CptS 415 Big Data

Group 6 Final Report

Jinyang Ruan, Rusu Wu, Yi Yao, Junqiao Mou, Brian Chan

December 12, 2021

1. Problem Statement

   The topic we chose is Airline Search Engine. The mainly problems we were focusing on are implement airport and airline data search engine supported by efficient MapReduce, SQL, and geospatial analytics. More specifically, the problem can be divided into two parts:

   a. Airport and airline search:
      - Find list of airports operating in the Country X.
      - Find the list of Airlines having X stops.
      - List of airlines operating with code share
      - Find the list of active airlines in the United States - Airline aggregation.
      - Which country (or) territory has the highest number of Airports.
      - Top K cities with most incoming/outgoing airlines.

   b. Geospatial analytics: Use Apache Sedona (https://sedona.apache.org/)
      - Find the closest airport to a city X's geospatial coordinate.
      - Find the airport in each US state's geospatial boundary.

   There are 5 members is our team, the roles of each member and actual tasks we performed in this project can be shown as following:

   - Jinyang Ruan: Infrastructure manager. Focusing on implementing and compiling milestone 1, 2, final competition and final report, organizing teamwork.
   - Rusu Wu: ETL programmer. Focusing on implementing and compiling milestone 1, 2, and final competition.
   - Yi Yao: Data analyst. Focusing on implementing and compiling milestone 1, 3, helping teammates finish milestone 4, and final competition.
   - Junqiao Mou: Visualization expert. Focusing on implementing and compiling milestone 2, 4, and final competition.
   - Brian Chan: Visualization expert. Focusing on implementing and compiling milestone 4 and final competition.

2. Datasets

   Three datasets were used in the project. The brief description of each dataset can be shown below:

   - Airports.
     The format of Airports dataset is .dat, and the size of the dataset is 1.08MB. Totally, it has 14110 tuples and 14 attributes which include "Airport ID", "Name", "City", "Country", "IATA", "ICAO", "Latitude", "Longitude", "Altitude", "Timezone", "DST", "Tz database time zone", "Type", and "Source".
   - Airlines.
     The format of Airlines dataset is .dat, and the size of the dataset is 388KB.

Overall, it has 21317 tuples and 8 attributes which include “Airline ID”, “Name”, “Alias”, “IATA”, “ICAO”, “Callsign”, “Country”, and “Active”.

- Routes.
  The format of Routes dataset is .dat, and the size of the dataset is 2.27MB. Specifically, it has 67663 tuples and 9 attributes which include “Airline”, “Airline ID”, “Source airport”, “Source airport ID”, “Destination airport”, “Destination airport ID”, “Codeshare”, “Stops”, and “Equipment”.

  We used relational data model when we were working on those datasets.

3. Two interesting queries report
   a. Find the country or territory has the highest number of Airports.
   We implemented 2 methods for this query.
   Method 1:
   - Firstly, Count the number of airports by country

```
#1. Count the number of airports by country.#
Territory_Airports_Count = spark.sql("select Country, COUNT(Country) as Number from airports GROUP BY Country")
Territory_Airports_Count.show(10)
Territory_Airports_Count.createOrReplaceTempView("Territory_Airports_Count")
```

   - Secondly, find the maximum number from the result we get in step 1

```
#2. Then, find the maximum number in the step 1 result column.#
Territory_HighestNoOfAirports_Number = spark.sql("select MAX(Territory_Airports_Count.Number) from Territory_Airports_Count")
Territory_HighestNoOfAirports_Number.show()
Territory_HighestNoOfAirports_Number.createOrReplaceTempView("Territory_HighestNoOfAirports_Number")
```

```
[Stage 93:=================================================>   (187 + 2) / 200]
```

```
+-----------+
|max(Number)|
+-----------+
|       1512|
+-----------+
```

   - Lastly, print the name of the maximum number country or territory.

```
#3. List the Name of the maximum number country (or) territory.#
Territory_HighestNoOfAirports = spark.sql("select Territory_Airports_Count.Country, Territory_Airports_Count.Number from Territor
Territory_HighestNoOfAirports.show()
Territory_HighestNoOfAirports.createOrReplaceTempView("Territory_HighestNoOfAirports")
```

```
+-------------+------+
|      Country|Number|
+-------------+------+
|United States|  1512|
+-------------+------+
```

