Project Proposal
CS6604: Spatial Databases
February 24, 2014

Kyle L. Schutt, Wei Wang, and Parang Saraf

***Abstract*— In this paper, we design an android application to help the traveller to efficiently and economically explore the city through the public transit system. The application will provide the real-time traffic information as well as the smart travel routes with diverse demands by combining the WMATA, Twitter with the local place information such as popular attractions, convenience store, and parking plots and so on.**

***Index Terms*— Data Mining, Android, Travel, WMATA, Twitter, Route, Transit**

I. Introduction

It's no surprise that the smart phones are swiftly taking over the mobile phone market. With Nielsen's mobile insight report, it shows that the smartphone owners in the US make up 62.5% of the market in 2013[1]. There are two mainstream Operation Systems for smartphone platform: Apple IOS and Google Android. The Apple IOS takes the 40.6 percent of the market share, while Google Android takes 52.2 percent.

Given the rapid adoption of smart phone in the worldwide, there is clear evidence that mobile phone are becoming the "second screen" to receive and public information. With hi-resolution and larger screen, the smart phone owners are more comfortable to with their mobile devices. With the hundreds and thousands applications in Apple App Store and Google Play, the smart phone can be used just about anything.

Accompanying with the booming of the smartphone apps, there are more and more enterprises and government departments provide their data or API to public free. For example, Twitter provides stream API [3] to access the real-time tweets data which can be used to monitor the current topic trends or detecting the abnormal events. The Metro provides WMATA API [2] to access the open metro data such as the schedule of the rail and bus, the routes information and the incidents. With the high accessibility and diversity of open data and the prevalence of the smart mobile device, it provides a great opportunity for developers and researchers to design the creative and novel approaches to solve the real world problems and meet the explicit or potential demand of consumers.

Image the following scenario, you are a traveler and this is the first time you come to Washington, DC. Everything here is new to you. You want to visit the museums in Capitol region, and after that you may want to go to a theater to watch a show. Then you may have such questions: Which route should I take with most efficiency and lowest cost? Where are the nearby bus/rail stops around my hotel or destinations? When is the next bus coming? Furthermore, if I want to get a cup of coffee in middle way, which stop should I get out and it would cost least time? As a newer to a strange city, you may ask a lot of questions like this.

In this paper, we propose a novel approach and design an Android application to solve the above questions through open data. The agility and efficiency of information spreading in Twitter make it an ideal platform to monitor the real time incident. In this paper, we predict the potential traffic congestion by analyzing the tweets content. For example, if one user tweets that there is an accident in highway I-66 east bound exit 71. Then the I-66 around exit 71 would have high probability to get congested in soon future. Through the WMATA API, we can get the real-time information of the entire metro transit system. For instance, we can get the Geo-location of each stop as well as the schedule of each line. We also can the rail and bus incidents appeared in the public display throughout the transit system. Through the diverse LBS provider, such as yelp, foursquare, Google Map, we can easily get the rich location information etc. popular attractions, restaurants, coffee café and convenience stores. By combing these three kinds of data sources, our approach could efficiently provide the users multiple real-time routes to satisfy different demands.

The major contributions of our paper are:

- Analyze the real-time traffic congestion through tweets and RSS.
- Combine transit routing information with rich location information.
- Dynamically design the route according to the real-time information and user requirements.

We organize the paper as below: in the section 2, we introduce the related work. In section 3, we will provide an overview of our approach. In section 4, we will talk about the detail of our system design. And finally in section 5, we will talk about our conclusion and future work.

II. Related Work

There are a lot of apps have been developed related to the transit system in cities. Some apps are focused on providing the bus and rail schedule information. For example, the app DC Metro and Bus provides the functionality of the next bus, next train, nearby stops, metro maps and metro alerts. Some apps try to utilize the tweets along with bus/rail schedule information. For instance, Metro Twitter will show the latest tweets about each of Washington DC's WMATA metrorail lines in addition to getting real time train arrivals. But it just simply output the original tweets without any analysis. In our approach, we will predict the congestion situation of each line according to the tweets and then suggest the alternative route to avoid the bad trip. Most of the apps provide the function to search the bus or rail stops nearby or given location. But in some situation, what around the stops are more important than where is the bus stop. Take our previous example, if you drive to a new city, you may more care about which stops has a parking plots around it rather than a specific stop.

III. Proposed Approaches

Since we are focused on WMATA Challenge #1 to provide near real-time information to riders, there are several different approaches we can take to handle the incoming data. Specifically, the data with the WMATA API [5] is a mixture of dynamic and semi-static information about rail and bus lines and associated metadata. For example, rail lines and stations do not change often while arrival information is updated much more frequently. Additionally, this data does not provide exact coordinates of the stations or the corresponding polylines for the rail and bus routes. Therefore, additional GIS layers are required to analyze the correct routes [4].

By creating a native Android application, we can take advantage of the processor more effectively than a simple web-based application in that some of the analysis can be offloaded to the client-side to improve the overall responsiveness. Initially, we thought to create a web-based application that could work cross-platform and was browser agnostic. However, because this application relies on real-time data ingestion and updates, we thought the majority of our development time would be spent creating a dynamic, single-page website instead of focusing on the GIS analytics as part of the system. Creating a native Android application provides many benefits over a web-based version of the system because of increased usability, performance, efficiency, and overall user experience. We could have easily created an iOS application instead of Android; however, Android is the cheaper and easier option for this project.

