FREIGHT MOBILITY RESEARCH INSTITUTE
College of Engineering & Computer Science
Florida Atlantic University

Project ID: Y2R1-18

INTERACTIVE WEB-BASED PLATFORM FOR ANALYZING FREIGHT DATA-PHASE I

Final Report

by
Principal Investigator
Evangelos Kaisar, PhD.
Professor
Department of Civil, Environmental and Geomatics Engineering
Florida Atlantic University

Co-Principal Investigator
Praveen Edara, Ph.D., P.E.
Professor
Department of Civil and Environmental Engineering
University of Missouri-Columbia

Co-Principal Investigator
Yaw Adu-Gyamfi, Ph.D.
Assistant Professor
Department of Civil and Environmental Engineering
University of Missouri-Columbia

Co - Principal Investigator
Carlos Sun, Ph.D., P.E., J.D
Professor
Department of Civil and Environmental Engineering
University of Missouri-Columbia

for
Freight Mobility Research Institute (FMRI)
777 Glades Rd.
Florida Atlantic University
Boca Raton, FL 33431
October, 2020

ACKNOWLEDGEMENTS

This project was funded by the Freight Mobility Research Institute (FMRI), one of the twenty TIER University Transportation Centers that were selected in this nationwide competition, by the Office of the Assistant Secretary for Research and Technology (OST-R), U.S. Department of Transportation (US DOT).

DISCLAIMER

Table of Contents

List of Figures

List of Tables

Acknowledgments

The researchers acknowledge the support provided by our research partners who provided guidance and ideas.

Copyright

Disclaimer

The opinions, findings, and conclusions expressed in this document are those of the investigators. They are not necessarily those of the U.S. Department of Transportation, or Federal Highway Administration. This information does not constitute a standard or specification.

Executive Summary

An exponential growth in relevant data streams has brought new opportunities and challenges in the realm of data warehousing. Increased data enables improved planning, monitoring, prediction, and management of transportation systems, but only if the manipulation of such gigantic datasets could be efficiently automated. With the increasing demand for modern data warehousing, there has been a significant growth in commercial and open-source tools.

The current research seeks to develop a user-friendly, interactive, web-based prototype platform that takes advantage of recent advances in spatial data analysis, big data and user-centered visualization to integrate freight data across different private and public databases for the purpose of improving freight planning activities and data driven decision making. The methodology includes a spatio-temporal conflation framework that enables seamless integration of three key freight data sources including: weigh-in-motion (WIM), freight facility, and traffic flow data. A massively parallel database is subsequently designed to store the integrated data on a cluster of servers enabled with Graphical Processing Units (GPUs). We leverage the immense computational power of the GPUs to carry out analytics and visual rendering on-the-fly via a Structured Query Language (SQL) which interacts with the underlying database. A web interface is designed for near-instant rendering of queries on simple charts and maps to enable decision makers to drill down insights quickly. Example of an application in the APPCENTER is shown in Figure ES 1.

ES. 1: Web-based Interactive Visualization and Analytics

The framework is capable of powering big freight performance data applications with over 100 million rows. Performance benchmarking experiments conducted showed that the framework developed is able to provide real time visual updates for big datasets in less than 100 milliseconds.

Chapter 1: Introduction

1.1 Background

The efficient movement of goods and timely provision of services is critical to the economic and sustainable development of a region. Public decision makers require a comprehensive picture of freight movements to understand how freight transportation supports economic development, how land use affects freight transportation, and how transportation infrastructure supply impacts private sector freight and commercial activity. Freight data is available from many public and private sources. However, the data may vary significantly in terms of collection method, data quality, existence of gaps, availability or timeframe (daily, monthly, and quarterly), format (shape files, documents, tables, etc.) and suitability. The lack of coordination among freight data vendors not only prevents the seamless integration of data sources but also hinders data-driven decision making.

Although agencies such as State Departments of Transportation (DOTs) have transportation data management systems for storing and processing data streams, they are not uniquely designed to handle such large, heterogeneous, and multi-resolution data streams. They have limited analytical capabilities that will enable them to integrate, mine, visualize and predict on large, multivariate datasets at reasonable speeds (Mostafa et al. 2018, Richardson et al. 2014). Traditional data warehouses are stretched to the limit due to the enormous size and speed, and the significant variety of datasets across different vendors in terms of collection method, data quality, availability (daily, monthly or quarterly), and format (shapefiles, documents, table, videos, etc.). The need for frameworks that can help integrate and visualize information from existing freight databases is therefore crucial. In the current report, we leverage recent advances in big data and user-centered visualization to develop a scalable framework that allows for multidimensional visualization and analytics to be carried seamlessly on large freight datasets.

Recent advances in Big Data Analytics is enabling organizations to digest large amounts of data and transform them into actionable insights (Kluger et al, 2013, Adu-Gyamfi et al, 2016). This innovation is being fueled by massive open data platforms, driven by machine learning and empowered by low cost cloud computing. This new wave of invention could be leveraged to enable transportation agencies to identify the usefulness of their diverse datasets and to explore previously untapped applications.

1.2 Project Objectives

Under the above context, the primary goal of this project is to deliver a prototype design and deployment of an interactive, web-based platform that will assist decision makers to seamlessly integrate and analyze its freight datasets. The prototype platform is designed to be significantly faster and cheaper (by using open-sourced software solutions for development) compared to conventional data warehouses, which are heavily reliant on relational databases housed in big, costly enterprise machines. The four key subobjectives that arise from the primary goal are:

1. Develop a spatio-temporal conflation framework that enables seamless integration of three key freight data sources including: weigh-in-motion (WIM), freight facility, and traffic flow data.
2. Leverage state-of-the-art big data frameworks to develop a massively parallel database to store the integrated data on a cluster of servers enabled with Graphical Processing Units (GPUs). We leverage the immense computational power of the GPUs to carry out analytics and visual rendering, on-the-fly, via a Structured Query Language (SQL) which interacts with the underlying database.
3. Offer a low-cost, but effective data integration and analytics platform by leveraging open-source software for designing, developing and deploying the platform.
4. Provide user-centered, web-based data visualization to allow for easy interaction with the platform. Additionally, provide near-instant rendering of queries on simple charts and maps to enable decision makers to drill down insights quickly.

4.1 Report Organization

The remainder of this report is organized as follows: Chapter 2 covers the design of the key components of developed platform. It covers the frameworks used: the software and databases deployed at a high level. In Chapter 3, we describe the different sources of freight data used for to test and evaluate the performance of the current system. Chapter 4 highlights the Applications Center (APPCENTER), which is the heart of the TITAN platform. A variety of apps for visual analytics of freight performance data is covered in this section. A pipeline on how freight agencies could integrate the key outcomes, recommendations and limitations of the project is presented in Chapter 5.

Chapter 2: Development of Freight Data Integration and Analytics Platform

Figure 2.1 illustrates the design framework that enables our software platform to handle fast, disparate, noisy data streams in a seamless manner. We followed a similar architecture used to design TITAN (Sun et al. 2019). It comprises of two key modules: first, a user-centered, interactive, web-based front end where users can interface and interact with the platform and second, a big data enabled back end which stores data and provides a computational framework for retrieving, processing and visualizing large datasets. The design architecture seeks to address key technological gaps—in data handling, archiving and analysis for decision support. The following section provides a detailed overview of the components of the web-based data integration and analytics platform.