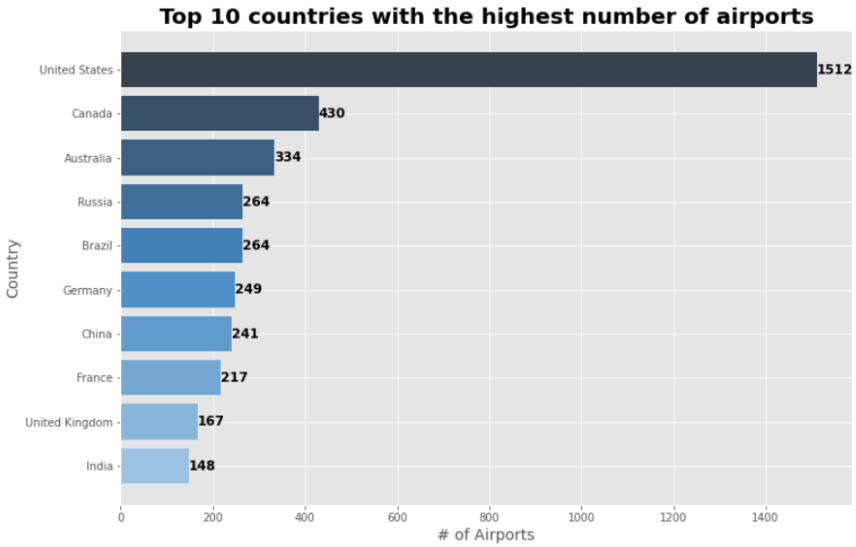
   The second method is more straight forward:
   Sort the result of the count by decreasing order then find the country or territory which has the most airports.

```
Territory_Number = spark.sql("select Country, COUNT(Country) as Number from airports GROUP BY Country ORDER BY Number DESC")
Territory_Number.show(10)
Territory_Number.createOrReplaceTempView("Territory_Number")
```

```
+--------------+------+
|       Country|Number|
+--------------+------+
| United States|  1512|
|        Canada|   430|
|     Australia|   334|
|        Russia|   264|
|        Brazil|   264|
|       Germany|   249|
|         China|   241|
|        France|   217|
|United Kingdom|   167|
|         India|   148|
+--------------+------+
only showing top 10 rows
```

   We used Bar chart for this query, the code and the chart can be show below:

```
plt.style.use("ggplot")
plt.figure(figsize = (12,8))
pal = sns.color_palette("Blues_d", len(airportsByCountry_top10["Airport_cnt"]))
hbars = plt.barh(width="Airport_cnt", y="Country", data=airportsByCountry_top10, color=np.array(pal[::-1]))
plt.gca().invert_yaxis()
plt.title("Top 10 countries with the highest number of airports", fontsize = 20, weight = 'bold')
plt.xlabel("# of Airports", size = 14)
plt.ylabel("Country", size = 14)
plt.bar_label(hbars, size = 12, weight = "bold")
plt.show()
```



We also make the color deeper when the number becomes larger in order to make the graph easy to read.

One interesting observation we find is the United States has the most airports in the world, and the number is much big than the others. By simple calculation, we can find even though we sum up the number of the 2nd to the 4th countries' airports, it still less than the number of the US airports.

b. The second query we did is to find the airport in each US state's geospatial boundary. Firstly, we load state boundaries in WKT TSV and convert WKT string column to a geometry column.

```
#Q8. Find the airport in each US state's geospatial boundary#
#1. Load state boundries in WKT TSV and convert WKT string column to a geometry column.#
airports_point = spark.sql("select *, ST_Point(CAST(Longitude as Double), CAST(Latitude as Double)) as Point from airports")
airports_point.show(5)
airports_point.printSchema()
airports_point.createOrReplaceTempView("airports_point")
```

Then we load city locations and convert the string column to a geometry column.

```
#2. Load city locations and convert the string column to a geometry column.#
states_wkt = spark.read.option("delimiter", "\t").option("header", "false").csv("boundary-each-state.tsv").toDF("s_name","s_bound
states_wkt.show(5)
states_wkt.printSchema()

states = states_wkt.selectExpr("s_name", "ST_GeomFromWKT(s_bound) as s_bound")
states.show(5)
states.printSchema()
states.createOrReplaceTempView("states")
```

