INTRODUCTION & BACKGROUND

Stratolaunch is taking big steps in the high-speed flight industry using the Roc, the world’s largest aircraft, as a mobile launch platform to deploy high speed vehicles into the hypersonic environment. In order to properly address our goals for the year, we decided to split up into two different teams. The visualization team is developing an interactive web app to replace a pre-existing dashboard, while the comparative analysis team is researching and testing databases to find the most performant database that matches Stratolaunch’s needs.

COMPARATIVE ANALYSIS

Flexible to allow for machine learning and artificial intelligence
Must be easy to use so we can continue to research and develop
Shift away from using Plotly for visualization
We must choose one of the many available databases for our project

FRONTEND

Applied query testing to all databases
Worked together to obtain a large list of data
Selected React (because developers had prior experience with React), when other frameworks were considered
Can only hold CSV data
Research rendering methods: dynamically generating plots (render the plot on the frontend) or image generation (render an image on the backend and send it to the frontend)
Utilized example code/documentation/tutorials for React

TECH STACK

BACKEND

Python
Flask

FRONTEND

Tailwind
React
JavaScript

CONCLUSIONS

Visualization

Our project is a data visualization dashboard (below) with the following features:

- Rendering data from multiple CSV files simultaneously (side-by-side or overlaid)
- Limits/Biases: Dynamic/Image Rendering: Chose image rendering to ensure no data would be sent between the client and server and for related performance reasons
- Specifying date and time range to plot
- Bokeh/Plotly: Plotly WebGL is considerably faster than Bokeh on all tested samples
- Goals: Aid flight testing progression by allowing complex flight data to be rendered easily
- Create a web application to replace a pre-existing dashboard, while the comparative analysis team is researching databases to find the most performant database that matches Stratolaunch’s needs
- Specify SQL database to analyze data effectively
- Developed React because developers had prior experience with React, when other frameworks could have been more performant

COMPARATIVE ANALYSIS

InfluxDB
- Ingest: Multiple Ingest Methods - InfluxDB Line Protocol for small data sets - Simple Ingest from CSV
- Queried: Uses Influx Query Language (QL)
- Lots of documentation
- Easy to translate from SQL and Python

QuestDB
- Ingest: Can be done within web console
- CSV ingest or as a replacement for the SQL database
- Automatic Data Type Detection and Conversion
- Queried: SQL
- Built-in timestamp functions
- Ease of Use: Easy to translate to Python

TimescaleDB
- Ingest: Can be done within web console
- CSV ingest or as a replacement for the SQL database
- Queried: SQL
- Built-in timestamp functions
- Ease of Use: Easy to translate to Python

Apache Druid
- Ingest: Manual ingest in terminal
- Requires all column data to be indexed
- Automatic Data Type Detection and Conversion
- Queried: SQL
- Built-in timestamp functions
- Ease of Use: Requires additional setup

DolphinDB
- Ingest: Simple ingest
- Unavailable to recognize data in CSV and convert to timestamps
- Queried: SQL
- Built-in timestamp functions
- Ease of Use: Easy to translate to Python

PLOTLY/BOKEH

The below is a table depicting our plotting packages and their time-to-render in seconds. Utilized 1 sample and 10 sample testing samples in the number of points plotted per second. b, v, a indicate the rendering method used.

FUTURE GOALS

Our goal is to use the selected database to analyze data effectively:
- We must choose one of the following databases: DolphinDB, TimescaleDB, QuestDB, Apache ruiid, InfluxDB
- The database we select must be capable of handling large-scale data efficiently and support real-time or near-real-time data processing and analytics
- We must be able to query metadata efficiently
- Flexible to allow for machine learning and predictive modeling
- Must be easy to use so we can analyze metadata provided by Stratolaunch