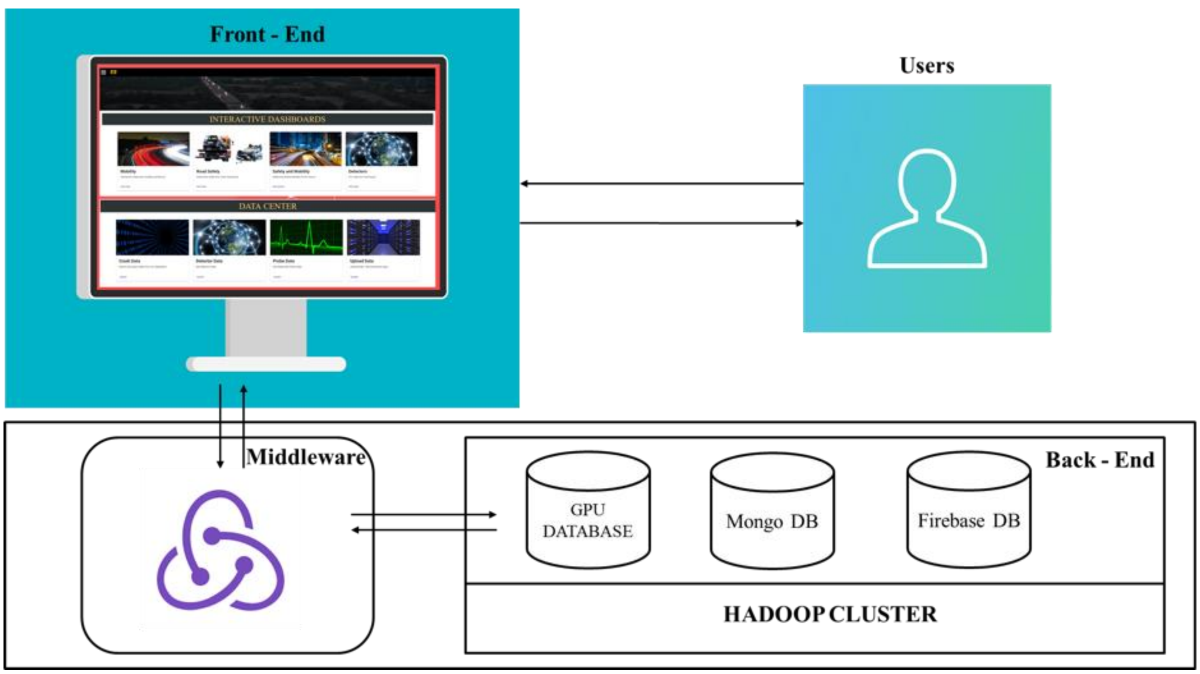

Figure 2. 5: Schematic of Platform's Key Components

2.1 Front-End Module Design

In order to be user-friendly, the front end of platform helps to mask software specifications and requirements by using a variety of layouts and user interface (UI) elements. This enables users to interact with applications directly on any computer, with any browser, and from any location with internet access. The main challenge associated with front end module designs is that the tools and frameworks used to create them change constantly. As a result, a key consideration for selecting a front-end development framework includes both the number of developers contributing to the development of libraries and its popularity among developers. Table 2.1 shows a list of the most popular front-end development frameworks, the number of contributors and its popularity among developers using the GitHub popularity metric. GitHub is a web-based computer code hosting service that is especially popular with open-source software projects.

Table 2. 2: Choosing a Front-End Library for Platform's Development

Front End Library	Number of Contributors	GitHub Popularity (number of stars)
React	1285	124,747
Aurelia	97	10,873
Angular	1596	59,434
Ember	753	20,772
Vue	268	131,290

In this study, the platform was developed with React (Todd, 2016), a JavaScript library developed by Facebook for building interactive user interfaces. React is a highly popular front-end development framework with about 125,000 stars on GitHub. With 1285 active contributors and increasing, the library is able to catch up with the constantly-changing requirements for front end development. Despite Vue's popularity, it was not used in this study as it is relatively new and has fewer developers contributing to its libraries.

There are 2 main components that users can access via the front-end user interface: the Applications Center (APPCENTER) and the Data Center. The following section describes these components, starting with the need to register an account.

2.1.1 Registration and Login

To take advantage of the full capabilities of the platform, all first time users of the platform must register with an email and password via a log-in or registration screen. The user will only be required to go through the log-in process once if the same computer is used to access the database. The credentials of each user are stored in a database to manage user activities. Figure 2.2 shows the login or register screen.

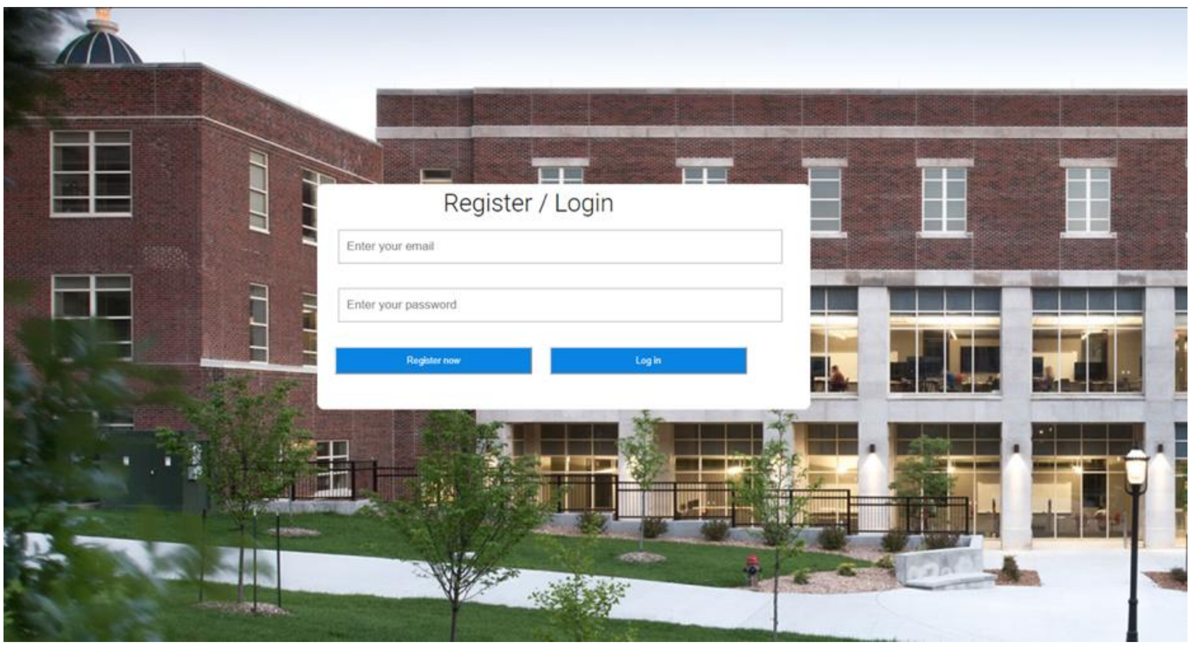

Figure 2. 6: User Registration and Login

The authentication framework, shown in figure 2.3, was developed to manage the types of modules that a user could access based on their credentials. Currently, the APPCENTER is only accessible to registered users. Other modules such as data querying and visualization are accessible to the general public.

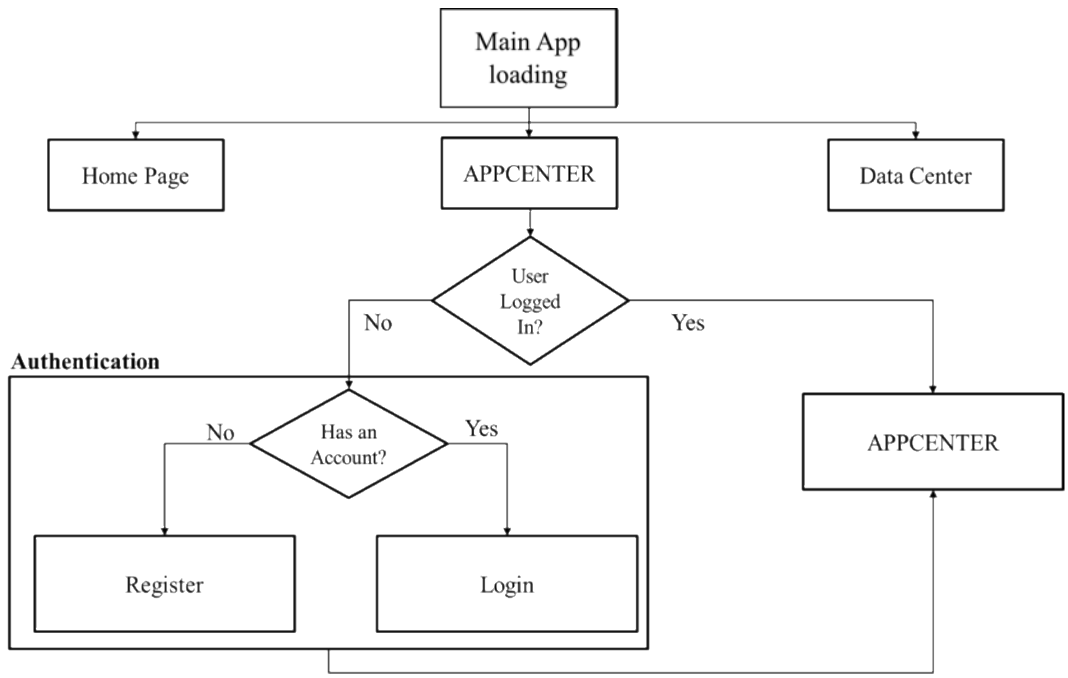

Figure 2. 7: Authentication Framework.

2.2 Back-End Module Design

The primary goal of the Back-End is to provide computational resources that could be used to quickly respond to user queries from the front end. The types of analytics carried out on the front end can be computationally expensive. For example, a user may request to calculate the average gross weight of trucks carrying goods from food processing centers during PM peak hours on all major highways in the Tampa area over a period of 5 years. This task will require a back end that is able to sift through approximately 100 gigabytes of data and provide a response to the user without any significant latencies. To enable the platform to carry out such highly complex analytics from the front end, we designed a scalable, cloud-based back end based on recent advances in big data analytics. In addition to providing computational resources, the back end is also used for archiving large datasets and managing functions such as user authentication, data security, and others. The structure of the back end and how it integrates into the overall framework is shown in figure 2.4. At the core of the back end is a Hadoop distributed file system (HDFS) (Shvachko et al. 2010) which enables the networking of a series of computers into cluster. By using the HDFS, we are able to maintain the processing speed of the platform even as the size of data grows. On top of the Hadoop framework are two different databases: Firebase (Singh et al. 2018), and a GPU database. The roles of each of these databases are defined next.

