Query optimization in data stream processing

Optimizarea interogărilor în procesarea fluxurilor de date

Abstract

Sabina Surdu
Supervisor: Prof. univ. Dr. Leon Țâmbulea

Faculty of Mathematics and Computer Science
Babeș-Bolyai University

Cluj-Napoca

2012

The thesis contains the following chapters[1]:

[1]We don't expand the sections in this abstract.

Keywords: data streams, continuous queries, data stream management systems, query optimization, reducing resource usage, performance optimization, pervasive applications, pervasive computing, heterogeneous data management

Publications related to this thesis

The results of our research and the original contributions presented in the thesis were published in journals or proceedings of the international conferences we attended (one of these papers is accepted for publication):

- **Sabina Surdu** and Vasile-Marian Scuturici, Addressing resource usage in stream processing systems: sizing window effect, IDEAS'11 Proceedings - 15th International Database Engineering & Applications Symposium, pages 247-248, Lisbon, 2011. Excellence in Research for Australia (ERA) indexed this symposium in the B category in its most recent hierarchy from 2010 [Era10]. (URL of the article: `http://dl.acm.org/citation.cfm?id=2076623.2076658&coll=DL&dl=ACM&CFID=63572418&CFTOKEN=57655636`)

- Yann Gripay, Frédérique Laforest, François Lesueur, Nicolas Lumineau, Jean-Marc Petit, Vasile-Marian Scuturici, Samir Sebahi and **Sabina Surdu**, ColisTrack: Testbed for a Pervasive Environment Management System, EDBT 2012 - The 15th International Conference on Extending Database Technology, Berlin, 2012. The conference is indexed in the A category by ERA in 2010 [Era10]. (URL accepted papers: `http://edbticdt2012.dima.tu-berlin.de/program/EDBT-papers/`)

- **Sabina Surdu**, A new framework for evaluating performance in data stream monitoring applications with StreamInsight: StreamEval, MaCS 2012 - Booklet of abstracts from The 9th Joint Conference on Mathematics and Computer Science (international conference), page 92, Siófok, 2012. (URL Booklet of abstracts: `http://macs.elte.hu/downloads/abstracts/booklet.pdf`)

- **Sabina Surdu**, A New Architecture Supporting The Sizing Window Effect With StreamInsight, Studia Universitatis Babeș-Bolyai Series Informatica,

LVI(4):111-120, 2011. The journal is indexed in the B+ category (BDI indexed) by CNCSIS in 2011 [CNC11].

- **Sabina Surdu**, Data stream management systems: a response to large scale scientific data requirements, Annals of the University of Craiova, Mathematics and Computer Science Series, 38(3):66-75, 2011. The journal is indexed in the B+ category (BDI indexed) by CNCSIS in 2011 [CNC11].

- **Sabina Surdu**, A new architecture for load shedding on data streams with StreamInsight: StreamShedder, University of Pitești Scientific Bulletin, Series Electronics and Computers Science, 11(2):57-64, 2011. The journal is indexed in the B+ category (BDI indexed) by CNCSIS in 2011 [CNC11].

- **Sabina Surdu**, A technique for constructing training sets in data stream mining: kSiEved Window Training Set, MDIS 2011 - Proceedings of The Second International Conference on Modelling and Development of Intelligent Systems, pages 180-191, Sibiu, 2011. (URL conference proceedings: `http://conferences.ulbsibiu.ro/mdis/2011/Doc/Proceeding_mdis2011.pdf`)

- **Sabina Surdu**, Towards an education monitoring platform based on data stream processing, Education and Creativity for a Knowledge Society International Conference, The fifth edition - Computer Science Section, pages 61-66, București, 2011. (URL conference program: `http://www.utm.ro/conferinta_2011/files/program_conferinta_2011.pdf`)

- **Sabina Surdu**, Online political communication, Interdisciplinary New Media Studies Conference Proceedings (international conference), pages 55-58, Cluj-Napoca, 2009. (URL conference program: `http://journalism.polito.ubbcluj.ro/inms/wp-content/uploads/2010/07/INMS_conference_prog.pdf`)

The following papers are under evaluation or about to be submitted to conferences or journals:

- **Sabina Surdu**, Yann Gripay, Jean-Marc Petit and Vasile-Marian Scuturici, paper sent to an A* international conference 2012, under evaluation.

- **Sabina Surdu**, A new framework for evaluating performance in data stream monitoring applications with StreamInsight: StreamEval, Annales Universitatis Scientiarum Budapestinensis de Rolando Eötvös Nominatae - Sectio Computatorica, 2012, under evaluation. The extended paper was sent together with the abstract having the same title, accepted at a previously mentioned international conference.

- **Sabina Surdu** și Vasile-Marian Scuturici, Assessing performance in data stream processing, to be submitted to IDEAS 2012 - The 16th International Database Engineering & Applications Symposium, Prague, 2012. The symposium is indexed in the B category by ERA in 2010 [Era10].

- **Sabina Surdu**, Data stream processing: traditional vs. dedicated systems (SQL Server vs. StreamInsight), to be submitted to Studia Universitatis Babeș-Bolyai Series Informatica. The journal is indexed in the B+ category (BDI indexed) by CNCSIS in 2011 [CNC11].

1 Thesis structure

In Chapter 1 we briefly describe the general problem of data stream processing using continuous queries and we present the ubiquitous network society, characterized by heterogeneous data processing in pervasive environments and applications. We present the challenge of optimizing queries when processing data streams from pervasive environments and we highlight the two main research directions we follow in this thesis: optimizing resource usage when processing queries on data streams and heterogeneous data management in pervasive application development. These applications contain static data, streams and functionalities [GLP10]. We discuss the original contributions from this thesis and we mention a list of papers, published in journals or presented at international conferences and published in proceedings. We also mention the papers we sent to conferences or journals, which are under evaluation or accepted for publication.

In Chapter 2 we present the state of the art in data stream processing. We introduce key data stream processing systems and discuss alternative approaches for query optimization, mostly oriented towards reducing system resource usage. We provide a comparative view on the discussed approaches.

In Chapter 3 we describe the techniques we propose in our thesis for optimizing resource usage in data stream processing. We analyse the sizing window effect, in order to determine an optimal window size for a query, so that resource usage is minimal and accuracy constraints are met. We discuss the kSiEved Window Training Set technique, a strategy for building training sets in data mining on data streams. This strategy aims at saving system resources and still meeting accuracy requirements.

In Chapter 4 we discuss the resource-aware architectures we designed, oriented towards reducing resource usage when processing continuous queries. StreamShedder and WindowSized are two such architectures for a Data Stream Management System, based on a commercial system for data stream processing. We briefly describe StreamEval, an application that evaluates performance variations when some

conditions in the environment change. We also discuss SCIPE and InstantSchool-Know, two proposals we made for Data Stream Management Systems for specific application domains.

In Chapter 5 we move on to pervasive application development. We present the testbed we developed as part of a team, at LIRIS, INSA Lyon, for a system that manages pervasive environments, based on a scenario designed for such environments, in a medical context. This testbed can be used to assess pervasive application development. We present the design of a data-oriented pervasive application, using the SoCQ system [GFLP09], thereby homogeneously handling the data in the pervasive environment. We describe the application we implemented, which enables a user to write continuous queries, combining heterogeneous data.

In Chapter 6 we evaluate the development of pervasive applications, using multiple systems. We describe the proposed benchmark and conduct an experimental study. The results of the research presented in this chapter are part of a paper we submitted to an international conference and that is currently under evaluation.

In Chapter 7 we summarize the results we obtained on the chosen research directions: optimizing resource usage when processing queries on data streams and managing heterogeneous data in pervasive application development. We briefly discuss the techniques we developed for optimizing resource usage in data stream processing and the resource-aware architectures we designed to save resource usage when processing continuous queries on data streams. We touch on the testbed for pervasive application development and the benchmark for evaluating these applications. We describe future research directions driven by the results we obtained.

