Smart Traffic Cloud: An Infrastructure for Traffic Applications

Wen Qiang Wang, Xiaoming Zhang, Jiangwei Zhang, Hock Beng Lim

Intelligent Systems Centre, Nanyang Technological University, Singapore

***Abstract*—With rapid development of sensor technologies and wireless network infrastructure, research and development of traffic related applications, such as real time traffic map and on-demand travel route recommendation have attracted much more attentions than ever before. Both archived and real-time data involved in these applications could potentially be very big, depending on the number of deployed sensors. Emerging Cloud infrastructure can elastically handle such big data and conveniently providing nearly unlimited computing and storage resources to hosted applications, to carry out analysis not only for long-term planning and decision making, but also analytics for near real-time decision support. In this paper, we propose Smart Traffic Cloud, a software infrastructure to enable traffic data acquisition, and manage, analyze and present the results in a flexible, scalable and secure manner using a Cloud platform. The proposed infrastructure handles distributed and parallel data management and analysis using ontology database and the popular Map-Reduce framework. We have prototyped the infrastructure in a commercial Cloud platform and we developed a real-time traffic condition map using data collected from commuters' mobile phones.**

I. Introduction

Monitoring and managing traffic, for real time, as well as long term decision making is desirable both for policy makers and the general public. Technical advancements on sensor and wireless network technologies, and proliferation of smart portable devices has made collection of the necessary data extremely easy. For example, experts from urban development authority may use archived traffic information for better road infrastructure planning; traffic police may use real-time road information to identify road congestion and figure out how to disperse traffic; public users may receive on-demand and dynamic route recommendation from on-board devices that have access to traffic information, which saves both time and fuel.

There are two primary sources of traffic related data information: (1) Direct data feed from sensors which function specifically to gather traffic information and are deployed by local transport authorities or chartered companies. These sensors include cameras, vehicle counters deployed on road side and Electronic Road Pricing (ERP) gates, etc. (2) Inferred data from readings of other sensors that do not measure traffic information directly. A typical example is GPS readings which can be used to compute the speed of the GPS device, which may allow an analyst to figure out if a particular road segment is congested or not.

Given the wide-scale deployment of GPRS/3G networks and over one billion deployed smart phones [1], [2] equipped with advanced GPS and acceleration sensors, *participatory sensing* [3] is a promising cost-effective approach for collecting traffic related data in a very large scale.

Building the back-end IT infrastructure for traffic related applications, however, is non-trivial. To cater the huge volume of incoming data and constant intensive computations, the infrastructure must be very robust and scalable. The infrastructure may also need to take the individual data owners' privacy into consideration, as certain information such as smart phone owners' locations are sensitive and should not be revealed to unauthorized entities. Last but not least, the infrastructure must be capable of seamlessly integrating both classes of data sources mentioned above, as well as other relevant sensor data and information from other categories (such as pollution level, weather conditions), so that more complicated applications that uses multiple data sources can be implemented. Traditional centralized data storage and client-sever based architectures no longer meet these requirements mainly because

1521-9097/12 $26.00 © 2012 IEEE
DOI 10.1109/ICPADS.2012.134

they are not scalable in the sense that their performance decreases rapidly if the need for resources grows rapidly. A cloud based infrastructure is a natural choice, since it provides elastic storage and computing resources. Its on-demand resource allocation capability makes it an ideal platform for both individuals and companies to easily manage huge amount of data from heterogeneous sources and rapidly prototype, test and deploy applications replying on these data. We believe that using cloud as the back-end data and application server for traffic related applications is a reliable and scalable solution to the challenges mentioned previously. Our contributions in this paper are:

1. We propose the *Smart Traffic Cloud* infrastructure for traffic data management and traffic related application based on data from multiple sources, including sensed in a participatory manner. The back-end infrastructure leverages on the popular Map-Reduce [4] model to realize distributed computing in cloud so that huge amount of traffic data can be processed in parallel to meet the needs of real-time decision making, as well as to scale in general.

2. We propose ontology based data management framework for Smart Traffic Cloud, which enables seamless integration of data from heterogeneous data sources.

3. We implemented Smart Traffic Cloud in a commercial cloud platform with real data from deployed smart phones. Based on the infrastructure and the collected data, we built a web-based prototype application that presents real-time traffic conditions on an interactive map interface.

The rest of the paper is organized as follows. Section II describes the proposed Smart Traffic Cloud infrastructure. Section III presents the data management framework and Section IV describes the design and implementation of the traffic condition map prototype. Section V discusses related works and finally we conclude in Section VI.

