BigData 2025 Group 17

Project Big Data is provided by University of Tartu.

Project Title: Project 3 - Flight Interconnected Data Analysis

Students: Siddiga Gadirova, Andres Caceres, Fidan Karimova, Moiz Ahmad.

Introduction

This report presents a network analysis of the 2009 flight dataset using Apache Spark and GraphFrames. The primary objective is to explore airport connectivity and detect structural patterns such as hubs, triangles, and influence in the flight network. We construct a directed graph where each airport is a node and each flight is a directed edge. The analysis includes degree statistics, triangle counts, centrality measures, and a custom PageRank implementation to evaluate airport importance

Files to be used

- BigGraph.ipynb: Main analysis notebook with all queries implemented.
- 2009.csv: Dataset containing flight records for the year 2009.
- Dockerfile and docker-compose.yml: Environment setup to support Spark + GraphFrames

Requirements

1. Dependencies are listed in the `requirements.txt` file

2. Run `docker compose up -d` in the terminal

User guide: How to run this Notebook

To run the notebook successfully and reproduce the results:

1. Use the provided Dockerfile and `docker-compose.yml` to launch the Spark + Jupyter environment. Run `docker compose up -d` in the terminal

2. Place the file `2009.csv` into the local folder “mnt”.

3. Running the Notebook

- Access Jupyter via `http://localhost:8888`.
- Open the notebook `BigGraph.ipynb`.
- Run all cells sequentially from top to bottom.
- Each query and computation are documented using markdown cells.

4. Output:

- Results (degree statistics, triangle counts, PageRank, etc.) are displayed inline.
- Key screenshots of results are provided in the report for reference.

Observations

The dataset revealed a vast network of 278 airports and over 4,000 unique flight routes across the United States.

Major hubs such as ATL, ORD, and DFW were identified through degree statistics, confirming their high connectivity.

Triangle detection revealed densely interconnected airport clusters, suggesting frequent regional circuit paths.

While degree measures highlighted activity, PageRank uncovered airports with broader network influence.

Together, the metrics provide a comprehensive view of the air transportation network structure in 2009.

Queries:

1. Graph Construction

The dataset `2009.csv` was used to construct the graph. Each row represents a flight, with `ORIGIN` and `DEST` fields used to define directed edges between airport vertices.

```
=== Graph Basic Statistics ===
```

Component	Count
Vertices (airports)	296
Edges (flights)	6429338

2. Query 1 – Degree Statistics and Triangle Count

For each airport node, we computed the in-degree (flights arriving), out-degree (flights departing), total degree (sum of in and out), and the number of triangles the node is part of. This helps identify highly connected and clustered airports.

Results are below:

```
=== Query 1: Degree Statistics ===
+----+----------+---------+------------+--------------+
|node|out_degree|in_degree|total_degree|triangle_count|
+----+----------+---------+------------+--------------+
| ATL|    417449|   417457|      834906|         12183|
| ORD|    313848|   313769|      627617|          2687|
| DFW|    264396|   264398|      528794|          5995|
| DEN|    235675|   235700|      471375|          4749|
| LAX|    192879|   192916|      385795|          2166|
| PHX|    183502|   183491|      366993|          1934|
| IAH|    182097|   182088|      364185|          2923|
| LAS|    153993|   153984|      307977|          2213|
| DTW|    152081|   152075|      304156|          4692|
| SFO|    136488|   136532|      273020|          1575|
| SLC|    131694|   131674|      263368|          1663|
| MCO|    120944|   120936|      241880|          1931|
| MSP|    119732|   119759|      239491|          2401|
| JFK|    119574|   119571|      239145|          1801|
| EWR|    118602|   118602|      237204|          2530|
| CLT|    116650|   116640|      233290|          2383|
| BOS|    110460|   110463|      220923|          1520|
| SEA|    100948|   100922|      201870|          1570|
| BWI|    100923|   100928|      201851|          1732|
| LGA|    100334|   100323|      200657|          1488|
+----+----------+---------+------------+--------------+
only showing top 20 rows
```

3. Query 2 – Total Number of Triangles in the Graph

We implemented a custom method using edge joins to detect triangle patterns without using GraphFrame's built-in `triangleCount()`. The result reflects the number of triangular interconnections in the entire graph.

```
=== Query 2: Triangle Count (Custom Implementation) ===
Total triangles in the undirected graph: 16015
```

For comparison we calculated triangle count with GraphFrame's functions as well and result is: **Exact triangle count: 16015**

It shows that our custom implementation is correct.

4. Query 3 – Centrality Measure

Degree centrality was chosen as the metric for evaluating the importance of airports. It was computed as: total_degree / (number of vertices - 1). The airports with the highest centrality values are central hubs in the network. Results are below (limited to top 10 – you can see more when you will run the cell):

```
=== Query 3: Degree Centrality ===
```

node	degree_centrality
ATL	2830.1898305084746
ORD	2127.515254237288
DFW	1792.5220338983052
DEN	1597.8813559322034
LAX	1307.7796610169491
PHX	1244.04406779661
IAH	1234.5254237288136
LAS	1043.9898305084746
DTW	1031.0372881355931
SFO	925.4915254237288

5. Query 4 – PageRank Implementation

We manually implemented the PageRank algorithm in PySpark using rank propagation and damping. This identifies the most 'important' airports based on their link structure, rather than just direct connections.

Here we manually implement the **PageRank algorithm**:

- Each airport starts with an equal rank.
- At each iteration, rank is redistributed from source to destination nodes.
- A damping factor (commonly 0.85) is applied to simulate teleportation probability.

We repeat this for 10 iterations and rank airports by their final scores.

The results for query 4 are below:

```
=== Query 4: Custom PageRank ===
```

id	rank
ATL	2.3561081790300507E47
ORD	2.1475525152182944E47
DFW	1.8337063895243112E47
DEN	1.7879699261447383E47
LAX	1.7327029344334827E47
PHX	1.5514294916968752E47
LAS	1.4656286648064942E47
IAH	1.3025335044602526E47
SFO	1.2498321865847688E47
BOS	1.1807420077410707E47

6. Query 5 – Most Connected Airports

The final task was to rank the most connected airports based on their total degree. This highlights major airport hubs.

We sort airports by their **total degree** to identify the most connected hubs in the network — those with the highest number of incoming and outgoing flights combined.

Here are results limited to the top 10.

```
=== Query 5: Most Connected Airports ===
```

node	out_degree	in_degree	total_degree
ATL	417449	417457	834906
ORD	313848	313769	627617
DFW	264396	264398	528794
DEN	235675	235700	471375
LAX	192879	192916	385795
PHX	183502	183491	366993
IAH	182097	182088	364185
LAS	153993	153984	307977
DTW	152081	152075	304156
SFO	136488	136532	273020

Conclusion

This project successfully applied graph-based analysis to the 2009 U.S. flight dataset using Apache Spark and GraphFrames. We constructed a directed graph from flight data, with airports as nodes and flights as edges, enabling a network-level understanding of the air transportation system.

Key findings include:

- **Atlanta (ATL), Chicago O'Hare (ORD),** and **Dallas/Fort Worth (DFW)** emerged as the most connected airports based on total flight volume.

- Triangle detection revealed numerous interconnected airport triplets, pointing to common travel circuits and regional clustering.

- Degree centrality highlighted the busiest airports, while PageRank helped uncover influential airports based on their broader connectivity in the network.

By avoiding built-in shortcuts and implementing triangle counting and PageRank manually, we demonstrated a deep understanding of distributed graph analytics. The results align with real-world expectations and showcase the value of Spark in handling large-scale, interconnected data.