Multi-model Databases and Tightly Integrated Polystores: Current Practices, Comparisons, and Open Challenges

Jiaheng Lu*
University of Helsinki
Finland
jiaheng.lu@helsinki.fi

Irena Holubová†
Charles University
Prague, Czech Republic
holubova@ksi.mff.cuni.cz

Bogdan Cautis
University of Paris-Sud
Orsay, France
bogdan.cautis@u-psud.fr

ABSTRACT

One of the most challenging issues in the era of Big Data is the "*Variety*" of the data. In general, there are two solutions to directly manage multi-model data currently: a single integrated multi-model database system or a tightly-integrated middleware over multiple single-model data stores. In this tutorial, we review and compare these two approaches giving insights on their advantages, trade-offs, and research opportunities. In particular, we dive into four key aspects of technology for both types of systems, namely (1) theoretical foundation of multi-model data management, (2) storage strategies for multi-model data, (3) query languages across models, and (4) query evaluation and its optimization. We provide a comparison of performance for the two approaches and discuss related open problems and remaining challenges. The slides of this tutorial can be found at http://udbms.cs.helsinki.fi/?tutorials/CIKM2018.

KEYWORDS

Multi-model databases, Category theory, Polystores, Big Data, Variety of data

ACM Reference Format:
Jiaheng Lu, Irena Holubová, and Bogdan Cautis. 2018. Multi-model Databases and Tightly Integrated Polystores: Current Practices, Comparisons, and Open Challenges. In *The 27th ACM International Conference on Information and Knowledge Management (CIKM '18), October 22–26, 2018, Torino, Italy.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3269206.3274269

1 INTRODUCTION

One of the most challenging issues of Big Data is their "*Variety*" [4, 6]. The data may be presented in various types and formats – structured, semi-structured and unstructured – and produced by different sources, and hence natively have various models. To address this challenge, probably the first type of respective specific database management systems (DBMSs) are *NoSQL databases* which can be further classified to *single-model* and *multi-model*. The latter type enables to store and process structurally different data, i.e., data with distinct models. Even the Gartner Magic quadrant for operational database management systems [5] assumes that, by 2017, all leading operational DBMSs offered multiple data models, relational and NoSQL, in a single DBMS platform.

An alternative to the single-store multi-model DBMS, yet bearing important similarities, has also made its way in recent research, e.g., [1, 2], which does not assume a single store capable of supporting various data models, but multiple dedicated ones under the "hood" of a tightly-integrated platform, which represents the single point of access for all data related and administration tasks. Such *tightly-integrated polystores* [3] can be seen as a particular kind of federated databases, which trade autonomy (of the model-dependent stores) for efficiency and usability in practical, possibly cloud-oriented enterprise scenarios. This can be seen as their main common trait to a multi-model DBMS, leading in many aspects to similar ideas and challenges, but presenting also major differences.

In this tutorial, we review the previous work on single-store multi-model database management systems (denoted hereafter, for simplicity, multi-model DBMSs) and tightly integrated polystores, giving insights on their advantages, trade-offs, and research opportunities. We set the tutorial's scope on these two areas in multi-model data management, which could be seen as two sides of the same coin, leaving other related areas aside in order to take a closer look at their performance and applicability in scenarios requiring centralized and efficient access to data in multiple formats.

First, we show that the idea of multi-model DBMS is not at all a novel approach. Indeed, it can be traced back to Object-Relational Data Management Systems (ORDBMSs) in the early 1990s and, in a more broader scope, even to federated and integrated DBMSs in the early 1980s. Recently, we can observe a similar trend among NoSQL databases, with the support of multiple data models against a single, integrated backend, while meeting the growing requirements for scalability and performance. Similarly, tightly-integrated polystores can be seen as the latest incarnation of the federated databases or multi-database systems [10], which have been studied extensively, yet failed to have the expected impact in industry. Like multi-model DBMSs, polystores are in part motivated by the advent of the NoSQL principles in recent years, but they are also motivated by the technological breakthrough that is *cloud data management*.

Second, we dive in the key aspects of technology for multi-model DBMS and tightly-integrated polystores, namely (1) category theory as a theoretical foundation, (2) storage strategies for multi-model data, (3) query languages accessing data across multiple models, and (4) query evaluation and its optimization in the context of multiple data models. Last but not least, we provide a comparison of features and performance for the two directions, and we discuss related open problems and remaining challenges.

*Supported by the Academy of Finland (310321).
†Supported by the MŠMT ČR grant PROGRES.

CIKM '18, October 22–26, 2018, Torino, Italy

ACM ISBN 978-1-4503-6014-2/18/10.
https://doi.org/10.1145/3269206.3274269

2 TUTORIAL ORGANIZATION

Motivation (10'). We motivate the need for multi-model data management by several examples in the era of Big Data.

History and classification (20'). We introduce the history and classification of multi-model databases, including ORDBMS, NoSQL databases, Polyglot persistence [12], as well as the one of federated databases [10] and polystores [12]

Category theory as a theoretical foundation (15'). We introduce some basic ideas from category theory and their application on multi-model data [11] .

Multi-model data storage (15'). We introduce various methods to store multi-model data, including data layouts such as object-relational, graph, document, or native hierarchical.

Multi-model data query languages (15'). We compare languages for multi-model data processing, such as AQL, SQL++, OrientDB SQL, and SQL/XML.

