OBDF: OBDA + Data Federation – Extended Abstract

Zhenzhen Gu
Nanchang University (China)
zhenzhen.gu@ncu.edu.cn

Diego Calvanese
Free University of Bozen-Bolzano (Italy)
Umeå University, Umeå, (Sweden)
calvanese@inf.unibz.it

Marco Di Panfilo
Free University of Bozen-Bolzano (Italy)
mdipanfilo@unibz.it

Davide Lanti
Free University of Bozen-Bolzano (Italy)
lanti@inf.unibz.it

Alessandro Mosca
Free University of Bozen-Bolzano (Italy)
mosca@inf.unibz.it

Guohui Xiao
University of Bergen (Norway)
guohui.xiao@uib.no

***Abstract*—Ontology-Based Data Access (OBDA) has emerged as a well-established approach to information management, facilitating access to a *sole relational* relational database via a high-level ontology and declarative mappings. In response to the challenges posed by Big Data, we propose the Ontology-Based Data Federation (OBDF) framework, which merges OBDA with Data Federation. This merging allows for the integration of numerous, distributed, and heterogeneous data sources in a virtual, uniform, and semantically coherent fashion.**

***Index Terms*—OBDA, VKG, OBDF, Data Federation, Query Optimization**

I. Introduction and Motivation

ONTOLOGY-Based Data Access (OBDA) [?], [?], [?], also known as Virtual Knowledge Graphs (VKGs) within the context of Semantic Web, is a well-established [?] paradigm that allows for transparent access to data via a mediating ontology. The ontology furnishes end-users with friendly vocabulary aligned with the application domain, concealing the intricacies of storage conventions. Such ontology is expressed in a lightweight conceptual language, such as OWL 2 QL [?], which has its formal foundations in the DL-Lite family [?] of description logics. In OBDA, it is assumed that the underlying data are stored in a single relational data source, to which the ontology elements are mapped in a declarative way.

With the rapid development of the digitalization process, enterprises and institutions typically have accumulated large amounts of multiple, heterogeneous and distributed data sources, like RDBs, NoSQL DBs, and CSV files. Data only becomes more valuable when considered together. Thus, we introduce Ontology-Based Data Federation (OBDF for short) with the motivation of integrating and accessing multiple, heterogeneous and distributed data sources in a uniform and semantic way. This work is an extended abstract of [?].

II. Contribution: OBDF

Figure 1 depicts the full process of query answering in OBDF. A SPARQL query is posed over the VKG exposed by the OBDA system. The OBDA system exploits the mappings and the ontology to *reformulate* the SPARQL query over the VKG into a SQL query over the federated data source. Such

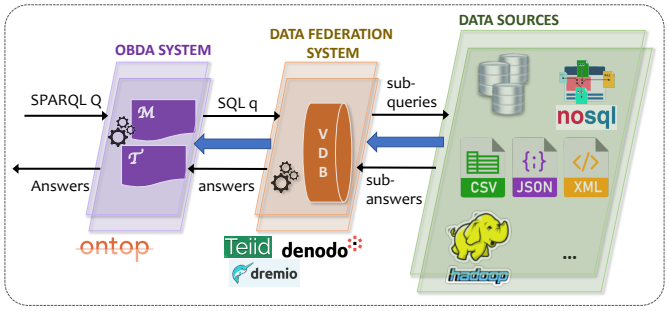

Fig. 1. The proposed framework for Ontology-Based Data Federation.

SQL query is forwarded to the data federation engine, like Denodo or Dremio, in order to be evaluated by the data sources. The challenge in this framework is to complement structural and semantic optimizations adopted in OBDA [?], [?], [?], [?], [?], [?] with the need of minimizing the number of federated operations, that is, of operations requiring the combination of data coming from multiple sources. To this aim, we devise novel optimization strategies, that we called *hint-based* [?], allowing for the generation of SQL queries that can be evaluated over multiple sources via federation systems in an efficient way. Hints are gathered through an offline analysis of the sources, driven by the mappings specification and axioms in the ontology.

