Design and Development of a Polystore System for Heterogeneous Biomedical Data

Mirco Cazzaro[1,*]

[1]Department of Information Engineering, University of Padova, Via Gradenigo 6/B, Padova, 35131, Italy

Abstract

Biomedical data management is increasingly complex due to the variety of storage systems and evolving data models. This heterogeneity presents obstacles to data integration and querying, crucial for advancing biomedical research and healthcare. A possible solution is to employ a recent idea in the database field: polystores. A polystore is a DBMS designed to integrate and manage multiple heterogeneous data stores, allowing for efficient data processing and querying across diverse data models and storage systems. Nevertheless, polystore systems may differ profoundly one with the other, both in their structure and in the interface provided to the users: this is usually due to the diverse landscapes where polystores may be applied, and thus the diverse nature of data available in different fields, and to the information needs that users may have. These limitations impede the adoption of existing polystores in the context of biomedical data. Moreover, as far of our knowledge, there exist no system offering an integrated viewpoint to biomedical data by means of a graph data model, which would instead provide a sharpened representation of this domain. In this paper, we outline the research challenges and the initial steps towards the development of a polystore system that provides efficient access to multiple heterogeneous biomedical data sources, addressing also critical privacy concerns by tracing data flow and ensuring the privacy and anonymity of individuals.

Keywords

Data Integration, Data Federation, Polystore Systems

1. Introduction

Managing biomedical data poses significant challenges. Practitioners are confronted with numerous disparate storage systems, each employing different data models that may evolve over time. This leads to a high degree of heterogeneity, where the same data domain can be represented using various models such as relational, hierarchical or graph-based. In particular, graphs are highly used for biomedical data, because their structure naturally mirrors the complex and interconnected relationships found in biological systems. Moreover, these systems are often managed by different Database Management Systems (DBMSs). Additionally, the data is described by diverse metadata schemas, leading to more heterogeneity. This low level of integration hinders the possibility of querying them together and infer new knowledge, unless experts manually integrate them, by defining a unified data model, achieving consensus among data stakeholders, manually matching existing data to this new model, migrating data accordingly, and then modifying applications to adapt to changes in the used query language. All these steps are time-consuming and expensive.

At the European level, numerous contexts exist where the aforementioned challenges are relevant. Specifically, the Department of Information Engineering at the University of Padova leads the EU project HEREDITARY, which involves collaboration with three medical centers, encompassing heterogeneous multimodal clinical and genomic data, which require integration. HEREDITARY underscores the pressing demand for a robust and efficient system capable of seamlessly integrating diverse biomedical datasets.

A possible solution is to employ a recent advancement in the database field: polystores. A polystore

SEBD 2025: 33rd Symposium on Advanced Database Systems, June 16-19, 2025, Ischia, Italy

[*]Corresponding author.

mirco.cazzaro@phd.unipd.it (M. Cazzaro)
https://dei.unipd.it/~cazzaromir (M. Cazzaro)
0009-0006-3856-7207 (M. Cazzaro)

is a system that integrates multiple heterogeneous databases (SQL, NoSQL, graph, document stores, etc.) under a unified query interface. It enables querying data across different types of databases without physically moving or transforming the data. Polystores uses a query federation approach, where queries are translated and executed across different underlying databases, leveraging their native query engines.

The state of the art is represented by BigDAWG [1], developed within the Intel Science and Technology Center on Big Data. BigDAWG's architecture consists of four main layers: the base layer, which includes diverse physical data stores such as relational databases and column-oriented DBMS; the island layer, containing independent software components designed to facilitate querying across different database types; the main BigDAWG layer, which manages query processing and dispatching; and the application layer, responsible for user interaction. Despite being well documented[1], BigDAWG functions more as a prototype, requiring significant effort to adapt to our needs. Another relevant system is CloudMdsQL [2], which also aims at querying heterogeneous data sources in cloud environments. However, CloudMdsQL presents several limitations that make it unsuitable for our purposes. Firstly, it supports queries primarily against unimodal databases, focusing mainly on tabular data, which limits its applicability to more diverse or multimodal data environments. Secondly, it relies on an ad-hoc query language, which necessitates additional training and effort from end-users, hindering ease of adoption. Finally, CloudMdsQL employs a Global-As-View (GAV) approach for schemas integration, which significantly increases maintenance overhead due to the complexity and rigidity of maintaining global views when underlying data schemas evolve.

