Privacy-Preserving OLAP-based Monitoring of Data Streams: The PP-OMDS Approach (Discussion Paper)

Alfredo Cuzzocrea[1], Assaf Schuster[2], Gianni Vercelli[3], and Massimiliano Nolich[1]

[1] University of Trieste, Trieste, Italy
alfredo.cuzzocrea@dia.units.it, mnolich@units.it
[2] Technion, Haifa, Israel
assaf@cs.technion.ac.il
[3] University of Genoa, Genoa, Italy
gianni.vercelli@unige.it

Abstract. In this paper, we propose PP-OMDS (*Privacy-Preserving OLAP-based Monitoring of Data Streams*), an innovative framework for supporting the *OLAP-based monitoring of data streams*, which is relevant for a plethora of application scenarios (e.g., security, emergency management, and so forth), in a *privacy-preserving manner*. The paper describes motivations, principles and achievements of the PP-OMDS framework, along with technological advancements and innovations. We also incorporate a detailed comparative analysis with competitive frameworks, along with a trade-off analysis.

Keywords: Privacy-Preserving OLAP over Data Streams, OLAP-based Monitoring of Data Streams, Privacy-Preserving OLAP-based Monitoring of Data Streams.

1 Introduction

The PP-OMDS (*Privacy-Preserving OLAP-based Monitoring of Data Streams*) framework focuses the attention on the research challenge represented by *models, techniques and algorithms for supporting privacy-preserving OLAP-based monitoring of data streams*. The investigated research context includes and integrates three distinct components, namely *privacy-preserving OLAP* (e.g., [1,14]), *data stream monitoring* (e.g., [2,15]), and *privacy-preserving data stream monitoring* (e.g., [3]). Even if some sporadic works on supporting *aggregate monitoring queries* in a privacy-preserving manner exist (e.g., [4,5]), the three related issues have not been investigated together under the umbrella of a common framework specially focused to OLAP analysis (rather than the underlying aggregate querying layer), which indeed defines a powerful reference application scenario for a wide spectrum of emerging applications over data streams, such as security (e.g., [6]) and emergency management (e.g., [2]).

Starting from this main motivation, the PP-OMDS framework aims at introducing models, techniques and algorithms for supporting privacy-preserving OLAP-based monitoring of data streams, which is relevant for modern distributed environments such as Clouds. This will fulfil actual limitations of state-of-the-art solutions that do not address the relevant application scenario represented by using OLAP tools and methodologies to support data stream monitoring in a privacy-preserving manner. Application scenarios of the PP-OMDS are many-fold (e.g., security and emergency management). Tangible results of the PP-OMDS framework in real-life applicative settings are represented by relevant research and innovation advancements in the context of models, techniques and algorithms for supporting privacy-preserving OLAP-based monitoring of data streams, which represent the main "result" of the framework. It should be noted that, nowadays, these topics play a critical role as they are "naturally" compliant with the EU H2020 research framework under the topic *Big Data*.

The methodology implemented within the PP-OMDS framework foresees a *multi-step approach* that comprises: (*i*) conceptual analysis and design of case studies and related use cases showing in details how and according to which tasks users/applications interact with the PP-OMDS framework; (*ii*) conceptual analysis and design of models, techniques and algorithms for supporting privacy-preserving OLAP-based monitoring of data streams, which provides a top-down overview of the different PP-OMDS framework's components to be defined and (prototypically) implemented; (*iii*) design of the OLAP-based models and related algorithms for supporting the monitoring of data streams, which represents a more-detailed view of the activity (*ii*) specifically focused on OLAP-analysis aspects of the PP-OMDS framework; (*iv*) design of the privacy-preserving version of the OLAP-based models and related algorithms for supporting the monitoring of data streams, which represents a further refinement of models and algorithms defined by the activity (*iii*) but targeted to embedding privacy-preserving aspects of the PP-OMDS framework in such models and algorithms; (*v*) integration of the OLAP component and the privacy-preserving OLAP component, both oriented to data stream monitoring, into the PP-OMDS framework, according to several alternative computing models.

