Open Access

Open Journal of Databases (OJDB)
Volume 6, Issue 1, 2019
http://www.ronpub.com/ojdb
ISSN 2199-3459

Special Issue on High-Level Declarative Stream Processing

Patrick Koopmann[A], Theofilos Mailis[B], Danh Le Phuoc[C]

[A]Institute of Theoretical Computer Science, Technische Universität Dresden, Nöthnitzer Straße 46, 01187 Dresden, Germany, patrick.koopmann@tu-dresden.de
[B]Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Zografou 161 22, Athens, Greece, tmailis@di.uoa.gr
[C]Institute of Telecommunication Systems, Technische Universität Berlin, Einsteinufer 25, 10587 Berlin, Germany, danh.lephuoc@tu-berlin.de

Abstract

Stream processing as an information processing paradigm has been investigated by various research communities within computer science and appears in various applications: realtime analytics, online machine learning, continuous computation, ETL operations, and more. The special issue on "High-Level Declarative Stream Processing" investigates the declarative aspects of stream processing, a topic of undergoing intense study. It is published in the Open Journal of Web Technologies (OJWT) (www.ronpub.com/ojwt). This editorial provides an overview over the aims and the scope of the special issue and the accepted papers.

Type of Paper and Keywords

Editorial: *Query processing on streams; Stream reasoning; Ontology based data access on streams; Context aware processing on streams; Complex event processing; Event recognition on streams; RDF/Linked Data stream processing; Data stream management systems; Benchmarking for querying/reasoning on streams;*

1 Introduction

We are pleased to present the special issue on "High-Level Declarative Stream Processing". Stream processing as an information processing paradigm has been investigated by various research communities within computer science and appears in various applications: realtime analytics, online machine learning, continuous computation, ETL operations, and more. Next to algorithmic oriented research on low-level stream processing, declarative stream processing frameworks, which are in the focus of this special issue, are also a topic of undergoing intense study.

Declarative stream processing frameworks, such as data stream management systems or systems for complex event processing, provide query languages with clear-cut semantics. Additionally, new demands such as expressive reasoning capabilities in stream processing pipelines have emerged, and attracted attention from both the data management and the knowledge representation and reasoning communities. Finally, combining declarative stream processing with provenance identification information, i.e., where and how a piece of data is produced, is an important factor in determining the quality of data items in data integration

This paper is accepted at the *Workshop on High-Level Declarative Stream Processing (HiDeSt 2018)* held in conjunction with the 41st German Conference on Artificial Intelligence (KI) in Berlin, Germany. The proceedings of HiDeSt@KI 2018 are published in the Open Journal of Databases (OJDB) as special issue.

applications.

The papers in this special issue discuss various aspects of declarative stream processing, focusing on: *(i)* knowledge representation, reasoning, and query answering over continuous data streams; *(ii)* the identification of data provenance in declarative stream-processing applications. All of the included papers indicate that currently, declarative stream processing is a fast growing and developing research area, with many different aspects worth being investigated.

2 Overview of Accepted Papers

Ontology-Based Data Access to Big Data [7]: *Ontology Based Data Access* (OBDA) is an approach to access information stored in multiple data sources via an abstraction layer that mediates between the data sources and data consumers. The OBDA layer uses an ontology to provide a uniform conceptual schema that describes the problem domain of the underlying data independently of how and where the data is stored, while declarative mappings are used to specify how the ontology is related to the data. The ontology and mappings are used to transform queries over ontologies, i.e., ontological queries, into data queries over data sources. Early research on OBDA use as backend sources simple relational database systems. However, recent theoretical and practical developments on distributed processing systems, and their extensive use in industry for statistical analytics on big data, have raised interest in considering specialised engines for big data processing as potential backends for OBDA.

Schiff et al [7]. present their insights in designing an OBDA system that uses a big-data-processing engine (Apache Spark) as a backend system and an OBDA framework for continuous queries (STARQL) as an ontology-level query language. Apache Spark is a unified analytics engine for big data processing, with built-in modules for streaming, SQL, machine learning, and graph processing [8]. The *Streaming and Temporal ontology Access with a Reasoning-based Query Language* (STARQL) framework allows to combine static conjunctive queries over $DL-Lite_{\text{agg}}$ ontologies with continuous diagnostic queries that involve simple combinations of time aware data attributes, time windows, and aggregation functions over streams of attribute values [6]. The STARQL-SPARK combination is evaluated in industrial scenarios for reactive diagnostics .

Multi-Shot Stream Reasoning in Answer Set Programming: A Preliminary Report [5]: A different methodology for knowledge representation and reasoning over continuous data streams constitutes in using *Answer Set Programming* (ASP) for stream reasoning. ASP is a well established paradigm for declarative problem solving. Rather than solving a problem by programming its solution, the basic idea of ASP is to declaratively describe a problem by a logic program and an ASP solver retrieves the corresponding solutions. One of the first systems that integrated ASP with stream reasoning was implemented in [2] as an exhaustive wrapper for the underlying ASP system clingo [3], enabling reasoning over continuous data streams. In this implenentation, the ASP solver has to be restarted for each new stream update, which leads to recomputations of the same implicit information between consecutive stream-processing cycles.

