Real-time spatio-temporal data mining with the "streamonas" data stream management system

P. A. Michael & D. Stott Parker
Computer Science Department, University of California Los Angeles, USA

Abstract

Data Stream Management Systems (DSMSs) have not yet reached a mature enough stage to effectively run data mining algorithms, as they still face challenges within the streaming environment. Streamonas DSMS, as presented in a recent publication, is the first DSMS to reach the maximum level of difficulty supported by the Linear Road Benchmark which is 10 Expressways. The powerful engine of Streamonas can manage an input stream of 20,368 tuples/second with an average query latency of 0.000026 seconds, 192,307 times faster when compared to the 5 seconds maximum query latency the benchmark allows. The on-line data mining over streams presented in this work, is the first effort to apply spatio-temporal data mining algorithms on the Streamonas DSMS system. Dynamic clustering of spatio-temporal subsequences in real-time has been performed successfully, within the large space, high bandwidth, heavy load linear road benchmark streaming platform. Dynamic clustering queries have been expressed in a novel SQL-like language, which we name Streamonas-SQL.

Keywords: real-time, data mining, spatio-temporal, dynamic clustering, pattern matching, streamonas, streamonas-SQL, Linear Road Benchmark, query latency, throughput, semantic space.

1 Introduction

This work uses as a platform the streaming environment of the Linear Road Benchmark (LRB) [1, 2, 12–15] at the maximum level of its difficulty, i.e. 10 XWays. The high-performance results of the Streamonas Data Stream

WIT Transactions on Information and Communication Technologies, Vol 42, © 2009 WIT Press
www.witpress.com, ISSN 1743-3517 (on-line)
doi:10.2495/DATA090121

Management System (DSMS) on the LRB have been published in [1, 2]. The engine of Streamonas while tested at the maximum level of difficulty the Linear Road Benchmark supports (10 XWays) managed an input stream of 20,368 tuples/second with an average query latency of 0.000026 seconds, 192,307 times faster when compared to the 5 seconds maximum query latency the benchmark allows. The results of this work, presented in the following sections, demonstrate that Streamonas can effectively perform large space, high bandwidth, heavy load dynamic clustering of spatio-temporal subsequences in real-time within the LRB streaming platform, at an average query latency of 0.000027 seconds. The dynamic clustering querying is expressed in a novel SQL-like language with the name Streamonas-SQL.

2 Previous work

Representation of the structure of an event sequence with Non-deterministic Finite Automata (NFA) has been used by [9, 11]. NFA are also used by the SASE event language [3, 4] in order to read query-specific event sequences efficiently from continuously arriving events. Other event systems are based on fixed data structures such as trees [8], finite automata [9] and Petri nets [10].

The researchers in [3] achieve excellent performance results based on their own event generator platform. In this work we present our first efforts to apply on-line spatio-temporal data mining over streams on the Streamonas DSMS platform within the streaming environment of the Linear Road Benchmark along with performance evaluation. For pattern matching we have used as similarity metric the correlation coefficient between a given pattern and the incoming streaming information. While the correlation coefficient is a simple widely used methodology for off-line similarity measurement, the work [6] referred by [5], analyze that for Complex Event Processing and pattern matching in sequences of rows, the correlation aggregate is illegal when applied on groups of rows, as the two groups, depending on their respective filtering predicates, may have different number of rows. In [19] it is emphasized that extensive bibliography exists on spatio-temporal databases. The researchers in [24] introduce a stream processing paradigm of functional transformations (transducers) on streams. Aurora [23] has introduced an architecture based on a data-flow model and an algebra with a set of operators to express its stream processing requirements. The project STREAM [25] has built a Data Stream Management System prototype which supports a large class of declarative continuous queries over continuous and traditionally stored data sets.