The scientific contribution of the research carried-out by the PP-OMDS framework is very high because state-of-the-art research, even though focused on the relevant problem of privacy-preserving OLAP over static data, lacks of proposals that are specifically focused on privacy-preserving OLAP over data streams, and, in addition to this, there is not a specific integration with monitoring aspects, which, contrary to this, are indeed very relevant for a wide spectrum of next generation data stream applications and systems.

Summarizing, the main research topics investigated by the PP-OMDS framework are the following:

- models, techniques and algorithms for supporting OLAP-based monitoring of data streams;
- models, techniques and algorithms for supporting the *privacy-preserving version* of OLAP-based monitoring of data streams;

- integration of models, techniques and algorithms devised in the context of the two previous topics/activities, according to several alternative computing models.

The paper describes motivations, principles and achievements of the PP-OMDS framework, along with technological advancements and innovations. We also incorporate a detailed comparative analysis with competitive frameworks, along with a trade-off analysis.

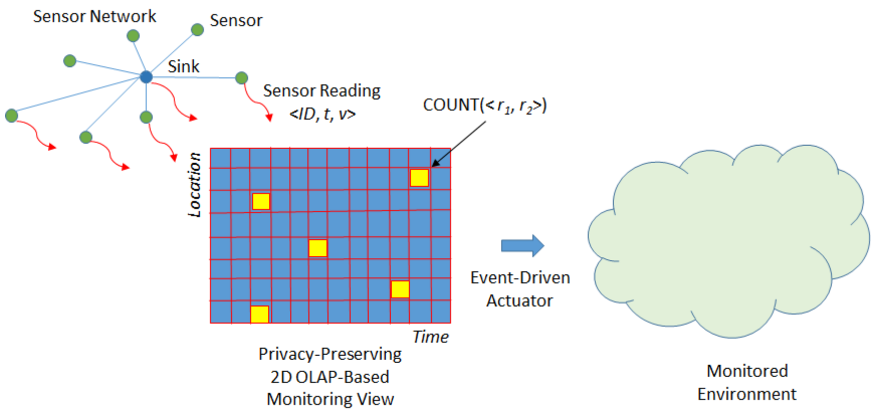

Fig. 1. A Typical Application Scenario for the PP-OMDS Project.

2 Reference Application Scenario

In the PP-OMDS framework, the main research focus is on the issue of supporting privacy-preserving OLAP-based monitoring of data streams, which, has highlighted above, is innovative in actual state-of-the-art research, and it is of relevant interest for a wide spectrum of data stream applications (e.g., security [6], emergency management [2], and so forth). Figure 1 shows a typical application scenario for the PP-OMDS framework. Here, a sensor network, composed by both *sink nodes* and *sensor nodes* (e.g., [7]), produces sensor readings of kind $\langle ID, t, v\rangle$ such that: (*i*) ID is the absolute identifier of the sensor node, (*ii*) t is the timestamp at which the sensor reading is produced, (*iii*) v is the proper reading (i.e., the value). A privacy-preserving 2D OLAP-based monitoring view V is interfaced to the sensor network directly, and it is used to monitor the reading value variable v, based on a complex multidimensional data model [8,9]. In the reference application scenario of Figure 1, the model introduces $Location$ as the first dimension and $Time$ as the second dimension, respectively, and the measure value is defined on top of a COUNT aggregate operator over a two-dimensional range $\langle r_1, r_2\rangle$, being r_1 and r_2 two one-dimensional ranges along $Location$ and $Time$, respectively, which define the data cube cell. The variable is monitored via suitable aggregate monitoring queries (e.g., [10]) over data cube cells that populate the OLAP view V.

With reference to the so-delineated application scenario, a critical challenge relies on effectively and efficiently computing the OLAP view V over sensor readings (i.e., streaming data), with the additional requirement that V must be computed in a privacy-preserving manner (e.g., [11] – for the static case, [13] – for the dynamic case), i.e. it must preserve the privacy of data sources that originate the sensor readings. The OLAP view V is meant for monitoring goals: when the COUNT-aggregated value of a data cube cell C_i exceeds a fixed threshold τ, then an *event* occurs and a suitable *action* (*trigger*, respectively) is activated in order to modify the target monitored environment. Obviously, the described application scenario is just an instance that represents even more sophisticated multidimensional settings, where the privacy-preserving OLAP-based monitoring view is characterized by multiple dimensions. It is worth to recognize that the described application scenario well-describes a plethora of modern data stream applications, ranging from event detection to complex monitoring queries support, from near-duplicate detection over multimedia streams to anomaly detection, and so forth, all with the innovative and challenging requirement of preserving the privacy of data streams, under OLAP analysis requirements (e.g., [12]), with monitoring purposes, being the latter the three main pillar of the PP-OMDS project.