2 Processing data streams in pervasive environments

In the last couple of years we witnessed a shift in the traditional data processing paradigm, from the classical model, characterized by static data, to a dynamic model, which encompasses data characterized by high dynamics. In a growing number of fields, information takes the form of continuous streams. These are potentially unending sequences of data, which cannot be efficiently handled by traditional DBMSs [ACC+03]. Research teams from the academic world have developed prototypes to manage and process data streams. These are called Data Stream Management Systems (DSMSs). Industrial players also designed and launched fully featured DSMSs (a recent example is StreamInsight, released by Microsoft [KDA+10]).

In traditional databases, data has a static nature. It takes the form of finite, stored data sets, which are queried when necessary [ABB+04]. On the other hand, data streams are dynamic by definition. They are not permanently stored in a system. A query in this context executes in a perpetual manner on temporary data, which arrives at the system, is processed and in the end is discarded. A DSMS can execute a great number of complex continuous queries [ABB+03], which take into account multiple data streams. The stream data rate can greatly vary in time. Limited system resources must cope with these issues, when data processing must consider the temporal nature of the data.

In the applications from the the *ubiquitous network society* [Mur09], the user not only interacts with other users, but it also relates to objects from the environment, equipped with computational-enabled devices [Uni05]. In this context, data streams coexist with data modelled in different manners. Systems that manage such environments must consider static data, functionalities and data streams; pervasive environments are composed of such ingredients and are used to model the surrounding reality [GLL+12]. The integration of static data, data streams and functionalities querying capabilities in a unified, declarative model, opens other directions for query optimization in this new context, but similar to those found in traditional databases, based on SQL-like languages [Gri09].

Data streams and pervasive applications developed in the context of pervasive environments are the newcomers in the ubiquitous network society scenarios. Efficient resource management when processing data streams and the smooth development of pervasive applications are the necessary prerequisites in order to achieve the ubiquitous network society.

In this abstract we briefly describe the most important original contributions we present in our thesis.

3 Problem statement

In this thesis we investigate the general problem of query optimization, in the context of data stream processing in pervasive environments. We identify to main research directions, materialized in our published papers, listed in the preamble of this abstract:

- optimizing resource usage when processing queries on data streams;
- investigating heterogeneous data management in pervasive application development.

4 Optimizing resource usage when processing queries on data streams

One of the main problems that stream processing system designers are dealing with today is intensive resource usage. A system with limited resources needs to be able to handle a huge number of data sources, considerable data volumes, fast data rates and unpredictable spikes in data (like we show in [SS11]). The number of data sources, the data rates and the data volumes are constantly increasing. The data distribution can be variable and the system needs to be able to execute multiple complex queries, in a continuous manner [ABB+03]. In this context, we raise the following questions: how can the system function properly under these circumstances and how can the system performance be assessed?

In our thesis we describe the novel solutions we proposed to tackle this problem, presented on two research directions: (1) resource usage optimizing techniques in data stream processing and (2) designing resource-aware architectures for data stream processing, oriented towards saving system resources.

We enumerate the original contributions from our thesis, which aim at reducing resource usage when processing data streams.

4.1 The sizing window effect

We develop the sizing window effect, an approach that aims at optimizing the resource usage (memory and CPU time), by computing an optimal window size for a given continuous query. We intend to improve this technique, so that the computation of the optimal window size can be performed by the system, in an automated manner. To the best of our knowledge, this is the first study that takes into account the size of the input window in order to reduce resource usage. We don't take into account windows that are semantically significant (for instance, a query that computes the average speed of cars on a road segment, in the last five minutes, needs a sliding window of fixed size). The semantics of these windows is derived from their

temporal dimension. In our case, the semantics of the window is not connected to this parameter.

We briefly present the sizing window effect (in our thesis, we rigorously formalize the temporal domain, the data streams, the notion of query equivalence, the ideal result, the approximated result, the distance function and other concepts we use; in this abstract we concisely present them). A sliding window in this context is a contiguous portion of data from a data stream S [BBD$^+$02]. If its temporal boundaries are instants t_i and t_j, we will denote this window by $SW_{ij}(S)$.

Let Q be a query whose execution produces a stream of aggregate results over time. $t_c \in T$ is the current timestamp, where T is the chosen temporal domain. $t_i \in T$ is a timestamp that marks the starting point of a window in time and t_0 is the timestamp of the first emitted element on stream S. Initially $t_i = t_c$. We denote by $CrtTS$ the set of all timestamps from T, which will be assigned to t_c. We go through the following stages:

1. We establish an accuracy threshold ϵ. In order to obtain equivalent queries and valid answers, the difference between the ideal results and the approximated results must not exceed the accuracy threshold.

2. We compute the ideal result R_{s_c} of query Q executed on data stream S, at current time instant t_c:

$$R_{s_c} = Q(S, t_c) = Q(SW_{0c}(S), t_c), R_{s_c} \in \mathbb{R} \tag{1.1}$$

   where $SW_{0c}(S)$ is a sliding window from the data stream S, whose temporal boundaries are t_0 and t_c. We say this result is *ideal*, because it takes into account all the elements arrived on the stream until the current time instant.

3. We constantly decrease t_i. We compute the approximated results $R_{w_{c\sigma_{ic}}}$ of query Q executed on the sliding windows $SW_{ic}(S)$, at current timestamp t_c. For each temporal value t_i, the window dimension $SW_{ic}(S)$ is σ_{ic}, representing the number of temporal instants contained by the window:

$$R_{w_{c\sigma_{ic}}} = Q(SW_{ic}(S), t_c), R_{w_{c\sigma_{ic}}} \in \mathbb{R} \tag{1.2}$$

We compute the distances between the ideal result and the approximated result, using a distance function, for each value of the t_i timestamp:

$$distance_{agg_{c\sigma_{ic}}}(R_{s_c}, R_{w_{c\sigma_{ic}}}) = \mid R_{s_c} - R_{w_{c\sigma_{ic}}} \mid \tag{1.3}$$

4. We repeat steps 2 and 3 for all the values the current timestamp t_c takes from $CrtTS$.

5. After finalizing step 4 (when t_c has taken all the values from $CrtTS$), we compute the average of the distances between the approximated results and the ideal results over time, for each window dimension σ_{ic}:

$$AvgDistance(\sigma_{ic}) = \frac{\sum\limits_{t_c \in CrtTS} distance_{agg_{c\sigma_{ic}}}(R_{s_c}, R_{w_{c\sigma_{ic}}})}{|CrtTS|} \tag{1.4}$$

The optimal window dimension for query Q is the smallest window dimension for which the average distance between the approximated results and the ideal results is less than the accuracy threshold ϵ.

In our experiments, conducted on a set of aggregate queries, we use the data from the Linear Road benchmark [ACG+04]. This is simulated data, related to road traffic on expressways, which are all divided into 100 segments.

For each of the following aggregate queries we will apply the previously described tactic.

Query 1: Calculate the average number of cars per time unit which have been travelling on a given segment.

Query 2: Calculate the average speed on a given segment.

Query 3: Calculate the average toll paid by a car (on all the segments).

In the case of Query 1 (figure 1.1, which we published in [Sur11a]), we notice that the average distance between the ideal results and the approximated results is less than the accuracy threshold 1, for all the window dimensions greater than 1000 time instants.

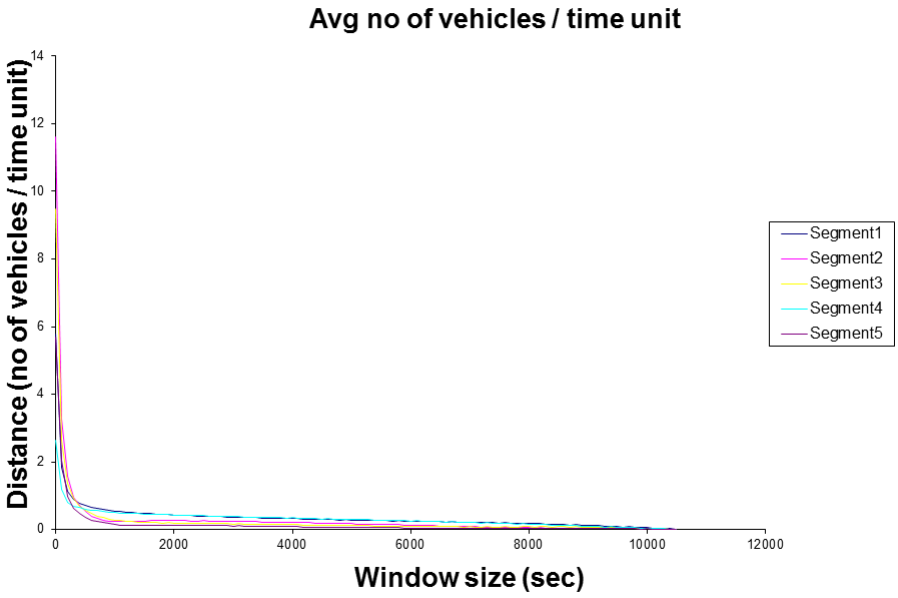


Figure 1.1: Average number of cars per time unit, separately computed for 5 segments.

