Towards the Stream Analysis Model in Grid-based Zero-Latency Data Stream Warehouse

Tho Manh Nguyen, A Min Tjoa, Josef Schiefer

Institute of Software Technology
Vienna University of Technology
Favoriten Straße 9/188, A-1040 Vienna, Austria
{tho,tjoa,schiefer}@ifs.tuwien.ac.at

Abstract. Recent emerging applications increasingly generate continuous, larger amounts of valuable data. The demand of conducting advanced analysis over fast and huge data streams to capture trends, patterns, and exceptions become crucial. However, fully extracting the latent knowledge within the data stream is a challenging task because of insufficient technology. While Data Warehouse (DWH) technologies have resulted in considerable information processing efficiencies, there is still a significant delay in the time to deliver mission critical information to data consumers. Traditional data stream processing focuses on statistical approaches hence produces approximate results. In this paper, we introduce the Stream Analysis Model with a Grid-based Zero-Latency Data Stream Warehouse (GZLDSWH) framework which allows to perform analytical processing on continuous data streams and to trigger relevant actions depending on patterns discovered in event streams without using statistical approximation. Essential data stream elements are captured, analysed on the fly and finally evaluated to detect abnormalities while the entire data streams are stored within a Grid and integrated into a virtual DWH for further analysis in the case of ambiguity or uncertainty.

1 Introduction

In recent years, advances in modern technologies have allowed us to automatically record daily transactions at a rapid rate. Such processes lead to large amounts of transactional data which grow at an “unlimited” rate and are available as continuous data streams. Data streams arise naturally and are used in the various scientific and business application domains. Examples include network packets, records of telephone calls, weather measurements, satellite imagery, and sensor networks. In daily life, sources of data streams such as Internet transactions, click streams, updates of stock quotes, toll booth observations are also ubiquitous. Data streams thus become now fundamental to many data processing applications and the need for complex analyses of these high-speed data streams is substantial. Further, the ability to make decisions and discover interesting patterns on-line (i.e., as the data stream arrives) is crucial for several mission-critical tasks that could be decisive for an organization (e.g., telecom fraud detection, stock market monitoring, etc.)

Data Warehouse (DWH) and Business Intelligence (BI) applications are normally used for strategic planning and decision making. However, existing DWH technologies and tools (e.g. ETL, OLAP) often rely on the assumption that data in the DWH can lag at least a day (if not a week or a month) behind the actual operational data and the decisions are based upon the analytical process on that "window on the past". For many business situations, especially, in data stream analysis, this decision making approach is too slow due to the fast pace of today's business.

The huge volumes of data streams arrive with irregular, high data rates, and can be read only once. Data stream processing thus entails special constraints. Firstly, data stream systems are typically characterized by the presence of multiple long-running continuous queries which may cause blocking query operator [1]. Secondly, linear scans are the only cost-effective access method because random access is prohibitively expensive due to the lack of resources. Approximate statistical based methods are used as general techniques for data reduction and synopsis construction in data stream processing such as sketches [6], random sampling, histograms [7], and wavelets [8]. Other approximate methods are applied to tackle the blocking operator such as Sliding Window [1], load shedding [10], punctuation [9].

We proposed a framework for building a Grid-based Zero-Latency Data Stream Warehouse [2] (GZLDSWH) in which entire data streams are captured with no loss and continuously stored within the Grid while performing the analytical processing without using approximation. In this paper, we introduce a Stream Analysis Model which allows conducting analytical process upon data streams and issuing the relevant actions depending on the evaluation results. The remainder of the paper is organized as follows. In section 2, we provide an overview of our GZLDSWH framework. Section 3 proposes the requirements for the analytical processing on data streams and describes the structure of the Stream Analytical Model. Finally, in section 4 we present our conclusion and future work.