Considering other polystore solutions, among the most interesting and actively maintained academic projects is Polypheny [3]. This open-source project appears promising but is still in the early stages of development and has a limited user base, raising concerns about its future sustainability and continued development. Additionally, Polypheny currently lacks essential features such as a logging system, user management capabilities, and support for graph data, crucial aspects for a polystore intended for application within the biomedical domain.

Another relevant aspect is how integrated data is modeled. As mentioned before, a graph-based data model is necessary to represent the intricate pattern of such a complex domain: this translates into the need of an ontology through which data is accessed. The Ontology-Based Data Access (OBDA) paradigm [4] combines the capabilities of polystore systems with those of graph databases. OBDA leverages ontologies to introduce a semantic layer, providing a conceptual representation that significantly simplifies data querying and interpretation. Moreover, ontologies enhance graph databases with inferential algorithms, enabling the derivation of new knowledge from existing data. OBDA acts by means of mappings, through which ontology patterns described in SPARQL queries are unfolded into relational queries.

This paper's sections are organized as follows: section 2 and 3 describe in details both the approach and the proposed framework, analyzing how its structures allows to tackle our challenges. Section 4 derives instead open research questions coming from the framework weaknesses, and how we plan to address them; section 5 concludes the paper.

2. OBDF: OBDA + Data Federation

Recent advancements in the research proposed a novel approach to polystores: the Ontology-Based Data Federation (OBDF) paradigm [5]. This paradigm is the result of combining the standard OBDA paradigm on top of a Data Federation system (e.g. Denodo, Dremio), through which it is possible to combine multiple data models into a Virtual Database (VDB), without duplicating or re-materializing data but rather relying on local DBMSs, accessing data in a streaming fashion. Figure 2 represent how a query in OBDF is transformed from its semantic form into multiple independent queries across the data sources, potentially in different formats, depending on the respective source type (e.g. relational, hierarchical, graph, etc.). To set up an OBDF-based infrastructure, users are required to have (or design) an ontology that models the domain of interest: ontological axioms are exploited at query time to explicitly retrieve