3 Spatio-temporal data-mining on streamonas

3.1 Architectural outline

As extensively analyzed in [1, 2] the Streamonas architecture follows a two layer architecture where streaming is decoupled from querying (Figure 1). The incoming serial data stream is decomposed into specially designed data-

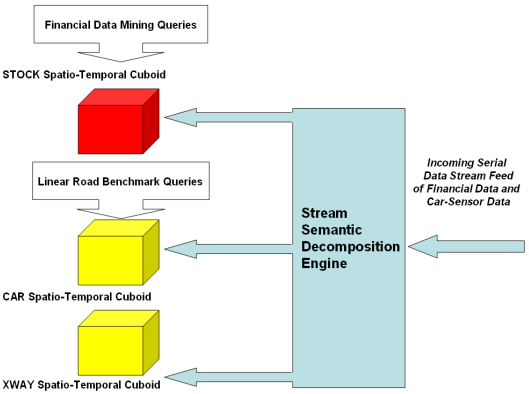


Figure 1: Architecture and database design of an integrated streamonas platform for simultaneous data mining queries, with different semantics, on financial and linear road benchmark streams.

structures, which store uniquely identified temporal sequences with the name Atomic Streams (ASTs). Special data structures called Spatio-Temporal Cuboids (ST-Cuboids) isolate and index the atomic streams. The incoming serial stream consists of data from the stock market, as also from car sensors. The Stream Semantic Decomposition Engine, decomposes the data into semantically meaningful temporal sequences (Atomic Streams), which populate the respective spatio-temporal cuboids.

3.2 Database design

ST-Cuboids have a role as fundamental for our DSMS as the role of a Relation in a Relational DBMS. Figure 1, presents the spatio-temporal cuboids of the database supporting simultaneously the Linear Road Benchmark queries as also the Financial Data Mining queries. We would like to emphasize the database centric nature of Streamonas which allows the integration of multiple streaming sources with different semantics within the same database model. This database centric logic, allows the reusability of the streaming information, by allowing queries of different applications to run in parallel and access the underlying database.

In an equivalent manner as in the Relational Databases theory, the ST-Cuboid Database Schema is the collection of schemas for the ST-Cuboids in the database. In order to support the data mining application within the streaming environment of the Linear Road Benchmark we define the following ST-Cuboid Database Schema:

STOCK (Stock_Id:int, Price(t):AST(10))
CAR (Car_Id:int, Speed(t):AST(6), Segment(t):AST(6))
XWAY (XWay:int, Segment:int, Direction:int, Lane:int,
AVGS:AST(6), LAV:AST(6))

The ST-Cuboid *STOCK* stores the financial atomic streams while the ST-Cuboids CAR and XWAY store atomic streams generated from the Linear Road Benchmark simulating generation of data from car sensors.

The ST-Cuboid schemas include attributes of type Atomic Stream as also their historical span in parentheses (e.g. historical span is 10 for the atomic stream STOCK.Price(t), while it is 6 for the atomic stream XWAY.AVGS(t)).

The attribute Stock_Id describes the unique identification of each stock. The attribute STOCK.Price(t) of type atomic stream, models the evolving price of each stock. The attribute CAR.Car_Id models the static over time unique identification of each car, while CAR. Speed(t) of type atomic stream, models the evolving speed of each car. CAR. Segment(t) models the evolving segment (location) of each car on an XWay. In an equivalent manner the attributes of the ST-Cuboid XWAY, XWay, Segment, Direction and Lane are defined. The attributes AVGS and LAV of the ST-Cuboid XWAY model statistical information based on the specifications of the LRB.

3.3 Real-time clustering of spatio-temporal subsequences based on the correlation similarity metric

As also emphasized in [17], central to all goals of cluster analysis is the notion of the degree of similarity. For real-time cluster analysis of spatio-temporal patterns on the Streamonas Data Stream Management System, we have used correlation as a similarity metric [17]. More specifically we have used the correlation coefficient computational version analyzed in [18] (Eq. (1)):

$$r = \frac{\Sigma xy - \frac{\Sigma x \, \Sigma y}{n}}{\sqrt{(\Sigma x^2) - \frac{(\Sigma x)^2}{n}} \sqrt{(\Sigma y^2) - \frac{(\Sigma y)^2}{n}}} \tag{1}$$

