TP Final - Base de Datos de Grafos

Alumna: Ximena Zuberbuhler (57287)

8 de Febrero 2023

Introducción

El objetivo del trabajo es representar grafos con múltiples tipos de vértices y ejes en los frameworks Graphframes o GraphX, por medio de la técnica de aplastamiento (flattening) y resolver algunas consultas.

Parte A: Carga de Datos

Para esta primera parte del trabajo se debe implementar la carga de los datos con la estructura del archivo *air-routes.graphml* provisto por la cátedra. La estructura de este grafo está dada de la siguiente manera:

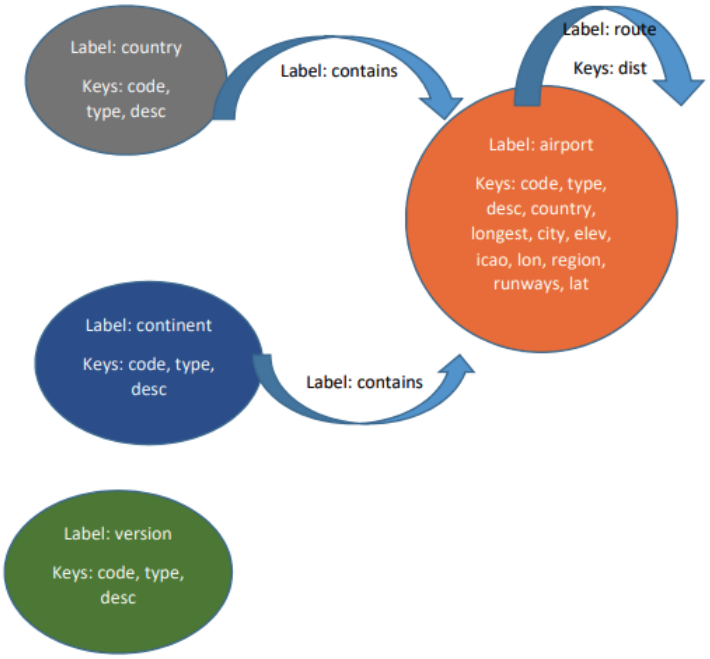

Fig 1. Esquema del grafo de vuelos.

Para este trabajo se decidió no cargar el nodo *version* ya que este solo posee metadatos del archivo que no aportan información necesaria para las consultas que se realizarán.

Como en un principio se consideró utilizar el framework GraphX, para la carga de los datos primero se lee el archivo graphml y se generan las listas de vértices y aristas como una lista de *Tuple2* para los vértices y una lista de *Edges* para las aristas. Luego, cuando se transforman a Listas de *Row* para ser cargadas como *DataFrame* y crear el *GraphFrame*.

Ejecución

El comando utilizado para ejecutar el programa dentro del cluster es:

> spark-submit --master yarn --deploy-mode cluster --class ar.edu.itba.graph.MainApp hdfs:///user/xzuberbuhler/final/TP-graphs-1.jar hdfs:///user/xzuberbuhler/final/files/<file-name>.graphml

Parte B: Procesamiento de Consultas

Consulta B1

Indicar para aquellos aeropuertos que tengan valores de latitud y longitud negativos, cuáles van al aeropuerto SEA (Seattle) usando a lo sumo una escala, y cuál es esa forma de llegar.

Para la verificación de la implementación de esta consulta se creó el archivo *test-1.graphml* el cual contiene el siguiente grafo:

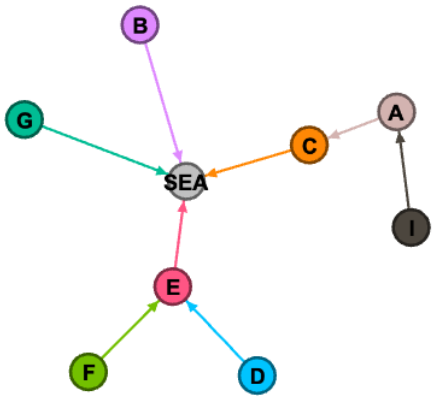

Fig 2. Grafo *test-1*

Donde en el caso de la Fig 2. las longitudes y latitudes de los aeropuertos se puede ver en la siguiente tabla:

Código del Aeropuerto	Longitud	Latitud
SEA	47	-122
A	33	-84
B	-1	-8
C	-33	-84
D	-3	-72
E	-93	4
F	0	-84
G	40	-104
I	-23	-14

En la tabla se coloreó en verde aquellos aeropuertos que cumplen con todas las condiciones de la consulta, es decir, su longitud y latitud son negativas y llegan a SEA con a lo sumo una escala.