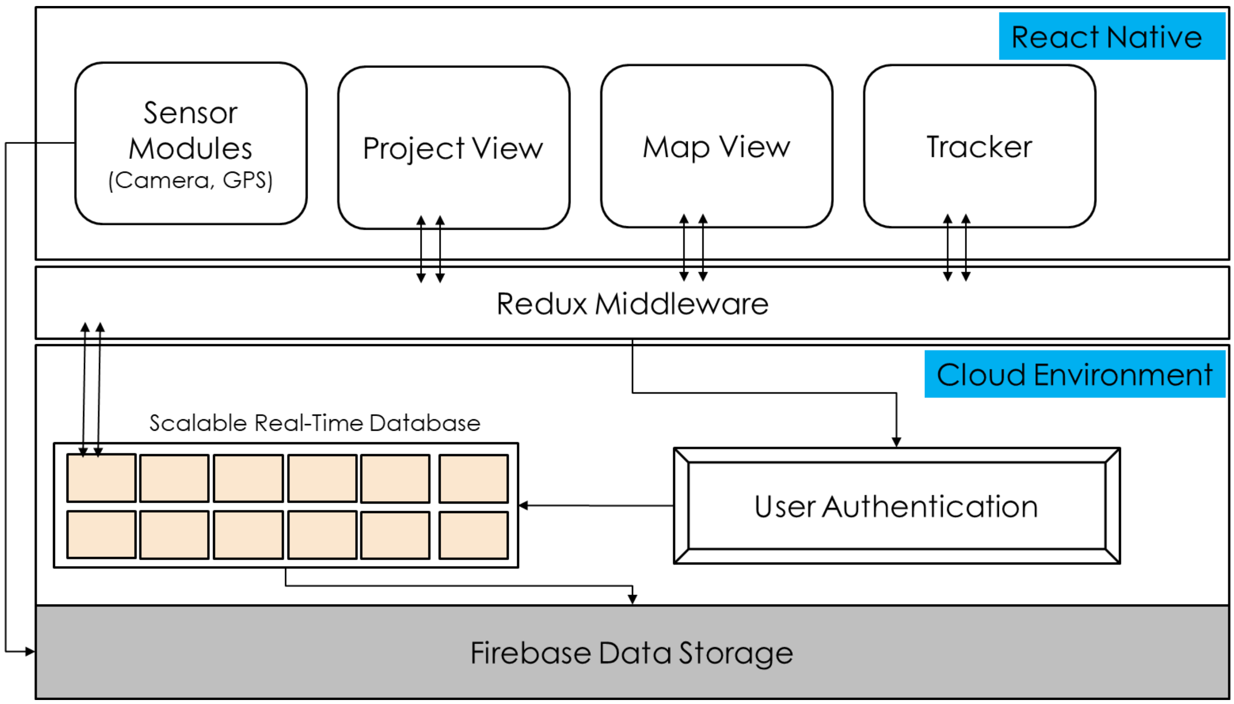

Figure 2. 8: Back end Integration into TITAN's Framework.

2.2.1 Firebase DB

Firebase is a NOSQL database for storing datasets that are not structured. NOSQL means nonrelational in contrast to traditional SQL databases. NOSQL also differs in how it is scaled by increasing the database server pool as opposed to increasing the horsepower of the hardware. Firebase serves two functions in our framework: user authentication and temporary data storage. Before a user can use the app, they must be authenticated. Firebase authenticates users via email/password, phone number, or Facebook, Google, Twitter, and Github accounts. To simplify our developments, users are authenticated only by email and password. Firebase is also used to temporarily store all user uploaded datasets. Since Firebase is a NOSQL database, it is able to consume all types of data formats: Shapefiles, CSVs, XML, Video, Images, etc. In contrast, a

SQL database requires a predefined format or schema. It is however, not designed for storing very large datasets. Firebase therefore redirects significantly large files (30GB or more) to the HDFS and MongoDB depending on the use of the data.

2.2.3 Graphical Processing Unit (GPU) DB

All front-end visualizations are carried out by using an open-source Graphical Processing Unit (GPU) database. GPUs have tremendous processing capabilities compared to CPUs. For example, a single NVIDIA Geoforce 1080ti GPU card has over 4000 processing cores, meaning 1 GPU is capable of processing information at a rate comparable to using a cluster of 500 8-core computers. The GPU database is the reason why the current platform is so fast on the front end. A Structured Query Language (SQL) database is used on top of this database to process data in the memory of the Graphical Processing Unit (GPU).

Chapter 3: Data Description and Conflation Framework

The freight-related datasets, for Florida, that were used to develop the analytics platform included the following: Weigh – in – Motion data, Automatic traffic count data, Freight Facility Center data, and static metadata on the location of road and freight facilities.

3.1 Truck TMS Data

The truck TMS data maintained by Florida DOT contains speed, classification and weight information for each truck. It combines data from both Weigh – In – Motion (WIM) stations, and Automatic Traffic Counters (ATR). Other attributes of this database included the vehicle type, vehicle length, gross weight, the weight of each of the axles of the truck and the spacing between them. We collected TMS data from 2011 through 2018 for this project. The total size of the data was approximately 300GB after applying some data reduction techniques to the original dataset. The data also includes static GIS information on the location and attributes of each weigh – in – motion station. Figure 3.1 shows the locations of WIM stations on major roads in Florida.

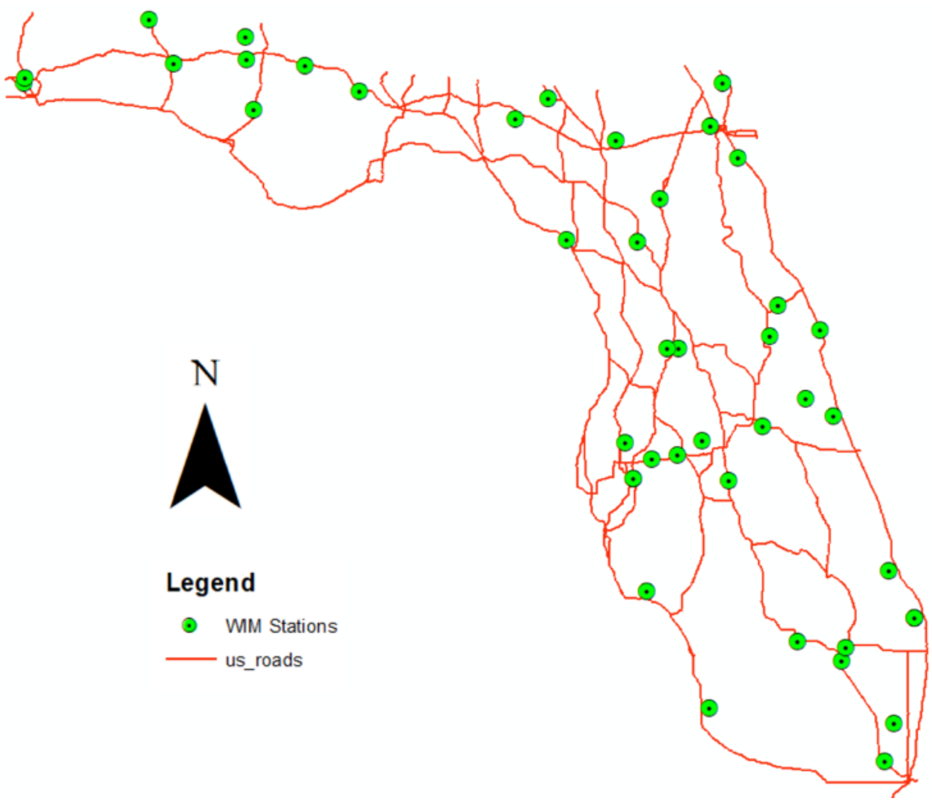

Figure 3.1: Weigh – In – Motion Locations on major roads in Florida.

3.2 Freight Facility Centers data

Florida DOT's data on freight facility centers (FFCs) were used in this project. FDOT initially obtained this data from the Florida Department of Revenue (DOR). Key data fields used to identify and describe the freight facilities included parcel ID, land use code, year built, total living or usable area, number of buildings, owner's name, short legal description, and physical address. The DOR dataset has been expanded using online mapping resources (such as Bing Maps, Bing StreetSide, Google Earth, Google Maps, Google StreetView, etc.) to visually scan parcels and identify large freight facility locations throughout Florida (FDOT, 2016). Information for the data fields added to each record during the scan included the name of the business in operation, data source of the information, the date indicated for the source information, railroad access availability, vacancy, estimated number of truck bays, industrial park or development, and multitenant business names. Figure 3.2 shows the locations of freight facilities in Florida. The complete set of data contains about 91,895 records, with over 114 attribute columns.

