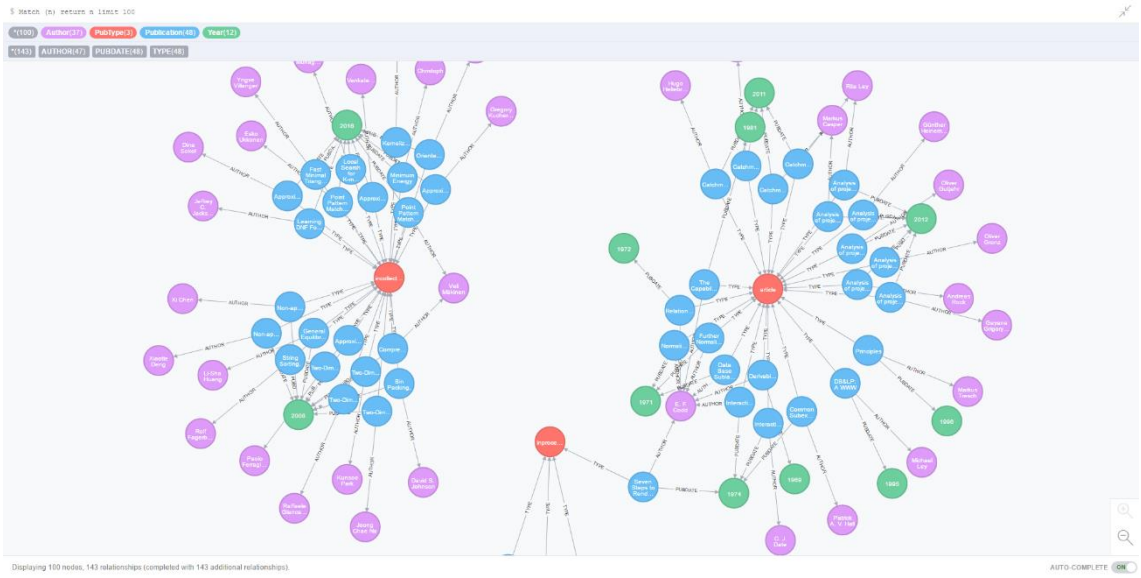

Figura 3. Muestra del grafo de publicaciones en Neo4j

A continuación, se adjuntan figuras especificas para cada tipo de nodo:

Figura 4. Nodos Author en Neo4j

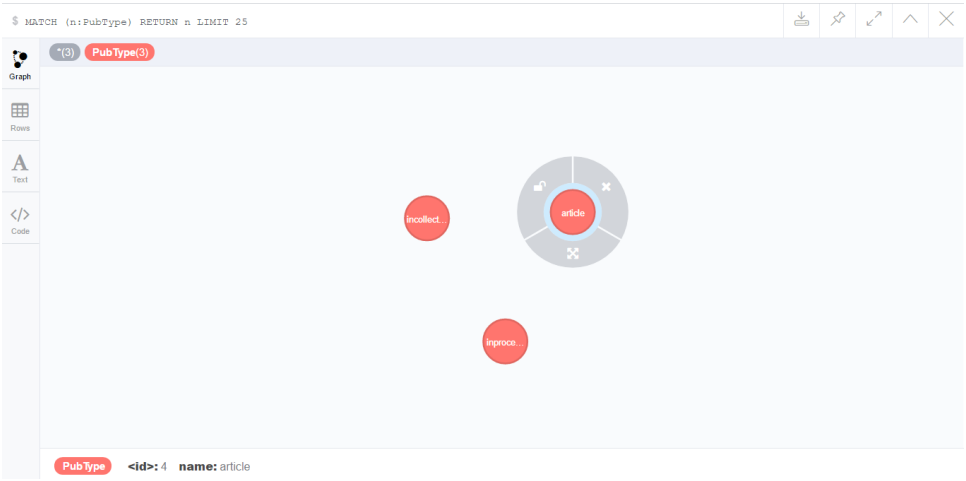

Figura 5. Nodos PubType(Tipo de publicación) en Neo4j.