Figure 3 presents the optimization rules developed for OBDF. These rules detail the application of hint-based optimizations, taking into account the precomputed data hints relevant to the process. Note that these data hints are specific to a particular data source instance, denoted by the subscript $\mathbb{D}$ in Figure 3. The optimizations transform relational algebra expressions with unions ($\cup$), joins ($\bowtie$), and left-joins (⟕) into equivalent expressions (modulo the hints for the considered data instance). The application of rules is decided according to the collected hints (labels of the edges), plus an additional condition $\dagger$ for rules **ce** and **eje** signaling the application of a cost function. Notation $\emptyset_{\mathsf{sig}(J)}$ denotes the empty (named) relation of signature J, and M denotes a set of materialized views produced during the hints pre-computation phase.

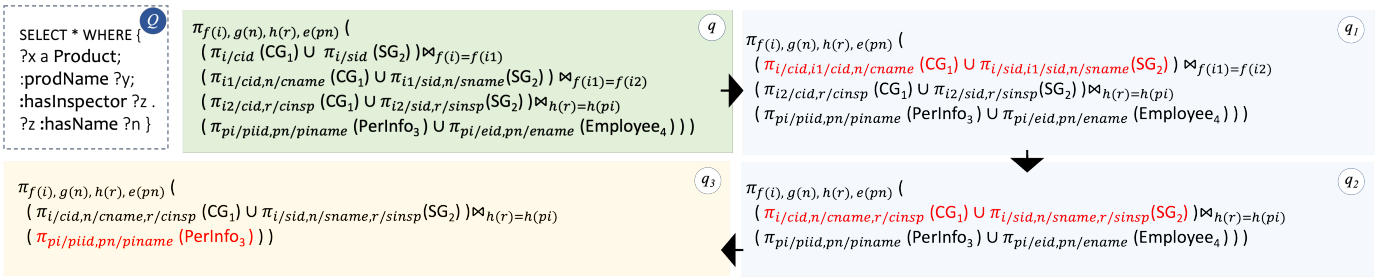

Fig. 2. An example of hint-based query optimization in OBDF: The sequence from step q to q_1, and further to q_2, utilizes the **eje** rule, supported by the data hint $\mathsf{CG}_1 \bowtie \mathsf{SG}_2 \equiv_{\mathbb{D}} \emptyset$. Subsequently, the progression from q_2 to q_3 leverages the **ce** rule, together with the data hint $\mathsf{Employee}_4 \subseteq_{\mathbb{D}} \mathsf{PerInfo}_3$.

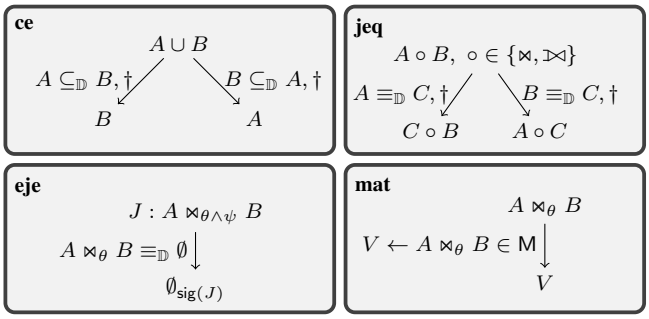

Fig. 3. List of optimization rules applied in OBDF. **ce**: containment elimination rule, **jeq**: join-equality rule, **eje**: empty join elimination rule, **mat**: materialization rule

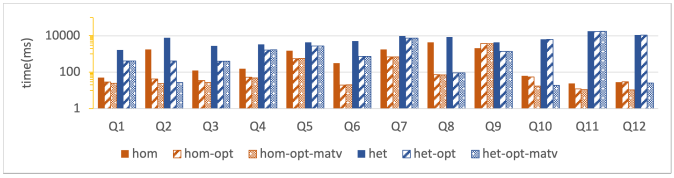

Fig. 4. SQL evaluation (in ms) with BSBM scale factor 20K.