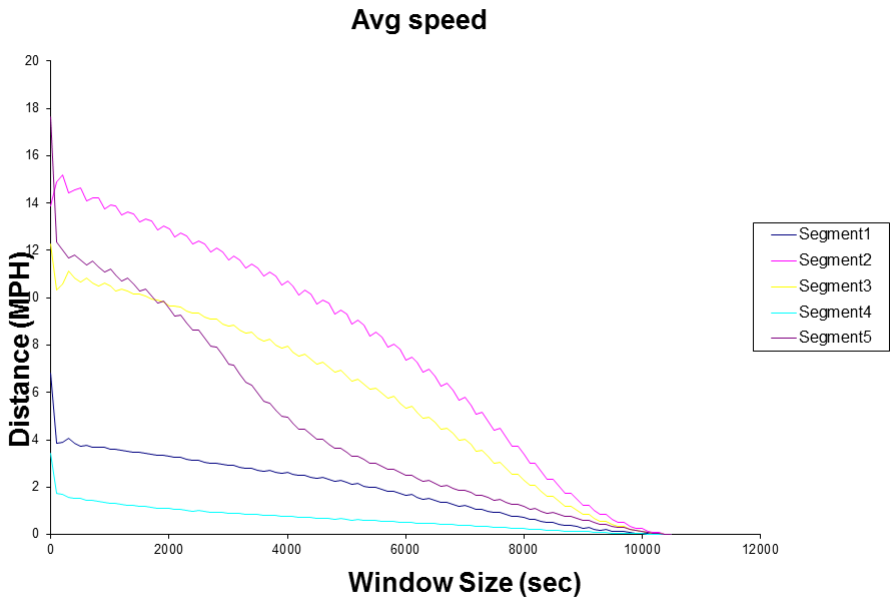


Figure 1.2: Average speed, separately computed for five segments

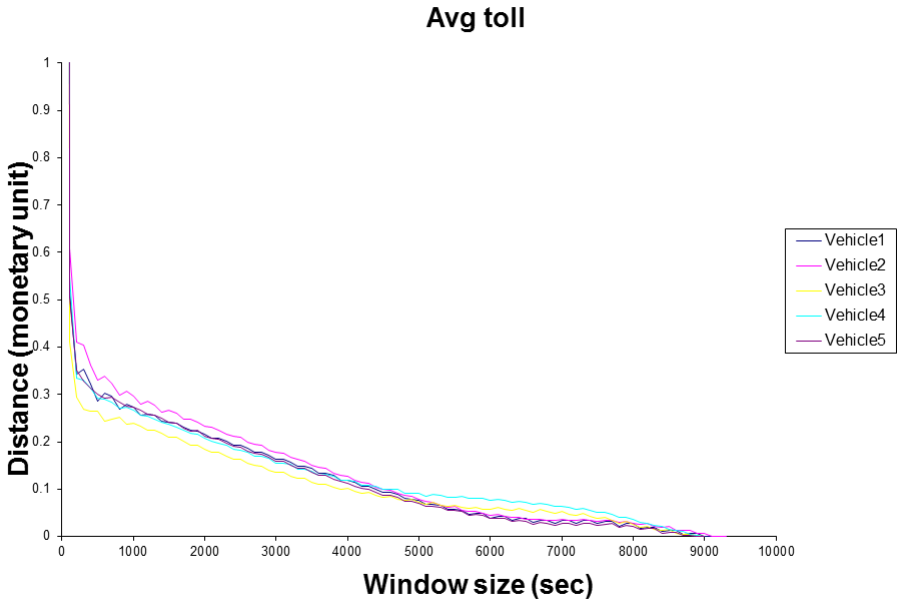


Figure 1.3: Average toll paid by 5 cars (computed separately for every car)

In the case of Query 2 (figure 1.2), the average distance between the ideal results and the approximated results is less than the accuracy threshold 1, for all the window dimensions greater than 10000 time instants.

In the case of Query 3 (figure 1.3), the average distance between the ideal results and the approximated results is less than the accuracy threshold 0.1, for all the window dimensions greater than 6000 time instants.

The administrator of an application that implements Linear Road can establish an accuracy threshold for Query 1 on the output data (from the query executed on sliding windows), so that it doesn't differ by more than 1 from the ideal result. The system can execute this query on a window of 1000 time instants. Similar constraints can be formulated for the other queries as well. The results of this research are published in [SS11].

4.2 kSiEved Window Training Set

One of the challenges encountered in data mining today is applying techniques specific to this process on continuous data streams [ZB03]. We develop a technique that takes into account system resources when constructing training sets for data mining algorithms on data streams: the kSiEved Window Training Set (kSEWT). This is the first method that "sieves" a data stream depending on some parameters, to construct training sets in this context, while meeting accuracy requirements. We define a new data model, the kSiEved model, which is based on kSiEved windows, built from sliding windows by applying functions that extract positions from a window. We rigorously define this model in our thesis.

kSEWT computes correct results, on sliding windows SW_{ic}, at every time instant t_c (we omit the stream S in the definition of these windows to simplify the notations). For each such window, kSEWT builds kSiEved windows $SEW_{ic}(k)$, based on a parameter k, which varies in time. This parameter generates a "sieve" with holes, which will "sieve" the elements from the window SW_{ic}, yielding the kSiEved window $SEW_{ic}(k)$. On this window we compute queries results as well. kSEWT estimates the accuracy of the obtained results on kSiEved windows when compared to the correct results by using a distance function. Depending on the average of the computed distances, we choose parameter k (its greatest value), for which the average of the distances from the correct result does not exceed a previously established error threshold δ. Parameter k provides the kSiEved Window Training Set, which contains all the kSiEved windows with parameter k obtained in the experiment.

We present the experimental results we obtained on a data set with a uniform distribution. By applying kSEWT we obtained the graph from figure 1.4. If we choose a threshold $\delta = 0.5$, from this graph we notice we can apply "sieves" with parameter k = 2, when building the training set. This means we discard half of the input tuples, while still meeting formulated accuracy requirements. This way we are reducing resource usage in a significant manner. This research is published in

[Sur11b].

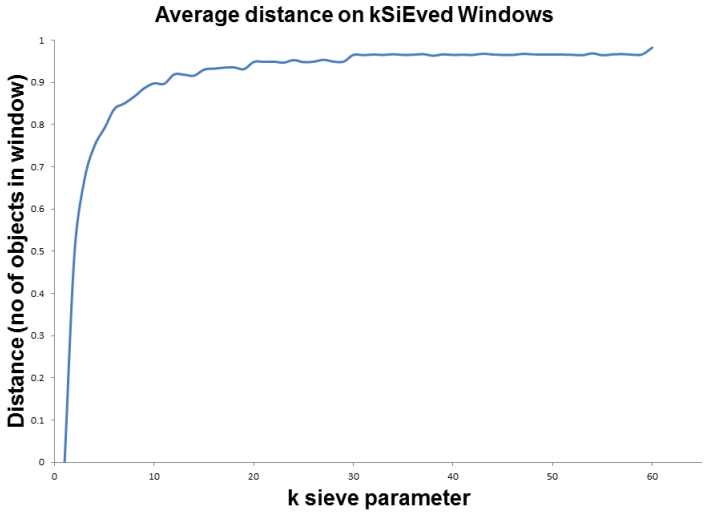


Figure 1.4: Average distance between the correct results and the results of queries executed on kSiEved windows

4.3 WindowSized

Deepening our study on the sizing window effect, we propose a new resource-aware architecture to implement this effect, using Microsoft StreamInsight [KDA+10]: WindowSized. The main contribution of this architecture is the integration of the WindowSizing module in a monitoring application developed with StreamInsight. WindowSizing interacts with the query engine, with the data sources interfaces - to modify the size of the window and with the output devices interfaces - to obtain the queries results.