2 Grid-based Data Stream Warehouse (GDSWH) framework

During the last years, Service Oriented Architecture (SOA) gained popularity as new software engineering paradigm. It arose from the necessity of creating components providing clearly defined small pieces of functionality that later can be assembled into complex (usually distributed) applications. The Web Services Model follows the SOA and allows applications to communicate using agreed, widely used standards and protocols independent of their implementation and platform. The Open Grid Service Architecture/Infrastructure (OGSA/OGSI) [5] represents the convergence of Web service and Grid computing technologies with the aim of describing the next generation of Grid Architecture in which the components are exchangeable on different layers. Based on the Globus Toolkit 3 (GT3) which implements most of OGSI specifications, our GZLDSWH system is composed of several specific Grid services as described in Fig.1. Each service conducts the specific task such as capturing, storing, constructing OLAP cube, performing analysis, issuing relevant actions or notifications, etc. Further details on GZLDSWH could be found in [2].

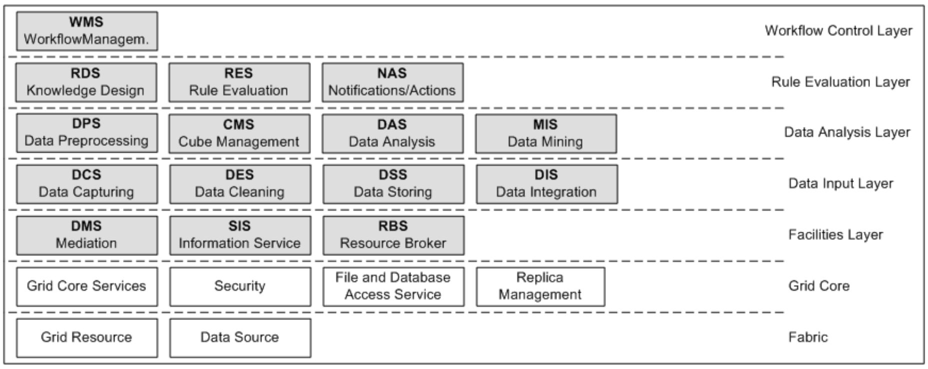

Fig.1. Grid service component in Grid-based Zero-Latency Data Stream Warehouse

3 Stream Analysis Model

Several important characteristics of data streams make them different than other data. They could be "infinite", and once a data element has arrived, it should be processed and then either archived or deleted; i.e. only a short history can be stored immediately in the database. It is also preferable to process data elements in the order of their arrival, because even sorting of sub streams of a limited size, is a blocking operation. Conducting analysis on data streams, hence have to deal with a variety of challenges:

- **Online processing**: Streams must be processed as soon as possible when they arrive because they only exist for limited time. In the DWH context, the processing can include any type of data transformation, data cleansing, calculation or evaluation metrics or storing the metrics into the DWH. Since the data has to be integrated with minimal delay, a light-weight architecture is necessary to facilitate streamlining and accelerate the data processing by moving the data between different processes steps without any intermediate file or database storage.
- **Correlating event streams:** Many analysis metrics require a set of related event streams for its calculation. The related events often stream into the system at different points in time. In order to generate metrics with minimal delay for analysis purposes, a mechanism to continuously gather related event data is required to trigger the metric calculation as soon as sufficient event data is available. A simple event correlation example is calculating the duration of a mobile call where event pairs of the time a call started and the time its finished have to be collected. As soon as a call completes, the call duration can be calculated. This example requires a mechanism for holding event data for a certain time period.
- **Multiple analysis levels:** In case of ambiguity or uncertainty in evaluating the rules during the online analysis process, it is necessary to conduct further analysis on data streams at multiple levels to reach the final decision. For example, a web site monitoring system analyzes its web clicking to balance the bandwidth between the servers. There are two scenarios to be considered:

a. If the average clicking traffic in North America is up to 30% higher than that of the last 24 hours, then no action is necessary.

b. If it is more than 50% higher, then more complex multi-dimensional analysis is needed to discover which topics cause the high traffic. The action rule pattern could be "If the average clicking traffic in North America on Sports in the last 15 minutes is 40% higher than that in the last 24 hours, then some actions are necessary to improve the bandwidth e.g. provide more Web server for Sport topic in North America area.