IV. System Design

In order to meet our three contributions described previously, we have to use several different types of tools to properly analyze the data and display it to the user. In order to expedite the usage of the basic GIS data layers, we will be using OpenGeo Suite as our database and middle layer management application. This will provide us a RESTful web-service for access to our data as needed and in real-time.

On the server side, the database will contain a variety of support datasets that are static. These datasets include Metro Bus Lines and Stops, Entrances, Rail Lines, Park-n-Ride Lots, Station Entrances, Stations, and Station-to-Line cross references [4]. These are all GIS-enabled data layers that we can use for further analysis and visualization. For example, the Rails Lines polyline layer will provide an accurate representation of the Metro Rail routes (as opposed to the skewed metro map).

Furthermore, these datasets only provide a partial picture of the entire system since it is also required to include points-of-interest, arrival and schedule times, weighted routing, fare pricing, and congestion analysis. For this, we must incorporate data from the WMATA API [5]; specifically, Station Predictions for Rail, and Schedule by Stop or Route for Bus. Additionally, we must incorporate fare pricing into the analysis as a weighted function in order to determine the shortest and cheapest route. This data is available from WMATA and the Washington Post as an API [6]. Additionally, this data can be accessed via a custom RESTful API built with Ruby on Rails (RoR) in such a way that minimizes the hit on the API limits by using simple caching of the static data. The RoR application will also act as an aggregator for social media data from Twitter and Yelp!. The above mentioned datasets shall be accessible from the Android application to improve performance and reliability of the system. The Android application shall be able to visualize this data in a useful manner.

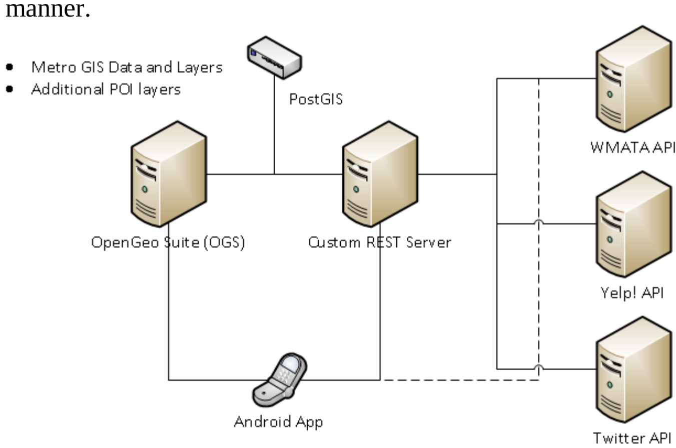

Figure 1: Preliminary Architecture Diagram

As mentioned above, some of the analysis is performed on the client-side and some is performed on the server-side. One of the more intensive calculations will be the routing of a user from one location to another. For this, we can utilize two different approaches: a web-based routing API (Google or MapQuest), or utilize PostGIS pgRouting [7] which provides database level routing functions. In fact, we might utilize both options to compare and visualize the differences for the user. The latter of the two approaches provides more flexibility and for us to analyze the route against the fare pricing and traffic congestion data already stored in the database. Additionally, it keeps the entire system more contained then it otherwise would be. It also allows us to perform analysis on a stop-by-stop basis to pull points-of-interest in the surrounding area (similar to walkscore.com).

On the client-side, it has already been mentioned that some analysis will occur with respect to the real-time data to provide a seamless user experience. The user shall be presented a list of the closest rail and bus stations based on the current location, or the user can select from an alphabetical list. The user shall select an origin and destination in the application. The application shall display multiple routing options based on price, congestion, length, duration, and other factors. These other factors may include a simple selection based on the user's mood. For example, they may select "Hungry" to get a list of restaurants or food places at each stop along their journey. They may also select "Get me home!" which will give them the fastest route without regard to price. There shall be similar options to these such as "Scenic route" for tourists or "Cheapest" for the budget conscious rider.

References

[1] http://www.comscore.com/Insights/Press_Releases/2013/12/comScore_Reports_October_2013_US_Smartphone_Subscriber_Market_Share
[2] http://developer.wmata.com/Application_Gallery
[3] https://dev.twitter.com/docs
[4] http://dcatlas.dcgis.dc.gov/catalog/results.asp?pretype=All&pretype_info=All&alpha=M
[5] http://developer.wmata.com/docs
[6] http://projects.washingtonpost.com/wmata/stations
[7] http://pgrouting.org/

Appendix A – Task Assignment

Kyle Schutt:

- Server-side management, i.e. OpenGeo
- GIS data management
- Twitter analytics

Wei Wang:

- Android (client-side) development
- …

Parang Saraf:

- …

Appendix B – Schedule

2/24/2014	Project Proposal
3/17/2014	OGS w/ data available
3/31/2014	Beta Android app
4/7/2014	Project Checkpoint
4/18/2014	Real-time traffic congestion data implemented
4/28/2014	Polished Android app
5/5/2014	Project Presentation
5/12/2014	Project Report