Real-Time Traffic Information Management using Stream Computing

Alain Biem, Eric Bouillet, Hanhua Feng, Anand Ranganathan, Anton Riabov, Olivier Verscheure
IBM TJ Watson Research Center, NY, USA
{biem, ericbou, hanhfeng, arangana, riabov, ov1}@us.ibm.com

Haris Koutsopoulos,Mahmood Rahmani
The Royal Institute of Technology, KTH, Stockholm, Sweden
{hnk,mahmoodr}@kth.se

Barış Güç
ETH, Zurich, Switzerland
baris@student.ethz.ch

Abstract

With the widespread adoption of location tracking technologies like GPS, the domain of transportation information management has seen growing interest in the last few years. In this paper, we describe a stream processing infrastructure for processing large volumes of sensor data in real time to derive useful traffic and travel planning information. We have used this infrastructure to process floating car data for the city of Stockholm in real-time. Our findings show that there is a great need for real-time traffic information management because of the tremendous variability in traffic conditions in a city like Stockholm. Also, our stream processing infrastructure can help meet this need by supporting the development of applications that can process large volumes of GPS and other data on a distributed cluster of machines.

1 Introduction

Intelligent Transportation Systems (ITS) have brought many advances in the transportation management field. An important development is the emergence and installation of sensor technologies for the collection of various types of data on the state of the transport system. GPS is an excellent example of this new generation of sensors that has the potential to provide high quality traffic data for real time traffic monitoring and management, as well as planning, policy, and applications, at a relatively low cost. An important characteristic of this new source is that it includes a fair amount of floating car data (FCD). FCD represent the location of vehicles collected by mobile sources, such as GPS devices installed in vehicles or cellular phones. This raw data can be sent to a central facility, where it can be processed in real-time.

In this paper, we briefly describe some of our recent work in supporting real-time Traffic Information Management using a stream computing approach. This work was made possible by access to GPS data from some taxis and trucks in the city of Stockholm. We highlight some of our findings on traffic variability in the city of Stockholm. We also show how we have used IBM's System S stream processing platform for the purpose of real-time traffic information management. We have developed applications on this platform that process real-time GPS data, generate different kinds of real-time traffic statistics, and perform customized analyses in

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

response to user queries. Examples of customized analyses include continuously updated speed and traffic flow measurements for all the different streets in a city, traffic volume measurements by region, estimates of travel times between different points of the city, stochastic shortest-path routes based on current traffic conditions, etc. Our system can handle large volumes of incoming GPS data. For instance, on a cluster of four x86 blade servers, it can process over 120000 incoming GPS points per second, combine it with a map containing over 600,000 links, continuously generate different kinds of traffic statistics and answer user queries.

2 Need for Real-Time Traffic Information

Obtaining real-time traffic information is extremely valuable. In fact, our experiments using the GPS data set from the city of Stockholm reveals that traffic in the city varies significantly during different time scales. Figures 1 and 2 provide a glimpse into the highly variable traffic situation. These figures illustrate the immense benefits that can be obtained for commuters, city agencies, commercial fleet operators and other parties if they have access to real time traffic information and predictions. We shall describe the actual data and the processing of the data to derive these statistics in Section 3.

Figure 1 shows the variability of the travel times between the center of the city of Stockholm and Arlanda Airport. These travel times were obtained by observing the same vehicle at the two locations within a certain time interval. The travel times are grouped by day of the week and time period (blue for the morning and red for the afternoon period). The results indicate that average (shown by dot) and standard deviation (std, shown by bold vertical lines) of travel times varies day-by-day. For example, as expected, the average is less during weekends compared to weekdays, but there is also variability among weekdays. Additionally, the travel time depends on the time of the day, both in terms of average and variability. For example, the average and standard deviation travel time in the afternoon (red) is increased by few minutes compared to the evening (blue).

