MXpress Software Requirements Specifications

October 19, 2019

Team MX

Table of Contents

1.1 Purpose

The purpose of the capstone project is to create an adaptable framework that allows the routing of vehicles. Time-permitting, the scope of the capstone will also involve the creation of an emergency vehicle routing system including web app that allows a dispatch agent to be presented with the fastest possible route based on true real-time data and calculations, in combination with ML based computations from saved historical data.

1.2 Intended Audience

The intended audience of the MXpress project is varied. The implementation of emergency vehicle routing using real time data is intended for use in emergency dispatch offices by municipalities. However, the scope of the framework we intend to create has a much broader audience. It would be designed to be able to be used by other developers in the creation of other apps, like ride sharing services, bus transit systems, taxi dispatching, and much more. The project would ideally have an audience to any developer or creator that requires some access to a routing algorithm that can process real time information.

1.3 Intended Use

Similarly to the audience, the intended use of the MXpress project is the emergency vehicle routing system. Additionally, the intended use would be as another component to be integrated into another project or application. Several industries could stand to benefit from a well created implementation of real time data to route vehicles and monitor road conditions.

1.4 Scope

Scope is relatively loosely defined in this project as the final project scope is varied. Our project aims to create an Apache Spark cluster that is able to process batch and stream data from the ontology from the University of Toronto. The minimum requirement for delivery is a framework that processes this batch and stream data and communicates with a routing processor such as OpenTripPlanner to generate routes. The scope also emcompasses an AI/ML component built on the libraries within Spark to better utilize the batch data in the routing. If time constraints permit, a web app and front end with dashboard will be created that allows interaction with the framework for the emergency vehicle routing. The scope is designed to be somewhat adaptable to the time and technological limitations that may be imposed.

1.5 Definitions and Acronyms

ESXi- VMWare product used for virtual machine deployment and server host management
Apache Spark - Open source Apache technology for processing big data using RDD. Contains several processing and query methods, along with several ML library components to be used.
RDD- Resilient Distributed Datasets (Parallelized data storage used by Spark)

2. Overall Description

2.1 User Needs

User Needs pose a challenge to define- the group does not have access to an emergency dispatch center to specifically design the application for any particular individual. The primary focus of user needs is on the main components of a framework that are not yet created, but have some purpose.

Applications such as OpenTripPlanner and other open source routing algorithms based on the OpenStreetMap suite generally do not include any consideration for real time data. Collection of reliable real time data is difficult and typically very processing heavy. The identified user need is an adaptable real time component and ML component that can work alongside the standard routing algorithm to offer enhanced data for routing vehicles without processing overhead on the existing algorithm. The Spark cluster is designed to handle all processing and computation needs as a black box where arguments can be fed as parameters, and a processed map of data is produced.

3. System Features and Requirements

3.1 Functional Requirements

Require the system to take input from some client for two geographic points for mapping. System must be able to process live stream data from the infrastructure sensor points. Additionally, the batch data from the sensors must be stored and processed to allow machine learning using a combination of the two data types. System is required to query and return route data for quickest path using some routing algorithm.

3.2 External Interface Requirements

External Interface Requirements are very demanding in the MXpress project. The key external interfaces that will be used are several Spark libraries, a routing algorithm, stream data from infrastructure sensors, and a front end to the user.

The Spark libraries that will be used include much of the built in functionality for simple ingest and processing of the stream and batch datasets. Additionally, Spark provides the RDD for storing this data across processing nodes, while maintaining fault tolerance. Spark also provides many components to allow implementation of machine learning.

3.3 Nonfunctional Requirements

Primary non-functional requirements include the performance of the system. Using a cluster of Apache Spark nodes to accelerate the query performance and processing is required. Performance of the cluster should allow routing paths to be created in no more than a minute, even when processing large datasets. ML features need to be implemented in a manner that does not bottleneck the stream data processing