En amarillo se marca un caso en el que se cumple la condición de longitud y latitud, pero requiere dos escalas para llegar SEA.

El resto son casos que cumplen con la condición de llegar a SEA con a lo sumo una escala pero no cumplen con tener longitud y latitud ambas negativas.

```
+-------+-------+
|Airport|  Route|
+-------+-------+
|      B|  B-SEA|
|      C|  C-SEA|
|      D|D-E-SEA|
+-------+-------+
```

Fig 3. Resultados de consulta B1 con *test-1.graphml*

En la Fig 3. se puede observar los resultados obtenidos en la consulta B1 utilizando como input *test-1.graphml*. Los aeropuertos devueltos en la consulta son B, C y D, que corresponden a los marcados en verde en la tabla.

Consulta B2

Listar por cada continente y país, la lista de valores de las elevaciones de sus aeropuertos. Debe aparecer una sola tupla por cada continente y país con la agrupación de los valores de las elevaciones registradas.

Al igual que para la consulta B1, se creó un archivo *test-2.graphml* para verificar la correctitud de la implementación. Este archivo posee el siguiente grafo:

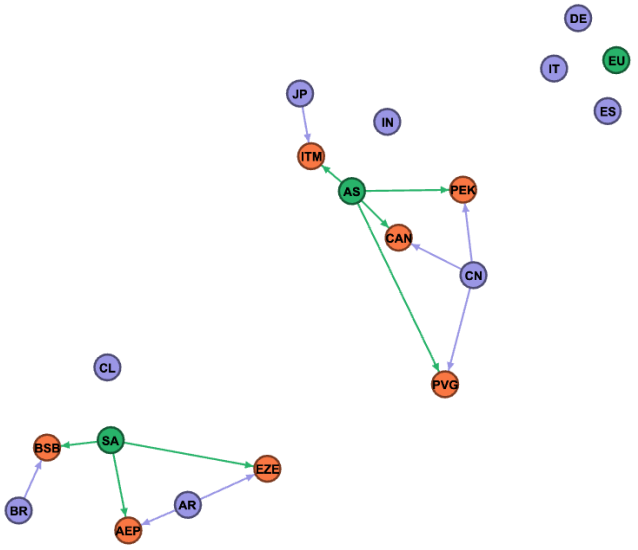

Fig 4. Grafo *test-2*

Donde las elevaciones de los aeropuertos se puede observar en la siguiente tabla:

Código del Aeropuerto	País	Elevación
PEK	China	116
PVG	China	13
CAN	China	50
ITM	Japón	50
EZE	Argentina	67
AEP	Argentina	18
BSB	Brasil	3497

```
+-------------+-------------+------------------+
|    Continent|      Country|Airport elevations|
+-------------+-------------+------------------+
|         Asia|    CN(China)|     [50, 13, 116]|
|         Asia|    JP(Japan)|              [50]|
|South America|AR(Argentina)|          [18, 67]|
|South America|   BR(Brazil)|            [3497]|
+-------------+-------------+------------------+
```

Fig 5. Resultados de consulta B2 con *test-2.graphml*

En la Fig 5. se observan los resultados obtenidos en la consulta B2 para el grafo en el archivo *test-2.graphml*. Comparando con la tabla anterior, se puede verificar que los resultados obtenidos son correctos.

Problemas

Se pide que los resultados obtenidos se almacenen en archivos de texto. Dado que cada nodo guarda su información en un archivo aparte, se optó por almacenar los archivos generados en una carpeta con el timestamp y número de consulta como nombre dentro de las carpeta *results*. Como solución, se encontró que la Ambari UI posee la funcionalidad de concatenar el contenido de los archivos en uno solo al momento de descargarlo, y así obtener los resultados en un solo archivo.