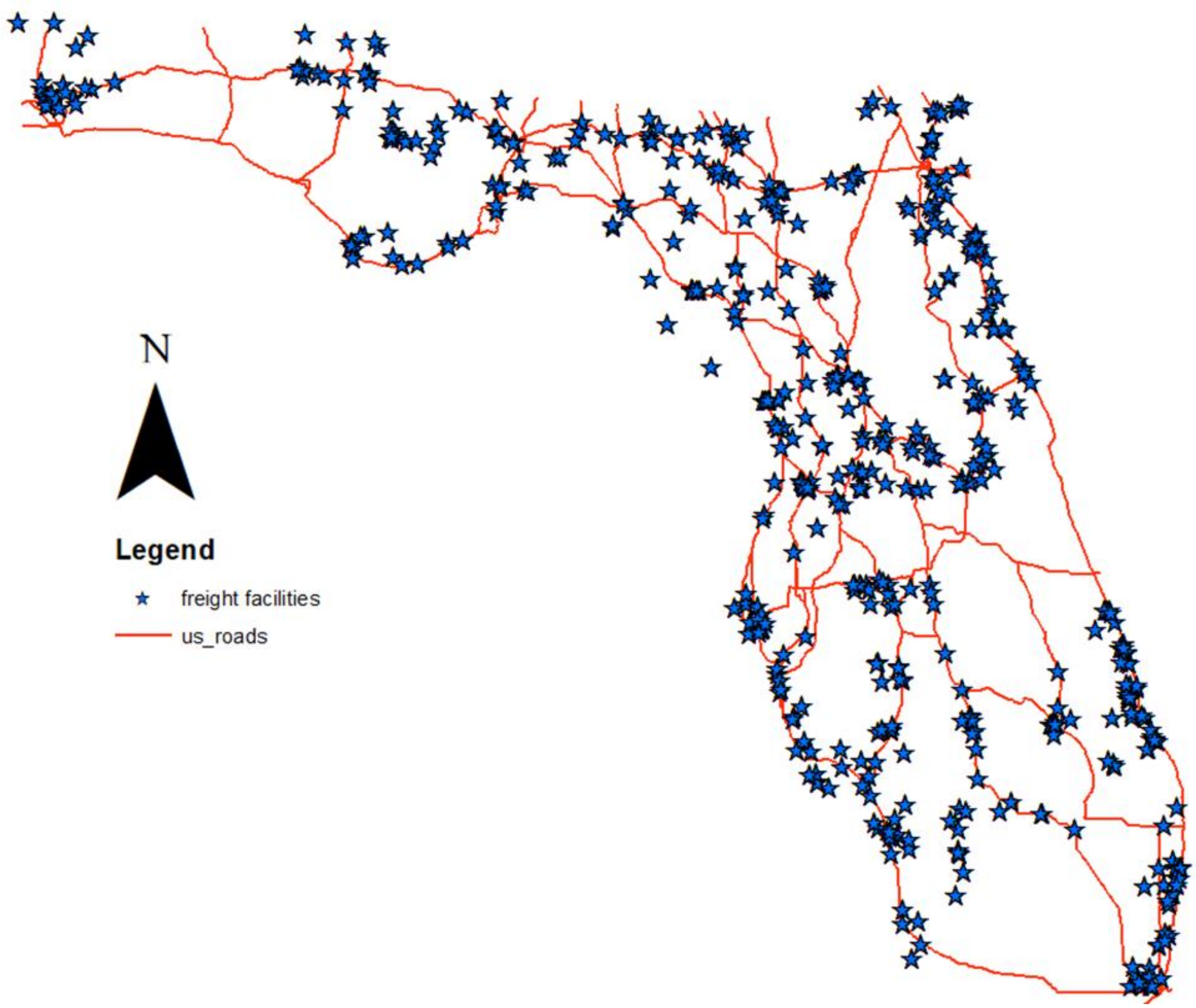

Figure 3.2: Freight Facility Centers across Florida.

3.3 Automatic Conflation of Freight Datasets

The primary goal of data conflation is to integrate the different freight databases into a single unified one for analytics. It involves the detection and mapping of spatial and temporal variations between different freight datasets. In this section, we discuss the framework used for integrating the freight facility data, weigh-in-motion data, traffic speed, count and classification data in space and time. The ideal way of conflating datasets is usually by manual mapping. This approach is more consistent and accurate. However, due to the size and scale of the freight datasets involved, an automated data conflation approach was developed in this study. Our framework for conflation involves four main steps as shown in figure 3.3. To make sure the process is fully automated, various libraries in Python language were used for coding the conflation framework.

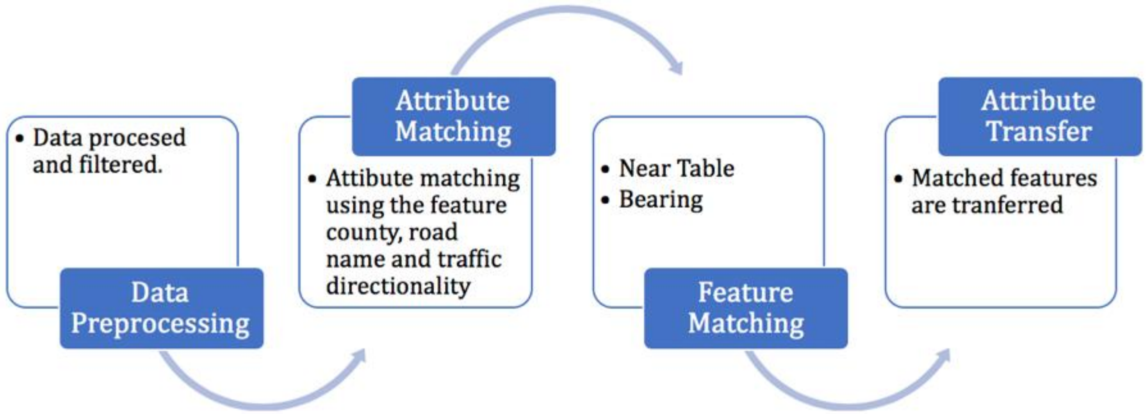

Figure 3.3: Proposed framework for data conflation

3.3.1 Data Processing and Attribute Matching

This step prepares the data for the best condition possible for further analysis. The different geographic freight data layers are collected from separate vendors. As a result, the potential for geometric and attribute data inconsistencies is high. The pre-processing stage includes validating data geometry and topology, selecting relevant attribute features (e.g. road names, segment ids, counties, etc.) for processing and using consistent map projections to ensure that the data layers are projected on the same geographic coordinate system. For temporal datasets, we aggregated the datasets by hour to reduce the size and the need to carry out conflation at very high time resolutions. Next, attributes in the different freight datasets describing the same features are matched. Three main attributes were used to define criterion for matching attributes from different layers: County, Road Name and Road Directionality. This basically means the road names as well as directionality in both layers must be matched to ensure that conflation is carried out only between similar road segments with the same directionality. The result of attribute matching is a mapping between freight facilities, count stations and road networks by county, sites and classification of roads.

3.3.2 Detect Feature Changes

Feature change detection is the second step of the conflation process. Knowing where and what the changes are between two datasets helps to assess how significant they are and whether or not to proceed with attribute transfer. To detect feature changes in two datasets, the Detect Feature Change (DFC) tool in ESRIs ArcGIS was used. This tool identifies spatial feature differences and outputs the type of change detected for each feature. Ideally, there are four (4) possible conditions that could occur: Spatial Change (topological difference), Attribute Change, No Change (1:1 match without any spatial or attribute changes) or new feature (unmatched feature). For features where a spatial change was detected, the following workflow is used to unify and consequently conflate into the base data.

3.3.3 Generate Near Features

This step uses buffering tools such as the near tool and linear reference tool in ArcGIS to associate different features base on spatial proximity. Although attribute matching is able to map for example FFCs to roads, it will be most valuable to know the closest weigh station, traffic counter or major interstate. To generate the near features, first, a 10, 20, and 30 mile buffer is created around all major interstates (see figure 3.4). All FFCs and WIM stations located within the buffer are mapped to the roads. This creates many – to – many mappings between the features. The distance between mapped features are included for use in post processing and filtering later.