II. The Smart Traffic Cloud Infrastructure

A. Architecture

Figure 1 shows the architecture of Smart Traffic Cloud, which consists of three layers: *Infrastructure*, *Data Processing/Analysing* and *Application*.

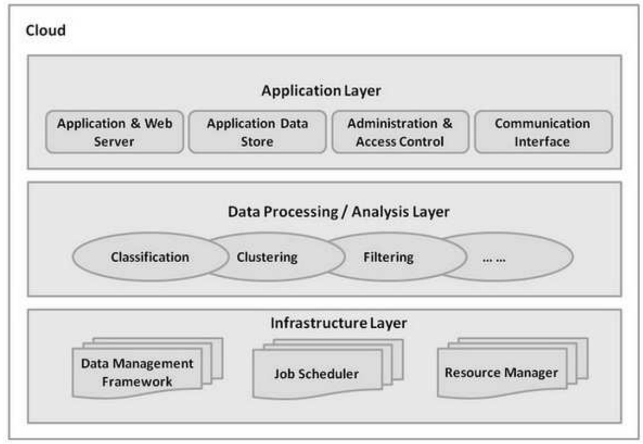

Fig. 1. Architecture of Smart Traffic Cloud

- The *infrastructure layer* aims to provide flexible, scalable and heterogeneous server clusters to higher level modules and services. It has components such as *data management framework*, *job scheduler* and *resource manager*. Data management framework manages data from various data sources such as traditional relational databases, relatively newly developed *No-SQL* databases or files with dumped data. The framework also incorporates ontology management to provide semantic aware access to sensor data and other traffic related information. The framework itself can be implemented in a distributed fashion to make data management scalable and extensible. The details of the framework can be found in section III. Job scheduler and Resource manager are core components in this layer to provide distributed and parallel processing capability. We adopt the Map-Reduce framework in this layer to implement these two modules.
- The *data processing/analysing layer* provides modules that carry out the data processing and analysis. These modules implement data analytic algorithms and machine intelligence techniques such as *classification*, *clustering* and *filtering* to extract valuable information from a huge amount of raw data. Such information is subsequently passed on to the application layer for presentation, visualization or other user specific tasks.
- The *application layer* contains software com-

ponents that directly support traffic related applications. These components include: *web/application server*, which receive or deliver contents from/to clients over the Internet; *application data store*, which stores application related information, such as user information, analysis results from data processing/analysing layer, etc.; *administration & access control* caters both administration tasks like authentication and logging, and access control enforcement to protect data owner's privacy; while the *communication interface* handles data exchange between the back-end infrastructure and the participating smart phones through GPRS/3G networks.

B. Applying Map-Reduce in Smart Traffic Cloud

Smart Traffic Cloud adopts Map-Reduce as the distributed computing model to process the data in parallel:

- ”Map” step: The master node takes a batch of traffic data files obtained from mobile phone users in a regular interval (e.g. 30 seconds), divides them into smaller tasks by users, and distributes them to slave worker nodes. The worker nodes process the assigned tasks, including raw data files parsing, GPS data filtering, road matching and traffic condition updating, then finally passes the processing results back to its master node. The output format of the map operation is a list of the following tuple: *(latitude, longitude, direction, speed)*
  In this output tuple, latitude and longitude represent the segment points given by Google Map service which feature a specific road while keeping a reasonable short list of the (latitude, longitude) pairs. Direction can be 0 or 1, meaning two different directions for a single road. Speed is the vehicle s travelling speed on the target road. For example, the map output derived from the vehicle A running on the AYE highway in Singapore can be:
  (1.31288, 103.76179, 1, 50)
  (1.31421, 103.76045, 1, 51)
  (... ...)
  In the ‘Map’ step, the major computing is around matching GPS points inputted from mobile user with the segment points featuring a specific road. Current solution is implemented by a rectangle method. Firstly a serial of thin rectangles along the traveling direction on the road are drawn by using two adjacent segment points as diagonal points. Next, the inputted GPS points are mapped with those rectangles. If an inputted GPS point drops into a specific rectangle, then one of the diagonal points of the rectangle is marked with the speed corresponding to the inputted GPS point.
- ‘Reduce’ step: The master node collects the processed results from distributed tasks and combines them to form a reasonable traffic condition estimation of the roads. The initial implementation is simple, and just computes the average traveling speed of the vehicles by combining the data from the different mobile users and tasks with overlapped road and time span. We note that the analysis can be improved by carrying out more intelligent traffic analysis, prediction and route recommendation by employing sophisticated machine learning algorithms.

