Real-Time Spatio-Temporal Analytics using Microsoft StreamInsight

Mohamed Ali[1], Badrish Chandramouli[2], Balan S. Raman[1], Ed Katibah[1]

[1]Microsoft SQL Server, {mali, sethur, edwink}@microsoft.com

[2]Microsoft Research, {badrishc}@ microsoft.com

ABSTRACT

Microsoft StreamInsight (StreamInsight, for brevity) is a platform for developing and deploying streaming applications that run continuous queries over high-rate streaming events. StreamInsight adopts a temporal stream model to handle imperfections in event delivery and define consistency guarantees on the output. This demo highlights the ability of StreamInsight to monitor, analyze and correlate spatio-temporal stream data that is generated by moving objects. The demo scenario is based on the Microsoft Shuttle Service where GPS readings are generated and streamed by shuttles as they move around the Microsoft main campus in Redmond, WA. The demo presents a set of relational continuous queries as well as various real-time analytics that help improve the efficiency of the shuttle service in terms of the average wait time per passenger and the average daily mileage per shuttle.

Categories and Subject Descriptors

H.2.4 [**DATABASE MANAGEMENT**]: Systems – Query Processing, Relational databases.

General Terms

Algorithms, Design, Theory.

Keywords

Data Streaming, SQL Server, StreamInsight, Spatio-temporal, Complex event processing, Analytics.

1. INTRODUCTION

Microsoft StreamInsight (StreamInsight, for brevity) [1, 2] is a platform for stream query processing. Thanks to its high-performance and low-latency stream query processing engine, StreamInsight can monitor, analyze, and correlate stream data from multiple sources to extract meaningful patterns and trends in real time. StreamInsight adopts a deterministic stream model that leverages a temporal algebra as the underlying basis for processing long-running *continuous queries* [3, 4]. In general, streaming applications demand the ability to cope with high input rates of data streams. Moreover, the input streams are usually characterized by imperfections in event delivery (i.e., incomplete or out-of-order data). StreamInsight is architected to natively handle imperfections in event delivery and to provide consistency guarantees on the output even if late and out-of-order stream events are still in transient.

Our analysis of customers' data, queries, and applications shows a remarkable diversity in customer requirements, which are imposed by their respective business domains. However, most customer requirements can be broadly categorized into a limited set of distinct workloads. Out of this limited set of workloads, we focus here on two sets:

1) *Stream-oriented workloads,* where stream data acquisition and processing has advanced the state of the art in many applications domains, e.g., web analytics, fraud detection, RFID monitoring, manufacturing and production line monitoring, smart power meters, financial algorithmic trading, and stock price analysis.
2) *Spatial-oriented workloads,* where the incoming data contains geographical locations and/or spatial extents such as point coordinates, lines, or polygons.

The first set of workloads has been addressed by existing features in StreamInsight. Some of the key distinguishing features in StreamInsight include a declarative query language based on LINQ [7] that adheres to a relational algebra with temporal extensions, the ability to provide consistency and correctness guarantees on the output in the presence of stream disorder and possible modifications to prior events, run-time query composability, and operator sharing. Moreover, *StreamInsight* is an extensible system that seamlessly integrates user defined modules into the execution query pipeline.

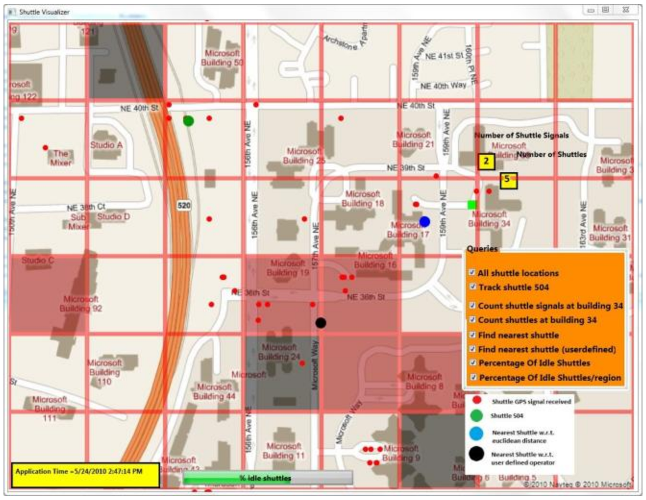
On the other hand, the *Microsoft SQL Server Spatial Library* [5] (SQL Spatial Library, for brevity) has been designed and implemented to address the requirements of the second set of workloads discussed above. SQL Spatial Library provides an easy to use, robust, and high performance environment for persisting and analyzing spatial data. SQL Spatial Library provides data type support for point, line and polygon objects. SQL Spatial Library adheres to the Open Geospatial Consortium Simple Feature Access specification [6], and is provided as part of the SQL Server Types Library.

In this demo, we highlight the ongoing effort at Microsoft to investigate the intersection zone of the previously mentioned workloads. In other words, we focus on the application domains of a streaming nature, where the incoming stream data contains spatial information. Traffic management, vehicle re-routing, and enhanced 911 emergency services are example applications that fall in the set of spatial and stream-oriented workloads. This demo

ACM GIS'10, November 2-5, 2010, San Jose, CA, USA.
Copyright 2010 ACM ISBN 978-1-4503-0428-3/10/11…$10.00.

highlights the value of having a full-fledged relational data streaming engine that is capable of executing queries with a standard set of operators (e.g., filter, project, joins, group-by, etc). Further, an extensible system that integrates the SQL Spatial library into the query execution pipeline is used to perform spatial-oriented operations over the spatial attributes of the incoming data streams. Various analytics are demonstrated to show how spatio-temporal stream processing can help make the right decision in real time.