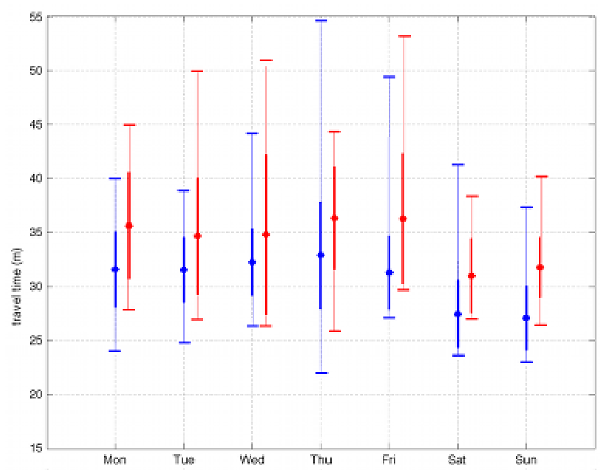


Figure 1: Travel time variability for the Stockholm-Arlanda route. Average (dot), standard deviation (bold line), and min/max (bar) of travel time. 7:00-9:00 (blue) and 16:00-18:00 (red).

Another experiment we ran with our GPS data set was to first derive the average speed on different road links at different 5 minute intervals of time during the day. We then used this average speed information to compute time-dependent shortest paths between various pairs of points in the city. We calculated the shortest paths between two points for 50 consecutive departure times between 6 AM and midnight, spaced 20 minutes apart. In this way, we can get an idea of how frequently the shortest path between two points on the road changes during the day. Each shortest path calculation uses time-varying link travel time information, where the estimated travel time on each link for each 5 minute interval during the day was calculated based on historical averages.

In Figure 2, we visualize an origin-destination pair where the time and traffic dependent shortest path changes frequently during the course of the day. The 50 shortest paths found for different departure times are displayed between the two points. The large number of changes for this pair is due to the fact that the points are in the central area of the city where there is a large number of links with traffic updates and due to the existence of many distinct alternative roads between the origin and the destination. The map shows how the shortest path switches between different highways and local roads during the day, and how smaller streets can be used as shortcuts instead of highways in some periods.

This figure shows that the best route between two points in the city can change very often, and in much more diverse ways than may have been imagined. Hence, making use of real time traffic information and good traffic prediction algorithms can result in massive savings of time and energy on the part of drivers.

Figure 2: Origin-destination pair with a large number of different shortest paths. 50 shortest paths between 6 AM and midnight for this origin-destination pair are displayed. The paths are drawn with offsets for better visualization. (©2010 Google - Map data ©2010 Tele Atlas)

3 Stream Processing Infrastructure

A key feature in our work has been the use of stream processing for the purpose of real time traffic information management. In particular, we have used System S [4] [1], which is an IBM research platform that supports high performance stream processing. It has been used in a variety of sense-and-respond application domains, from environmental monitoring to algorithmic trading. It offers both language and runtime support for improving the performance of streaming applications via a combination of optimized code generation, pipelining and parallelization. It supports a component-based programming model that simplifies the development of complex applications.

System S supports structured as well as unstructured data stream processing and can be scaled to a large number of compute nodes. The runtime can execute a large number of long-running jobs (queries) that take the form of data-flow graphs. A data-flow graph consists of a set of operators connected by streams, where each stream carries a series of Stream Data Objects (SDOs). The operators communicate with each other via their input and output ports, connected by streams. The operator ports as well as the streams connecting them are typed.

System S supports a declarative language called SPADE [3] to program stream-processing applications and to define the data-flow graph. SPADE supports a modular, component-based programming model, which allows reuse, extensibility and rapid prototyping. It supports a toolkit of all basic stream-relational operators with rich windowing semantics. It allows extending the set of built-in operators with user-defined ones, programmable in either C++ or Java. It also supports a broad range of stream adapters used to ingest data from outside sources and publish data to outside destinations, such as network sockets, relational and XML databases, etc. It also allows developing applications that offer high-availability through replicated processing and operator checkpointing.

