MASTER THESIS IN COMPUTER ENGINEERING

Federated Data Analytics for Genomics Data

MASTER CANDIDATE

Mirco Cazzaro

Student ID 2076745

SUPERVISOR

Prof. Gianmaria Silvello

University of Padova

CO-SUPERVISOR

Dott. Robert Kahn

University of Princeton

To my parents
and friends
not to alberto

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. The project will involve designing a system architecture that supports the integration of heterogeneous data stores under a unified federated system. The system will also utilize principles from the Ontology-Based Data Access (OBDA) paradigm to facilitate semantic querying capabilities. By developing a federated data analytics system for genomics data, this thesis will contribute to reducing the complexities involved in biomedical data management. The system will enable more effective data integration and querying processes, thereby supporting faster and more accurate genomics research. The completion of this project will result in a prototype of a federated data analytics system capable of handling the specific needs of genomics data.

Sommario

Contents

CONTENTS

Acknowledgments

List of Figures

List of Tables

List of Algorithms

List of Code Snippets

List of Acronyms

RDF Resource Description Framework

URI Uniform Resource Identifier

KG Knowledge Graph

OWL Ontology Web Language

DBMS Database Management System

SQL Structured Query Language

OBDA Ontology-Based Data Access

API Application Program Interface

JDBC Java Database Connectivity

ODBC Open Database Connectivity

FOSS Free and Open Source

ARP Advanced Relational Pushdown

1

Introduction

2

Background

2.1 Methodological Background

2.1.1 Resource Description Framework

The Resource Description Framework (RDF) is a standard[1] by the World Wide Web Consortium (W3C) designed for data interchange on the web. Its definitive syntax was lastly defined on 2003 [1].

RDF is based on the idea of making statements about resources, expressed as triples: for example, a triple might consist of "Gene123" (subject) "hasFunction" (predicate) "DNA Repair" (object). These triples are stored in a graph, making RDF uniquely suited for modern data analytics where relationships and linkages are crucial: this structure is by design flexible, and it is used to represent information in a way that makes it easier to aggregate, integrate, and manage diverse data from various sources.

The standard utilizes Uniform Resource Identifier (URI) to ensure that each element in the triple is uniquely identifiable, allowing to link data across different datasets seamlessly. RDF also supports literal values and datatypes, enabling detailed descriptions of properties and values.

In summary, RDF provides a robust and flexible framework for describing and interlinking data on the web, which is crucial for any federated data system

---

[1]https://www.w3.org/RDF/

that aims to integrate diverse data sources effectively.

2.1.2 KNOWLEDGE GRAPHS

Knowledge Graphs represent an innovative way of structuring and querying interconnected data. A Knowledge Graph (KG) organizes data in a graph format, where entities (nodes) and their interrelations (edges) are defined according to a schema that encapsulates both the entities and the possible links between them. This structure allows not only to better visualize data, but is also very well suited for data exploration and analysis.

Central to the concept of KG is the idea of enhancing search and data discovery capabilities beyond simple data retrieval. By semantically linking data entities, KG allow for more intuitive and sophisticated queries that are closer to natural language questions. This capability makes them useful in complex domains like biomedical research, where users may need to uncover hidden relationships and patterns among vast datasets.

Furthermore, KG well suits in scenarios requiring data integration from disparate sources. They support the combination of structured and unstructured data and they can scale well as new data need to be integrated. This flexibility is crucial in fields such as genomics, where new data attributes and relationships are continuously being discovered and need to be rapidly integrated into existing datasets.

In practice, KG are usually powered by technologies such as RDF and other standards discussed earlier, leveraging the strengths of these frameworks to ensure robust data handling and scalability.

2.1.3 ONTOLOGIES

Ontologies are a shared vocabulary for a particular domain. They serve as a crucial tool in structuring data by defining classes where data entities can be categorized. This classification system not only standardizes data representation but also simplifies the communication between different systems and users by providing a common understanding of the domain.

Ontologies involve the use of first-order logic to define rules about the relationships between different classes. These rules allow for the logical inference of new information from the data that is already known, which can significantly

deepen data analysis and querying capabilities. This set of first-order logic rules used to model ontologies composes the Ontology Web Language (OWL) [9].

With ontologies integrated within a KG, it is possible to employ inferential algorithms either at query time or at preprocessing time. This capability enables the derivation of new knowledge that isn't explicitly stated in the data but can be inferred based on the ontological representations.