Figure 1.5 depicts the main components of such an architecture. The lower level from the architecture (containing *Event sources*, *Input adapters*, *StreamInsight query engine*, *Output adapters* and *Event targets*) is taken from the architecture proposed by Microsoft, in order to implement an application with StreamInsight [SIA]. WindowSized is based on a tipical StreamInsight application design priciples, with the following elements: data sources, input adapters, continuous queries on the server,

output adapters and data consumers [GSK+09]. Our contribution is represented by the integration of the WindowSizing module in such an architecture. The obtained results are published in [Sur11a].

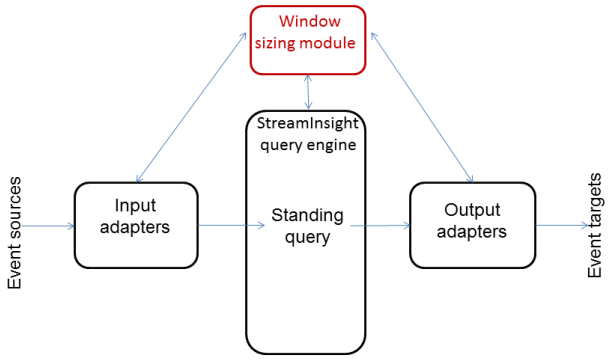


Figure 1.5: The WindowSized architecture

4.4 StreamShedder

We develop the StreamingTraffic traffic monitoring application. We propose a new architecture for such a monitoring application implemented with the StreamInsight platform [KDA+10]. We develop the load shedding [ABB+04] module StreamShedder and we recommend its integration in the architecture of StreamingTraffic. StreamShedder performs data elimination operations in a parameterized manner, taking into account system resources and query latency. The obtained architecture integrates load shedding strategies with a commercial stream processing system to obtain superior performances when processing continuous queries.

Figure 1.6 depicts the modified architecture of a monitoring application implemented with StreamInsight, which encompasses the StreamShedder module. Like in the case of WindowSized, the lower level from the architecture (with *Event sources*, *Input adapters*, *StreamInsight query engine*, *Output adapters* and *Event targets*) is taken from the architecture proposed by Microsoft, in order to implement an application with StreamInsight [SIA]. Our contribution is represented by the integration of the StreamShedder module in such an architecture. We will call this enriched

architecture StreamShedder as well, based on the module that performs the load shedding operations.

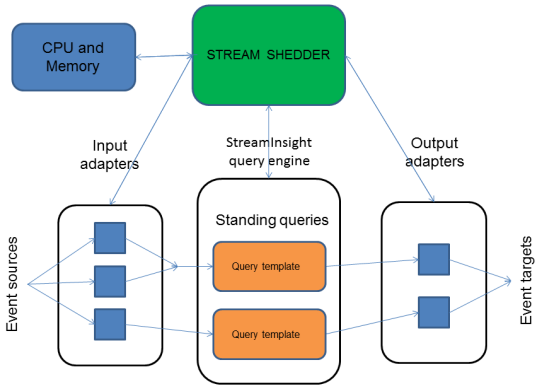


Figure 1.6: The StreamShedder architecture

StreamShedder is a software module implemented in C#. It communicates with a memory and processor monitoring module (CPU and Memory), which provides the used system resources. Depending on the user specified memory and CPU time thresholds, StreamShedder can command the input adapters to eliminate some tuples. StreamShedder also monitors the latency of the queries on the server. Based on the data it receives, it "tells" the input adapters what tuples to eliminate. To assess the impact on the application semantics, this module also communicates with output adapters, obtaining queries results. The results of this research are published in [Sur11d].

4.5 StreamEval

We develop a solution that evaluates performance variations when different conditions from the environment change (for instance, when we manipulate the data rate of the data sources that feed the continuous queries on the server): StreamEval. In the implementation of the monitoring application and of our proposed framework we use the previously mentioned commercial platform developed by Microsoft in the last couple of years: StreamInsight [AGR+09]. We use the StreamingTraffic monitoring application we developed for the StreamShedder architecture (section 4.4).

We denote by DR the data rate of the input stream, defined as the number of elements that arrive on the input stream S every second. We will use this notation when we change the data source's data rate. We use the (slightly changed) notation *ConsumedGate* from [Mon] to refer to the point immediately following the input adapters (at the first operator from a continuous query Q).

We use the query monitoring attributes provided by the ManagementService API [Mon]. Like in [Mon], we are interested in monitoring the average consumed latency, between two time instants t_1 and t_2. Therefore, we evaluate the number of processed tuples *TupleCount* and the response time *Latency* at *ConsumedGate*, at moments t_1 and t_2. We denote the average consumed latency by *AvgLat*. We compute *AvgLat* by applying the formula from [Mon]:

$$AvgLat = (Latency_{t_2} - Latency_{t_1})/(TupleCount_{t_2} - TupleCount_{t_1}). \quad (1.5)$$

We change the data rate of the data source as follows. We start with value 1 for DR (one event every second) and we evaluate the corresponding *AvgLat* value. We increase DR up to 500 events per second. The metric *AvgLat* stays under one millisecond. Even for values of 1000 events / second for DR, which exceed the requirements of StreamingTraffic, *AvgLat* maintains around the same value. This research is briefly described in the abstract [Sur12a]. The extended paper is under evaluation [Sur12b].

4.6 SCIPE

We develop a set of principles, SCIPE (*SCIentific data stream processing PrinciplEs*), oriented towards the realization of a Sci-DSMS, a Data Stream Management System that handles very large data from domains connected with exact sciences. Research communities from exact sciences have been working with data in the order of petabyes and this data is expected to exceed exabytes dimensions in the years to come [BLW09]. In this context, we investigate the possibility of implementing a DSMS, which answers the needs of exact sciences communities. Considering the objectives

of a particular domain can lead to optimizing resource usage in continuous queries on data streams.

We describe SCIPE:

1. When possible, the system should process and subsequently summarize or eliminate arriving elements. This principle has a significant impact on memory usage, by keeping elements in the form of a summary, if they are required for future processing.

2. If each element needs to be individually stored, then only the elements from the recent past should be kept and old elements should be eliminated or summarized.

3. The designed system should encompass elements revision possibilities (this strategy is borrowed from [AAB+05]).

4. Load shedding should be performed in a semantic manner, dependent on the application domain (Aurora [ACC+03] is a system that performs semantic load shedding).

5. Queries should be written in a user-friendly manner, combining visual languages and a declarative SQL interface (we combine the approaches from [ACC+03] and [ABW06]).

SCIPE and its motivation are published in [Sur11c].

4.7 InstantSchoolKnow

We analyse the educational field and the ways in which data streams can optimize the educational processes. We develop EdStream, a set of rules that can be applied when implementing an educational monitoring platform based on data stream processing. We propose the design of an educational monitoring platform, InstantSchoolKnow. Its purpose is to continuously acquire data from educational institutions (registered on the platform), to analyse this data using a continuous processing paradigm and to publish the results of this analysis in real time. To achieve

this goal the following stages must be accomplished: the registration on the InstantSchoolKnow platform, data acquisition, data analysis and data publishing. Unlike current approaches, InstantSchoolKnow aims at unifying e-learning and students monitoring capabilities in a single platform. This research is published in [Sur11e].

4.8 A platform for accessing data on smart mobile decives

We propose an architecture for the implementation of an online platform with content oriented towards political communication. In its initial stage the data has a static nature and can be accessed from smart mobile devices. We intend to extend this new media platform with stream and services processing capabilities, in the context of a pervasive environment. The research is published in [Sur09].