[1] https://bigdawg-documentation.readthedocs.io/en/latest/

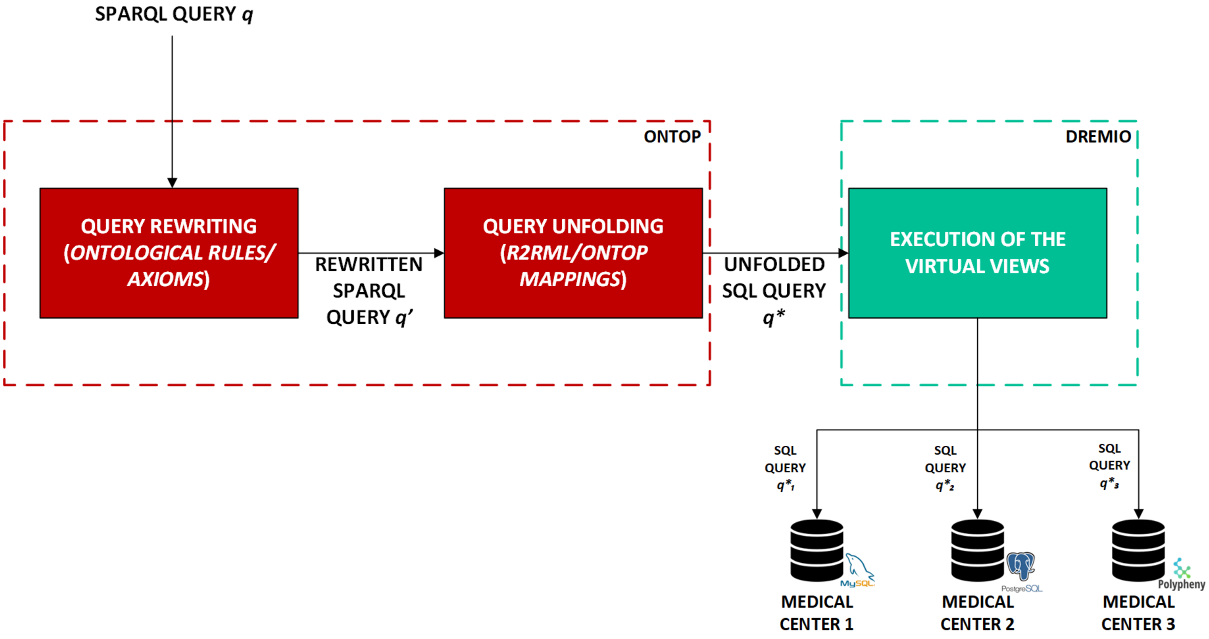

Figure 1: Query Flow Diagram representing the main transformation phases of a query in the OBDF scenario.

implicit results. In many OBDA implementations (e.g. Ontop [6]), this process is the result of multiple phases, where the query undergoes through multiple rewriting [7] phases, each of which produces an Intermediate Query (IQ). Users then have to define mappings between the ontology and the VDB. Although there exist many standard languages that allows to define mappings (e.g. RML, R2RML, Ontop Native), they share inherent properties: target ontological triples are matched with placeholders, that must correspond to the source VDB queries fields. By allowing to perform semantic queries over a graph data model, OBDF capabilities aligns perfectly with the HEREDITARY WP3 Federated Analytics task requirements, where multiple data sources with different models feeds data to the consortium without duplicating it, as a necessary condition for guaranteeing user's privacy and anonymity.

3. Federated Architecture Infrastructure

Identifying functional requirements is essential to shape the research design, workflow, and system architecture. These requirements are fundamental to shape the system architecture and to drive the choice of the different components. To address them, we explore the detailed design components essential for setting up a system that meets the project objectives. In Figure 3, we show how different components plug together: the SPARQL endpoint, the OBDA Data Integration platform (with ontology and mappings) and the Data Virtualization and Federation layer. In this section we will inspect each of these layers from a bottom-up perspective and we will discuss the implementation choices. There exist different solutions implementing a VDB such as Denodo[2], which although it is very robust, is an enterprise solution. An alternative is Dremio[3], which supports several sources, and allows for the development of custom adapters for unsupported sources. Considering that Dremio is open-source, we will focus on it as our federation component. Going into details, Dremio is a VDB with several features: it is as a data virtualization system that enables seamless access to data across various storage systems like RDBMSs, NoSQL databases and cloud storage without duplicating data, thereby optimizing query performance. The platform also includes a user management system that allows administrators to control access and manage permissions effectively, ensuring that data is accessible only to authorized personnel and facilitating collaboration. Additionally, Dremio incorporates a robust logging system,

