Memoria práctica BBDD no convencionales

Descripción del proceso y definición de la BBDD

Para el desarrollo de la práctica se ha escogido el fichero de datos completo, aproximadamente 2.10GB.

Una vez descargado el fichero de datos el proceso se encarga de leerlo, crear las estructuras JSON necesarias para poder realizar la pposterior inserción en MongoDB. Dado el tamaño del fichero, para asegurar el correcto funcionaniento se realizan inserciones en BBDD cada 5000 registros.

La BBDD se ha organizado de la siguiente manera:

- 2 colecciones:
  - **publicaciones**: contiene la información de cada una de las publicaciones. Compuesta por los siguientes campos:
    - **_id**: campo *key* del *tag*
    - **tipo**: tipo de publicación, es el *tag* del *XML*
    - **titulo**: nombre de la publicación. Nodo *title* del *tag* tratado.
    - **year**: año de la publicación. Nodo *year* del *tag* tratado.
    - **autores**: lista de autores de la publicación. Lista de cada uno de los nodos *author* o *editor* del *tag* tratado.
  - **autores**: lista de todos los autores del fichero de datos. Compuesta de los siguientes campos:
    - **nombre**: nombre del autor. Extraído del *tag author* o *editor* de cada una de las publicaciones.
    - **year_ini**: año de la publicación con fecha más antigua.
    - **year_fin**: año de publicaciones con fecha más reciente.
    - **edad**: número de años transcurridos entre los datos year_ini y year_fin.
    - **num_publicaciones**: número de publicaciones en las que se ha encontrado a este autor.
    - **publicaciones**: lista de las publicaciones en las que se ha encontrado a este autor.
    - **coautores**: lista de los autores que aparecen junto a este autor en cada na de las publicaciones en las que este se ha encontrado.
- Se han creado los siguientes índices:
  - **publicaciones**: *year*, *tipo* y *autores*.
  - **autores**: *nombre*, *num_publicaciones* y *edad*

Esta distribución permite realizar todas las consultas solicitadas minimizando la necesidad de realizar consultas en ambas colecciones y simplificando las operaciones necesarias para poder realizar las mismas, es decir, se puede extraer la información de las publicaciones de un autor de forma directa, filtar los autores por el año de sus publicaciones por el rango de años de publicación de las mismas, extraer información de las publicaciones de un autor sin necesidad de tratar el conjunto completo de publicaciones.

Los 6 índices creado optimizan las consultas solicitadas, permitiendo la extracción de la información de forma rápida.

Ambas colecciones se actualizan con la misma frecuencia, es decir, cada 5000 registros, pero la de publicaciones se lanza según se va leyendo, a diferencia de la de autores, que se almacena de forma completa en memoria y posteriormente se trocea para su envío a la BBDD. El motivo es que la información de las publicaciones está contenida en cada uno de los tag's de forma independiente, pero la información de los autores se encuentra repartida por todo el fichero de datos.

Análisis de los datos

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

   Consulta: ***db.autores.find({"nombre":"Sanjeev Saxena"},{_id:0,publicaciones:1})***

   Resultado:


```
    "publicaciones" : [
        "Parallel Integer Sorting and Simulation Amongst CRCW Models.",
        "Optimal Sublogarithmic Time Parallel Algorithms on Rooted Forests.",
        "Parallel Algorithms for Finding the Most Vital Edge in Weighted Graphs.",
        "An efficient parallel algorithm for building the separating tree.",
        "Fast parallel edge colouring of graphs.",
        "Faster algorithm to find anti-risk path between two nodes of an undirected graph.",
        "Maximum cardinality neighbourly sets in quadrilateral free graphs.",
        "Maximal independent sets in a generalisation of caterpillar graph.",
        "Algorithms for testing occurrences of length 4 patterns in permutations.",
        "A still simpler way of introducing interior-point method for linear programming.",
        "Efficient VLSI Parallel Algorithm for Delaunay Triangulation on Orthogonal Tree Network in
Two and Three Dimensions.",
        "Two-Coloring Linked Lists is NC^1-Complete for Logarithmic Space.",
        "Dominance made simple.",
        "An optimal parallel algorithm for general maximal matchings is as easy as for bipartite
graphs.",
        "Parallel Algorithms for Connectivity Problems on Interval Graphs.",
        "On finding fundamental cut sets.",
        "Corrigendum: Optimal Parallel Algorithms for Coloring Bounded Degree Graphs and Finding
Maximal Independent Sets in Rooted Trees.",
        "Optimal Parallel Algorithms for Coloring Bounded Degree Graphs and Finding Maximal
Independent Sets in Rooted Trees.",
```

```
        "Local Nature of Brooks' Colouring for Degree 3 Graphs.",
        "Improved Deterministic Parallel Integer Sorting",
        "Optimal Parallel Algorithm for Brooks' Colouring Bounded Degree Graphs in Logarithmic Time on EREW PRAM.",
        "Parallel algorithms for separable permutations.",
        "On Parallel Prefix Computation.",
        "Ellipsoid Method for Linear Programming made simple.",
        "A Still Simpler Way of Introducing the Interior-Point Method for Linear Programming.",
        "Still Simpler Way of Introducing Interior-Point method for Linear Programming.",
        "A simple introduction to Karmarkar's Algorithm for Linear Programming.",
        "Maximal Independent Sets in Generalised Caterpillar Graphs.",
        "Maximum Cardinality Neighbourly Sets in Quadrilateral Free Graphs.",
        "Faster replacement paths algorithms in case of edge or node failure for undirected, positive integer weighted graphs.",
        "Parallel Vertex Colouring of Interval Graphs.",
        "Splay Trees.",
        "Parallel algorithms for single row routing in narrow streets.",
        "Efficient solutions for finding vitality with respect to shortest paths.",
        "Ultra-wideband antenna for a ground penetrating radar.",
        "Faster Replacement Paths Algorithm for Undirected, Positive Integer Weighted Graphs with Small Diameter.",
        "Parallel Algorithms for Testing Length Four Permutations.",
        "On Parallel Sorting and Addition with Concurrent Writes.",
        "Swap Edges of Shortest Path Tree in Parallel.",
        "Parallel algorithms for the longest common subsequence problem.",
        "Parallel algorithms for vehicle routing problems.",
        "Algorithms for Testing Length Four Permutations."
    ]
```


