DV8: Interactive Geo-Spatial Analysis of Aviation Data

Yifan Song, Keyang Yu, Seth Young, Ph.D. (Adviser) {song.1221, yu.2049, young.1460} @ osu.edu

Department of Computer Science & Engineering / Center for Aviation Studies, The Ohio State University, Columbus, Ohio, USA

THE OHIO STATE UNIVERSITY

DV8 is an interactive visualization framework which provides immediate and interactive visualizations of spatiotemporal aviation data. *DV8* is built on a database structure designed to optimize spatiotemporal data from the Federal Aviation Administration SWIM traffic flow management system (TFMS) data stream. *DV8* is designed with analysis tools to allow for interactive filtering, color coding, and selecting of various flight characteristics, allowing users to gain valuable insights into flight routing performance.

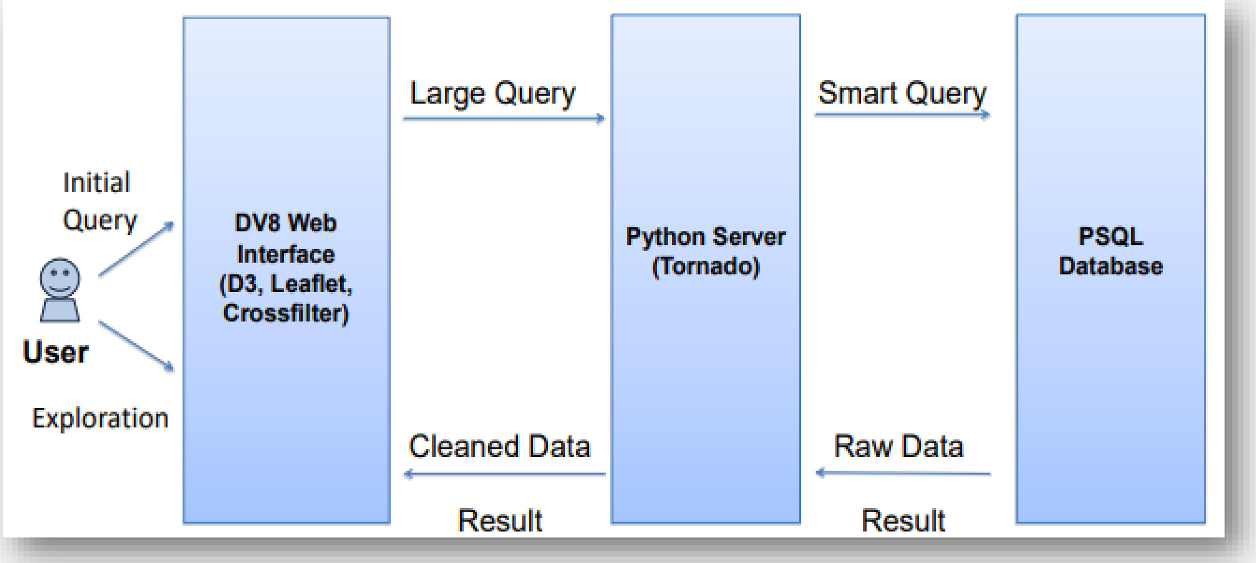
System Architecture

User
The user first issues their initial query to pull data into the interface. Then they explore the data and make iterative queries to refine their selection. DV8's dedication to speed and touch controls make data exploration natural for experts.

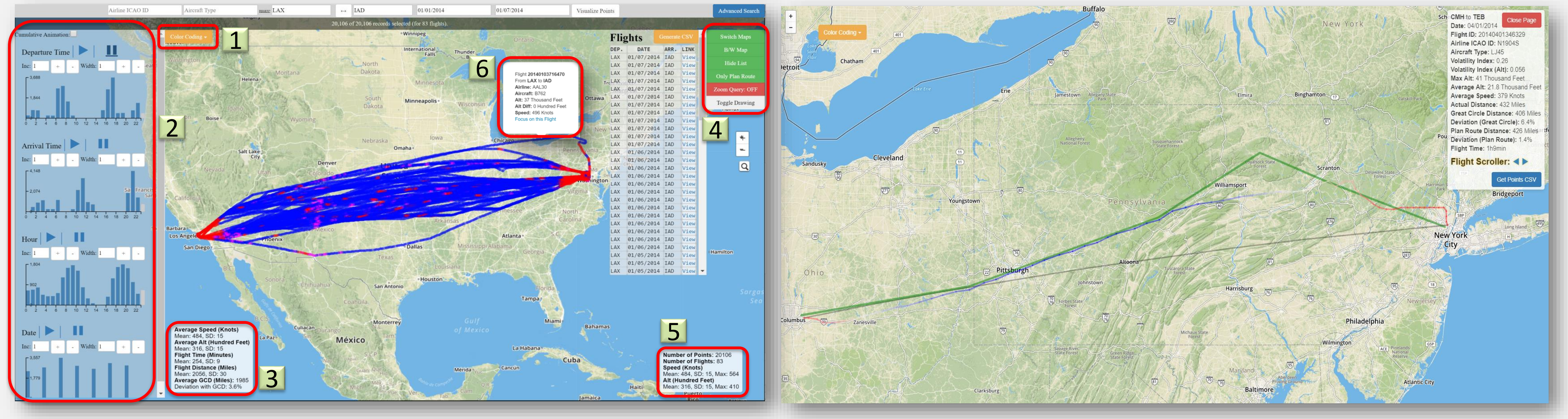
Interface
The Web Interface visualizes the data and provides analysis tools. The tools are designed to be modular so any expert can create custom tools for their specific spatiotemporal dataset.