Multi-model query processing (15'). We overview the multi-model extensions of traditional query processing approaches, such as B+ tree, schema discovery, and cross-model query processing.

Overview on tightly integrated polystores (20'). We describe several of the most recent polystore systems, with a particular focus on the models they rely on, their storage and querying capabilities.

Query processing in tightly integrated polystores (20'). We then discuss the various query processing and optimization approaches that are employed by the reference systems.

Advanced aspects of tightly integrated polystores (15'). Current advances and challenges in aspects such as self-tuning, data ingestion and placement, distributed transactions are discussed.

Comparison of multi-model databases and tightly integrated polystores (15'). We give a real application scenario to compare multi-model databases and tightly integrated polystores for their features and architecture, trades-off and differences [7, 9, 13].

Open problem and challenges (20'). We conclude with a discussion of the major open problems and challenges.

3 PREVIOUS TUTORIAL

A tutorial with a similar topic was prepared by J. Lu and I. Holubová for EDBT 2017 [8]. This proposed tutorial is a logical continuation, which differs in the following aspects:

(1) An entirely new part on the tightly-integrated polystores.
(2) An entirely new part on the theoretical foundation – category theory on multi-model data management
(3) Information on new functionalities and changes of the described multi-model DBMSs made since Spring 2017.
(4) New systems that have recently become multi-model following the Gartner's market hypothesis [5].
(5) Comparison of the key features of the described systems related to the support of multiple models.

In general, the estimated new content is more than 50%.

4 INTENDED AUDIENCE

This tutorial is intended for a wide scope of audience, e.g., for developers and architects to get insights from the emerging industrial trends and its connections to scientific research, for stakeholders to make wise and informed decisions on investments in multi-model data management products, for motivated researchers and developers to select new topics and contribute their expertise on multi-model data, and for new developers and students to quickly gain a comprehensive picture and understand the new trends and the state-of-art techniques in this field.

5 SHORT BIBLIOGRAPHIES

Jiaheng Lu is an Associate Professor at the University of Helsinki, Finland. His main research interests lie in the Big Data management and database systems. He has published more than eighty journal and conference papers. He has published several books, on XML, Hadoop and NoSQL databases. His book on Hadoop is one of the top-10 best-selling books in the category of computer software in China in 2013. He has frequently served as a PC member for conferences including SIGMOD, VLDB, ICDE, EDBT, CIKM etc.

Irena Holubová is an Associate Professor at the Charles University, Prague, Czech Republic. Her current main research interests include Big Data management and NoSQL databases, evolution and change management of database applications, analysis of real-world data, and schema inference. She has published more than 80 conference and journal papers; her works gained 4 awards. She has published 2 books on XML and NoSQL databases.

Bogdan Cautis is a Professor at the Department of Computer Science of University of Paris-Sud, France, since Sept. 2013. Before that, he was an Associate Professor at Telecom Paristech, Paris (2007–2013). His current research interests lie in the broad area of data management and information retrieval, including social data management and database theory.

REFERENCES

[1] Azza Abouzeid, Kamil Bajda-Pawlikowski, Daniel Abadi, Avi Silberschatz, and Alexander Rasin. 2009. HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads. *VLDB* 2, 1 (2009), 922–933.
[2] Raphaël Bonaque et al. 2016. Mixed-instance querying: a lightweight integration architecture for data journalism. *PVLDB* 9, 13 (2016), 1513–1516.
[3] Carlyna Bondiombouy and Patrick Valduriez. 2016. Query processing in multi-store systems: an overview. *IJCC* 5, 4 (2016), 309–346.
[4] Jinchuan Chen, Yueguo Chen, Xiaoyong Du, Cuiping Li, Jiaheng Lu, Suyun Zhao, and Xuan Zhou. 2013. Big data challenge: a data management perspective. *Frontiers Comput. Sci.* 7, 2 (2013), 157–164.
[5] Donald Feinberg, Merv Adrian, Nick Heudecker, Adam M. Ronthal, and Terilyn Palanca. 2015. Gartner Magic Quadrant for Operational Database Management Systems. (2015).
[6] Zhen Hua Liu, Jiaheng Lu, Dieter Gawlick, Heli Helskyaho, Gregory Pogossiants, and Zhe Wu. 2018. Multi-Model Database Management Systems - a Look Forward. *Poly* (2018).
[7] Jiaheng Lu. 2017. Towards Benchmarking Multi-Model Databases. In *CIDR*.
[8] Jiaheng Lu and Irena Holubová. 2017. Multi-model Data Management: What's New and What's Next?. In *EDBT*. 602–605.
[9] Jiaheng Lu, Zhen Hua Liu, Pengfei Xu, and Chao Zhang. 2016. UDBMS: Road to Unification for Multi-model Data Management. *CoRR* abs/1612.08050 (2016).
[10] M. Tamer Özsu and Patrick Valduriez. 1999. *Principles of Distributed Database Systems, Second Edition.* Prentice-Hall.
[11] David I. Spivak and Ryan Wisnesky. 2015. Relational foundations for functorial data migration. In *DBPL*. 21–28.
[12] Michael Stonebraker. 2015. The Case for Polystores. (2015). http://wp.sigmod.org/?p=1629
[13] Chao Zhang, Jiaheng Lu, Pengfei Xu, and Yuxing Chen. 2018. UniBench: A Benchmark for Multi-Model Database Management Systems. In *TPCTC*.