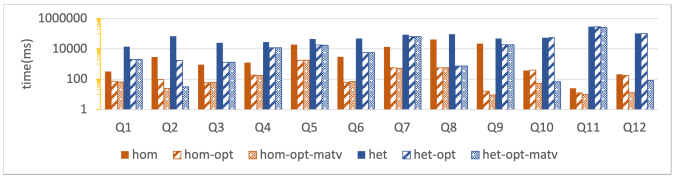

Fig. 5. SQL evaluation (in ms) with BSBM scale factor 200K.

In [**?**], we introduce the algorithm $\mathsf{unfold}_{\mathsf{OBDF}}$, demonstrating the application of these novel rules to the reformulation process of a query and their combination with standard optimization techniques such as self-join elimination.

Figure 2 shows an example of hint-based query rewriting. The subscripts for the relations in the figure indicate the data source. For instance, CG_1 denotes relation CG over data source S_1. Federated joins, which involve multiple sources, are known to be time-consuming [**?**], [**?**]. With an empty federated join hint $\mathsf{CG}_1 \bowtie \mathsf{SG}_2 \equiv_{\mathbb{D}} \emptyset$ and a redundancy hint $\mathsf{Employee}_4 \subseteq_{\mathbb{D}} \mathsf{PerInfo}_3$, the complex SQL query q (output of a "standard" reformulation procedure) can be simplified to q_3, containing a single federated join.

III. Experiments

Extensive experiments have been conducted to evaluate the effectiveness of the proposed hint-based query optimization approach. We utilized mainstream database engines, such as PostgreSQL, MySQL, and MongoDB, along with the OBDA system Ontop and the data federation system Teiid. Based on the BSBM benchmark [**?**], two OBDF specifications were created: a homogeneous one, employing five relational databases, and a heterogeneous one, in which some of these five sources are converted into NoSQL databases and CSV files. Figures 4 and 5 display the results from the experiments [**?**] for the BSBM scales 20k and 200k, respectively. In these figures, hom and het represent the results of answering queries in the homogeneous and heterogeneous settings without hint-based optimization. The term -opt is used to indicate the application of only empty federated join hints and redundancy relation hints in query rewriting, while -opt-matv denotes the results where all hints, including those for materialized views, are applied. This configuration demonstrates a significant improvement in query evaluation time, confirming the efficiency of the proposed optimization techniques across both homogeneous and heterogeneous data sources.

IV. Conclusions and Future Work

This study introduces OBDF and proposes techniques for optimizing query answering. Our empirical evaluation demonstrates how applying these techniques has a huge impact on performance.

Our approach is not devoid of bottlenecks. Note that, although hints provide ways of optimizing the queries, they are dependent of the data instance being considered. Provided that computing hints at query time is unfeasible, hints should be computed in an offline phase and therefore only involve sources that are not expected to change frequently. Another issue to pay attention to is how to determine which federated operations should be materialized, as applying **mat** rule blindly can potentially lead to an explosion in storage size.

Future research will focus on implementing the approach into the state-of-the-art OBDA system Ontop [**?**], [**?**], devising further optimization strategies, as well as effective "filters" to tame the explosion of materialized views.

ACKNOWLEDGMENT

This work has been carried out while Marco Di Panfilo was enrolled in the Italian National Doctorate on Artificial Intelligence run by Sapienza University of Rome in collaboration with Free University of Bozen-Bolzano. This research has been partially supported by the EU-funded project CyclOps (grant agreement No. 101135513).

REFERENCES

[1] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati, "Linking data to ontologies," *J. on Data Semantics*, vol. 10, pp. 133–173, 2008.

[2] D. Calvanese, B. Cogrel, S. Komla-Ebri, R. Kontchakov, D. Lanti, M. Rezk, M. Rodriguez-Muro, and G. Xiao, "Ontop: Answering SPARQL queries over relational databases," *Semantic Web J.*, vol. 8, no. 3, pp. 471–487, 2017.