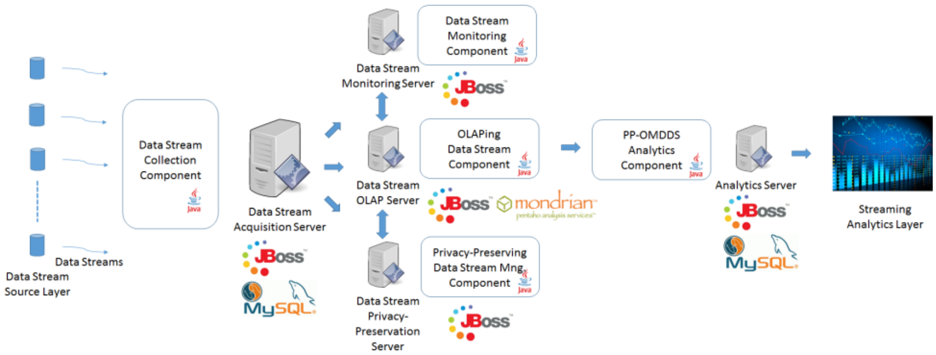

Fig. 2. PP-OMDS Framework Detailed Architecture.

3 PP-OMDS Architecture and Main Functionalities

Figure 2 shows the detailed architecture of the PP-OMDS framework, along with its components and reference technologies. These are:

- *Data Stream Source Layer* – The layer where data streams are produced by a target process (e.g., sensor networks, logistic networks, etc.). Technology used: arbitrary (at the input layer).
- *Data Stream Acquisition Server* – The server devoted to acquire data streams (by also providing the initial pre-processing – e.g., cleaning) and to make the necessary data model transformations. Technology used: *Java programming language*; *JBoss application server*, *MySQL database server*. In particular: (*i*) JBoss provides the necessary Java execution runtime environment for the *Data Stream Collection Component*, which is the component that implements classes and functions supporting the data

stream acquisition phase; (*ii*) My SQL database server provides the necessary data storage (even buffer-oriented) and management functionalities.

- *Data Stream Monitoring Server* – The server devoted to the monitoring of data streams, by implementing the algorithms proposed in the PP-OMDDS framework. It fully interacts with the *Data Stream OLAP Server* and the *Data Stream Privacy-Preservation Server*, respectively, in order to achieve the overall privacy-preserving OLAP-based monitoring of data streams pursued by the PP-OMDDS project. Technology used: Java programming language; JBoss application server. In particular, JBoss provides the necessary Java execution runtime environment for the *Data Stream Monitoring Component*, which is the component that implements classes and functions supporting the data stream monitoring phase.
- *Data Stream OLAP Server* – The server devoted to support OLAP over of data streams, by implementing the algorithms proposed in the PP-OMDDS framework. It fully interacts with the *Data Stream Monitoring Server* and the *Data Stream Privacy-Preservation Server*, respectively, in order to achieve the overall privacy-preserving OLAP-based monitoring of data streams pursued by the PP-OMDDS project. Technology used: Java programming language; JBoss application server, *Mondrian OLAP server*. In particular: (*i*) JBoss provides the necessary Java execution runtime environment for the OLAPing Data Stream Component, which is the component that implements classes and functions supporting OLAP analysis over data streams; (*ii*) Mondrian OLAP server provides the necessary multidimensional data storage and management functionalities.
- *Data Stream Privacy-Preservation Server* – The server devoted to support privacy-preserving management of data streams, by implementing the algorithms proposed in the PP-OMDDS project. It fully interacts with the *Data Stream Monitoring Server* and the *Data Stream OLAP Server*, respectively, in order to achieve the overall privacy-preserving OLAP-based monitoring of data streams pursued by the PP-OMDDS project. Technology used: Java programming language; JBoss application server. In particular, JBoss provides the necessary Java execution runtime environment for the *Privacy-Preserving Data Stream Management Component*, which is the component that implements classes and functions supporting the privacy-preserving data stream management phase.
- *Analytics Server* – The server devoted to support data stream analytics, according to the privacy-preserving OLAP-based vision pursued by the PP-OMDS framework. Technology used: Java programming language; JBoss application server; MySQL database server. In particular: (*i*) JBoss provides the necessary Java execution runtime environment for the *PP-OMDS Analytics Component*, which is the component that implements classes and functions supporting the PP-OMDS analytics phase; (*ii*) My SQL database server provides the necessary data storage (even buffer-oriented) and management functionalities.