5 Managing heterogeneous data in pervasive application development

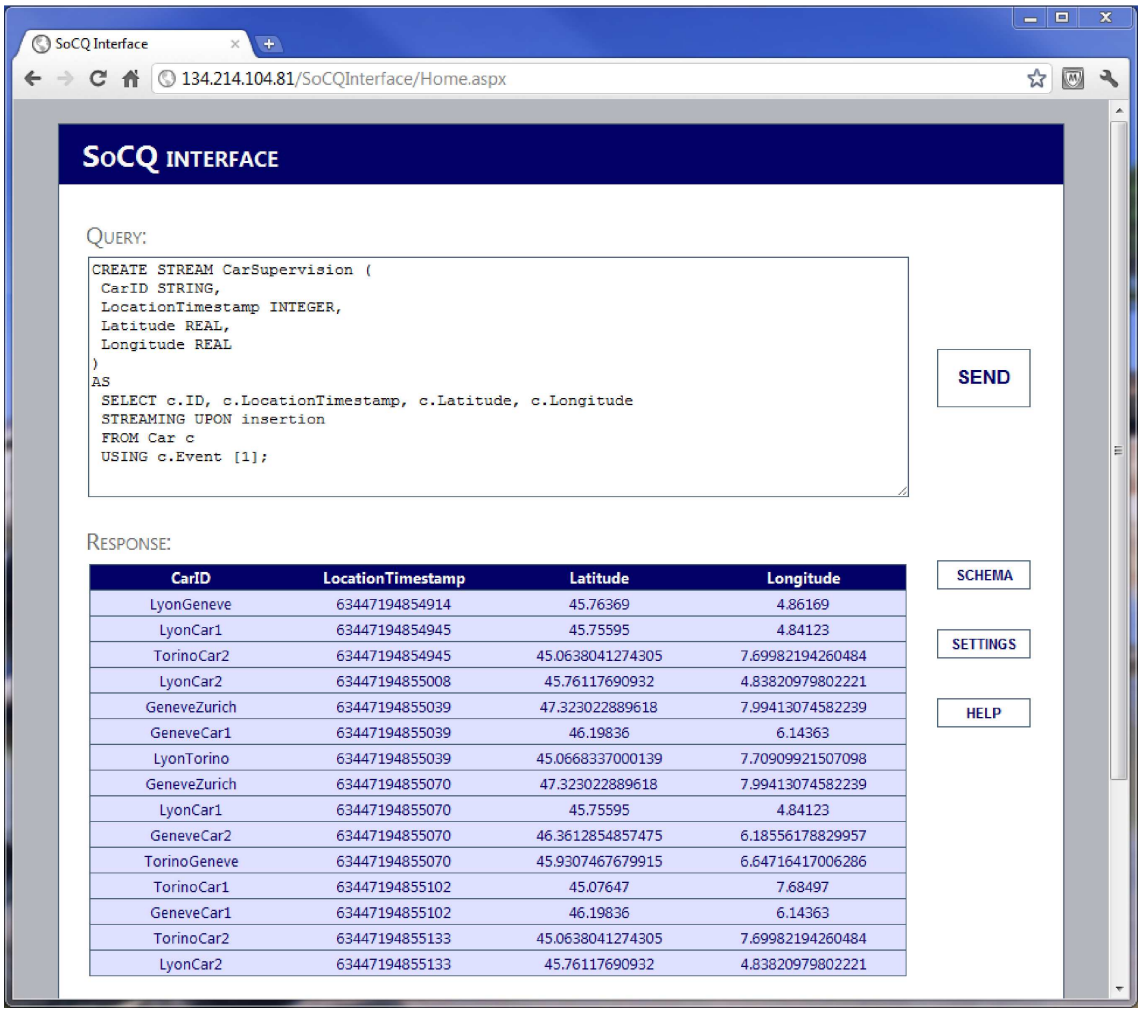
A considerable number of scenarios and pervasive applications based on these scenarios contain static data (similar to data from relational databases), data streams and distributed functionalities or services [GLP10], according to the real, everyday life situations they model. In order to manage all these elements from a pervasive environment, ad hoc programming is usually used, which integrates multiple programming models: imperative and declarative languages and network protocols [Gri09]. Solutions developed in this manner are complex in their implementation and the development time is high. We investigate alternative approaches to pervasive application development and methods to assess the development process.

We enumerate the original contributions from our thesis, which we obtained in the context of heterogeneous data management in pervasive applications.

5.1 Heterogeneous data management in a pervasive environment

We tackle one of the main challenges in pervasive computing: facilitating pervasive application development. We describe a scenario for container monitoring in a medical context, which refers to the transportation of medical content in containers equipped with sensors. Based on this scenario, we discuss a testbed, which can be used when developing applications and assessing the development process and we show how to build a pervasive application using the SoCQ (Service-oriented Continuous Query) system [GFLP09]. The scenario, its simulation as a testbed, its visualization and the developed pervasive application represent the intrinsic contributions of this research, which we developed together with the team we worked with, at LIRIS, INSA Lyon. The results of this research are presented in a paper accepted at an international conference and will be published [GLL+12].

To interact with the query engine, we implement a Web ASP.NET application. This application allows a developer to write continuous queries, which mix hete-



CarID	LocationTimestamp	Latitude	Longitude
LyonGeneve	63447194854914	45.76369	4.86169
LyonCar1	63447194854945	45.75595	4.84123
TorinoCar2	63447194854945	45.0638041274305	7.69982194260484
LyonCar2	63447194855008	45.76117690932	4.83820979802221
GeneveZurich	63447194855039	47.323022889618	7.99413074582239
GeneveCar1	63447194855039	46.19836	6.14363
LyonTorino	63447194855039	45.0668337000139	7.70909921507098
GeneveZurich	63447194855070	47.323022889618	7.99413074582239
LyonCar1	63447194855070	45.75595	4.84123
GeneveCar2	63447194855070	46.3612854857475	6.18556178829957
TorinoGeneve	63447194855070	45.9307467679915	6.64716417006286
TorinoCar1	63447194855102	45.07647	7.68497
GeneveCar1	63447194855102	46.19836	6.14363
TorinoCar2	63447194855133	45.0638041274305	7.69982194260484
LyonCar2	63447194855133	45.76117690932	4.83820979802221

Figure 1.7: The Web application that allows continuous query writing

rogeneous data from the environment, by using a query language similar to SQL, specific to the SoCQ system [Gri09]. If we want to monitor the location of each car at each instant, we write a query in this language, which generates all the car locations as a result, in a real time manner. Figure 1.7 depicts the Web application, a query and its results.

5.2 AgilBench

We propose a benchmark for evaluating pervasive application development. We use multiple systems for this purpose and we conduct an experimental study. The results of this research were included in a paper we submitted to an international

conference and that is currently under evaluation [SGPS12].

6 Conclusions and future research directions

The two main research directions we followed led to developing architectures, strategies and techniques to optimize resource usage when processing data streams, but also to the implementation of a testbed and a benchmark in the context of pervasive application development. These contributions were published in journals or proceedings of international conferences. One additional paper is accepted for publication and two other papers are currently under evaluation.

The fields of data streams and pervasive applications are advancing on a daily basis. We can certainly expect that our proposals will suffer changes in time. We intend to automate the sizing window effect, so that the system can automatically choose the optimal window size and to enhance the resource-aware architectures we proposed, so that all the decisions are taken by the system, with no user intervention. We want to add new services in the testbed and enrich the benchmark we implemented to evaluate pervasive application development. We assess the possibility of designing a system, capable of managing pervasive environments, which allows the total replacement of the scenario that models a pervasive environment, which no changes in its implementation. The most powerful systems (like SoCQ) need new modules if the data access mechanisms change, once the scenario is replaced.

Thesis bibliography

[AAB+05] Daniel J. Abadi, Yanif Ahmad, Magdalena Balazinska, Ugur Cetintemel, Mitch Cherniack, Jeong-Hyon Hwang, Wolfgang Lindner, Anurag S. Maskey, Alexander Rasin, Esther Ryvkina, Nesime Tatbul, Ying Xing, and Stan Zdonik. The Design of the Borealis Stream Processing Engine. In *CIDR 2005, Proceedings of Second Biennial Conference on Innovative Data Systems Research*, pages 277–289, 2005.

[ABB+03] Arvind Arasu, Brian Babcock, Shivnath Babu, Mayur Datar, Keith Ito, Rajeev Motwani, Itaru Nishizawa, Utkarsh Srivastava, Dilys Thomas, Rohit Varma, and Jennifer Widom. STREAM: The Stanford Stream Data Manager. *IEEE Data Engineering Bulletin*, 26(1):19–26, 2003.