The use of ontologies in a KG is crucial in complex domains like genomics. Here, the ability to infer new relationships and properties can lead to understanding intricate biological connections and processes.

In summary, ontologies provide the foundational structure for managing complex information systems. They not only facilitate a standardized approach to data handling and integration but also enhance the capability of systems to derive and utilize new knowledge effectively.

2.1.4 DATA FEDERATION

Data Federation is a technology that organizes data from multiple different and autonomous data sources and makes it accessible under a uniform data model, allowing for real-time data retrieval from various sources without requiring data deduplication. This approach creates a unified data access layer that abstracts the underlying technical details of each Database Management System (DBMS). Users can query this virtual layer using standard data querying languages, like Structured Query Language (SQL), without needing to know where the data is physically stored or in what format.

A key strength of Data Federation lies in its capacity to harmonize heterogeneous data sources, ranging from traditional relational databases to modern big data solutions and cloud services. This integration is seamless and dynamic, scaling well as soon as new sources come in, without requiring significant reconfiguration. Such agility is crucial in fields like genomics, where data formats and sources evolve rapidly alongside scientific advancements. Moreover, it minimizes the risks associated with data duplication and movement, such as data loss or corruption. It also ensures that data remains current, reflecting real-time changes without delay. This is particularly valuable for decision-making processes where up-to-date information is critical.

Different existing Data Federation systems may have different characteristics, and the choice of which system suits better certain requirements may depend on

many different factors. A recent comparative study on Data Virtualization Systems [5] highlights and evaluates diverse features of many systems coming both from the academia as well enterprise solutions, such as supported query languages, supported data sources, data security, exposed interfaces and software support.

2.1.5 SEMANTIC DATA INTEGRATION

Semantic Data Integration is an approach that aims to integrate data from a single relational source to an ontology that models a specific domain. This integration is achieved by defining mappings that establish semantic relationships among data entities.

The core idea behind semantic data integration involves establishing a global schema, represented by the ontology, which acts as a blueprint for how data from various sources is viewed and accessed. This ontology defines not just the entities and their attributes but also the relationships and constraints between these entities. The mapping between the ontology (global schema) and the relational data source is critical as it dictates how data is interpreted in a meaningful way.

The concept of Data Integration, firstly introduced without a semantic focus [7], has evolved with research advancements in this field. These advancements have led to the development of tools and mapping languages specifically designed for semantic data integration, laying the groundwork for a paradigm known as Ontology-Based Data Access (OBDA) [11].

The OBDA framework, that has been precisely formalized [2], is composed of a chain of procedures that can be summarized as follows:

1. input query $q(\vec{x})$ is rewritten [8] into q' over the virtual ABox A' (i.e. the first-order logic specification of the ontology);

2. the rewritten query q' is unfolded using the *mappings* into an SQL query q^*;

3. the optimised SQL query q^* is evaluated over the data instance D (e.g. a relational DBMS).

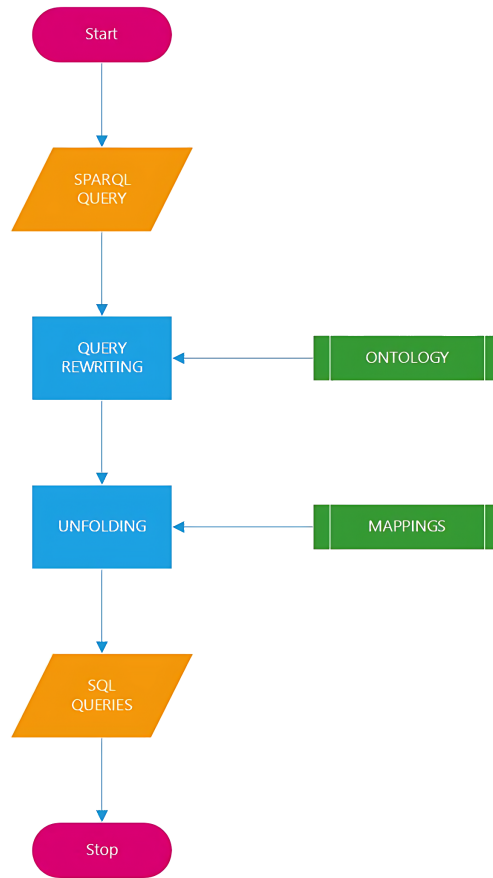Figure 2.1 outlines the aforementioned algorithm within a flowchart.