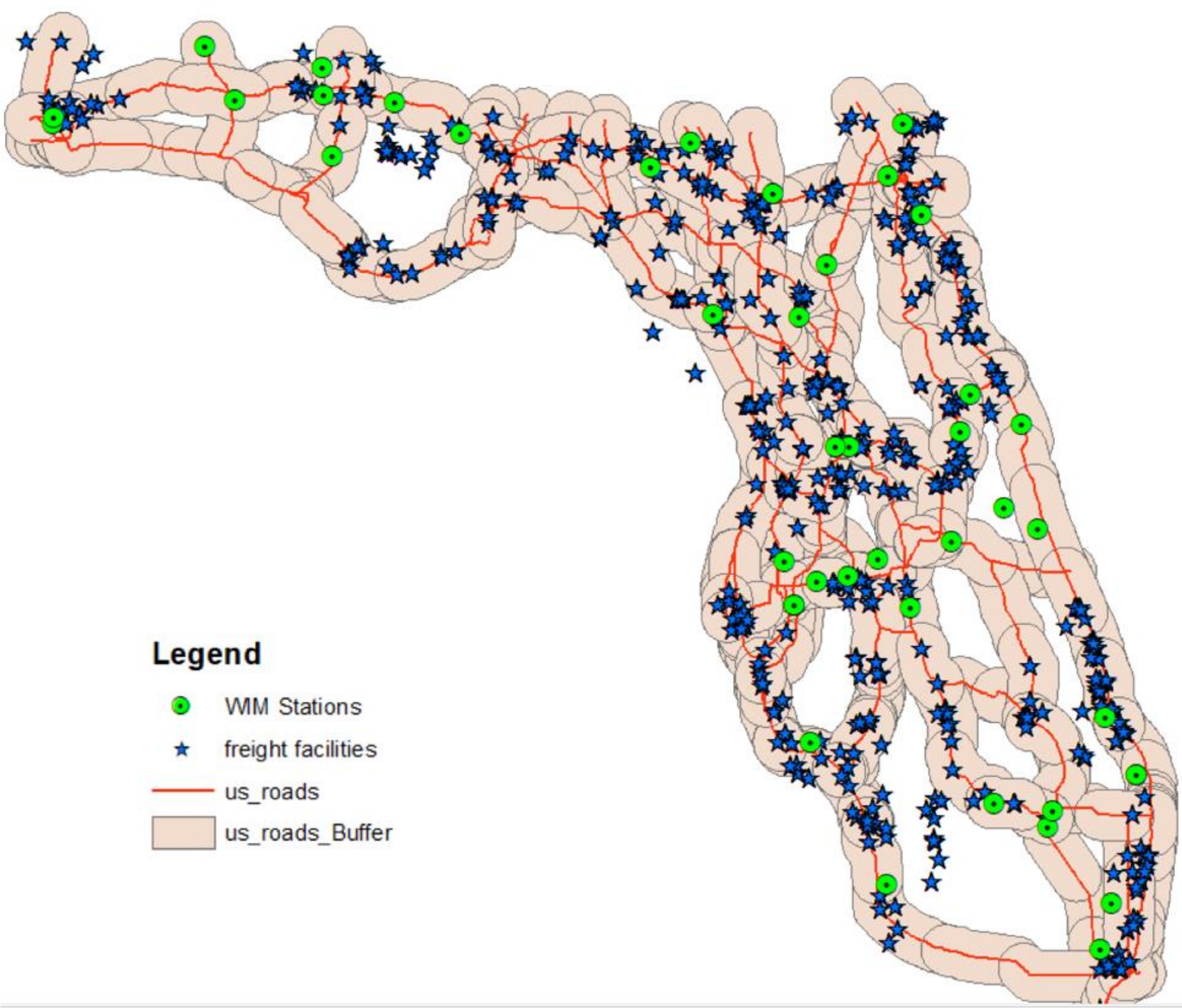

Figure 3.4: Generating near features using different buffer distances.

3.3.4 Attribute Transfer

Finally, once features between the geographic data layers have been matched, specific feature attributes from the source layer are transferred to the matching target features. Tables 3.3 and 3.4 shows the final conflation table and a description of the attributes in the data layer respectively. It integrates information from all the key freight data sources including truck weight, speed, and volume by time, freight facility centers, and roadway information. After attribute transfer, the size of the integrated data layer is approximately 180GB with 620 million rows of data.

Table 3.3: Transfer Attribute Output

Route	DistanceWSFC	FC_Long	FC_Lat	DIR	c17_year	c18_month	c19_date	c20_hour	SPD	GROSS_WT	GROSS_WT1
I 95	15.62786	-80.17	26.69	N	2017	10	4	1	67.18185	5417366	13
I 275	5.211796	-82.47	28.19	N	2016	11	3	21	66.47421	1927328	12
I 275	14.15145	-82.33	28.33	S	2016	8	3	9	68.05601	4090148	23
I 95	15.62786	-80.17	26.69	N	2017	1	14	6	67.25877	5140534	19
I 275	5.211796	-82.47	28.19	N	2016	12	19	10	68.3459	6597693	36
I 275	14.15145	-82.33	28.33	S	2016	8	3	21	70.69405	1390966	14
I 275	5.211796	-82.47	28.19	N	2016	11	17	17	68.13622	2033848	22
I 95	15.62786	-80.17	26.69	N	2016	2	1	13	66.63043	16923084	42
I 275	14.15145	-82.33	28.33	S	2017	12	2	3	67.42857	1691221	7
I 275	22.80733	-82.67	28.38	S	2016	3	11	1	68.02292	1019262	16
I 95	15.62786	-80.17	26.69	N	2016	2	19	17	64.74535	9724018	25
I 275	5.211796	-82.47	28.19	N	2016	3	29	9	66.51021	4021633	27
I 275	14.15145	-82.33	28.33	N	2017	12	31	3	68.52857	757187	7
I 275	22.80733	-82.67	28.38	N	2017	2	8	15	70.44886	3921189	25
I 95	15.62786	-80.17	26.69	N	2015	1	27	23	67.70969	6548168	9
I 275	22.80733	-82.67	28.38	N	2017	3	21	2	67.34013	1327470	7
I 275	14.15145	-82.33	28.33	S	2017	7	15	9	69.74107	1659591	12
I 275	5.211796	-82.47	28.19	N	2016	10	25	11	67.13743	4064319	22
I 95	9.331313	-80.09	26.79	N	2017	4	24	18	61.88128	11294346	26

RailAccess	EstBays	FC_CO_NO	FC_Landuse	LND_SQFOOT	TOT_LVG_AR	dis_WS_R	CountySite	WS_Long	WS_Lat	R_Category
\N	0	60	Heavy Manufacturing	1757470	6112	1E-06	939952	-80.14391	26.915	Interstaes
\N	0	61	Lumber Yards	346737	22374	5E-06	109955	-82.41382	28.13385	Interstaes
\N	0	61	Open Storage	157687	924	5E-06	109955	-82.41382	28.13385	Interstaes
\N	0	60	Heavy Manufacturing	1757470	6112	1E-06	939952	-80.14391	26.915	Interstaes
\N	0	61	Lumber Yards	346737	22374	5E-06	109955	-82.41382	28.13385	Interstaes
\N	0	61	Open Storage	157687	924	5E-06	109955	-82.41382	28.13385	Interstaes
\N	0	61	Lumber Yards	346737	22374	5E-06	109955	-82.41382	28.13385	Interstaes
\N	0	60	Heavy Manufacturing	1757470	6112	1E-06	939952	-80.14391	26.915	Interstaes
\N	0	61	Open Storage	157687	924	5E-06	109955	-82.41382	28.13385	Interstaes
\N	0	61	Materials Processing	43560	300	5E-06	109955	-82.41382	28.13385	Interstaes
\N	0	60	Heavy Manufacturing	1757470	6112	1E-06	939952	-80.14391	26.915	Interstaes
\N	0	61	Lumber Yards	346737	22374	5E-06	109955	-82.41382	28.13385	Interstaes
\N	0	61	Open Storage	157687	924	5E-06	109955	-82.41382	28.13385	Interstaes
\N	0	61	Materials Processing	43560	300	5E-06	109955	-82.41382	28.13385	Interstaes
\N	0	60	Heavy Manufacturing	1757470	6112	1E-06	939952	-80.14391	26.915	Interstaes
\N	0	61	Materials Processing	43560	300	5E-06	109955	-82.41382	28.13385	Interstaes
\N	0	61	Open Storage	157687	924	5E-06	109955	-82.41382	28.13385	Interstaes
\N	0	61	Lumber Yards	346737	22374	5E-06	109955	-82.41382	28.13385	Interstaes
2	20	60	Bottlers	873414	255594	1E-06	939952	-80.14391	26.915	Interstaes

Table 3.4: Definition of attributes in the conflated dataset

Attribute	Definition	Attribute	Definition	Attribute	Definition
RailAccess	Accessible by rail	R_Category	Road category	SPD	Vehicle Speed
EstBays	Estimated number of bays	Route	Route name	GROSS_WT	Truck gross weight
FC_CO_NO	County number	DistanceWSFC	Distance from weigh station to facility		
FC_Landuse	Land use code	FC_Long	Facility longitude		
LND_SQFOOT	Square footage	FC_Lat	facility latitude		
TOT_LVG_AR	Total usable area	DIR	Road direction		
dis_WS_R	Distance from weigh station to road	c17_year	Year		
CountySite	Joined county and site number	c18_month	month		
WS_Long	Longitude of weigh station	c19_date	Date		
WS_Lat	Latitude of weight station	c20_hour	Hour		

Chapter 4: Application Center – APPCENTER

The APPCENTER is the heart of the platform. It provides a non-programmatic GUI access to underlying algorithms of the platform while guiding users to derive powerful analytical insights from the integrated dataset. The design framework of APPCENTER, shown in figure 4.1, follows a big data architecture which synergistically utilizes the power of distributed computing on the server side and GPU strengths of data rendering on the client end. The simultaneous use of GPU data frames and SQL enables fast, interactive queries on the front end. Cluster resources are used for filtering, aggregating and integrating large datasets. Layout designs such as grid and list views are used to improve the user-friendliness of each application within the APPCENTER. In its current state, the APPCENTER can be used for two main activities: Performance Measurement and Predictive Analytics. In the following section, the APPCENTER will be used to deploy an interactive web-based application for analyzing the freight datasets integrated in the preceding sections.