2. DEMO DESCRIPTION



Shuttle Visualizer
Number of Shuttle Signals
Number of Shuttles
2
5
Queries
Shuttle GPS signal received
Shuttle 504
Nearest Shuttle w.r.t. euclidean distance
Nearest Shuttle w.r.t. user defined operator
Application Time =5/24/2010 2:47:14 PM
% idle shuttles
©2010 Navteq © 2010 Microsoft

☑ All shuttle locations
☑ Track shuttle 504
☑ Count shuttle signals at building 34
☑ Count shuttles at building 34
☑ Find nearest shuttle
☑ Find nearest shuttle (userdefined)
☑ Percentage Of Idle Shuttles
☑ Percentage Of Idle Shuttles/region

Figure 1. The Microsoft shuttle service visualizer.

The demo scenario is based on the shuttle service provided at the Microsoft main campus in Redmond, Washington, USA. StreamInsight has been deployed to monitor the location, speed and direction of shuttles as they move across the campus. The goal of the demo is to illustrate the usability of StreamInsight in deploying simple continuous queries as well as complex analytics on streams of spatio-temporal data. The proposed demo features a user interface that connects to *Bing Maps* and retrieves a map of the region to be monitored (Figure 1). Each shuttle is equipped with a GPS system and a transmitter that feeds StreamInsight with a continuous stream of the shuttle information. The shuttle information includes the shuttle id, its position (latitude and longitude), its speed and direction. The stream of shuttle information is fed into StreamInsight, where several continuous queries are registered to continuously process the incoming stream readings. Examples of these queries are summarized as follows:

The *all-shuttle-location* query is a simple query that reads all shuttle signals and visualizes the shuttle locations as they move around the campus. The *track-a-shuttle* query tracks a single shuttle as it moves from one location to another. This query helps notify passengers through short messages on their mobile phones as the shuttle approaches their pick up location. More interestingly, the *count-shuttle-signal* query at a specific building indicates how many shuttle signals have been received in the parking zone of that building. Thanks to the Microsoft SQL Server Spatial Library [5], the intersection and containment operations are invoked during the course of the continuous query execution. These operations identify the existence of a shuttle in the spatial region that denotes the building parking zone. Note that a shuttle that parks in front of a building generates multiple stream readings per minute. The *count-shuttle* query counts how many *unique* shuttles are in front of the building by grouping the received readings by the shuttle id. The *count-shuttle-signal* and *count-shuttle* queries demonstrate the strength of the system's underlying relational algebra and the ability to run aggregation, join, and group-by queries over the incoming data streams.

To help scheduling the pick-ups and drop-offs of passengers, the *find-nearest-shuttle* query returns the nearest vehicle to a building. Note that the query can simply use the shortest route as a notion of proximity. However, in practice, complex logic that has been built over years of domain expertise in transportation systems is applied to measure the closet (or the cheapest) shuttle to pick up a passenger. This logic takes into account the number of vacant seats in the shuttle, the shuttle size, its fuel consumption, and the overall schedule of multiple passengers travelling simultaneously. Thanks to the extensibility framework of StreamInsight, domain experts get the power and flexibility to write their own user defined modules and to integrate these modules with the continuous query pipeline. This query demonstrates the use of *Bing Maps* to pick up the shortest route as well as the extensibility of StreamInsight to incorporate user's logic as part of the query.

Finally, this demo presents several analytics that help improve the efficiency of transportation systems. More precisely, StreamInsight help decision makers respond to anomalies and unexpected conditions (e.g., accidents or traffic jams) in real time. The *percentage-of-idle-shuttle* query measures the percentage of idle shuttles over multiple spatial granularities. It becomes apparent that shuttles in some regions are over-committed while shuttles in other regions remain under-utilized. This pattern is common in situations where public events are held without a priori knowledge or proper estimation of the expected attendance. Figure 1 illustrates the shuttle utilization per region using a color scheme. StreamInsight helps redistribute the shuttles across campus to rebalance the load and to minimize the wait time per passenger. Once the average wait time is restored to its normal level, the system may decide to keep the shuttles in an idle mode to minimize the shuttles running cost. Real-time spatio-temporal analytics has proven to be crucial to traffic management where immediate response to changes in traffic conditions is required.

3. REFERENCES

[1] Mohamed Ali et al.: Microsoft CEP Server and Online Behavioral Targeting. In VLDB 2009.

[2] B. Chandramouli, J. Goldstein, and D. Maier. On-the-fly Progress Detection in Iterative Stream Queries. In VLDB, 2009.

[3] Roger S. Barga, Jonathan Goldstein, Mohamed Ali, and Mingsheng Hong. Consistent Streaming Through Time: A Vision for Event Stream Processing. In CIDR, 2007.

[4] Jonathan Goldstein, Mingsheng Hong, Mohamed Ali, and Roger Barga. Consistency Sensitive Streaming Operators in CEDR. Technical Report, MSR-TR-2007-158, Microsoft Research, Dec 2007.

[5] SQL Server Spatial Libraries. *http://www.microsoft.com/sqlserver/2008/en/us/spatial-data.aspx.*

[6] Open Geospatial Consortium. *http://www.opengeospatial.org*

[7] Paolo Pialorsi, Marco Russo. *Programming Microsoft LINQ*, Microsoft Press, May 2008.