2. Número de publicaciones de un autor determinado.

   Consulta: ***db.autores.find({"nombre":"Sanjeev Saxena"},{_id:0,num_publicaciones:1})***
   Resultado:


```
        "num_publicaciones" : 42
```


3. Número de artículos en revista para el año 2017.

   Consulta: ***db.publicaciones.find({$and:[{"year":2017},{"tipo":"article"}]}).count()***
   Resultado:


```
        145839
```

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

   Consulta: ***db.autores.find({"num_publicaciones":{$lt:5}}).count()***

   Resultado:

   ```
   1641949
   ```

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

   Consulta:

```
db.autores.aggregate([{$sort:{num_publicaciones:-1}},{ $limit : 10 },{$project:{"nombre":1, "_id":0}},
{$lookup:{
        from:"publicaciones",
        localField:"nombre",
        foreignField:"autores",
        as: "publicaciones_selec"
}
},
{$project:{
   "nombre":1,
   "articles_selec":{
      $filter:{
         input:"$publicaciones_selec",
         as:"publicacion",
         cond:{$eq:["$$publicacion.tipo","article"]}
      }
   },
   "inproceedings_selec":{
      $filter:{
         input:"$publicaciones_selec",
         as:"publicacion",
         cond:{$eq:["$$publicacion.tipo","inproceedings"]}
      }
   }
}
},
{$project:{
   "nombre":1,
```

```
   "articles":{$size: "$articles_selec"},
   "inproceedings":{$size: "$inproceedings_selec"}
}
}
])
```

Resultado:

```
/* 1 */
{
    "nombre" : "H. Vincent Poor",
    "articles" : 1031,
    "inproceedings" : 479
}

/* 2 */
{
    "nombre" : "Philip S. Yu",
    "articles" : 395,
    "inproceedings" : 708
}

/* 3 */
{
    "nombre" : "Mohamed-Slim Alouini",
    "articles" : 607,
    "inproceedings" : 537
}

/* 4 */
{
    "nombre" : "Wei Wang",
    "articles" : 460,
    "inproceedings" : 634
}

/* 5 */
{
    "nombre" : "Wen Gao 0001",
    "articles" : 338,
    "inproceedings" : 742
}

/* 6 */
{
    "nombre" : "Yu Zhang",
    "articles" : 457,
```

```
    "inproceedings" : 619
}

/* 7 */
{
    "nombre" : "Lajos Hanzo",
    "articles" : 639,
    "inproceedings" : 407
}

/* 8 */
{
    "nombre" : "Wei Li",
    "articles" : 470,
    "inproceedings" : 552
}

/* 9 */
{
    "nombre" : "Li Zhang",
    "articles" : 430,
    "inproceedings" : 582
}

/* 10 */
{
    "nombre" : "Yang Liu",
    "articles" : 441,
    "inproceedings" : 560
}
```

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

   Consulta:

```
db.publicaciones.aggregate([
{$group:{
   _id:"totales",
   "total_publicaciones":{$sum:1},
   "total_autores":{$sum:{$size:"$autores"}},
   "media":{$avg:{$size:"$autores"}}
}
}
])
```

Resultado:


```
{
    "_id" : "totales",
    "total_publicaciones" : 4101376.0,
    "total_autores" : 11898231,
    "media" : 2.90103394568067
}
```