- *Streaming Analytics Layer* – The layer where streaming analytics functionalities are finally implemented and delivered to the client applications/users, based on specific analytics goals (e.g., emergency detection and management, security, etc.). Technology used: arbitrary (at the output layer).

The PP-OMDS framework exploits software and hardware solutions that are currently available and completely technologically-feasible (even considering open-source solutions, in the case of software). Indeed, looking at the hardware architecture of the PP-OMDS framework, we identify an (hardware) architecture that makes use of standard solutions for (*i*) collecting data streams, (*ii*) monitoring data streams, (*iii*) aggregating data streams, (*iv*) providing privacy-preserving methods over data streams, and (*v*) supporting OLAP-like query answering over data streams. All these components are commonly delivered on top of well-known architectures composed by (*i*) data servers, (*ii*) OLAP servers, (*iii*) application servers, (*iv*) client components (e.g., desktop computers, laptops, mobile devices, etc.). As regards the software architecture of the PP-OMDS framework, we identify a (software) architecture populated by software components that can be developed by means of standard high-level programming languages (e.g., Java, C++, etc.) and standard data access and manipulation methods (e.g., JDBC, ODBC, etc.). It should be noted that, for both the hardware architecture and the software architecture of the PP-OMDS framework, the components to be developed adhere to well-assessed and mature technologies (both hardware and software) that clearly make the PP-OMDS framework completely-feasible. In addition to this, all the components of the PP-OMDS framework are already Cloud-enabled, hence the framework can be easily extended towards a Cloud-based application (hence, improving performance, reliability and availability).

4 Comparative Analysis with Competitive Frameworks

The software product context of the PP-OMDS framework is represented by the wide area of streaming analytics tools and systems, and, in particular, those devoted to support stream monitoring. Nevertheless, it does not exist a solution that specifically focuses on the relevant problem of supporting privacy-preserving, OLAP-based monitoring of distributed data streams, as delineated by the PP-OMDS framework's motivations. This confirms to us the innovativeness of our proposal.

In the following, we focus the attention on the specific block-founding problems that characterizes the PP-OMDS framework. For what regards the basic distributed data stream monitoring problem, some relevant tools and systems that are currently available are the following ones:

- *APAMA Streaming Analytics*, from Software AG;
- *Stream Computing*, from IBM;
- *Event Stream Processor*, from SAP;
- *Apache Spark*, from Cloudera.

For what regards both the basic OLAP analysis over data streams problem and the basic privacy-preserving data stream management problem, there do not exist direct tools and systems currently available, although vertical solutions on top of existing streaming analytics platforms can be devised.

The most critical limitation of competitor frameworks is represented by the fact that they all adhere to a common deployment model, i.e. providing a general platform for supporting data stream analytics. While on top of such (common) platform personalized solutions can still be developed (via ad-hoc IDE and functional programming environments), they do not focus on the specialized problem of supporting privacy-preserving OLAP-based monitoring of distributed data streams, despite the relevance of this problem and its target application scenarios (e.g., emergency management, security, and so forth).