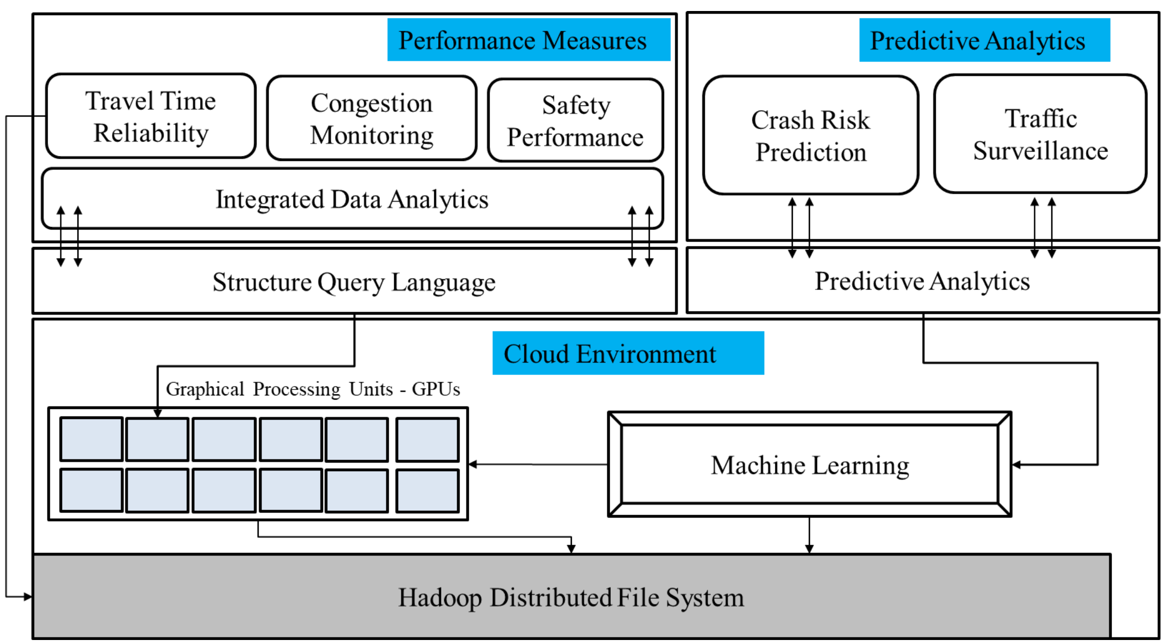

Figure 4. 2: APPCENTER Design Framework

4.1 Visual Analytics of Freight Performance Measures

Freight transportation performance measures drive strategic investments and policy decisions that are aimed at providing a safe and reliable transportation system for communities. One of the key limitations of traditional web-based methods used for communicating performance measures is their inability to provide dynamic and interactive trends at both aggregated and disaggregate levels. As a result, local, spatio-temporal trends are usually not captured. APPCENTER communicates a variety of performance measures via dynamic, interactive dashboards that enables users to explore both localized and generalized trends in large datasets. Users can configure routes and perform customized analysis such as travel time index, computing congestion hours, travel time reliability or speed performance. APPCENTER's performance management dashboards also hosts functions that enable users to select, drop, pan, zoom and filter their respective areas of interest. A variety of custom-build visualization charts were built using reactjs and d3js. Examples of charts that were created are maps, histograms, bars, lines and pie charts. Figures 4.2 through 4.5 shows examples of charts that were generated from the integrated freight dataset. In figure 4.2, a row chart shows the top 15 county sites with the highest gross weight.

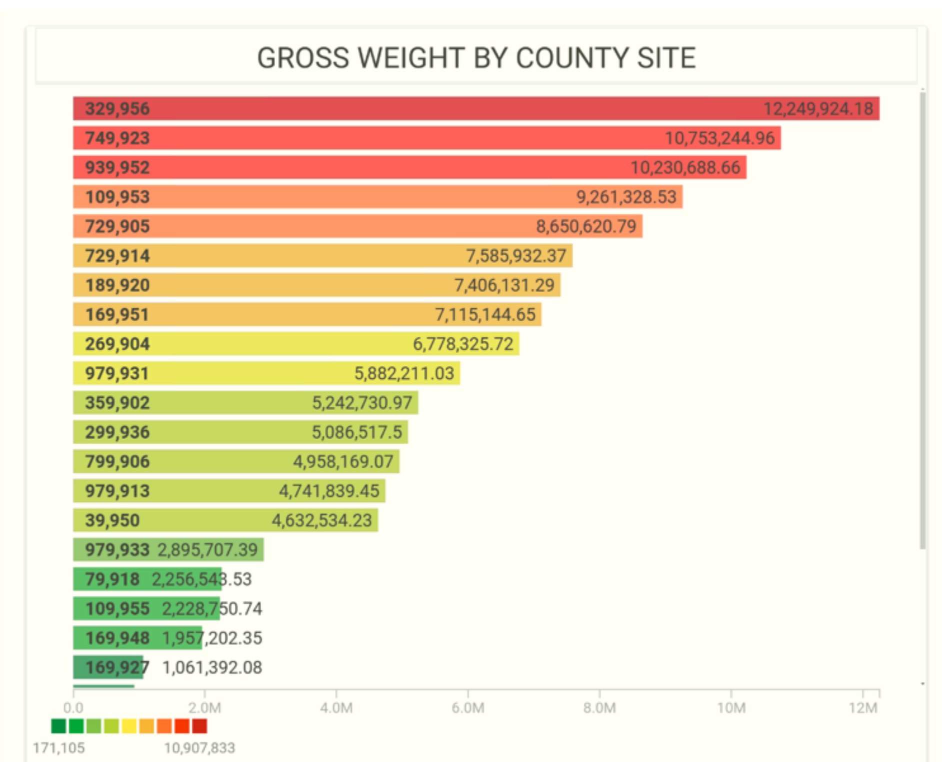

Figure 4.2: Average gross weight at each of the top 15 county sites

Figure 4.3 shows all the freight facility centers in the State, colored by their proximity to a highway. The color bar at the bottom left shows the range of distances vary between 0 and 30 miles. The size of each of the bubbles shows the total square foot area occupied by the facility. The chart also has additional features for filtering and zooming into regions of interest. Figure 4.4 uses a scatter plot to describe the different types of freight facilities, their proximity to major highways and their average annual gross weight. The trends shown indicates that bottling facilities which generate the largest proportion of the gross weight are typically located nearest to the major highways. Warehouses and light manufacturing facilities are typically about 20 miles away from major roads. Figure 4.5 shows line plots for top 6 high volume WIM stations. The volumes are averaged by hour of day over the 8-year period of analysis. It is important to note that all these charts have filters that enables the user to interact with the chart.

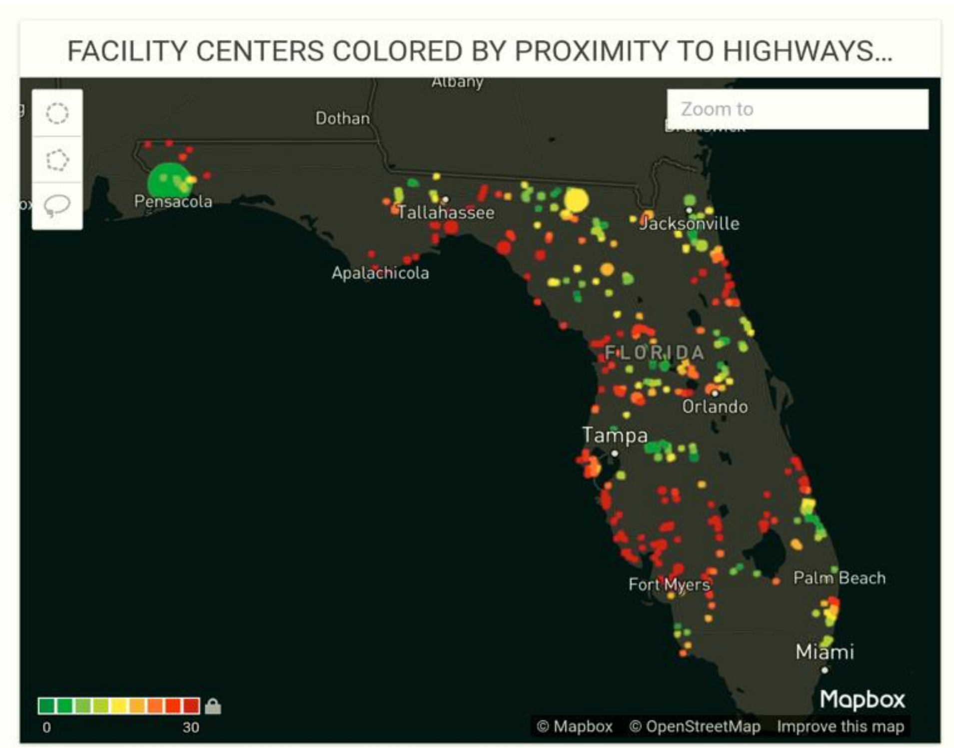

Figure 4.3: Freight facility centers colored by the proximity to highways and sized by their total square footage.