3.4 Simplicity of querying with streamonas-SQL

Querying on Streamonas is expressed with an SQL-like language with the name Streamonas-SQL. Figure 2 shows the SQL statement expressing the query used during our experiments. While we shall extensively analyze Streamonas-SQL in a future publication, in this work we provide a general description of the Streamonas-SQL statement presented in Figure 2, emphasizing its simplicity. Streamonas-SQL manages tuples of atomic streams, rather than tuples of non-temporal information. The SELECT clause of the statement, returns the atomic streams that satisfy the predicate over time, the FROM clause of the statement refers to the ST-Cuboid STOCK and the predicate Correlation_function (Price, Pattern1) > 0.95, receives as arguments two atomic streams (the stock price and the pattern) and returns their correlation based on equation Eq (1). The query is being re-evaluated upon each arrival of a new tuple. Efficiency is achieved by evaluating only the delta results of the query within the scope of the new arrival. It is important to mention that Streamonas-SQL differs from other research

works such as [3, 4, 6] as it does not include the pattern definition within the Query Expression. The object-oriented design of Streamonas allows the pattern to be stored as a data object in the database (as an atomic stream) and encapsulates the processing logic of each atomic stream in a respective function which we name Atomic Temporal Function (ATF). Correlation_function (Price, Pattern) is an example of an ATF. The query in figure 2 dynamically evaluates the members of a cluster of spatio-temporal subsequences for any stock in the database, where the correlation coefficient of the subsequence with the pattern is larger than 0.95. As shown in the experimental results, the population of the cluster with new members is performed with an average query latency of 0.000016 seconds.

4 Experimental results

4.1 Preparation of the serial data stream and testbed

We have used stock market data from [16]. Real data from seven indexes (NASDAQ 100, SP 500, Dow Jones, Diamonds, QQQQ and Spyder) have been multiplexed together and stored in a single file of 122.7MB. This single file has been used as an initial stress-test of the system at high-bandwidth. The tuples were also enumerated for reference purposes during the experiment. A number of 6,385,295 tuples were included in the file. A second file was also created by multiplexing the stock data with the car sensor data of the Linear Road Benchmark at a level of 10XWays. This second file has a size of 6.3GB consisting of 126,717,975 tuples. The Linear Road Benchmark environment was used in order to test the effectiveness of Streamonas under heavy load. A single PC was used having as CPU a Pentium4 processor running at 3GHz with 4GB RAM and a 100GB hard disk. During the second experiment we have used the same data driver tool of the LRB as in [1] and [2].

4.2 High bandwidth dynamic clustering of spatio-temporal subsequences in real-time with standalone performance evaluation on streamonas

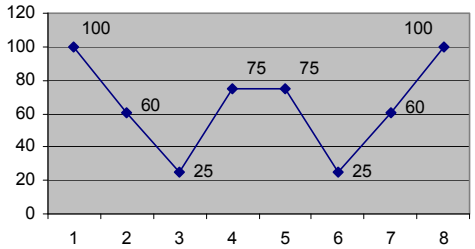
Our first experiment measured the performance of the Streamonas DSMS when applying dynamic clustering on a number of 7 stocks at high bandwidth.

A "W" shaped pattern (Figure 2) was chosen as the centre of the cluster. The cluster was being populated in real-time with a number of spatio-temporal patterns from any stock that would satisfy the predicate. Two members of the cluster and their respective correlations are presented in Figure 2. By increasing the threshold to $r > 0.99$, two only spatio-temporal patterns satisfied the predicate. The bandwidth of the experiment was 63,461 tuples/second while the average query latency (of the sample points) was 0.000016 seconds (Figure 3a). The first experiment was performed by reading data directly from disk and includes the overhead from the stream semantic decomposition. A second experiment was performed by buffering decomposed stock data in memory and then streaming it into the system. The clustering was performed at an average

Streamonas-SQL Query:

```
SELECT *
FROM Stock
WHERE
Correlation_function
( Price, Pattern1 ) > 0.95
```