Thirdly, get the list of the airport in each US state's geospatial boundary.

```
#3. We can get the list of the airport in each US state's geospatial boundary.#
city_per_state = spark.sql("select states.s_name, states.s_bound, airports_point.Name, airports_point.City, airports_point.Point
city_per_state.show(100)
```

Lastly, we find the airport in each US state and count them.

```
airports_point = spark.sql("""
                           SELECT *,
                               ST_Point(CAST(Longitude as Double), CAST(Latitude as Double)) as Point
                           FROM airports""")
airports_point.createOrReplaceTempView("airports_point")
```

```
airport_per_state = spark.sql("""
                           SELECT states.s_name,
                               states.s_bound,
                               airports_point.AirportID,
                               airports_point.Name,
                               airports_point.City,
                               airports_point.Point
                           FROM states, airports_point
                           WHERE ST_Contains(states.s_bound, airports_point.Point)
                           ORDER BY states.s_name""")
airport_per_state.createOrReplaceTempView("airport_per_state")
```

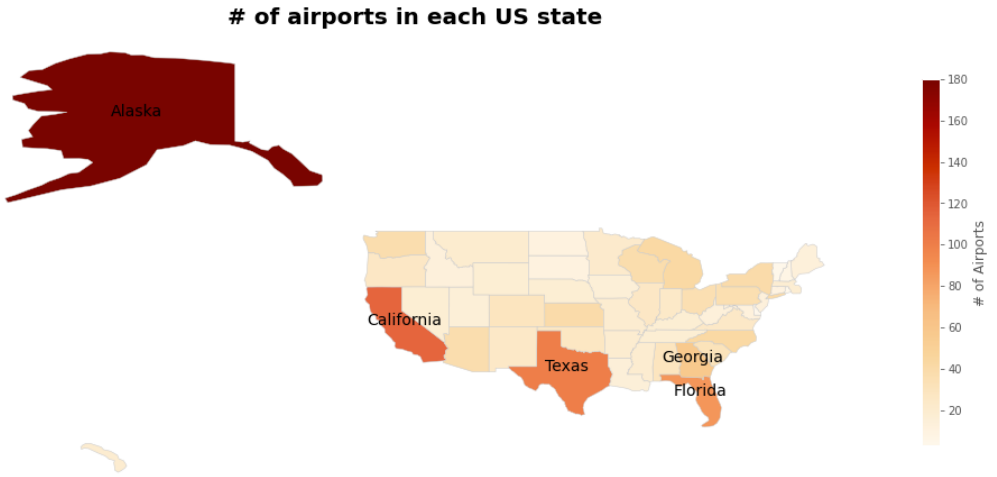
We can count the number of airports in each state and visualize using choropleth maps.
The state having the most number of airports has the darkest color.

```
airport_cnt_per_state = spark.sql("""
                                  SELECT s_name as State,
                                      s_bound,
                                      COUNT(DISTINCT AirportID) as airport_cnt
                                  FROM airport_per_state
                                  GROUP BY 1,2
                                  ORDER BY 3 DESC""")
airport_cnt_per_state_pd = airport_cnt_per_state.toPandas()
airport_cnt_per_state_gpd = gpd.GeoDataFrame(airport_cnt_per_state_pd, geometry='s_bound')
airport_cnt_per_state_gpd.head(10)
```

```
21/12/07 02:28:49 WARN JoinQuery: UseIndex is true, but no index exists. Will build index on the fly.
```

We use map visualization for this query, for the state has more airports, the color becomes darker.

```
airport_cnt_per_state_gpd.plot("airport_cnt", cmap = 'OrRd', figsize = (18,15), edgecolor = "0.8",
                               legend = True,
                               legend_kwds = {"label": "# of Airports","shrink": 0.4})
plt.axis("off")
plt.title("# of airports in each US state", fontsize = 20, weight = 'bold')
airport_cnt_per_state_gpd.apply(lambda x: plt.annotate(text=x["State"], xy=x.s_bound.centroid.coords[0], ha="center", fontsize =
plt.show()
```



The interesting observation is Alaska has the most airports in the US, which is not expected at the beginning. After searching on Google, Alaska has the 48th population in the US, even it has such small population, it still has the most airports over the country.