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

Consulta: ***db.autores.find({"nombre":"Sanjeev Saxena"},{_id:0,coautores:1})***

Resultado:


```
  "coautores" : [
        "Krzysztof Diks",
        "Pramod Chandra P. Bhatt",
        "Anjeneya Swami Kare",
        "K. V. R. C. N. Kishore",
        "K. Nandan Babu",
        "K. Jeevan Madhu",
        "Yijie Han",
        "M. R. Tripathy",
        "Tomasz Radzik",
        "Sudarshan Banerjee",
        "V. C. Prasad",
        "N. Malahal Rao",
        "Sagar Kalra",
        "Torben Hagerup",
        "Niloy Chakrabarti",
        "Neethi K. S.",
        "P. C. P. Bhatt",
        "V. Yugandhar",
        "Jay Mahadeokar",
        "G. Sajith",
        "Xiaojun Shen",
        "S. Das",
        "Kurt Mehlhorn"
    ]
```


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).

Consulta:

```
db.autores.aggregate([
{$sort:{"edad":-1}},
{$limit:5},
{$project:{
   edad:1,
   _id:0
}
}
])
```

Resultado:

```
/* 1 */
{
    "edad" : 75
}

/* 2 */
{
    "edad" : 71
}

/* 3 */
{
    "edad" : 68
}

/* 4 */
{
    "edad" : 64
}

/* 5 */
{
    "edad" : 64
}
```

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.

Consulta:

```
db.autores.aggregate([
{$match:{"edad":{$lt:5}}},
{$group:{
   _id: "total",
   count:{$sum:1}
}
}
])
```

Resultado:

```
{
    "_id" : "total",
    "count" : 1565770.0
}
```

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

Consulta:

```
db.publicaciones.aggregate([
{$project:{
   no_article: {$cond: { if: { $ne: [ "$tipo", "article" ] }, then: 1, else: 0 }},
   article:{$cond: { if: { $eq: [ "$tipo", "article" ] }, then: 1, else: 0 }}
}
},
{$group:{
   _id:"tipo",
   no_article:{$sum:{$add:["$no_article","$article"]}},
   article:{$sum:"$article"}
}
},
{$project:{
   _id:0,
   total:"$no_article",
```

```
    articles:"$article",
    porcentaje:{$divide:[{$multiply:["$article",100]},"$no_article"]}
}
}

]
)
```

Resultado:

```
"total" : 4101376.0,
 "articles" : 1789952.0,
 "porcentaje" : 43.6427189314025
```

Parte II: Neo4J

Al mismo tiempo que el proceso trata el XML y prepara el JSON que va insertando en la BBDD MongoDB, va generando 4 CVS's, dos de nodos y 2 de relaciones:

- *nodosPub.csv*: fichero de nodos de las distintas publicaciones.
- *nodosAut.csv:* fichero de nodos de cada uno de los autores.
- *relacionesPub.csv*: fichero de relaciones entre cada publicación y sus autores
- *relacionesAut.csv:* fichero con las relaciones entre los autores

esta forma de preparar los ficheros permite realizar directamente las inserciones de los dos ficheros de nodos, si necesidad de realizar validaciones, y tratar los fichero de relaciones teniendo la seguridad de que ya existen cada uno de los nodos.

Los fichero se crearán en la misma ruta desde donde está leyendo el XML origen.

Llegados a este punto, y uniéndolo al de comentar las diferencias entre MongoDB y Neo4J, he de decir que sólo he sido capaz de realizar las inserciones de los nodos, han tardado mucho tiempo pero ambas han finalizado, pero la operaciones de creación de las relaciones no ha finalizado en ningún caso, ni dejando el proceso ejecutándose más de un día.

Estas son las sentencias ejecutadas para realizar las inserciones, una vez copiados los ficheros a la ruta de importación de Neo4J:

```
USING PERIODIC COMMIT 500
LOAD CSV WITH HEADERS FROM "file:///nodospUB.csv" AS csvReg
CREATE (p:PUBLICACION {nombre: csvReg.nombre })


USING PERIODIC COMMIT 500
LOAD CSV WITH HEADERS FROM "file:///nodosAut.csv" AS csvReg
CREATE (p:AUTOR {nombre: csvReg.nombre })

USING PERIODIC COMMIT 500
LOAD CSV WITH HEADERS FROM "file:///relacionesPub.csv" AS csvReg
MATCH (autor:AUTOR { id: toInt(csvReg.nodo1)}),
(publicacion:PUBLICACION { id: toInt(csvReg.nodo2)})
CREATE (autor)-[:REL_{role: csvReg.relacion }]->(publicacion)

USING PERIODIC COMMIT 500
LOAD CSV WITH HEADERS FROM "file:///relacionesAut.csv" AS csvReg
MATCH (autor1:AUTOR { id: toInt(csvReg.nodo1)}),
(autor2:AUTOR { id: toInt(csvReg.nodo2)})
CREATE (autor1)-[:REL_{role: csvReg.relacion }]->(autor2)
```