Cluster Center: Pattern 1

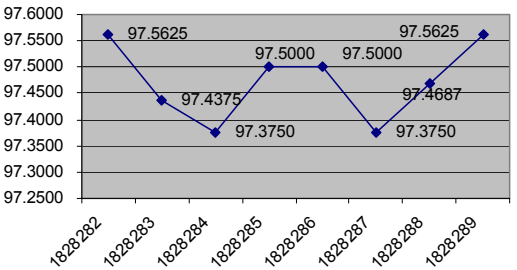


Dynamic Cluster for r>0.95

Average Bandwidth of streaming data: 63,461 tuples/sec

Average Query Latency 0.000016 seconds

`Stock: 4 Correlation Coef.: 0.991831 Tick#: 1828289`



`Stock: 7 Correlation Coef.: 0.954671 Tick#: 1157213`

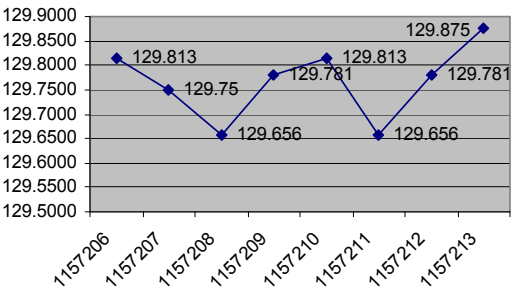


Figure 2: A streamonas-SQL query developing dynamic clusters of spatio-temporal subsequences in real-time.

bandwidth of 225,371 tuples/second with an average query latency (of the sampled points) 0.000004 seconds (measurements do not include the Stream Semantic Decomposition overhead) (Figure 3b).

4.3 Large space - high bandwidth – heavy load dynamic clustering of spatio-temporal subsequences in real-time within the linear road benchmark streaming platform

The experiments in section 4.2 were performed over a relatively small number of stocks (7 stocks). We wanted to stress test the Streamonas system under the heavy load of the Linear Road Benchmark as described in [1, 2]. We have applied the same dynamic clustering described in 4.2 on the financial information multiplexed with the car-sensor data, stored in a 6.3GB file as described in section 4.1.

4.3.1 Dynamic clustering under the heavy load of the LRB

A first experiment measured the performance of the system within the Linear Road Benchmark environment at its maximum level of difficulty (10XWays), loaded also with the dynamic clustering queries for the financial data. The average, maximum and minimum query latencies for this experiment (based on all tuples streamed into the system) are: 0.000026 seconds, 0.109826 seconds and 0.00003 seconds respectively.

4.3.2 Large space – high bandwidth – heavy load dynamic clustering

We wanted to stress test Streamonas in a scenario where patterns are searched within a larger space than the semantic space [2] defined by the 7 stocks. For this reason we performed a second experiment where we applied dynamic clustering of spatio-temporal subsequences over the large semantic space of the speeds of the 1,373,327 cars of the LRB (10 XWays), simultaneously with the dynamic clustering of the 7 stocks (historical span was 6 for all ASTs). In an example application we wanted to identify patterns of driving behaviour (e.g. any car accelerating or decelerating based on the predefined pattern). The average, maximum and minimum query latencies for this experiment are: 0.000027 seconds, 0.188051 seconds and 0.00003 seconds respectively, evaluated based on each one of the tuples streamed into the system. The bandwidth mapping of the system during the 3hr simulation is presented in Figure 4.

5 Conclusions

We have performed Large Space (more than 1.3 million evolving entities), High Bandwidth (63,461 tuples/sec on average), Heavy Load (6.3GB of data) Dynamic Clustering of spatio-temporal subsequences in real-time on the Streamonas Data Stream Management System. The experiments were performed within the environment of the Linear Road Benchmark at its maximum level of difficulty (10 XWays). In all our experiments, Streamonas performed excellently with an average query latency of 0.000027 seconds. Dynamic clustering was expressed in the novel Streamonas-SQL language.