Figure 2.1: The OBDA framework

2.2 Technical Background

2.2.1 Data Federation Systems

As discussed in section 2.1.4, Data Federation Systems are sophisticated softwares and thus evaluable under many aspects. In this background analysis we will focus on four main aspects, considering their importance as the Data Federation System will be part of a broader framework. In particular, by the aforementioned comparison, we will present three most prominent systems considering:

- software support & documentation;

- scalability;

7

- logging capabilities;

- performances.

These features are also briefly summarized in Tab. 2.1.

Denodo

Denodo[2] is an enterprise virtualization platform that serves as a data federation system, integrating data from diverse sources to provide a unified view without requiring physical data deduplication. It allows for real-time access to structured and unstructured data from various sources including relational, column and No-SQL databases, web (! ((!)API), and flat files.

Denodo provides robust security features, including hashing, encrypting functions and user privileges, ensuring that sensitive data is protected according to compliance standards.

Moreover, it utilizes advanced query optimization techniques, such as caching and query rewriting, to enhance performance. These optimizations ensure efficient data retrieval, reducing latency and improving the overall speed of data access across the federated sources.

Although it is highly scalable, capable of accommodating new data sources, and it is provided also with a custom source wrapper template for unsupported data sources, being it an enterprise solution allows for maximum five connections, unless a premium plan is signed.

Teiid

Teiid[3] is an open-source Java component developed by Red Hat[4] that provides integrated access to multiple data sources through a single uniform API. Rather than a DBMS, Teiid is more a query engine for integrating data from multiple sources, accessible both through Application Program Interface (API) and Java Database Connectivity (JDBC)/Open Database Connectivity (ODBC) interfaces.

---

[2]https://denodo.com/en
[3]https://teiid.io/
[4]https://www.redhat.com/

It comes in different shapes: there are Teiid implementations as an Eclipse plugin as well as deployable packages on web containers such as Wildfly and OpenShift.

One of the main issues with Teiid is poor documentation when it is not deployed alongside enterprise solutions (like OpenShift). Moreover, no major releases have been published since four years, and many open issues on the official GitHub repository[5] are not being addressed.

Dremio

Similarly to Denodo, Dremio[6] is a virtualization platform that serves as a data federation system. Although it is developed within an enterprise context, the standard version, comprehensive of most of the Dremio features, it is Free and Open Source (FOSS) under the Apache 2.0 license, combining typical enterprise software robustness with a strong community active on continuous maintenance.

Even if it is possible to use Dremio as a standalone instance (e.g. on a server), it is by design a distributed system and thus it can run on clusters up to more than 1000 nodes. In case of standalone configurations, in linux server environments it is possible to install it using packet managers, but the easiest way is to use the Docker image it comes with.

Dremio makes use of Apache Arrow to enhance processing speeds and reduce latency. Moreover, it optimizes query performance through its advanced query planner and execution engine, which can push down operations to the data source level, minimizing data movement and speeding up response times.

It provides comprehensive security features that include encryption, access controls, and data masking to ensure privacy. It also maintains detailed logs of all queries, which are crucial for compliance purposes in many fields such as the clinical domain, where patient medical data is managed alongside their personal information.

As in Denodo, but under an open-source perspective, it is possible to build custom connectors for unsupported data sources and Dremio. This is realizable through Advanced Relational Pushdown (ARP) connectors: a public repository[7]

---

[5]https://github.com/teiid/teiid
[6]https://www.dremio.com/
[7]https://github.com/dremio-hub/dremio-sqllite-connector

provides a Maven template, that is customizable for each data source for which a JDBC driver is available.

In conclusion, Dremio offers an open-source virtualization system with the typical robustness of enterprise softwares; it is highly scalable both in the sense of computation distribution over clusters and in the types of supported sources, and it has a comprehensive logging system that suits well for tracking data flow.

	Support	Free and Open Source	Well Documented	Scalability	Solid Logging Capabilities
Denodo	✓		✓	✓	✓
Teiid		✓			
Dremio	✓	✓	✓	✓	✓

Table 2.1: Comparative among Data Federation Systems

2.2.2 OBDA Systems

OBDA, as shown in Fig. 2.2, allows to seamlessly integrate a relational source within an ontology, so to add a semantic layer. Research on this specific topic has been performed not only by the academy, but also from Research & Development branches of big companies, considering the impact on the industry: a preliminary analysis [6] by Siemens[8] that lead subsequently to a custom platform, analyzed existing systems. Table 2.2 summarizes the content of this analysis.