On the other hand, developing personalized solutions on general-purpose streaming analytics platforms poses critical challenges for what regards a wide spectrum of issues, ranging from complexity overheads to (software) maintenance problems, from heterogeneous data format integration issues to analytical front-end tools that are not focused to the peculiarities of the target application (thus making the whole knowledge discovery from big streaming data harder), and so forth. All these considerations clearly suggest the adoption of an all-inside, self-contained analytical framework that, still being focused on the specific goal of supporting privacy-preserving OLAP-based distributed data stream monitoring, can be further specialized on particular, vertical application settings (still falling in the reference technological area).

Table 1. Comparative Analysis of Competitor Frameworks and the PP-OMDS Framework.

	Distributed Data Stream Monitoring	OLAP Analysis over Data Streams	Privacy-Preserving Data Stream Management	Support for General-Purpose Applications	Support for Vertical Applications
APAMA Streaming Analytics	✓	×	×	✓	×
Stream Computing	✓	×	×	✓	×
Event Stream Processor	✓	×	×	✓	×
Apache Spark	✓	×	×	✓	×
PP-OMDS	✓	✓	✓	✓	✓

In addition to this, the competitor frameworks do not provide an explicit support neither to OLAP analysis tools over data streams neither to privacy-preserving data stream management, making for them hard to follow the paradigms dictated by the PP-OMDS framework's paradigms. Table 1 proposes a summary on the comparative analysis of the competitor frameworks against the PP-OMDS framework, along the above-discussed parameters/functionalities.

5 Conclusions

In this paper, we propose have proposed PP-OMDS, an innovative framework for supporting the OLAP-based monitoring of data streams, which is relevant for a plethora of application scenarios (e.g., security, emergency management, and so forth), in a privacy-preserving manner.

References

1. R. Agrawal, R. Srikant, D. Thomas: Privacy-Preserving OLAP, *SIGMOD*, pp. 251-262, 2005
2. M. Gan, H. Dai: Detecting and monitoring abrupt emergences and submergences of episodes over data streams. *Information Systems 39*:277-289, 2014
3. C. Dwork, T. Pitassi, M. Naor, G.N. Rothblum: Differential privacy under continual observation, *STOC*, 2010
4. E. Shi, T.-H. H. Chan, E.G. Rieffel, R. Chow, D. Song: Privacy-preserving aggregation of time-series data, *NDSS*, 2011
5. L. Fan, L. Xiong. Real-time aggregate monitoring with differential privacy, *CIKM*, 2012
6. J. Zhou, O.C. Au, G. Zhai, Y.Y. Tang, X. Liu: Scalable Compression of Stream Cipher Encrypted Images Through Context-Adaptive Sampling. *IEEE Transactions on Information Forensics and Security 9(11)*:1857-1868, 2014
7. A. Cuzzocrea: Optimization issues of querying and evolving sensor and stream databases. *Information Systems 39*:196-198, 2014
8. Gray J., et al. Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. *Data Mining and Knowledge Discovery, 1(1)*: 29-54, 1997
9. A. Cuzzocrea: CAMS: OLAPing Multidimensional Data Streams Efficiently. *DaWaK*, pp. 48-62, 2009
10. A. Bulut, N. Koudas, A. Meka, A.K. Singh, D. Srivastava: Optimization Techniques for Reactive Network Monitoring. *IEEE Trans. Knowl. Data Eng. 21(9)*: 1343-1357, 2009
11. A. Cuzzocrea, V. Russo, D. Saccà: A Robust Sampling-Based Framework for Privacy Preserving OLAP. *DaWaK*, pp. 97-114, 2008
12. A. Cuzzocrea: Retrieving Accurate Estimates to OLAP Queries over Uncertain and Imprecise Multidimensional Data Streams. *SSDBM*, 2011, pp. 575-576
13. A. Cuzzocrea, E. Bertino: Privacy Preserving OLAP over Distributed XML Data: A Theoretically-Sound Secure-Multiparty-Computation Approach. *Journal of Computer and System Sciences 77(6)*: 965-987, 2011
14. A. Cuzzocrea, V. Russo: Privacy Preserving OLAP and OLAP Security. *Encyclopedia of Data Warehousing and Mining 2009*, pp. 1575-1581, 2009
15. A. Beimel, K. Nissim, E. Omri. Distributed private data analysis: Simultaneously solving how and what, *CRYPTO*, pp. 451-468, 2008