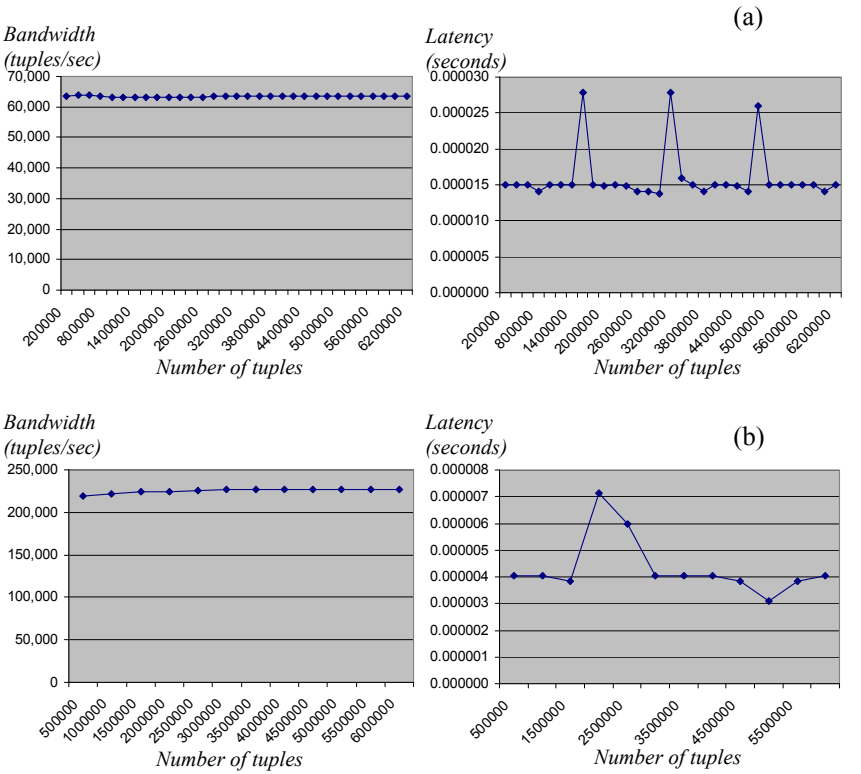


Figure 3: Average bandwidth and average latency during dynamic clustering of spatio-temporal subsequences on streamonas.

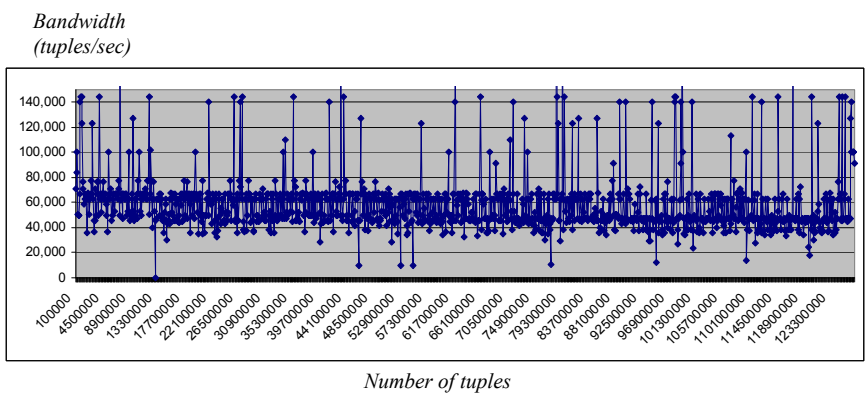


Figure 4: Bandwidth mapping during dynamic clustering of spatio-temporal subsequences on streamonas for financial data and car-sensor data within the environment of the linear road benchmark.

Acknowledgements

Panayiotis Michael would like to thank his parents, Adamos Michael and Andriani Zachariadou-Michael for their continuous support to his research work.

References

[1] P. A. Michael, D. S. Parker. "Architectural Principles of the Streamonas Data Stream Management System and Performance Evaluation based on the Linear Road Benchmark". *In Proc. of the 2008 International Conference on Computer Science and Software Engineering (2008 CSSE)*, IEEE, Wuhan, China, December 2008.

[2] P. A. Michael, D. S. Parker. "The Semantic Space-time models of the Streamonas Data Stream Management System". *In Proc. of the 2009 World Congress on Computer Science and Information Engineering (2009 CSIE)*, IEEE, Los Angeles / Anaheim, USA, March 2009 (to appear).

[3] E. Wu, Y. Diao, S. Rizvi. "High-Performance Complex Event Processing over Streams". *In Proc. of the 2006 ACM SIGMOD Intl. Conf. on Management of Data,* Chicago, Illinois, USA, 2006.

[4] D. Gyllstrom, Y. Diao, E. Wu, P. Stahlberg, H. Chae, G. Anderson. "SASE: Complex Event Processing over Streams", *CIDR*, 2007.

[5] N. Jain, J. Gehrke, J. Widom, H. Balakrishnan, U. Cetintemel, M. Cherniack, R. Tibbetts, S. Zdonik. "Towards a Streaming SQL Standard". *In Proc. of the 34th VLDB Conference*, Auckland, New Zealand, 2008.