Python Server
The Server contains the algorithms that make data retrieval fast. The smart query has three parts: the server requests meta-data from the DB, the DB responds, then the server exploits the meta-data and the DB's construction to create a fast query which retrieves a manageable subsection of the large query. The two queries are completed in sub-second time.

PSQL Database
The Database currently contains 2+ billion data points from 40+ million flights, more live data is coming. Proper indexing and logical sharding make the database scalable.



Application Interface



Multi Flight Visualizer

1. Color coding is applied to the points on the map in many ways helpful to visualization, including: altitude, speed, airline, aircraft, day of the week and cluster *(more on this in the next section).* The current view is color coded by *altitude difference* which can show turbulence.
2. Cross-filter charts. Selecting a range of data on these charts will filter out the unselected data. These charts organize data by time (hour and date); filtering can reveal patterns in time. Animating the charts also shows data over time.
3. Statistic summary for flights including: speed, altitude, time, distance, and deviation to great circle distance.
4. Command buttons include functions: a) switch to heatmap, b) switch to b/w map, c) show flight list and generate aggregate csv file, d) toggle drawing to select points on the map.
5. Statistic summary for the query including number of points and flights searched.
6. Pop-up information when clicking on a certain point. Display point and related flight information; link to single visualizer focusing on specific flight.

Single Flight Visualizer
Selecting an individual flight on the map can bring up the Single Flight View. More information and special metrics can be shown here: volatility index, deviation - the amount an airplane deviates from the most efficient path or planned path. New metrics can be added depending on what analysis needs. Color coding is still available. The green line representing planned route and the gray line representing direct path are shown for comparison.

Optimization

- *Quantized Granularity* – Quantized Granularity delivers every n^{th} point along a flight path. The Server delivers approximate data, but the relevant aspects of a flight path (shape, arrival/departure) are preserved.
- *Caching* – Data is saved locally so repeat queries or sub-queries (which are common in analytics) are completed immediately.

Route Categorization

Machine learning development to categorize route by flight path clustering

Hierarchical Agglomerative Clustering: No need to provide the expected number of clusters beforehand, hierarchical clustering finds the appropriate number of flight groups based on threshold input. Our hierarchical clustering algorithm provides two ways of computing similarity, geographical distance and cosine similarity.

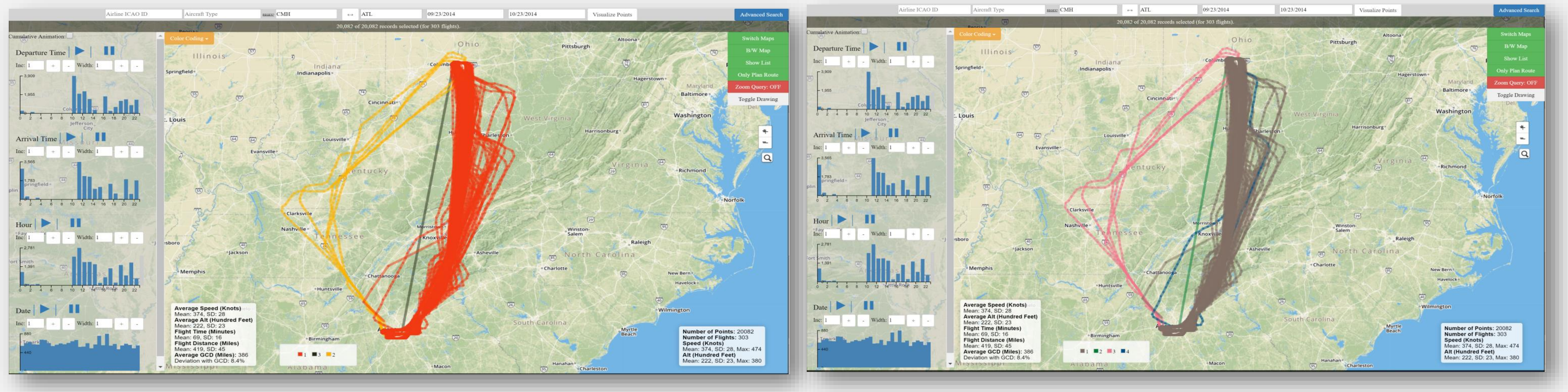
Two metrics:

- *Geographical Distance*: We used great circle distance to measure the distance along the surface of the earth. If two flight routes' geographical distance (average by point pairs) is greater than certain value, then they are categorized into two groups.
- *Cosine Similarity*: For every two consecutive data points in a flight route, subtract longitude and latitude to get a vector of flight direction. We then calculate cosine similarity based on every pair of vector to measure the direction similarity between two flight routes.

Sample Points Extraction: Based on the extremely large data size of each query, our application allows users to save running time by selecting a small proportion of points for each flight that keep the shape and location.

User-selected threshold: After select the similarity function, users can enter specific threshold for clustering. They can then compare different clustering results based on different metrics and thresholds.

Further Steps
We plan to divide the flight routes into 3 segments, departure, middle flying and arrival. Cosine similarity is a good measurement for orientation of flight route, which is particularly useful for clustering during departure and arrival. Geographical distance is a good measurement to calculate the distance in the middle process. By combining these two measurements together, our clustering algorithm can possibly yield a better result.



Color coding by cluster result, generated by geographical distance (left) and cosine similarity (right)

The research team would like to thank the Federal Aviation Administration and NetJets, Inc. for their support on this project. More about this project may be found at http://u.osu.edu/young.1460/dv8 and https://dv8.osu.edu

99^{th} Annual TRB Meeting
Washington, D.C, January 13, 2020