[ABB+04] Arvind Arasu, Brian Babcock, Shivnath Babu, John Cieslewicz, Mayur Datar, Keith Ito, Rajeev Motwani, Utkarsh Srivastava, and Jennifer Widom. STREAM: The Stanford Data Stream Management System. Technical report, Stanford InfoLab, 2004.

[ABC+05] Yanif Ahmad, Bradley Berg, Ugur Cetintemel, Mark Humphrey, Jeong-Hyon Hwang, Anjali Jhingran, Anurag Maskey, Olga Papaemmanouil, Alex Rasin, Nesime Tatbul, Wenjuan Xing, Ying Xing, and Stanley B. Zdonik. Distributed operation in the Borealis stream processing engine. In *SIGMOD Conference*, pages 882–884, 2005.

[ABW06] Arvind Arasu, Shivnath Babu, and Jennifer Widom. The CQL continuous query language: Semantic foundations and query execution. *The VLDB Journal*, 15(2):121–142, 2006.

[ACC+03] Daniel J. Abadi, Donald Carney, Ugur Cetintemel, Mitch Cherniack, Christian Convey, Sangdon Lee, Michael Stonebraker, Nesime Tatbul, and Stanley B. Zdonik. Aurora: a new model and architecture for data stream management. *The VLDB Journal*, 12(2):120–139, 2003.

[ACG+04] Arvind Arasu, Mitch Cherniack, Eduardo Galvez, David Maier, Anurag S. Maskey, Esther Ryvkina, Michael Stonebreaker, and Richard Tibbetts. Linear Road: A Stream Data Management Benchmark. In *VLDB'04, Proceedings of The Thirtieth International Conference on Very Large Data Bases*, pages 480–491, 2004.

[Adm] Federal Highway Administration. Congestion Pricing: A Primer. `http://www.ops.fhwa.dot.gov/publications/congestionpricing/congestionpricing.pdf`.

[Agg07] Charu C. Aggarwal. An Introduction to Data Streams. In *Data Streams - Models and Algorithms*, pages 1–8. 2007.

[AGR+09] Mohamed H. Ali, Ciprian Gerea, Balan Sethu Raman, Beysim Sezgin, Tiho Tarnavski, Tomer Verona, Ping Wang, Peter Zabback, Asvin Ananthanarayan, Anton Kirilov, Ming Lu, Alex Raizman, Ramkumar Krishnan, Roman Schindlauer, Torsten Grabs, Sharon Bjeletich, Badrish Chandramouli, Jonathan Goldstein, Sudin Bhat, Ying Li, Vincenzo Di Nicola, Xianfang Wang, David Maier, Stephan Grell, Olivier Nano, and Ivo Santos. Microsoft CEP Server and Online Behavioral Targeting. *Proceedings of the VLDB Endowment*, 2(2):1558–1561, August 2009.

[AIS93] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining association rules between sets of items in large databases. In *SIGMOD '93,*

Proceedings of the 1993 ACM SIGMOD international conference on Management of data, pages 207–216, 1993.

[AMT06] Serge Abiteboul, Ioana Manolescu, and Emanuel Taropa. A Framework for Distributed XML Data Management. In *EDBT 2006, Proceedings of The 10th International Conference on Extending Database Technology*, pages 1049–1058, 2006.

[AW04] Arvind Arasu and Jennifer Widom. A Denotational Semantics for Continuous Queries over Streams and Relations. *SIGMOD Record*, 33(3):6–12, 2004.

[BBC+04] Hari Balakrishnan, Magdalena Balazinska, Donald Carney, Ugur Cetintemel, Mitch Cherniack, Christian Convey, Eduardo F. Galvez, Jon Salz, Michael Stonebraker, Nesime Tatbul, Richard Tibbetts, and Stanley B. Zdonik. Retrospective on Aurora. *The VLDB Journal*, 13(4):370–383, 2004.

[BBD+02] Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom. Models and Issues in Data Stream Systems. In *PODS*, pages 1–16, 2002.

[BBD+04] Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Dilys Thomas. Operator scheduling in data stream systems. *The VLDB Journal*, 13(4):333–353, 2004.

[BBDM03] Brian Babcock, Shivnath Babu, Mayur Datar, and Rajeev Motwani. Chain: Operator Scheduling for Memory Minimization in Data Stream Systems. In *SIGMOD Conference*, pages 253–264, 2003.

[BBS04] Magdalena Balazinska, Hari Balakrishnan, and Michael Stonebraker. Load management and high availability in the Medusa distributed stream processing system. In *SIGMOD '04, Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 929–930, 2004.

[BDM04] Brian Babcock, Mayur Datar, and Rajeev Motwani. Load Shedding for Aggregation Queries over Data Streams. In *ICDE 2004, Proceedings of the 20th International Conference on Data Engineering*, pages 350–361, 2004.

[BH07] Don Box and Anders Hejlsberg. LINQ: .NET Language-Integrated Query. `http://msdn.microsoft.com/en-us/library/bb308959.aspx`, 2007.

[BLW09] Jacek Becla, Kian-Tat Lim, and Daniel Liwei Wang. Report from the 3rd Workshop on Extremely Large Databases. *Data Science Journal*, 8:MR1–MR16, 2009.

[CCC+02] Don Carney, Ugur Cetintemel, Mitch Cherniack, Christian Convey, Sangdon Lee, Greg Seidman, Michael Stonebraker, Nesime Tatbul, and Stan Zdonik. Monitoring Streams - a New Class of Data Management Applications. In *VLDB '02, Proceedings of the 28th International Conference on Very Large Data Bases*, pages 215–226, 2002.

[CCD+03] Sirish Chandrasekaran, Owen Cooper, Amol Deshpande, Michael J. Franklin, Joseph M. Hellerstein, Wei Hong, Sailesh Krishnamurthy, Sam Madden, Vijayshankar Raman, Fred Reiss, and Mehul Shah. TelegraphCQ: Continuous Dataflow Processing for an Uncertain World. In *CIDR 2003, Proceedings of the First Biennial Conference on Innovative Data Systems Research*, 2003.

[CCR+03] Don Carney, Ugur Cetintemel, Alex Rasin, Stan Zdonik, Mitch Cherniack, and Michael Stonebraker. Operator Scheduling in a Data Stream Manager. In *VLDB '03, Proceedings of the 29th International Conference on Very Large Data Bases*, pages 838–849, 2003.

[CDTW00] Jianjun Chen, David J. DeWitt, Feng Tian, and Yuan Wang. NiagaraCQ: A Scalable Continuous Query System for Internet Databases. In

Proceedings of ACM SIGMOD International Conference on Management of Data, pages 379–390, 2000.

[CEP] Complex Event Processing. `http://www.complexevents.com/`.

[CG05] Graham Cormode and Minos N. Garofalakis. Sketching Streams Through the Net: Distributed Approximate Query Tracking. In *VLDB 2005, Proceedings of the 31st International Conference on Very Large Data Bases*, pages 13–24, 2005.

[Cha] Nicholas Chase. The ultimate mashup – Web services and the semantic Web, Part 1: Use and combine Web services. `http://www.ibm.com/developerworks/xml/tutorials/x-ultimashup1/`.

[CNC11] Consiliul Național al Cercetării Științifice din Învățământul Superior. Situația curentă a revistelor recunoscute CNCSIS. `http://www.cncsis.ro/userfiles/file/CENAPOSS/Bplus_2011.pdf`, 2011.

[Cor08] Computing Research and Education. `http://core.edu.au/cms/images/downloads/conference/Astar.pdf`, 2008.

[CSD11] Alfredo Cuzzocrea, Il-Yeol Song, and Karen C. Davis. Analytics over large-scale multidimensional data: the big data revolution! In *DOLAP'11, Proceedings of the ACM 14th international workshop on Data Warehousing and OLAP*, pages 101–104, 2011.

[CVC+10] Víctor Cuevas-Vicenttín, Genoveva Vargas-Solar, Christine Collet, Noha Ibrahim, and Christophe Bobineau. Coordinating Services for Accessing and Processing Data in Dynamic Environments. In *OTM'10, Proceedings of the 2010 International Conference on On the move to meaningful internet systems - Volume Part I*, pages 309–325, 2010.