[2] https://www.denodo.com/en
[3] https://www.dremio.com/

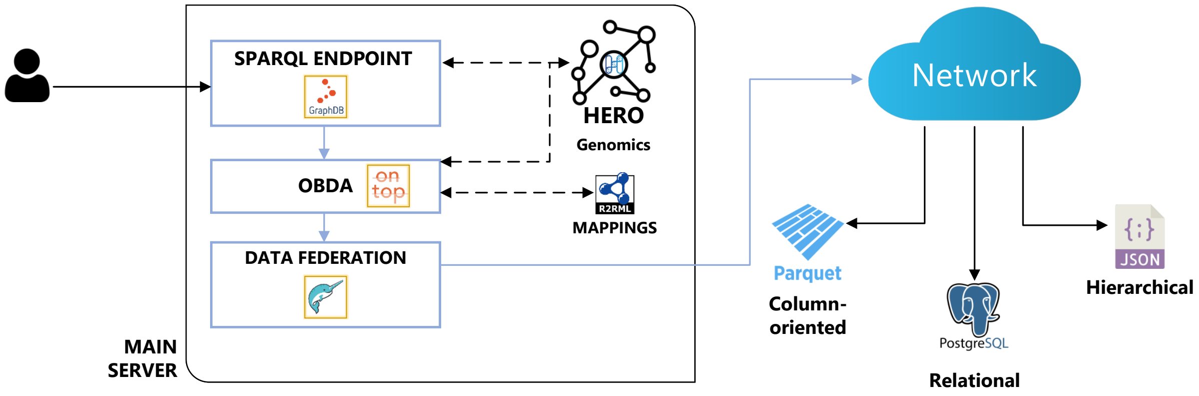

Figure 2: The three-layer architecture of the proposed OBDF polystore system.

which is essential for recording user activities and troubleshooting. This logging capability helps in analyzing usage patterns and tracking data flow.

Different OBDA systems have been implemented within time to support SPARQL-to-SQL translations. However, recent studies focused on comparing performances between Mastro and Ontop, as they are among the few OBDA systems which support ontology reasoning. In general, Mastro shown faster responses in scenarios [8] requiring extensive in terms of timings, while Ontop performs better in scenarios where a considerable number of mappings is involved for unfolding a certain query. This means that a choice on which system fits better depends on how constraining are timings in the query processing phase and on how many mappings have to be unfolded on average, that strictly depends on the heterogeneity of the underlying relational source. In our early setup, we chose Ontop as our OBDA component, as long as at this stage it is still unknown which are the requirements in terms of mappings needed, while it is known to us the research team behind it is already investigating possible optimizations when employed in a federated setup. Also, Ontop supports the R2RML standard, allowing to migrate seamlessly to another system, if necessary.

Between these, an ontology must act as a shared dictionary. As a first step towards the integration of heterogeneous data, the HEREDITARY consortium provided the genomic component of the HEReditary Ontology (HERO) [9]. We also defined mappings between the ontology and a VDB schema covering the genomic domain (samples, variants, zygosities, etc.).

4. Research Challenges

Considering the connection points between HEREDITARY challenges and OBDF key features, our plan is to adopt it as the backbone of our polystore. Nevertheless, this does not come without challenges. In this section we present preliminary results from an initial implementation of an OBDF-based polystore system, focusing on bottlenecks and hurdles we encountered in the process we aim to tackle.

Query Optimization We set up a federated environment with two data sources hosting genomic data, and we federate them in an OBDF setup, choosing Ontop as the OBDA layer, and Dremio as the Data Federation layer. Then, we derived 4 query of interest in the domain of genomics, in particular mimicking those provided by Beacon V2[4] for variants discovery. We ran each of these multiple times, recording timings for each execution phase, and we then computed average and standard deviation. Figure 4 showcases a portion of our results. As it is instantly evident, there is a huge bottleneck in the last phase, namely the Dremio Running phase. What we observed is that queries that come out from the OBDA layer present an unnecessary amount of self joins, that could possibly collapse into a single interrogation, grouped by identical queries. Moreover, cross join between federated sources are performed, retrieving uselessly the whole databases from the sources multiple times, often risking