To support these complex multi-dimensional analysis queries, data streams needed to be stored without loss within the Grid. DWH repositories and online analytical processing (OLAP) cubes, wide-spread technologies for storing data in an analysis-centric way, are built on the fly from these Grid node's data. The significant parts are: (1) the creation and maintenance the OLAP Cube, (2) the OLAP query engine that executes analytical queries on OLAP data, and optionally (3) the OLAP Data Mining Engine for execution of the on-line analytical mining (OLAM) algorithms.

- **Automated response mechanisms**. The monitoring or analysis of data streams often entails a direct or indirect feedback for applications, users or operational systems. This response can be done manually or automatically (via reactive rule evaluation process) and enhances the target system with business intelligence.

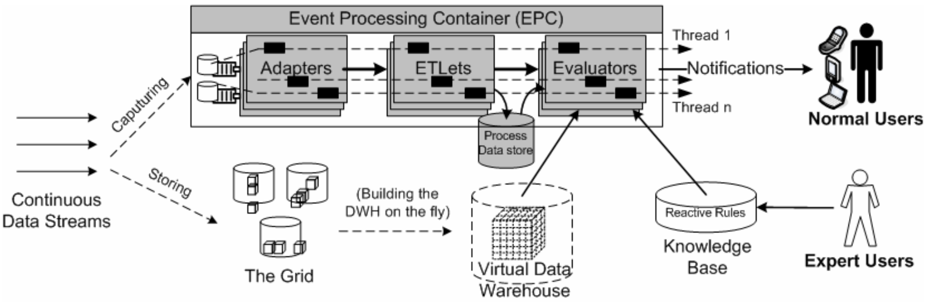

Fig.2. The Stream Analysis Model in Grid-based Zero-Latency Data Stream Warehouse

Our proposed Stream Analysis Model (described in Fig. 2) aims to satisfy the above requirements. Arriving continuous data streams are captured and processed via two paths. On one path, the light-weight Process Data Store (PDS) [4] approach is applied for online processing purpose. The EPC container manages three components 1) event adapters, 2) ETLets and 3) evaluators. Event adapters tap into data streams and extract essential event data in standard XML format. ETLets use the extracted XML event data as input and perform the data processing tasks to calculate metrics that can be evaluated by evaluator components. The evaluators access the reactive rules in Knowledge base, evaluate the metrics and trigger appropriate relevant actions (e.g. sending out notifications to business people or triggering business operations) dependent on the rules and metric values. If the ambiguity or uncertainty exists, the evaluators conduct the complete analysis process on the data extracted from the virtual DWH (built from the second path) before issuing the final actions. The ETL container handles incoming events with a lightweight Java thread, rather than a heavyweight operating system process thus the event processing can be performed without using any intermediate storage.

On the other path, the data streams are stored in variant distributed Grid nodes dependent on the Grid resource status. The OLAP cubes are built from the Grid-based sources which contain the whole streaming data. The OLAP Cube Management Service [3] are implemented to manage the creating, updating and querying of the associated cube portions distributed over the Grid nodes. The data cube structure consists of an increasing number of chunks, which consist of a fixed number of measures. A measure is the smallest, atomic element of the cube that contains a metric value. The chunk is a part of the whole cube and has the same dimension like the cube. Therefore, it contains measures associated with a number of positions of each dimension.