III. Data Management

The design of Smart Traffic Cloud is guided by a longer term plan, where we envision data from multiple sources are fused together for more complex analysis and sophisticated applications. In order to realize such a vision, it is essential to go beyond low level data storage and access methods. The semantics of the data becomes equally important, since it provides essential indicators on how the data can be used by third parties. Data from heterogeneous sources are usually coded in different formats. For example, GPS data obtained from participatory sensing projects may be kept in small files, each line represents a data record with schema like *(timestamp, latitude,longitude,altitude,bearing,instant_speed)*, whereas data obtained from transport authority's road sensors may be in XML format with a set of predefined tags. In addition to sensor data, other information such as accident information published by traffic police should also be incorporated as they are vital to applications such as intelligent travel planning. To make all these information available to application developers in a systematic and organized manner and reduce the application development overheads, an approach that is capable of handling both the data and the semantics of the data is essential.

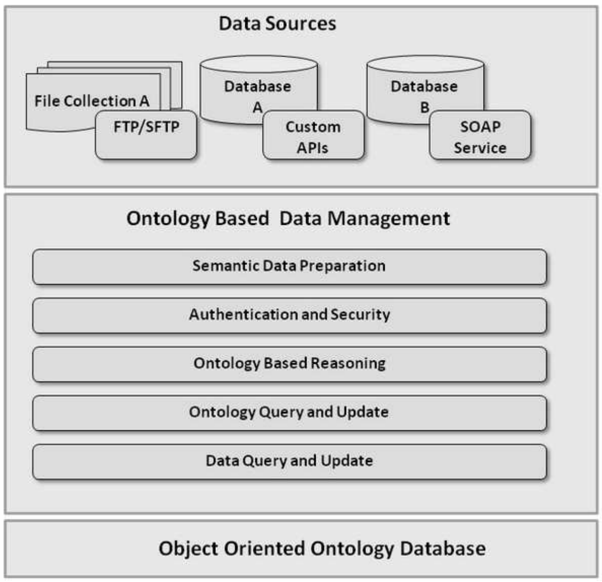

Fig. 2. The ontology based data management

Figure 2 shows our proposal of a ontology-based data management framework. We make use of ontology to formally describe heterogeneous data sources and their data. Ontologies are generated by data owners or designated data analysts in standard OWL [5] documents and imported into an object-oriented database as a hierarchical structure with a set of objects that represent various attributes of the data sources and the data, including sensor deployment details, sensor data formats, calibration functions, available time slots, etc. When a particular type of sensor data is queried, ontologies of all the sensors of that type are retrieved from the ontology database so that users are able to chose the sensors he is interested in and obtain both the actual sensor readings as well as the relevant ontology, which allow him to correctly interpret and use the data. Authorization information can also be included in ontologies which enables access control on both ontology and data.

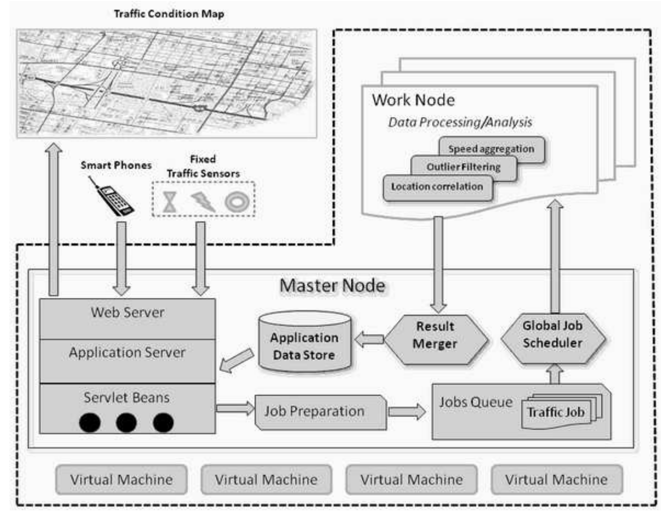

Fig. 3. Implementation of Real-Time Traffic Condition Map

IV. Real-Time Traffic Condition Map

A. Application Implementation

Figure 3 illustrates the implementation design of the real-time traffic condition map, which conforms to the Smart Traffic Cloud architecture defined in Figure 1. The data analytic layer is implemented in Hadoop [6], which is an open-source realization of the Map-Reduce framework. The interface layer is currently implemented in master node and is providing support to various applications and web services such as data acquisition from smart phones and other traffic sensors, user privilege administration and real-time traffic data feeds.