[4] https://docs.genomebeacons.org/variant-queries/

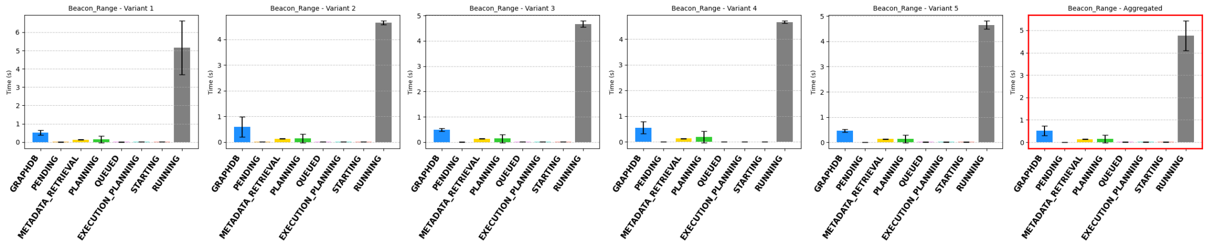

Figure 3: Preliminary results of the OBDF Dremio-based polystore system, in the context of genomics. The Figure showcases average and standard deviation of 10 executions of a Beacon Range query, in 5 different parameters settings. The last plot then shows the average and the standard deviation of all the 50 previous runs.

to fill the available memory. This occurs because OBDA systems have traditionally been designed for single-source scenarios and have not been optimized for federated settings. Recently, new research advancements [10] have proposed some optimization strategies that we aim to adopt to reduce execution bottlenecks. Another crucial aspect regards the choice of the Data Federation VDB component. Here, we need a system that exploits efficiently sources indexes and data structures, and that provides efficient functions necessary to realize the translation from relations to triples (e.g. IRI encoding).

Mapping Accuracy How mappings between the ontology and the VDB are defined is paramount. Although automatic mapping bootstrapping tools are available, they usually operate in a classical OBDA scenario. Hence, in a federated context, especially in a sophisticated domain such as the biomedical one, it is essential to have an human agent manually defining them. This process is inherently error-prone, and thus it will be necessary to put into action accuracy estimation techniques. Recent advancements in the research [11] proposed a novel approach to estimate KG accuracy using sampling strategies and providing estimates within confidence intervals. We plan to extend these techniques also in the domain of Virtual Knowledge Graphs, to assess their accuracy with respect to the original data sources, and thus to determine mappings quality.

System Accessibility The aim of having a Federated Analytics infrastructure is providing to project partners all the tools and the services to get access to the data in a streamlined way. This means that on top of the architecture, an user-friendly interface should be available to the audience. We plan to do this by means of a web application. This application will not only ease data retrieval, but will allow to export query results in many formats. Additionally, a comprehensive logging system will monitor data access, contributing to explainability. As a preliminary result, we effectively set up a web portal allowing to interact with the aforementioned embryonal OBDF-based setup[5] for genomic variant discovery.

5. Conclusions and Future Works

In this paper, we have discussed the challenges posed by the management and integration of heterogeneous biomedical data. We have critically evaluated existing polystore and OBDA solutions, highlighting their limitations in meeting the complex requirements of the biomedical domain. Consequently, we proposed adopting the OBDF approach, combining OBDA techniques and Data Federation technologies. We described an initial implementation leveraging Dremio as a federated virtualization layer and Ontop as the semantic integration engine, showcasing preliminary results within the genomic domain. Our analysis identified critical performance bottlenecks, inaccuracies in ontology-data mappings, and the need for improved system accessibility. In future work, we aim to address the identified challenges. Specifically, we will investigate query optimization techniques to reduce computational bottlenecks in federated query execution, develop methods to rigorously estimate and improve mapping accuracy, and design intuitive user interfaces complemented by logging and privacy-preserving mechanisms. By addressing these aspects, we will significantly advance toward an efficient and accessible polystore solution tailored for biomedical research requirements.

[5] https://ontobeacon.bitsei.it

Acknowledgments

This project has received funding from the HEREDITARY Project, as part of the European Union's Horizon Europe research and innovation programme under grant agreement No GA 101137074.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] V. Gadepally, P. Chen, J. Duggan, A. J. Elmore, B. Haynes, J. Kepner, S. Madden, T. Mattson, M. Stonebraker, The bigdawg polystore system and architecture, in: 2016 IEEE High Performance Extreme Computing Conference, HPEC 2016, Waltham, MA, USA, September 13-15, 2016, IEEE, 2016, pp. 1–6. URL: https://doi.org/10.1109/HPEC.2016.7761636. doi:10.1109/HPEC.2016.7761636.