Both data in PDS and Virtual DWH are used for analysis purposes and decision support. The “brain” of the system which enable it to automatically react are the Knowledge base reactive rules and the rule evaluators. The reactive rules follow the basic Event-Condition-Action (ECA) rule structure, but carry out the complex OLAP analysis instead of evaluating the simple conditions as in ECA rules in OLTP. The reactive rule for web clicking analysis example above is described in Fig. 3. The Rule Evaluator operates as an engine that interacts with the Knowledge base, PDS, and DWH to evaluate the rules and controls the final actions. Hence, the engine must support the flexible decision branches which could happen when the criteria for decision making are ambiguous or uncertain. The ontology-based approach is considered to maintain and enrich the Knowledge base as well as to evaluate the rules.

```
  <?xml version="1.0" encoding="UTF-8"?>
  <ReactiveRule name = “Web Clicking Analysis">
  <variables>
   <variable name=”average_traffic”> </variable>
   <variable name=”location”> </variable>
    <variable name=”topic”> </variable>
   <variable name=”time_point”> </variable>
  <variables>
  <Events>
   <Event name = “high traffic">
     <fact > location = “North America”  topic = “All”
     average_traffic in “current_time” > 1.5 * average_traffic in “last 24h”   </fact >
   </Event>
  </Events>
  <Conditions>
      <fact > location = “North America”
       topic = “Sports”
       average_traffic in “last 15 mi” > 1.4 * average_traffic in “last 24h”  </fact >
  </Conditions>
  <Actions>
   <Action name = "add more web servers" on topic “Sports”/>
  </Actions>
</ ReactiveRule>
```

Fig.3. The Reactive analysis rule for the Web clicking monitoring system

4 Conclusion and Future Work

We have presented a model for continuous data streams within GZLDSWH framework. The model allows conducting both online analysis and complex multi-levels analysis on data streams by integrating the light-weight Process Data Store (PDS) and Grid-based OLAP Cube services. The next step in our on going work will be to refine the prototype implementation with the use of Ontology approach in maintaining and evaluating the rules. Another consideration is to improve the scheduling of the online process by utilizing the idle Grid resources to decrease the main memory usage when doing online analysis and evaluation processes (pipeline or parallel processing, etc).

Acknowledgement

The first author is supported by a Technology Grant South East Asia (No. 322/1/EZA/2002) of the Austrian Council Research and Technology and ASEA-UNINET.

References

1. Babcock, B., Babu, S., Datar, M., Motwani, R., Widom, J. : Models and Issues in Data Stream Systems. Proc. 2002 ACM Symp on Principles of Database Systems, June 2002.
2. Tho, N., Tjoa, A.: Grid-Based Zero-Latency Data Warehousing for continuous data streams processing, 6th Intl. Conf. on IIWAS2004, Jakarta, Sept. 2004.
3. Fiser, B., Onan, U., Elsayed, I., Brezany, P., Tjoa, A.: On-Line Analytical Processing on Large Databases Managed by Computational Grids. Invited paper DEXA2004, Zaragoza.
4. Schiefer, J., Beate, L., Bruckner, R.: Process Data Store: A real-time Data Store for Monitoring Business Process. DEXA2003, Prague, 2003.
5. The Globus Project, "Open Grid Service Architecture" at http://www.globus.org/ogsa/
6. Dobra, A., Garofalakis, M., Gehrke, J., Rastogi, R.: Processing complex aggregate queries over data streams", Proc. of the 2002 ACM SIGMOD, 2002.
7. Muthukrishnan, S., Strauss,M.: Maintenance of Multidimensional Histograms. 23rd Conf. Found. of Soft. Tech. and Theoretical Computer Science FSTTCS2003, India, 2003.
8. Chakrabarti, K., Garofalakis, M., Rastogi, R., Shim, K.: Approximate query processing using wavelets, The VLDB Journal vol. 10 (2001).
9. Tucker, P., Maier, D., Sheard, T.: Applying Punctuation Schemes to Queries Over Continuous Data Streams, Bull of the IEEE Comp. Soc. Tech. on Data Engineering, March 2003.
10. Babcock, B., Datar, M., Motwani., R.: Load Shedding for Aggregation Queries over Data Streams, In Proc. of Intl. Conference on Data Engineering (ICDE 2004).