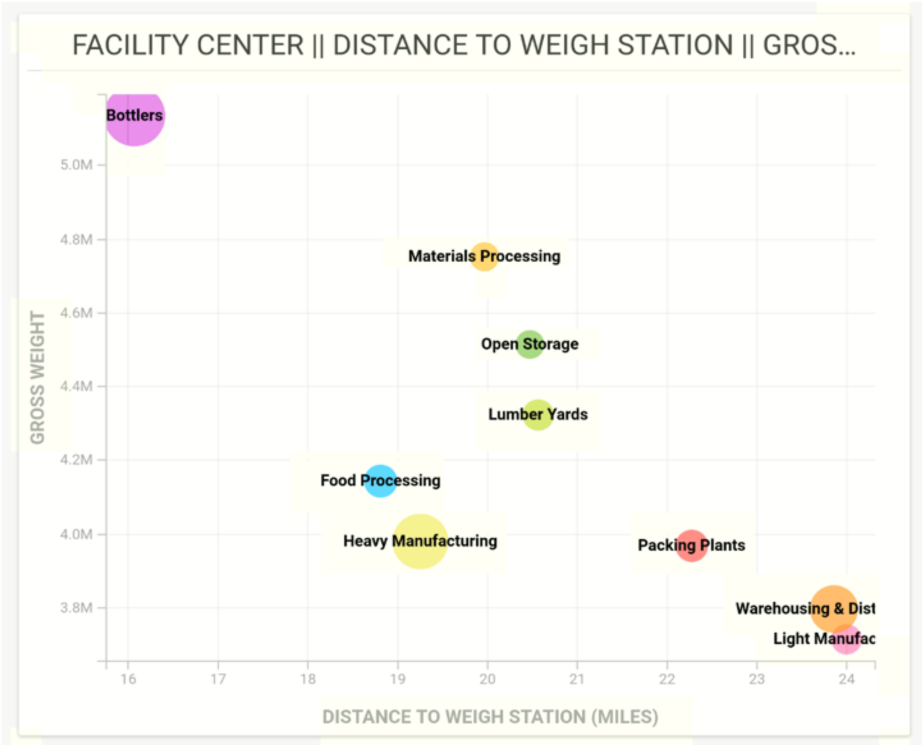

Figure 4.4: Proximity Analysis of different facility types to weigh stations.

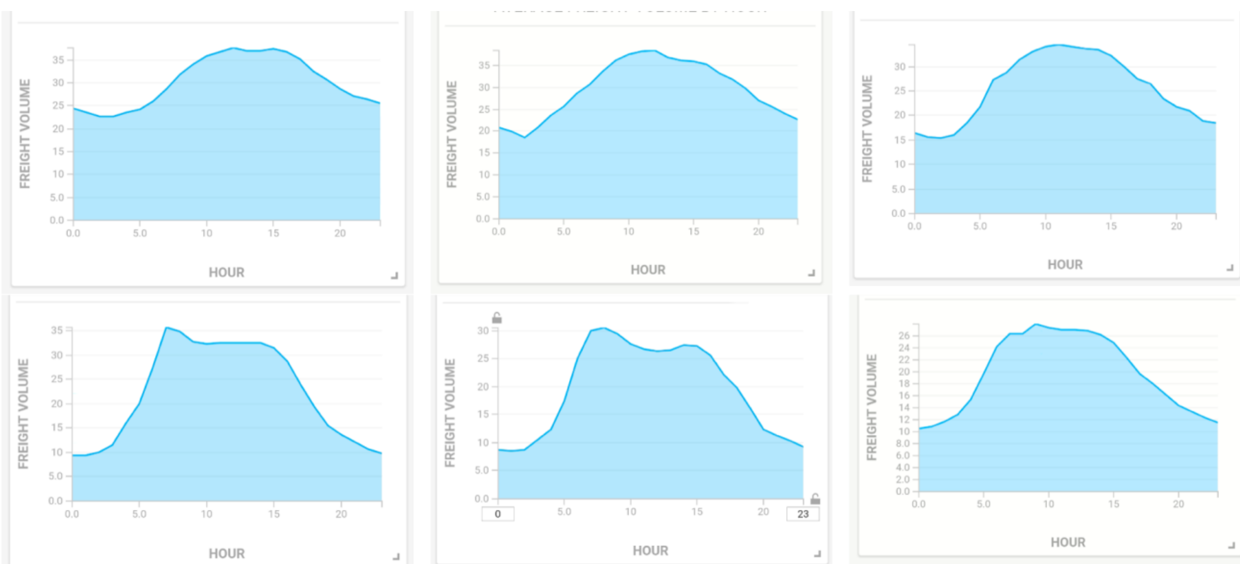

Figure 4.5: Line charts for top 6 high volume WIM stations - volume averaged by hour over 8-year period.

In figure 4.6, we show a dashboard that enables the user to interact with all the different charts on a single page. This application relies on the integrated freight data with approximately 100 million rows of data. The size of the freight data generated is approximately 80GB. APPCENTER is able to visualize and perform queries on this data without latencies in the web browser. The chart on row 1 column 2, shows the locations of weigh-in-stations, colored by the average speed of traffic and sized by the average gross weight. Other charts have also been included to explore different trends in the freight dataset. Each of the accompanying charts can be used to filter and select various areas of interest. Figure 4.7 illustrates customized functionalities that can be used to perform local analysis. The dashboard is updated to show the performance measures for the selected regions on the map. The analysis can also be configured for specific routes, counties, freeways, arterials, and other categories.

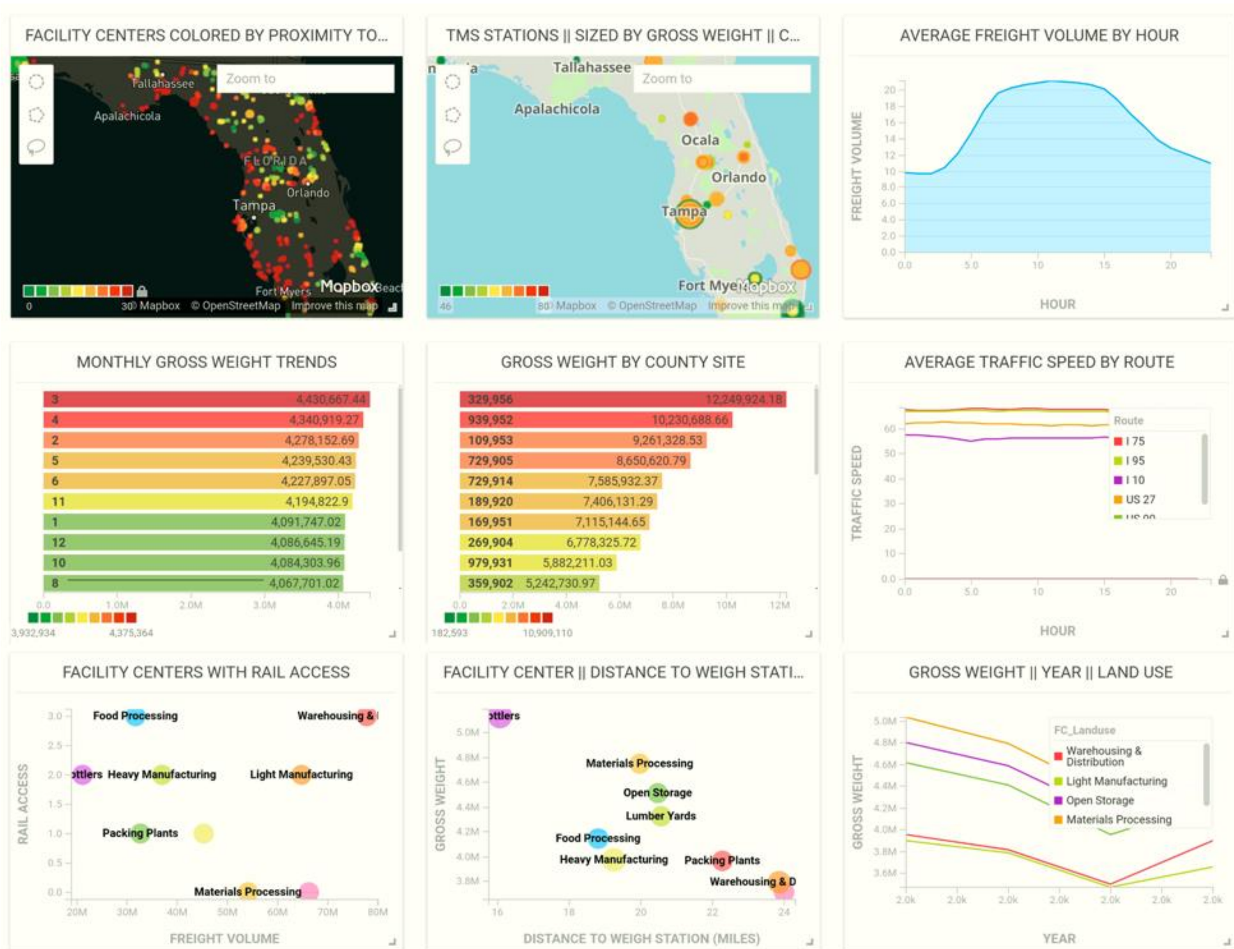

Figure 4. 6: An interactive dashboard for exploring statewide mobility trends.