[6] Anonymous: Pattern Matching in Sequences of Rows, SQL standard proposal, http://asktom.oracle.com/tkyte/row-pattern-recogniton-11-public.pdf, March, 2007.

[7] S. Reza, C. Zaniolo, A. Zarkesh, J. Adibi: Expressing and optimizing sequence queries in database systems. ACM *Transactions Database Systems. 29 (2): 282-318*, 2004.

[8] S. Chakravarthy, V. Krishnaprasad, E. Anwar, S. Kim. "Composite events for active databases: Semantics, contexts and detection". *In VLDB*, 1994.

[9] N.H. Gehani, H.V. Jagadish, O. Shmueli. "Composite event specification in active databases: Model and implementation". *In VLDB*, 1992.

[10] S. Gatziu and K.R. Dittrich. Events in an active object-oriented database system. *In Proc. of the 1st International Conference on Rules in Database Systems*, 1993.

[11] Y. Diao, M. Altinel, H. Zhang, M.J. Franklin, P.M. Fisher. "Path sharing and predicate evaluation for high-performance XML filtering". *TODS, 28(4), 467-516*, December 2003.

[12] A. Arasu, M. Cherniack, E. Galvez, D. Maier, A. S. Maskey, E. Ryvkina, M. Stonebraker, R. Tibbets. "Linear Road: A Stream Data Management Benchmark". *In Proc. of the 30th Intl. Conference on Very Large Data Bases*, 2004.

[13] Linear Road Benchmark, http://www.cs.brandeis.edu/~linearroad/

[14] N. Jain, L. Amini, H. Andrade, R. King, Y. park, P. Selo, C. Venkatramani. "Design, Implementation, and Evaluation of the Linear Road Benchmark on the Stream Processing Core". *In Proc. of the 2006 ACM SIGMOD Intl. Conference on Management of Data*, 2006.
[15] M. Svensson. "Benchmarking the performance of a data stream management system". *MSc thesis report*, Uppsala University, November 2007.
[16] Price-Data, http://www.price-data.com/
[17] T. Hastie, R. Tibshirani, J. Friedman. *The Elements of Statistical Learning. Data Mining, Inference, and Prediction.* Springer, pp. 453-457, 2001.
[18] S. Bernstein, R. Bernstein. *Elements of Statistics II*. McGraw Hill, pp. 344-349, 1999.
[19] C. Zaniolo, S. Ceri, C. Faloutsos, R.T. Snodgrass, V. S. Subrahmanian and R. Zicari. *"Advanced Database Systems"*. Morgan Kaufmann, pp. 100-124, 1997.
[20] J. Li, K. Tufte, V. Shkapenyuk, V. Papadimos, T. Johnson, and D. Maier. "Out-of-Order Processing: A New Architecture for High-Performance Stream Systems". *In Proc. of the 34th VLDB Conference*, Auckland, New Zealand, 2008.
[21] D. S. Parker, R. R. Muntz, H. L. Chau. "The Tangram stream query processing system". *In Proc. of the Fifth International Conference on Data Engineering*, 1989.
[22] H. Thakkar, B. Mozafari, and C. Zaniolo. "Designing an Inductive Data Stream Management System: the Stream Mill Experience". *In Proc. of the Second International Workshop on Scalable Stream Processing Systems*, Nantes, France, 2008.
[23] D.J. Abadi, D. Carney, U. Cetintemel, M. Cherniack, C. Convey, S. Lee, M. Stonebraker, N. Tatbul, S. Zdonik. "Aurora: a new model and architecture for data stream management". *The VLDB Journal*, 2003.
[24] D. Stott Parker. "Stream Data Analysis in Prolog". In L. Sterling, ed., *The Practice of Prolog*, Cambridge, MA: MIT Press, 1990 (abstract available at: http://www.cs.ucla.edu/~stott/sdb/abstracts.html).
[25] A. Arasu, B. Babckok, M. Datar, K. Ito, I. Nishizawa, J. Rosenstein and J. Widom. STREAM: "The Stanford Stream Data Manager". *In Proc. of the 2003 ACM SIGMOD Intl. Conf. on Management of Data*, 2003.