System S includes a scheduler component that decides how best to partition a data-flow graph across a distributed set of physical nodes [8]. The scheduler uses the computational profiles of the operators, the loads on the nodes and the priority of the application in making its scheduling decisions.

3.1 The Raw Data

We obtained historical GPS data traces from Trafik Stockholm [7] for the year of 2008. This data included traces from about 1500 taxis and 400 trucks that plied the streets of Stockholm. In total, there was about 170 million GPS probe points for the whole year. Each taxi produces a GPS probe reading once every 60 seconds that includes taxi identification and location information. Also, for privacy reasons, taxis produce fewer readings

[1] System S is the basis for the IBM InfoSphere Streams product

when they are carrying passengers. Trucks use more recent and more accurate GPS devices, that produce readings once every 30 seconds and include identification location, speed and heading information. The data rate for the whole city was over 1000 GPS readings per minute. However, our system can handle much larger input rates.

In addition, we are now receiving real-time data on GPS from taxis as well. The data rate is still in the order of 1000 GPS readings per minute. In fact, Figure 1 was derived from this real-time data feed over a 4 month period from Dec 2009 to Mar 2010, and Figure 2 was derived from the historical GPS data from all of 2008.

The Stockholm city road network which has 80735 polylines (i.e. road segments) and 37458 nodes (i.e. intersections). The maximum link length on the network is 10756 meters while the average is 142 meters. These 80735 polylines translate into over 600,000 individual line segments.

3.2 Overall Application Description

Having large numbers of vehicles sending real-time GPS data for the city allows us to create a picture of the traffic condition in time and space [6]. We now describe the stream processing applications that allow us to come up with the traffic information, as well as provide various value-added services on top of the basic information.

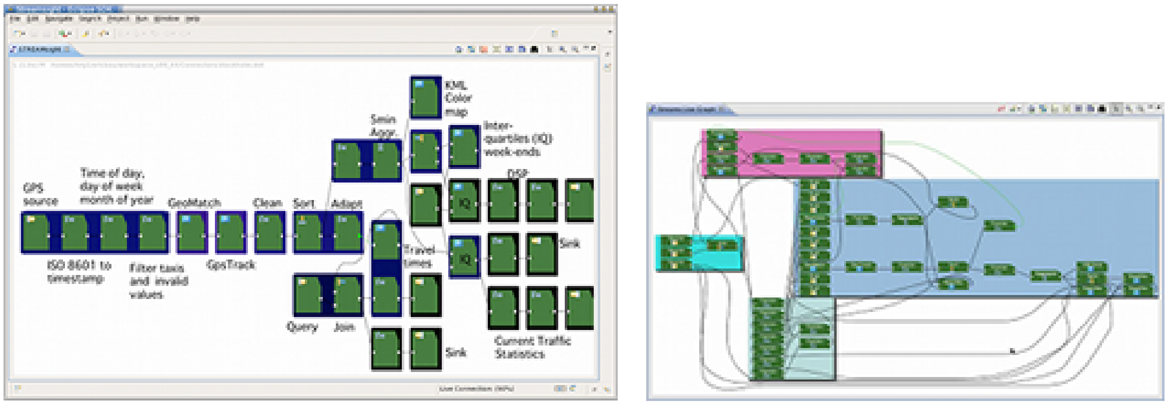
The application processes the data in three distinct phases. The first phase consists of real-time processing of the data. This includes obtaining, cleaning, de-noising, and matching the GPS data to the underlying road network or specified regions. In the second phase, the data is aggregated to produce traffic statistics per link and per time interval. The traffic statistics are in the form of medians and quartiles of vehicle speeds and vehicle counts on the link or region for the time interval. In the final phase, we make use of the statistics to compute different kinds of derived information such as the estimated travel times and shortest paths between different parts of the city. The final outputs can be sent to the user on different kinds of visualization platforms such as Google Earth, or may be stored in a database for additional offline analysis. Further details of this process is described in [1]. A video describing the prototype is available at [5].