[3] G. Xiao, D. Calvanese, R. Kontchakov, D. Lembo, A. Poggi, R. Rosati, and M. Zakharyaschev, "Ontology-based data access: A survey," in *Proc. of the 27th Int. Joint Conf. on Artificial Intelligence (IJCAI)*. IJCAI Org., 2018, pp. 5511–5519.

[4] G. Xiao, L. Ding, B. Cogrel, and D. Calvanese, "Virtual Knowledge Graphs: An overview of systems and use cases," *Data Intelligence*, vol. 1, no. 3, pp. 201–223, 2019.

[5] B. Motik, B. Cuenca Grau, I. Horrocks, Z. Wu, A. Fokoue, and C. Lutz, "OWL 2 Web Ontology Language profiles (second edition)," World Wide Web Consortium, W3C Recommendation, Dec. 2012, available at http://www.w3.org/TR/owl2-profiles/.

[6] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati, "Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family," *J. of Automated Reasoning*, vol. 39, no. 3, pp. 385–429, 2007.

[7] Z. Gu, D. Lanti, A. Mosca, G. Xiao, J. Xiong, and D. Calvanese, "Ontology-based data federation," in *Proc. of the 11th Int. Joint Conf. on Knowledge Graphs (IJCKG)*. ACM, 2022, pp. 10–19.

[8] J. F. Sequeda and D. P. Miranker, "Ultrawrap: SPARQL execution on relational data," *J. of Web Semantics*, vol. 22, pp. 19–39, 2013.

[9] M. Rodriguez-Muro, R. Kontchakov, and M. Zakharyaschev, "Ontology-based data access: Ontop of databases," in *Proc. of the 12th Int. Semantic Web Conf. (ISWC)*, ser. Lecture Notes in Computer Science, vol. 8218. Springer, 2013, pp. 558–573.

[10] D. Lanti, G. Xiao, and D. Calvanese, "Cost-driven ontology-based data access," in *Proc. of the 16th Int. Semantic Web Conf. (ISWC)*, ser. Lecture Notes in Computer Science, vol. 10587. Springer, 2017, pp. 452–470.

[11] D. Bilidas and M. Koubarakis, "In-memory parallelization of join queries over large ontological hierarchies," *Distributed Parallel Databases*, vol. 39, no. 3, pp. 545–582, 2021. [Online]. Available: https://doi.org/10.1007/s10619-020-07305-y

[12] G. Xiao, D. Lanti, R. Kontchakov, S. Komla-Ebri, E. Güzel-Kalayci, L. Ding, J. Corman, B. Cogrel, D. Calvanese, and E. Botoeva, "The virtual knowledge graph system Ontop," in *Proc. of the 19th Int. Semantic Web Conf. (ISWC)*, ser. Lecture Notes in Computer Science, vol. 12507. Springer, 2020, pp. 259–277.

[13] A. Schwarte, P. Haase, K. Hose, R. Schenkel, and M. Schmidt, "FedX: Optimization techniques for federated query processing on linked data," in *Proc. of the 10th Int. Semantic Web Conf. (ISWC)*, ser. Lecture Notes in Computer Science, vol. 7031. Springer, 2011, pp. 601–616.

[14] M. Saleem and A. N. Ngomo, "Hibiscus: Hypergraph-based source selection for SPARQL endpoint federation," in *The Semantic Web: Trends and Challenges - 11th International Conference, ESWC 2014, Anissaras, Crete, Greece, May 25-29, 2014. Proceedings*, ser. Lecture Notes in Computer Science, V. Presutti, C. d'Amato, F. Gandon, M. d'Aquin, S. Staab, and A. Tordai, Eds., vol. 8465. Springer, 2014, pp. 176–191. [Online]. Available: https://doi.org/10.1007/978-3-319-07443-6_13

[15] C. Bizer and A. Schultz, "The Berlin SPARQL benchmark," *Int. J. on Semantic Web and Information Systems*, vol. 5, no. 2, pp. 1–24, 2009.