Given that ontology reasoning is one of the most important requirements in the OBDA approach (without this, using this paradigm would allow just to run queries in SPARQL language, without any other real advantage), and considering that the Optique system is an enterprise, on-premise and closed source solution, more in-depth comparisons [10] studied Ontop and Mastro. Their comparison can be summarized as follows.
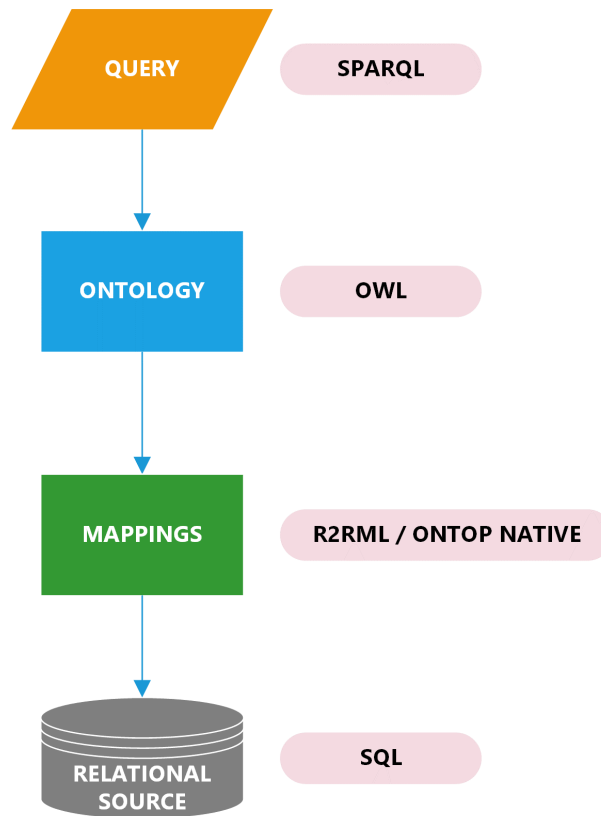
---

[8]https://www.siemens.com/

Figure 2.2: The OBDA framework

MASTRO

By our knowledge, Mastro [4] was the first OBDA tool, developed at La Sapienza University of Rome. It supports a subset of SPARQL queries and integrates a custom XML-based mapping syntax. Mastro's mappings were not initially compliant with R2RML standards, which was a strong limitation in standard OBDA environments. The tool used a complex system of view predicate mappings, which may affect its adaptability to standardized environments. Although recent updates have integrated R2RML support, the system's architecture still lacks certain optimizations due to its design constraints. For instance, Mastro cannot perform advanced semantic query optimizations because it does not support detailed data constraints in its mappings. This limitation could lead to less efficient query processing and increased execution times, especially with complex queries involving multiple joins or extensive data operations. Moreover, Mastro is less efficient handling of datatype operations and IRI constructions within SQL, leading to potential slowdowns in query processing.

System	Characteristics
Optique (Siemens)	Supports ontology reasoning; supports both static and streaming relational DBMS
Ontop	Supports ontology reasoning
Mastro	Supports ontology reasoning
morph-RDB	Supports ontology reasoning
D2RQ	Does not support ontology reasoning
OntoQF	Does not support ontology reasoning
Virtuoso	Does not support ontology reasoning
Spyder	Does not support ontology reasoning
Ultrawrap	Does not support ontology reasoning

Table 2.2: Comparative among OBDA Systems

ONTOP

Ontop [3] is an open-source OBDA framework developed at the Free University of Bolzano. It is compliant with W3C standards, including R2RML for ontology mapping, OWL for ontology representation, and SPARQL for querying. Ontop is designed for high-performance query answering over virtualized RDF graphs. Ontop provides an efficient query answering system that leverages R2RML mappings and supports comprehensive optimizations. The system uses advanced query rewriting techniques, which are highly effective in reducing query complexity and execution time. Ontop's strengths are particularly evident in its handling of complex SPARQL queries, where it efficiently translates these into SQL, utilizing T-mappings and database integrity constraints for optimization. Regardless of its adherence to standards, Ontop comprises also a custom mapping language (Ontop Native Language).