For the shortest path calculation, we use the method described in [2] to adapt A* algorithm for time-dependent networks. The A* algorithm is an improvement over Dijkstra algorithm for the one origin, one destination shortest path problem and it utilizes a heuristic function that decreases the search space. In our implementation, A* is adapted using the following modification: as paths are extended with links during the execution of A*, time is advanced and therefore future delay values of links are used as link costs. In addition, the algorithm is adapted to accept continuous changes in the estimated travel times for each link based on the current and predicted traffic conditions.

Figure 3(a) shows a screenshot of the deployed application. In the figure, boxes represent operators, interconnections represent data streams, and the resulting topology represents the entire application flow-graph. It also shows how some operators are fused together to form a PE (Processing Element), represented as large dark background rectangles that contain one or more individual operators. The purpose of fusing operators is to reduce the data transfer latency by having these operators be part of the same process with a shared address space. This fusion of operators is one of the many performance enhancing optimizations supported by the System S platform.

Figure 3(b) shows how those PEs are distributed across various hosts (nodes). In this example, the distribution is based on instructions in the SPADE program assigning different operators to various nodes, but it can also be done automatically by the System S scheduler.

This particular application is designed to deal with GPS data, but can accommodate other sources to better estimate and predict traffic conditions. These include induction loops, weather data, road incident information, video cameras, etc. The addition and processing of various types of data is a particular challenge for real time transportation information management. The System S platform, with its component based and modular programming model, does simplify this process and allows incrementally adding new kinds of data and processing.



(a) Application flow graph of operators, with labels describing the operations performed by different groups of operators.

(b) Application flow graph showing PE within hosts.

Figure 3: Different visualizations of the application.

4 Conclusion

In this paper, we have motivated the need for real time traffic information management using examples of traffic variability in the city of Stockholm. We have also briefly described a scalable stream processing approach for supporting real time traffic information management. Our implementation is based on IBM's System S platform, which is well suited to deal with scalability and adaptability challenges associated with real-time traffic information management. As part of future work, we are investigating several enhancements to our current implementation, including traffic prediction, multi-modal travel planning and multi-sensor data fusion.

Acknowledgements: The authors would like to thank Inga-Maj Eriksson from the Swedish Transport Administration and Tomas Julner from Transport Administration and Trafik Stockholm for their support, feedback, and provision of the data for this study, and Erling Weibust of IBM for facilitating the IBM/KTH collaboration.

References

[1] A. Biem, E. Bouillet, H. Feng, H. Koutsopoulos, C. Moran, A. Ranganathan, A. Riabov, and O. Verscheure. IBM InfoSphere Streams for Scalable, Real-Time, Intelligent Transportation Services. In SIGMOD 2010

[2] I. Chabini and S. Lan. Adaptations of the A* algorithm for the computation of fastest paths in deterministic discrete-time dynamic networks. *IEEE Trans on Intelligent Transportation Systems*, 3(1):60–74, 2002.

[3] B. Gedik, H. Andrade, K.-L. Wu, P. S. Yu, and M. Doo. SPADE: the System S declarative stream processing engine. In *SIGMOD 2008*, pages 1123–1134, 2008.

[4] IBM InfoSphere Streams. http://www-01.ibm.com/software/data/infosphere/streams/.

[5] IBM InfoSphere Streams. InfoSphere Streams enables smarter transportation at the city of Stockholm. http://www.youtube.com/watch?v=i2yCfLQV6M8.

[6] R. Kuehne, R.-P. Schaefer, J. Mikat, K. Thiessenhusen, U. Boettger, and S. Lorkowski. New approaches for traffic management in metropolitan areas. In *Proceedings of IFAC CTS Symposium*, 2003.

[7] Trafik Stockholm. http://www.trafikstockholm.com.

[8] J. Wolf et al. SODA: an optimizing scheduler for large-scale stream-based distributed computer systems. In *Middleware*, pages 306–325, 2008.