[dDCK+06] Scott de Deugd, Randy Carroll, Kevin E. Kelly, Bill Millett, and Jeffrey Ricker. SODA: Service Oriented Device Architecture. *IEEE Pervasive Computing*, 5(3):94–96, 2006.

[DG08] Jeffrey Dean and Sanjay Ghemawat. MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, January 2008.

[DGIM02] Mayur Datar, Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Maintaining Stream Statistics over Sliding Windows. In *SODA 2002, ACM-SIAM Symposium on Discrete Algorithms*, pages 635–644, 2002.

[ECPS02] Deborah Estrin, David Culler, Kris Pister, and Gaurav Sukhatme. Connecting the Physical World with Pervasive Networks. *IEEE Pervasive Computing*, 1(1):59–69, January 2002.

[Era10] Excellence in Research for Australia 2010 (Australian Research Council). Ranked Conference List. `http://www.arc.gov.au/era/era_2010/archive/key_docs10.htm`, 2010.

[FHA10] Fatima Farag, Moustafa Hammad, and Reda Alhajj. Adaptive query processing in data stream management systems under limited memory resources. In *Proceedings of the 3rd workshop on Ph.D. students in information and knowledge management*, pages 9–16, 2010.

[FHL+11] Nicolas Ferry, Vincent Hourdin, Stephane Lavirotte, Gaetan Rey, Michel Riveill, and Jean-Yves Tigli. Wcomp, a middleware for ubiquitous computing. In *Ubiquitous Computing*, pages 151–176, 2011.

[FPSS96] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, 17(3):37–54, 1996.

[GAE06] Thanaa M. Ghanem, Walid G. Aref, and Ahmed K. Elmagarmid. Exploiting predicate-window semantics over data streams. *SIGMOD Record*, 35(1):3–8, 2006.

[Geh09] Johannes Gehrke. Technical perspective - Data stream processing: when you only get one look. *Communications of the ACM*, 52(10):96, 2009.

[GFLP09] Yann Gripay, Frédérique Laforest, and Jean-Marc Petit. SoCQ: a Pervasive Environment Management System. In *UbiMob'09, 5èmes Journées Francophones Mobilité et Ubiquité*, pages 87–90, 2009.

[GLL+12] Yann Gripay, Frédérique Laforest, François Lesueur, Nicolas Lumineau, Jean-Marc Petit, Vasile-Marian Scuturici, Samir Sebahi, and Sabina Surdu. ColisTrack: Testbed for a Pervasive Environment Management System. In *EDBT 2012, The 15th International Conference on Extending Database Technology. Accepted and will be published*, 2012.

[GLP07] Yann Gripay, Frédérique Laforest, and Jean-Marc Petit. Towards Action-Oriented Continuous Queries in Pervasive Systems. In *BDA'07, Bases de Données Avancées 2007*, pages 1–20, 2007.

[GLP09] Yann Gripay, Frédérique Laforest, and Jean-Marc Petit. SoCQ: a Framework for Pervasive Environments. In *ISPAN 2009, 10th International Symposium on Pervasive Systems, Algorithms and Networks*, pages 154–159, 2009.

[GLP10] Yann Gripay, Frédérique Laforest, and Jean-Marc Petit. A Simple (yet Powerful) Algebra for Pervasive Environments. In *EDBT 2010, Proceedings of The 13th International Conference on Extending Database Technology*, pages 1–12, 2010.

[Gooa] Google Maps API Family. `http://code.google.com/apis/maps/index.html`.

[Goob] The Google Directions API. `http://code.google.com/apis/maps/documentation/directions/`.

[Gri08] Yann Gripay. Service-oriented Continuous Queries for Pervasive Systems. In *EDBT 2008 PhD Workshop (unofficial proceedings)*, pages 1–7, 2008.

[Gri09] Yann Gripay. *A Declarative Approach for Pervasive Environments: Model and Implementation*. PhD thesis, Institut National des Sciences Appliquées de Lyon, 2009.

[GS10] Yann Gripay and Vasile-Marian Scuturici. Managing Distributed Service Environments: a Data-oriented approach. In *UbiMob'10, 6èmes Journées Francophones Mobilité et Ubiquité*, pages 1–4, 2010.

[GSK+09] Torsten Grabs, Roman Schindlauer, Ramkumar Krishnan, Jonathan Goldstein, and Rafael Fernández. Introducing Microsoft StreamInsight. Technical report, Microsoft, 2009.

[GZK05] Mohamed Medhat Gaber, Arkady Zaslavsky, and Shonali Krishnaswamy. Mining data streams: A review. *ACM SIGMOD Record*, 34(2):18–26, 2005.

[HL11] Martin Hilbert and Priscila Lopez. The World's Technological Capacity to Store, Communicate, and Compute Information. *Science*, 332(6025):60–65, February 2011.

[HMS01] David J. Hand, Heikki Mannila, and Padhraic Smyth. *Principles of Data Mining*, pages 1–24. The MIT Press, Cambridge, MA, USA, 2001.

[IGLS06] Jon Espen Ingvaldsen, Jon Atle Gulla, Tarjei Laegreid, and Paul Christian Sandal. Financial News Mining: Monitoring Continuous Streams of Text. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 321–324, 2006.

[IM06] Edurne Izkue and Eduardo Magana. Sampling time-dependent parameters in high-speed network monitoring. In *PM2HW2N 2006, Proceedings of the ACM International Workshop on Performance Monitoring, Measurement, and Evaluation of Heterogeneous Wireless and Wired Networks*, pages 13–17, 2006.

[Int] Ovidiu Vermesan, Mark Harrison, Harald Vogt, Kostas Kalaboukas, Maurizio Tomasella, Karel Wouters, Sergio Gusmeroli

and Stephan Haller. Internet of Things. Strategic Research Roadmap. http://www.grifs-project.eu/data/File/CERP-IoT%20SRA_IoT_v11.pdf.

[JMHA10] Oana Jurca, Sebastian Michel, Alexandre Herrmann, and Karl Aberer. Continuous query evaluation over distributed sensor networks. In *ICDE'10, Proceedings of The 26th IEEE International Conference on Data Engineering*, pages 912–923, 2010.

[KDA+10] Seyed J. Kazemitabar, Ugur Demiryurek, Mohamed H. Ali, Afsin Akdogan, and Cyrus Shahabi. Geospatial Stream Query Processing using Microsoft SQL Server StreamInsight. *Proceedings of the VLDB Endowment*, 3(2):1537–1540, September 2010.

[KG10] Ramkumar Krishnan and Jonathan Goldstein. A Hitchhiker's Guide to Microsoft StreamInsight Queries. Technical report, Microsoft, June 2010.

[Kog07] Jacob Kogan. *Introduction to Clustering Large and High-Dimensional Data*, pages 98–99. Cambridge University Press, NY, USA, 2007.

[Lan09] Marc Langheinrich. A survey of RFID privacy approaches. *Personal and Ubiquitous Computing*, 13(6):413–421, August 2009.

[Lin] LINQ documentation. http://msdn.microsoft.com/en-us/library/bb397926.aspx.

[LMT+05] Jin Li, David Maier, Kristin Tufte, Vassilis Papadimos, and Peter A. Tucker. Semantics and Evaluation Techniques for Window Aggregates in Data Streams. In *SIGMOD Conference*, pages 311–322, 2005.

[MCP+02] Alan M. Mainwaring, David E. Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pages 88–97, 2002.

[Mea] Text REtrieval Conference (TREC). Common Evaluation Measures, 2011. `http://trec.nist.gov/pubs/trec19/appendices/measures.pdf`.

[Mei11] Erik Meijer. The World According to LINQ. *Communications of the ACM*, 54(10):45–51, October 2011.

[Mon] StreamInsight documentation. Monitoring the StreamInsight Server and Queries. `http://msdn.microsoft.com/en-us/library/ee391166.aspx`.

[Mur09] Teruyasu Murakami. The Age of Ubiquitous. *Highlighting Japan through articles*, 2(10):8–9, February 2009.