Figura 6. Nodos Publication en Neo4j

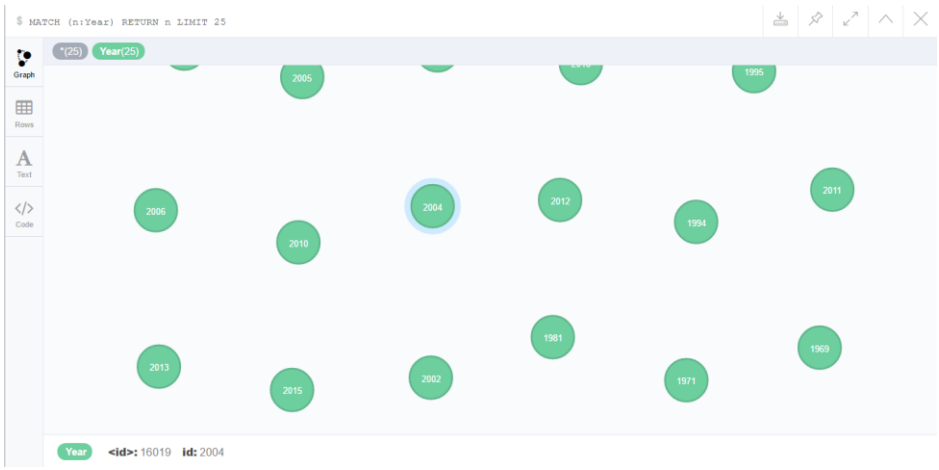

Figura 7. Nodos Year en Neo4j

Análisis

A continuación se comentan tres consultas en las que Neo4j es claramente más eficiente que MongoDB

1. Basadas en número de nodos de un tipo:

Ejemplo: Número de autores únicos. Dado que cada uno de los autores es un nodo en mongo, el número de autores únicos es simplemente el número de nodos autor

```
MATCH (n:Author) RETURN (count(n))
```

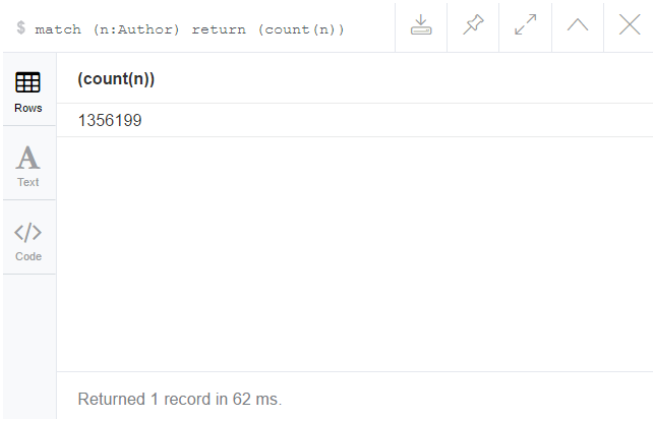

Figura 8. Número de autores en Neo4j

Este razonamiento es aplicable al resto de tipos de nodos que no son raíz en los documentos de mongo(publicaciones) como son tipos de publicación y años de las publicaciones.

2. Basadas en nodos incidentes en un nodo:

Ejemplo: Número de artículos de revista. Cada publicación tiene una arista que le comunica con el tipo de publicación asignado.

```
MATCH (a:Publication)-[:TYPE]->(:PubType { name:"article"})
RETURN COUNT(a)
```

Figura 9. Número de articulos de revista

3. Basadas en nodos incidentes a un nodo común

Ejemplo: Coautores de un autor

```
MATCH (:Author { name:"E. F. Codd"})--(article)--(n:Author)
RETURN distinct(n.name)
```

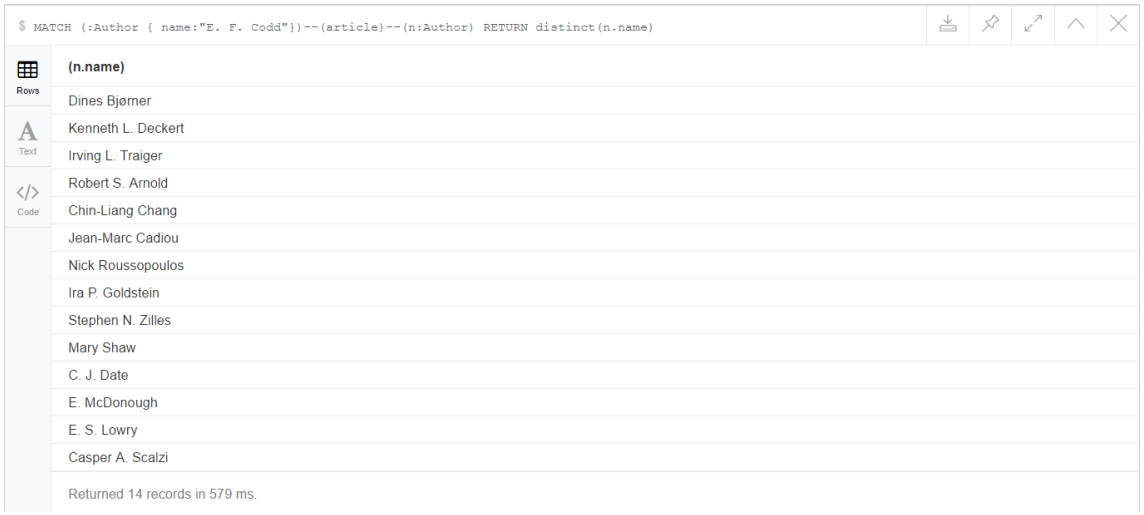

Figura 10. Coautores de E.F.Codd