[2] B. Kolev, C. Bondiombouy, O. Levchenko, P. Valduriez, R. Jiménez-Peris, R. Pau, J. Pereira, Design and implementation of the cloudmdsql multistore system, in: CLOSER 2016 - Proceedings of the 6th International Conference on Cloud Computing and Services Science, Volume 1, Rome, Italy, April 23-25, 2016, SciTePress, 2016, pp. 352–359. URL: https://doi.org/10.5220/0005923803520359. doi:10.5220/0005923803520359.

[3] M. Vogt, A. Stiemer, H. Schuldt, Polypheny-db: Towards a distributed and self-adaptive polystore, in: IEEE International Conference on Big Data (IEEE BigData 2018), Seattle, WA, USA, December 10-13, 2018, IEEE, 2018, pp. 3364–3373. URL: https://doi.org/10.1109/BigData.2018.8622353. doi:10.1109/BIGDATA.2018.8622353.

[4] D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, A. Poggi, R. Rosati, Ontology-based database access, in: Proceedings of the Fifteenth Italian Symposium on Advanced Database Systems, SEBD 2007, 17-20 June 2007, Torre Canne, Fasano, BR, Italy, 2007, pp. 324–331.

[5] Z. Gu, D. Calvanese, M. D. Panfilo, D. Lanti, A. Mosca, G. Xiao, OBDF: OBDA + data federation - extended abstract, in: 40th International Conference on Data Engineering, ICDE 2024 - Workshops, Utrecht, Netherlands, May 13-16, 2024, IEEE, 2024, pp. 381–383. URL: https://doi.org/10.1109/ICDEW61823.2024.00060. doi:10.1109/ICDEW61823.2024.00060.

[6] M. Rodriguez-Muro, R. Kontchakov, M. Zakharyaschev, Ontology-based data access: Ontop of databases, in: The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I, volume 8218 of *Lecture Notes in Computer Science*, Springer, 2013, pp. 558–573. URL: https://doi.org/10.1007/978-3-642-41335-3_35. doi:10.1007/978-3-642-41335-3_35.

[7] M. Rodriguez-Muro, R. Kontchakov, M. Zakharyaschev, Query rewriting and optimisation with database dependencies in ontop, in: Informal Proceedings of the 26th International Workshop on Description Logics, Ulm, Germany, July 23 - 26, 2013, volume 1014 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2013, pp. 917–929. URL: https://ceur-ws.org/Vol-1014/paper_24.pdf.

[8] M. Namici, G. D. Giacomo, Comparing query answering in OBDA tools over w3c-compliant specifications, in: Proceedings of the 31st International Workshop on Description Logics, Tempe, Arizona, US, October 27th - 29th, 2018, volume 2211 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2018. URL: https://ceur-ws.org/Vol-2211/paper-25.pdf.

[9] M. Cazzaro, I. G. Gut, L. Menotti, M. Rueda, G. Silvello, Hero-genomics: An ontology for integration and access of multicenter genomic data, in: 16th International Conference on Semantic Web Applications and Tools for Health Care and Life Sciences, SWAT4HCLS 2025, Barcelona, Spain, February 24th to 27th, 2025. URL: http://www.dei.unipd.it/~menottilau/papers/25-CGMRS-SWAT4HCLS.pdf.

[10] M. D. Panfilo, Towards optimizing ontology-based data federation: Performance insights from experimental studies, in: Companion Proceedings of the 8th International Joint Conference on

Rules and Reasoning, Bucharest, Romania, September 16-18, 2024, volume 3816 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024. URL: https://ceur-ws.org/Vol-3816/paper73.pdf.

[11] S. Marchesin, G. Silvello, Efficient and reliable estimation of knowledge graph accuracy, Proc. VLDB Endow. 17 (2024) 2392–2404. URL: https://www.vldb.org/pvldb/vol17/p2392-marchesin.pdf. doi:10.14778/3665844.3665865.