Parallel and Distributed Computing
Lab9
GraphFrames

In this lab, you will solve problems with Spark GraphFrames.

Importing GraphFrames

GraphFrames is not installed on Databricks by default. You will have to install the jar file as a library. You will be given the jar file. To install it, go to Clusters->running cluster->install new->then drop the jar on the landing pad.

DataSets

You will be answering a few specific questions about airline routes. There are 2 csv files you will need to perform this task:

routes.csv

airports.csv

These datasets were obtained from https://www.kaggle.com/open-flights/flight-route/database and https://openflights.org/data.html.

The routes.csv file contains the all the airline routes between airports anywhere in the world. The fields are:

airline, airlineID, sourceAirport, sourceAirportID, destinationAirport, destinationAirportID, codeshare, stops, planeType

The file contains 59k routes. Every route in the dataset has 0 stops. Note that there may be more than one route between 2 cities if multiple airlines have the same route. The important pieces in this file are the sourceAirport and destinationAirport (IATA codes like 'DEN').

An example of the file is:

2B,410,ASF,2966,MRV,2962,,0,CR2
2B,410,CEK,2968,KZN,2990,,0,CR2
2B,410,CEK,2968,OVB,4078,,0,CR2
2B,410,DME,4029,KZN,2990,,0,CR2
2B,410,DME,4029,NBC,6969,,0,CR2
2B,410,DME,4029,TGK,\N,,0,CR2

2B,410,DME,4029,UUA,6160,,0,CR2
2B,410,EGO,6156,KGD,2952,,0,CR2
2B,410,EGO,6156,KZN,2990,,0,CR2
2B,410,GYD,2922,NBC,6969,,0,CR2

The airports.csv file contains information about every airport in the world. The fields are:

airportID, name, city, country, IATA, ICAO, Lat, Long, Alt, timeZone, DST, databaseTimeZone, type, source

This file contains 10k airports. The fields we care about are country and the IATA (3 letter airport code).

An example of the file is:

3410,"Borg El Arab International Airport","Alexandria","Egypt","HBE","HEBA",30.917699813842773,29.696399688720703,177,2,"U","Africa/Cairo","airport","OurAirports"

3411,"Barter Island LRRS Airport","Barter Island","United States","BTI","PABA",70.1340026855,-143.582000732,2,-9,"A","America/Anchorage","airport","OurAirports"

3412,"Wainwright Air Station","Fort Wainwright","United States",\N,"PAWT",70.61340332,-159.8600006,35,-9,"A","America/Anchorage","airport","OurAirports"

3413,"Cape Lisburne LRRS Airport","Cape Lisburne","United States","LUR","PALU",68.87509918,-166.1100006,16,-9,"A","America/Anchorage","airport","OurAirports"

Creating the GraphFrame

In order to make a GraphFrame, you will need to create 2 DataFrames: one for edges and one for vertices. Let's focus on the edges first.

The routes.csv file contains all the edge information. In particular, there are fields for source and destination IATA codes. These are the codes we need in our DataFrame. But there are two issues.

First, multiple airlines might fly the same route, so you will want to remove duplicate routes. This should cut your number of routes about in half. Second, these are codes from all over the world. For this lab, we want to restrict routes to only those that start and end in the United States (partially to give you more practice cleaning data and partially to make the graph much smaller for the processing below).

For this second issue, we have the airports.csv file. This file contains IATA codes that match those in the routes.csv file. It also contains a country field that you can use to determine which IATA codes are from the United States.

Once you have your edge data cleaned, you will want to make the DataFrame that goes with the edges. Remember that you need 2 columns called "src" and "dst" to have this DataFrame work properly with GraphFrames.

Next you will need to create a DataFrame for the vertices. This should be a list of all the vertices in your graph. That is (assuming there are no isolated vertices with no edges - which there aren't for this lab – that would be odd having an airport that no one flew into), you need to get all the distinct src and dst values from your edges. Your vertex DataFrame needs to have a field called "id" to work properly with GraphFrames.

Finally, create the GraphFrame from your vertex and edge DataFrames.

Queries on the GraphFrame

Now that you have cleaned the data and constructed the graph, there are a number of queries you need to answer for this lab.

First, you should print out the number of US airport as well as the number of US to US routes in your graph. Your answer should be:

Number of US airport: 549
Number of US to US routes: 5450

Next, you should use Motifs to find all US airports that only have one way - but not round trip - flights to Denver (DEN). That is, airports where you can fly directly into DEN from but not back there directly or the reverse. Note that this doesn't include airports that have no flights at all (either to or from DEN). Your answer should be:

```
Airports with no direct roundtrip to or from DEN:
+-----+
| IATA|
+-----+
|[AIA]|
|[ORF]|
|[CDR]|
+-----+
```

And finally, you are to use the built-in shortest path algorithms to find all the US airports that require 4 or more flights to get to from DEN. Your results should be:

```
Airports that take 4 or more flights to get to from DEN:
+----+----+
|IATA|Hops|
```

```
+----+----+
| ANV|   4|
| AUK|   4|
| CEM|   4|
| HPB|   4|
| HSL|   4|
| KAL|   4|
| MLL|   4|
| NUL|   4|
| OOK|   4|
| SHG|   4|
| TNC|   4|
| TOG|   4|
| VAK|   4|
| VEE|   4|
| WNA|   4|
| WSN|   4|
| ARC|   5|
| NME|   5|
+----+----+
```

Submission

You are to attach 3 documents to Canvas as your submission. The first two are your Python source code (PY) and DataBricks archive (DBC) files. Make sure to have your name in a comment at the top of your source code. Third is a document that has a cut-paste of your results from running your code.