Obermeier et al [5]. suggest a different approach for integrating an ASP system into a streaming environment that refrains from recomputing implicit information during each stream-processing cycle. The specific approach exploits the multi-shot capabilities of modern ASP systems such as clingo. Multi-shot solving is a technique for processing continuously changing logic programs. In the context of stream reasoning, this allows to directly implement seamless sliding-window-based reasoning over emerging data by reusing the running clingo process and directly updating its internal knowledge base accordingly. In the paper of this special issue, Obermeier et al [5]. detail a barebones approach to leverage clingo's multi- shot solving capabilities for stream reasoning: they give an introduction to multi-shot solving, devise a basic stream reasoner based on clingo's multi-shot capabilities, and showcase the stream reasoner's workflow via job shop scheduling as a running example.

Provenance Management over Linked Data Streams [4]: The above-presented two papers [7, 5] of the special issue focus on employing declarative semantics in order to perform query-answering operations over continuous data streams. In many real-world environments, such as *Internet of Things Sensor Network* and *Smart Cities*, operations on data are often performed by multiple uncoordinated parties, each potentially introducing or propagating errors. These errors cause uncertainty of the overall data analytics process that is further amplified when many data sources are combined and errors get propagated across multiple parties. Understanding where and how a piece of data is produced, represented by its provenance, has long been recognized as an important factor in determining the quality of a data item particularly in data integration systems.

Qian Liu et al. [4] propose a methodology to

compute provenance of continuous queries executed over complete Linked Data streams [1]. Unlike traditional provenance management techniques, which are designed for processing static information, their approach focuses on the dynamicity and heterogeneity of Linked Data streams. Qian Liu et al. define the stream provenance model, propose and implement a system architecture for processing continuous queries over linked data streams, identify execution strategies that focus on executing continuous queries and tracing their provenance, and evaluate their techniques and implementations on two different datasets, the Billion Triples Challenge and the Web Data Commons.

3 Conclusions and Outlook

While the above articles focus on different levels of abstraction of stream data and processing operations, they all use layering approach to translate their declarative languages to processing operations that are later handled by the targeted processing engines or components. This commonality shows that high-level declarative stream processing is a popular paradigm for dealing with complex stream pipelines across different disciplines. As as result, a very complex stream processing query can be made significantly simpler via the declarative programming models that hide several levels of abstraction. Such abstraction levels are enabled by different compiling mechanisms that map the corresponding processing components to execution pipelines like the ones discussed in these articles.

This brings us to the first interesting outlook on a cross-disciplinary effort to tightly integrate different levels of processing abstractions to enable a very high-level declarative stream processing model, i.e. a system and a unified declarative language that will cover the features of all three above articles. The second outlook is what are the theoretical impossibilities and possibilities which lead to the right balance between expressiveness and scalability. As all of these articles are in an early stage of development, it will be interesting to see the practicality of these proposals and the likes in another outlook of this special issue.

Acknowledgements

The editors would like to extend their gratitude to all the reviewers for their comprehensive reviews, as well as all other people who are directly and indirectly involved in getting this issue ready for publication.

References

[1] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data: The story so far," in *Semantic services, interoperability and web applications: emerging concepts*, 2011, pp. 205–227.

[2] M. Gebser, T. Grote, R. Kaminski, P. Obermeier, O. Sabuncu, and T. Schaub, "Stream reasoning with answer set programming: Preliminary report." *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning (KR)*, vol. 12, pp. 613–617, 2012.

[3] M. Gebser, R. Kaminski, B. Kaufmann, and T. Schaub, "Clingo= asp+ control: Preliminary report," *arXiv preprint arXiv:1405.3694*, 2014.

[4] Q. Liu, M. Wylot, D. L. Phuoc, and M. Hauswirth, "Provenance management over linked data streams," *Open Journal of Databases (OJDB)*, vol. 6, no. 1, pp. 5–20, 2019. [Online]. Available: https://www.ronpub.com/ojdb/OJDB_2019v6i1n02_QianLiu.html

[5] P. Obermeier, J. Romero, and T. Schaub, "Multi-shot stream reasoning in answer set programming: A preliminary report," *Open Journal of Databases (OJDB)*, vol. 6, no. 1, pp. 33–38, 2019. [Online]. Available: https://www.ronpub.com/ojdb/OJDB_2019v6i1n04_Obermeier.html

[6] Ö. L. Özçep, R. Möller, and C. Neuenstadt, "A stream-temporal query language for ontology based data access," in *Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)*. Springer, 2014, pp. 183–194.

[7] S. Schiff, R. Möller, and Ö. L. Özcep, "Ontology-based data access to big data," *Open Journal of Databases (OJDB)*, vol. 6, no. 1, pp. 21–32, 2019. [Online]. Available: https://www.ronpub.com/ojdb/OJDB_2019v6i1n03_Schiff.html

[8] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin *et al.*, "Apache spark: a unified engine for big data processing," *Communications of the ACM*, vol. 59, no. 11, pp. 56–65, 2016.

AUTHOR BIOGRAPHIES

Patrick Koopmann is a researcher at the Technische Universität Dresden. His research interests include Description Logics, temporal and probabilistic logics, non-classical reasoning and ontology-based data access. Patrick Koopmann received his PhD at The University of Manchester, and is currently supported by the DFG within the collaborative research center SFB 912 (HAEC).

Theofilos Mailis is an Adjunct Researcher at National and Kapodistrian University of Athens. His research interests include most aspects of knowledge representation and automated reasoning, focusing on crisp and fuzzy Descriptions Logics, ontology based technologies, stream processing, and data analysis.

Danh Le Phuoc is a Marie Sklodowaka-Curie Fellow at the Technical University of Berlin. His research interests include linked data and the Semantic Web for the future internet, big data for the Internet of Everything, databases, and pervasive computing. Le Phuoc received a PhD in computer science from the National University of Ireland.