The work flow is as following: participating smart phones upload data files with timestamp, GPS coordinates, speed and acceleration information to the master node through web services. The master node then takes a batch of uploaded data files to generate a list of job files and put them in a job queue. Next, the global job scheduler dispatches each job file to a work node with suitable resource. When a work node receives a job file, it basically perform a three-step task:

1) Find out corresponding road segments the uploaded GPS coordinates belongs to. We first use Google's reverse Geocoding service [7] to identify the street name. Once the street name is determined, we retrieve pre-defined segments of the road from Google Map API

and compute which segment the coordinate falls in. The last step is to use speed vector to decide the driving direction of the vehicle, hence to locate the actual track on that road segment, i.e whether it is on inbound track or outbound track.

2) Filter out invalid or erroneous speed or acceleration readings. Threshold based heuristic decision tree is used to find correct and relevant GPS readings.
3) Calculate vehicle speed on that road segment. From the second step, a number of speed and acceleration readings are extracted from raw data files and these readings are clustered according to road segments. Although in the previous step, invalid readings have already been filtered, the speed readings may still vary much. Algorithms based on statistical model and consideration of historical data and instant acceleration reading can be used to remove outliers, hence producing a relatively accurate information.

Results from each work node are combined in the result merger module, where the final result, i.e., speed on all road segments, is formed. The application database is then updated accordingly and a web services is initiated to send updated information to external Map APIs that generates the real-time traffic condition on street map interface.

B. Deployment & Test Runs

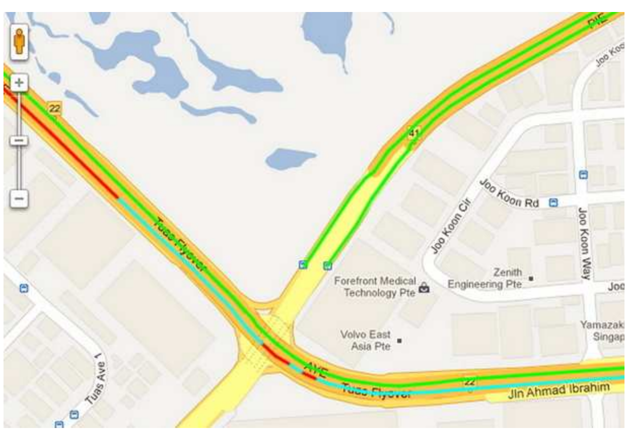

Fig. 4. A sample of Real Time Traffic Condition Map

A small program was developed for both Android and iOS platforms to enable participatory sensing for traffic related applications. The program collects GPS readings, which includes geographical location, instant speed and bearing, and acceleration readings from smart phones continuously and transmit these data to back-end infrastructure through GPRS/3G networks. Data transmit frequency is set to be significantly lower than that of data collection so that network bandwidth and device battery can be better utilized. The back-end infrastructure itself is deployed on the commercial Alatum [8] Cloud operated by Singapore Telecom. A number of *Virtual Machines (VM)* are utilized, among which one VM with largest memory and number of CPU cores is assigned the role of *Master Node* and rest of the VMs are used as *Work Node*. Vehicle speed information is used to classify traffic condition into several categories that are represented in different colors in the map interface, e.g., green stands for smooth traffic, blue stands for crowded traffic and red stands for congestion. Figure 4 is a snapshot from the real time traffic map and it illustrates traffic condition on major express ways at a selected time right before evening peak hour based on real data collected.

V. Related Work

There has been much work on traffic monitoring. CarTel [9] and Surface Street Traffic Estimation [10] leverage vehicle-based GPS units, tracking the movement of vehicles and reporting the information back to a server for aggregation and analysis. Washington State SmartTrek-Busview service [11] and the INRIX system [12] track vehicles and predict traffic condition using GPS information as well as other information from fixed traffic sensors deployed along the road, such as inductive loop vehicle detectors, traffic cameras, Doppler radar, etc. FASTMOD [13] applies Map Reduce technique through Hadoop to achieve near real time processing of large data from user mobility and vehicular networks, which has a similar back-end system as Smart Traffic Cloud, except that the traffic data are not obtained from participatory sensing network. Mobile Millennium [14] and TrafficSense [15] uses mobile phones carried by users as traffic probes.