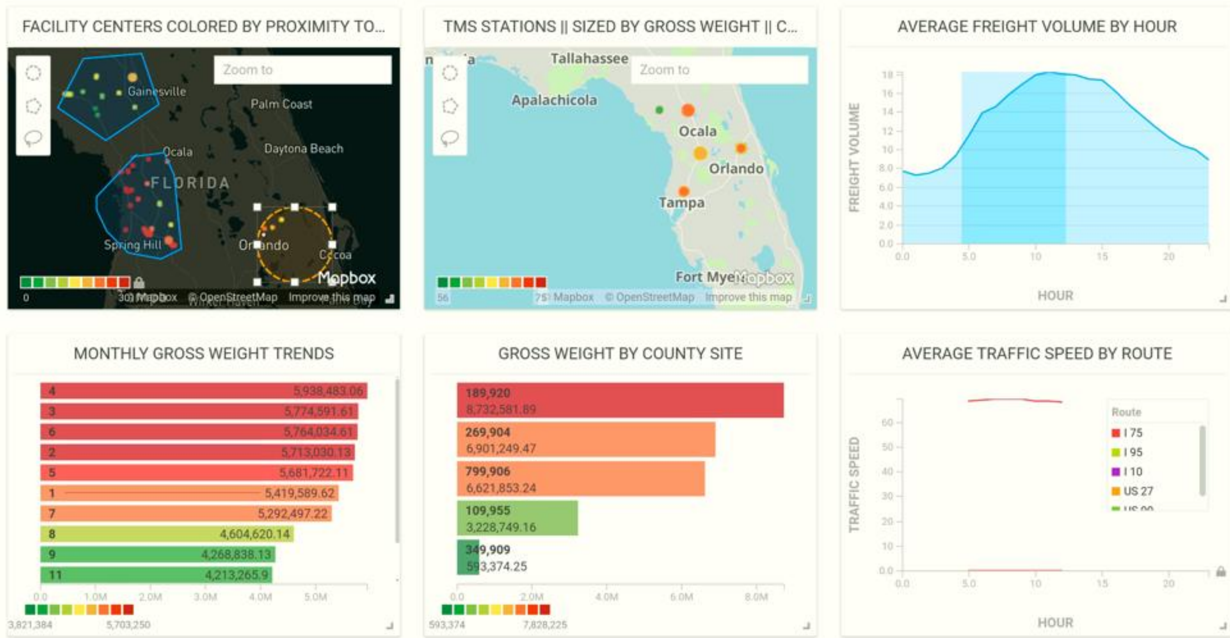

Figure 4. 7: Customized interactivity with dashboards.

4.2 Comparative Analysis of the Developed Freight Data Platform versus ORACLE-Based Systems

The developed freight data platform brings together multiple databases to enable seamless integration of a variety of freight datasets. In this section, we compare the current platform with the most common traditional relational database systems used for managing transportation data, ORACLE. We compare both platforms based on expected costs and capabilities as shown in Table 4.1.

The current platform was developed with open-source software tools. Hence, the software costs are relatively low compared to any enterprise platform. The downside of relying heavily on open-source software is that considerable effort will have to be expended in the development of the platform. Developers will have to spend more time on integrating all the bits and pieces of code together. This increases the development cost of the platform compared to Oracle-based applications. Cloud costs for our platform is expected to be slightly higher than Oracle because of the use of GPUs and computer clusters. The platform will require at least 2 GPUs (costing about $0.5 an hour) and 5 servers in a cluster (costing about $0.35 an hour). Oracle does not need GPUs and clusters, hence will be cheaper in terms of cloud costs. Administration costs will mostly be dependent on where the platform is deployed. On the cloud, server administration is usually done by the cloud service provider. Hence, both platforms will save money. If the server is built in-house, a relational database service such as Oracle will cost significantly higher. Most of the processes and application deployed are automated. All apps are updated instantaneously as the underlying data changes. Hence, the role of an administrator for the current platform is significantly reduced. For Oracle, as data changes, the administrator must rewrite queries, keep track of tables, and redo models. This increases the responsibilities of the administrator, a relatively high cost.

With regards to capabilities, our platform offers so much more. Although interactive visualizations can be carried out on Oracle, the size of data that can be visualized is limited (not more than 2GB). The freight data platform developed in this study is able to visualize up to 80GB of data interactively without any significant latencies. It also has modules for automatically integrating data from multiple sources. Data integration on platforms such as Oracle is heavily manual driven and dependent on the size of the datasets involved. Due to the lack of GPUs and clusters, the time needed for responding to queries and visualizations are orders of magnitudes higher for Oracle based applications. The developed platform also offers geospatial capabilities to help users deal with geographical datasets.

Table 4. 2: Comparing Platform with Traditional Data Warehouses

	Freight Platform	Oracle
Costs		
Software Costs	↓ (low)	↑ (high)
Development Costs	↑ (high)	↓ (medium-low)
Cloud Deployment Costs	↑ (high)	↑ (high)
In-house Deployment Costs	↓ (medium - low)	↓ (medium-low)
Administration Costs	→ (variable)	↑ (high)
Capabilities		
Interactive Visualizations	√	√
Data Integration	√	→
Platform Speed	↑	↓
Geospatial	√	×
Flexibility	↑	↓
Predictive Analytics	√	×

Summary and Conclusions

The current project successfully designed and deployed a fully-functional, interactive web application for storing, retrieving, integrating and visualizing a variety of large freight datasets. By leveraging recent advances in big data, the platform was developed with the future in mind. As data grows exponentially, the platform scales along with it, making it an extremely fast tool for data analytics and visualization. Relying heavily on open-source tools for development resulted in dynamic and easily customizeable platform compared to enterprise software solutions (e.g. Oracle) currently used by most transportation agencies. Additionally, software costs are significantly cheaper, compared to conventional platforms such as Oracle. The costs for cloud deployments are however comparable, while development costs are slightly higher than off-the-shelf platforms.

A modular design approach was used to develop the web-based platform. It consists of a front-end and a back-end module. A user makes requests and updates by connecting to the front-end user interface. All actions from the user are subsequently passed on to the back end which sends an appropriate response back to the user on the front end. The platform's design framework seeks to minimize the latency in communication between the front and back ends. To achieve this, data visualization on the front end was carried out by using GPUs (Graphic Processing Units), on the back end, a distributed cluster powered by Hadoop and Mongo DB.

The developed platform for freight data analytics has two main components: a data center and an applications center. The Data Center stores different streams of datasets and provides a user-friendly, non-programmatic interface for querying the different databases. By leveraging cluster computing and recent advances in big data analytics, we are able to generate responses to different forms of user queries at a much faster rate compared to traditional data warehouses. The second component is APPCENTER (Applications Center) which hosts a variety of applications for performance monitoring, data integration and predictive analytics. The APPCENTER is powered with fast, interactive visualizations that enables users to identify trends and discover insights quickly for decision making. The APPCENTER was developed on top a Graphical

Processing Unit (GPU) database which enables it to perform computations on large datasets within fractions of a second.

Practically, how could transportation agencies take advantage of this platform? First, the enormous amounts of data being collected by agencies has great potential for improving data-driven decision making. However, such data needs to be made accessible to various staff who do not have the time and resources to dig into these massive databases. The current platform provides the tools to shrink down the effort required for data integration and analysis. Second, it also provides graphical tools that simplifies data querying and analysis. Querying involves simple steps such as drawing a circle around a region of interest, which then automatically generates relevant perofrmance measures for the selected region. Finally, the platform takes advantage of modern computer cluster technology to reduce and eliminate latencies in software response. The platform's speed is a factor in making the software user-friendly so that its use can be incorporated into the regular workflow of agency employees.

References

Mostafa Amin-Naseri, Stephen Gilbert, "Evaluating the Reliability, Coverage, and Added Value of Crowd Sourced Traffic Incident Reports from Waze," Transportation Research Record, Washington, DC.

Richardson T., Gilbert S., Holub, J., Thompson, F., MacAllister, A., Radkowski, R., Winer, E., Davies, P., Terry, S. (2014) "Fusing Self-Reported and Sensor Data from Mixed-Reality Training", The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC).

Kluger, R., and Smith, B. 2013. *Next Generation Traffic Management Centers*. Final Report UVA-2012-02.

Adu-Gyamfi, Y., Michael, J., Skylar, K. 2016. "A Comprehensive Data Driven Evaluation of Wide Area Probe Data: Opportunities and Challenges." Transportation Research Board, Washington, DC.

Carlos S., Praveen E., Adu-Gyamfi, Y. (2019) "TITAN – AN INTERACTIVE WEB-BASED PLATFORM FOR TRANSPORTATION DATA INTEGRATION AND ANALYTICS", available at: https://www.modot.org/sites/default/files/documents/cmr19-006.pdf

Shvachko, K., Kuang, H., Radia, S., Chansler, R. (2010) "The Hadoop Distributed File System." *Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies*, Washington DC.

Singh, Harmeet, Tanna, M. (2018) "Serverless Web Applications with React and Firebase: Develop real-time applications for web and mobile platforms", *Packt Publising*.

Todd, A. (2016) "ReactJS: Become a professional in web app development," ISBN: 1533118825 9781533118820.

FDOT (2016) "Freight Facility Dataset Methodology Document", Transportation Statistics Office. Available at: https://fdotwww.blob.core.windows.net/sitefinity/docs/default-source/statistics/docs/freight-facility-dataset-methodology.pdf?sfvrsn=960e858b_0