[MWA+03] Rajeev Motwani, Jennifer Widom, Arvind Arasu, Brian Babcock, Shivnath Babu, Mayur Datar, Gurmeet Singh Manku, Chris Olston, Justin Rosenstein, and Rohit Varma. Query Processing, Resource Management, and Approximation in a Data Stream Management System. In *CIDR 2003, Proceedings of the First Biennial Conference on Innovative Data Systems Research*, 2003.

[Nas09] Hebah H. O. Nasereddin. Stream Data Mining. *International Journal of Web Applications*, 1(4):183–190, December 2009.

[Pug08] William Pugh. Technical perspective: A methodology for evaluating computer system performance. *Communications of the ACM*, 51(8):82–82, August 2008.

[RMCZ06] Esther Ryvkina, Anurag S. Maskey, Mitch Cherniack, and Stan Zdonik. Revision Processing in a Stream Processing Engine: A High-Level Design. In *ICDE 2006, Proceedings of the 22nd International Conference on Data Engineering*, pages 141–143, 2006.

[Rys11] Michael Rys. Scalable SQL. *Communications of the ACM*, 54(6):48–53, June 2011.

[Sch07] Sven Schmidt. *Quality-of-Service-Aware Data Stream Processing*. PhD thesis, Dresden University of Technology, Department of Computer Science, 2007.

[Sch09] Arnd Schröter. Modeling and optimizing content-based publish/subscribe systems. In *Proceedings of the 6th Middleware Doctoral Symposium*, pages 5:1–5:6, 2009.

[Scu09] Marian Scuturici. Dataspace API. Technical report, LIRIS, September 2009.

[SGPS12] Sabina Surdu, Yann Gripay, Jean-Marc Petit, and Vasile-Marian Scuturici. Paper under evaluation. International Conference A*, 2012.

[SIA] StreamInsight Server Architecture. `http://msdn.microsoft.com/en-us/library/ee391536.aspx`.

[Sima] Mark Simms. 101'ish LINQ Samples for StreamInsight (part 1 - filtering and aggregation). `http://blogs.msdn.com/b/masimms/archive/2010/09/16/101-ish-linq-samples-for-streaminsight.aspx`.

[Simb] Mark Simms. Using SQL Server for reference data in a StreamInsight query. `http://windowsazurecat.com/2011/08/sql-server-reference-data-streaminsight-query`.

[SM03] Debashis Saha and Amitava Mukherjee. Pervasive Computing: A Paradigm for the 21st Century. *IEEE Computer*, 36(3):25–31, March 2003.

[Soc] The SoCQ Project. `http://socq.liris.cnrs.fr/`.

[SS11] Sabina Surdu and Vasile-Marian Scuturici. Addressing resource usage in stream processing systems: sizing window effect. In *IDEAS'11 Proceedings, 15th International Database Engineering & Applications Symposium*, pages 247–248, 2011.

[Stra] StreamInsight documentation. Creating Input and Output Adapters. http://msdn.microsoft.com/en-us/library/ee378877.aspx.

[Strb] StreamInsight documentation. Microsoft StreamInsight. http://msdn.microsoft.com/en-us/library/ee362541.aspx.

[Sur09] Sabina Surdu. Online Political Communication. In *Interdisciplinary New Media Studies Conference Proceedings*, pages 55–58, 2009.

[Sur11a] Sabina Surdu. A New Architecture Supporting The Sizing Window Effect With StreamInsight. *Studia Universitatis Babeș-Bolyai Series Informatica*, LVI(4):111–120, 2011.

[Sur11b] Sabina Surdu. A technique for constructing training sets in data stream mining: kSiEved Window Training Set. In *MDIS 2011, Proceedings of the Second International Conference on Modelling and Development of Intelligent Systems*, pages 180–191, 2011.

[Sur11c] Sabina Surdu. Data stream management systems: a response to large scale scientific data requirements. *Annals of the University of Craiova, Mathematics and Computer Science Series*, 38(3):66–75, 2011.

[Sur11d] Sabina Surdu. A new architecture for load shedding on data streams with StreamInsight: StreamShedder. *University of Pitești Scientific Bulletin, Series Electronics and Computers Science*, 11(2):57–64, 2011.

[Sur11e] Sabina Surdu. Towards an education monitoring platform based on data stream processing. In *Education and Creativity for a Knowledge Society International Conference, The fifth edition - Computer Science Section*, pages 61–66, 2011.

[Sur12a] Sabina Surdu. A new framework for evaluating performance in data stream monitoring applications with StreamInsight: StreamEval. In *MaCS 2012, Booklet of abstracts from The 9th Joint Conference on Mathematics and Computer Science*, page 92, 2012.

[Sur12b] Sabina Surdu. A new framework for evaluating performance in data stream monitoring applications with StreamInsight: StreamEval. Under evaluation at Annales Universitatis Scientiarum Budapestinensis de Rolando Eötvös Nominatae, Sectio Computatorica, 2012.

[SW04] Utkarsh Srivastava and Jennifer Widom. Flexible Time Management in Data Stream Systems. In *PODS '04*, pages 263–274, 2004.

[TAC+06] Nesime Tatbul, Yanif Ahmad, Ugur Cetintemel, Jeong-Hyon Hwang, Ying Xing, and Stanley B. Zdonik. Load Management and High Availability in the Borealis Distributed Stream Processing Engine. In *GSN*, pages 66–85, 2006.

[Tam03] Leon Țâmbulea. *Baze de date*. Universitatea Babeș-Bolyai, Cluj-Napoca, Romania, 6th edition, 2003.

[Tat02] Nesime Tatbul. Qos-driven load shedding on data streams. In *EDBT '02, Proceedings of the Workshops XMLDM, MDDE, and YRWS on XML-Based Data Management and Multimedia Engineering-Revised Papers*, pages 566–576, 2002.

[TCZ+03] Nesime Tatbul, Ugur Cetintemel, Stan Zdonik, Mitch Cherniack, and Michael Stonebraker. Load shedding in a data stream manager. In *VLDB '03, Proceedings of the 29th International Conference on Very Large Data Bases*, pages 309–320, 2003.

[TCZa+03] Nesime Tatbul, Ugur Cetintemel, Stan Zdonik, Mitch Cherniack and Michael Stonebraker. Load Shedding on Data Streams. In *MPDS'03, ACM Workshop on Management and Processing of Data Streams*, 2003.

[Tib03] Richard S. Tibbetts. Linear Road: Benchmarking Stream-Based Data Management Systems. MSc Thesis. Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 2003.

[Tpc] TPC Benchmarks. http://www.tpc.org/information/benchmarks.asp.

[TTPM] Pete Tucker, Kristin Tufte, Vassilis Papadimos, and David Maier. NEXMark - a benchmark for queries over data streams. Technical report. OGI School of Science and Engineering at OHSU, 2002.

[TZ06] Nesime Tatbul and Stan Zdonik. Window-aware load shedding for aggregation queries over data streams. In *VLDB '06, Proceedings of The 32nd International Conference on Very Large Data Bases*, pages 799–810, 2006.

[Uni05] International Telecommunication Union. *The Internet of Things*. ITU Internet Reports. International Telecommunication Union, 2005.

[Wei91] Mark Weiser. The Computer for the 21st Century. *Scientific American*, 265(3):94–104, September 1991.

[XL05] Wenwei Xue and Qiong Luo. Action-Oriented Query Processing for Pervasive Computing. In *CIDR 2005, Proceedings of The Second Biennial Conference on Innovative Data Systems Research*, pages 305–316, 2005.

[XLD] XLDB - Extremely Large Databases. http://www.xldb.org/.

[YK96] Qi Yang and Haris N. Koutsopoulos. A microscopic traffic simulator for evaluation of dynamic traffic management systems. *Transportation Research Part C*, 4(3):113–129, 1996.

[ZB03] Qiankun Zhao and Sourav S. Bhowmick. Sequential Pattern Mining: A Survey. Technical report, Nanyang Technological University, Singapore, 2003.

[ZSC+03] Stanley B. Zdonik, Michael Stonebraker, Mitch Cherniack, Ugur Cetintemel, Magdalena Balazinska, and Hari Balakrishnan. The Aurora and Medusa Projects. *IEEE Data Engineering Bulletin*, 26(1):3–10, 2003.