The comparison between the two systems has been performed in two different scenarios. In general, Mastro shown faster responses in scenarios 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.

3

Analysis

4

Conclusions and Future Works

References

[1] Dave Beckett and Brian McBride. "RDF/XML syntax specification (revised)". In: *W3C recommendation* 10.2.3 (2004).

[2] Elena Botoeva et al. "A Generalized Framework for Ontology-Based Data Access". In: *AI*IA 2018 - Advances in Artificial Intelligence - XVIIth International Conference of the Italian Association for Artificial Intelligence, Trento, Italy, November 20-23, 2018, Proceedings*. Ed. by Chiara Ghidini et al. Vol. 11298. Lecture Notes in Computer Science. Springer, 2018, pp. 166–180. DOI: `10.1007/978-3-030-03840-3\_13`. URL: `https://doi.org/10.1007/978-3-030-03840-3%5C_13`.

[3] Diego Calvanese et al. "OBDA with the Ontop Framework". In: *23rd Italian Symposium on Advanced Database Systems, SEBD 2015, Gaeta, Italy, June 14-17, 2015*. Ed. by Domenico Lembo, Riccardo Torlone, and Andrea Marrella. Curran Associates, Inc., 2015, pp. 296–303.

[4] Diego Calvanese et al. "The MASTRO system for ontology-based data access". In: *Semantic Web* 2.1 (2011), pp. 43–53. DOI: `10.3233/SW-2011-0029`. URL: `https://doi.org/10.3233/SW-2011-0029`.

[5] Zhenzhen Gu et al. "A systematic overview of data federation systems". In: *Semantic Web* 15.1 (2024), pp. 107–165. DOI: `10.3233/SW-223201`. URL: `https://doi.org/10.3233/SW-223201`.

[6] Evgeny Kharlamov et al. "How Semantic Technologies Can Enhance Data Access at Siemens Energy". In: *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I*. Ed. by Peter Mika et al. Vol. 8796. Lecture Notes in Computer Science. Springer, 2014, pp. 601–619. DOI: `10.1007/978-3-319-11964-9\_38`. URL: `https://doi.org/10.1007/978-3-319-11964-9%5C_38`.

[7]   Maurizio Lenzerini. "Data Integration: A Theoretical Perspective". In: *Proceedings of the Twenty-first ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 3-5, Madison, Wisconsin, USA*. Ed. by Lucian Popa, Serge Abiteboul, and Phokion G. Kolaitis. ACM, 2002, pp. 233–246. DOI: `10.1145/543613.543644`. URL: `https://doi.org/10.1145/543613.543644`.

[8]   Konstantinos Makris et al. "Ontology Mapping and SPARQL Rewriting for Querying Federated RDF Data Sources - (Short Paper)". In: *On the Move to Meaningful Internet Systems, OTM 2010 - Confederated International Conferences: CoopIS, IS, DOA and ODBASE, Hersonissos, Crete, Greece, October 25-29, 2010, Proceedings, Part II*. Ed. by Robert Meersman, Tharam S. Dillon, and Pilar Herrero. Vol. 6427. Lecture Notes in Computer Science. Springer, 2010, pp. 1108–1117. DOI: `10.1007/978-3-642-16949-6\_32`. URL: `https://doi.org/10.1007/978-3-642-16949-6%5C_32`.

[9]   Deborah L McGuinness, Frank Van Harmelen, et al. "OWL web ontology language overview". In: *W3C recommendation* 10.10 (2004), p. 2004.

[10]   Manuel Namici and Giuseppe De Giacomo. "Comparing Query Answering in OBDA Tools over W3C-Compliant Specifications". In: *Proceedings of the 31st International Workshop on Description Logics co-located with 16th International Conference on Principles of Knowledge Representation and Reasoning (KR 2018), Tempe, Arizona, US, October 27th - to - 29th, 2018*. Ed. by Magdalena Ortiz and Thomas Schneider. Vol. 2211. CEUR Workshop Proceedings. CEUR-WS.org, 2018. URL: `https://ceur-ws.org/Vol-2211/paper-25.pdf`.

[11]   Guohui Xiao et al. "Ontology-Based Data Access: A Survey". In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*. Ed. by Jérôme Lang. ijcai.org, 2018, pp. 5511–5519. DOI: `10.24963/IJCAI.2018/777`. URL: `https://doi.org/10.24963/ijcai.2018/777`.

Acknowledgments