For above mentioned applications that reply on traffic data collected from pre-deployed sensor networks and vehicular GPS devices, deploying full scale monitoring infrastructure is very costly and time consuming, making them less likely to be practical in real life. Mobile Millennium and Traffic-Sense demonstrate the growing tend of participatory sensing for data acquisition. The proposed Smart Traffic Cloud infrastructure is designed to satisfy requirements raised from both participatory sensing and dedicated sensor networks. It is designed in Big Data scenario and leverage cloud computing to archive scalability and flexibility. Recently, Google has also deployed a traffic map system on it's map interface [16], [17], which essentially uses a similar approach that we are using for the Real-Time Traffic Condition Map application.

VI. Conclusion and Future Work

In this work, we proposed Smart Traffic Cloud that exploits cloud computing to provide flexible and scalable support to traffic related applications. The infrastructure adopts the Map-Reduce framework for distributed and parallel processing of traffic data and incorporates ontology based data management so that data from heterogeneous sources can be seamlessly integrated. Based on the proposed infrastructure, we developed a participatory sensing application that produces real-time road traffic information by analyzing data collected from smart phones.

There are several directions we would like to explore in the future work: (1) Security and privacy remains one of the top concerns in the cloud environment, and participatory sensed data magnifies the concern. Proper measures need to be applied for sensitive information such as smart phone owner's exact locations, daily travel patterns, etc. to be kept confidential from unauthorized access, in order to encourage participatory sensing. Techniques such as encryption or location cloaking may be applied to that end, but how that works in conjunction with the needs for fusing data from multiple sources, and carrying out sophisticated analytics is an open question. (2) We plan to develop more applications with the proposed infrastructure by fusing collocated data from more sources to demonstrate its advantage over a traditional architecture. For example, a travel planning decision support application that takes an user's historical travel patterns, real-time road condition and transport authorities' notification to generate personalized travel recommendations dynamically.

VII. Acknowledgments

This work has in part been supported by the A*Star TSRP grant 102 158 0038 for 'pCloud: Privacy in data value chains using peer-to-peer primitives' project. The authors would like to thank Professor Anwitaman Datta of School of Computer Engineering, Nanyang Technological University, for his feedbacks on this paper.

References

[1] Wikipedia, http://en.wikipedia.org/wiki/Smart_phone.
[2] ——, http://en.wikipedia.org/wiki/Mobile_phone.
[3] J.Burke, D.Estrin, M.Hansen, A.Parker, N.Ramanathan, S.Reddy, M.B.Srivastava, "Participatory sensing," in *First Workshop on World-Sensor-Web: Mobile Device Centric Sensory Networks and Applications, Sensys 2006*.
[4] Jeffrey Dean, Sanjay Ghemawat, "Mapreduce: Simplified data processing on large clusters," in *OSDI'04: Sixth Symposium on Operating System Design and Implementation*.
[5] Wikipedia, http://en.wikipedia.org/wiki/Web_Onto_Language.
[6] ——, http://en.wikipedia.org/wiki/Apache_Hadoop.
[7] Google Maps API Web Services, https://developers.google.com/maps/documentation/geocoding/.
[8] SingTel Alatum, http://www.alatum.com.sg/index.aspx.
[9] B. Hull, V. Bychkovsky, K. Chen, M. Goraczko, A. Miu, E. Shih, Y. Zhang, H. Balakrishnan, S. Madden., "The cartel mobile sensor computing system," in *Sensys 2006*.
[10] J. Yoon, B. Noble, M. Liu, "Surface street traffic estimation," in *MobiSys 2007*.
[11] D. J. Dailey, "Smarttrek: A model deployment initiative," Washington State Transportation Center (TRAC), University of Washington, http://www.wsdot.wa.gov/Research/Reports/500/505.1.htm, Tech. Rep., May 2001.
[12] INRIX Dynamic Predictive Traffic, http://www.inrix.com/technology.asp.
[13] K. L. Gautam S. Thakur, Yibin Wang, "Fastmod: A framework for realtime spatio- temporal monitoring of mobility data," Department of Computer & Information Science & Engineering, University of Florida, Tech. Rep., 2011.
[14] J. Herrera, D. Work, R. Herring, X. Ban, Q. Jacobson, and A. Bayen, "Evaluation of traffic data obtained via gps-enabled mobile phones: the mobile century field experiment," *Transportation Research Part C*, vol. 18, no. 4, pp. 568–583, August 2010.
[15] Prashanth Mohan, Venkata N. Padmanabhan, Ramachandran Ramjee, "Nericell: rich monitoring of road and traffic conditions using mobile smartphones," in *Sensys 2008*.
[16] Google Maps Support: Directions and Traffic, http://support.google.com/maps/bin/topic.py?hl=en&topic=1687356.
[17] Mashable Tech, http://mashable.com/2012/03/29/google-maps-traffic-data.