Eurostars Project

SPARQL-ML: Machine Learning for SPARQL Query Optimization over Centralized and Distributed RDF Knowledge Graphs

Project Number: 5736 **Start Date of Project:** 2024/11/01 **Duration:** 36 months

Deliverable 1.1
Requirement Specification

Dissemination Level	Public
Due Date of Deliverable	January 31, 31/01/2025
Actual Submission Date	January 31, 31/01/2025
Work Package	WP1, Requirements Elicitation & Conceptual Architecture
Deliverable	D1.1
Type	Report
Approval Status	Final
Version	1.0
Number of Pages	14

Abstract: The SPARQL-ML project aims to improve SPARQL query optimization for centralized and distributed RDF Knowledge Graphs using machine learning techniques. WP1, led by OpenLink, brings together project partners to identify and define the functional requirements needed to support ML-driven improvements in SPARQL query performance. This deliverable, D1.1, focuses on the requirements elicitation process, the initial phase in which we systematically explore selected use cases and relevant research developments. Through a review of the state of the art, we capture the current challenges and outline the functional requirements that will guide the subsequent architectural design and development phases. The insights and specifications presented in D1.1 establish a foundation for aligning project goals with effective query optimization strategies.

History

Version	Date	Reason	Revised by
0.1	11/11/2024	Initial Template & Structure	Milos Jovanovik
0.2	16/12/2024	OpenLink Use Case	Milos Jovanovik & Mirko Spasić
0.3	27/01/2025	UPB Use Case	Muhammad Sohail Nisar
0.4	29/01/2025	eccenca Use Case	Edgard Marx
0.9	29/01/2025	Alignment of Functional Requirements to Work Packages	Milos Jovanovik, Muhammad Sohail Nisar, Edgard Marx
1.0	30/01/2025	Finalizing	Milos Jovanovik

Author List

Organization	Name	Contact Information
OpenLink Software	Milos Jovanovik	mjovanovik@openlinksw.com
OpenLink Software	Mirko Spasić	mspasic@openlinksw.com
eccenca GmbH	Edgard Marx	edgard.marx@eccenca.com
University of Paderborn	Muhammad Sohail Nisar	msohail@mail.uni-paderborn.de

Contents

1 Introduction

Modern applications increasingly rely on extensive datasets, often too large for single-server environments, leading to a shift towards distributed solutions for efficient data management and query handling. In RDF Knowledge Graphs, distributed architectures present specific challenges for optimizing SPARQL query performance, particularly in large, complex datasets. The SPARQL-ML project addresses these issues by developing machine learning-based techniques to improve SPARQL query processing across centralized and distributed RDF Knowledge Graphs.

This deliverable, D1.1, documents the requirements elicitation phase in which the project partners collaborate to identify and define functional requirements that are critical for improvements in optimizing queries through machine learning. The requirements are based on specific use cases provided by each partner and are refined through an analysis of relevant research and real-world challenges in large-scale RDF data management.

The main use cases are:

- **eccenca's Benchmarking Use Case**: Focusing on improving query performance in enterprise knowledge graphs, eccenca will benchmark the project's methods and integrations against real-world enterprise data. The aim is to address query runtime issues in triplestores by using federated query optimization in distributed setups. Query logs will be collected, with client consent, to refine the developed solutions and enhance data management capabilities for enterprise applications.
- **Paderborn's Linked Cancer Genome Atlas Use Case**: This use case targets the RDF version of the Cancer Genome Atlas (Linked TCGA), which encompasses 20.4 billion triples from cancer patients across numerous tumor types. Given the dataset's scale, a single endpoint approach is infeasible, making this a valuable case for applying SPARQL-ML's distributed optimization techniques to demonstrate performance benefits in a high-stakes, data-intensive healthcare domain.
- **OpenLink's LinkedGeoData and DBpedia Use Case**: OpenLink will focus on optimizing query performance for LinkedGeoData and DBpedia, two large RDF Knowledge Graphs connected to the Linked Open Data (LOD) Cloud. With data sizes reaching billions of RDF triples, this use case will demonstrate how a Deep Reinforcement Learning (DRL) agent, developed for the Virtuoso SPARQL engine, can improve query performance for LinkedGeoData and DBpedia endpoints under heavy workloads.

These use cases and the associated requirements, detailed in Section 2, provide a foundation for the specification of functional requirements that will guide the design and implementation of SPARQL-ML components. Section 3 aligns these requirements with specific project components and work packages, establishing a baseline for evaluating and measuring the project's success.

2 Use Cases Descriptions

2.1 Benchmarking for Disruptive Innovation in Data Management

Effective management of disruptive innovation requires precise benchmarking strategies to evaluate and optimize the performance across cutting-edge technologies and systems. Eccenca, a leading innovator in the field of semantic technologies, focuses on leveraging benchmarking to address challenges in SPARQL query optimization, particularly for federated and centralized RDF Knowledge Graphs. With the increasing complexity of data-driven innovation, Eccenca's Corporate Memory (CMEM) platform plays a pivotal role in facilitating seamless

data integration and management. The aim is to employ benchmarking tools to evaluate solutions' scalability, efficiency, and ability to handle large-scale federated queries effectively.

The FedShop benchmark [3], discussed in the associated study, is central to this use case. This benchmark enables a realistic e-commerce scenario for testing the scalability of SPARQL federation engines, making it instrumental for evaluating how well Eccenca's solutions perform under various query workloads. By combining the FedShop benchmarking with Eccenca's proprietary testing suite—comprising Build KG Integration Tests, Build KG Performance Tests, Explore KG Integration Tests, and cmemc Integration Tests—the use case ensures a comprehensive evaluation of the platform's robustness and scalability.

Elicitation Procedure

The elicitation procedure for benchmarking Eccenca's solutions involves structured processes to derive comprehensive performance metrics under realistic use cases. Initially, specific requirements are identified through collaboration with stakeholders, including clients and consortium partners from SPARQL-ML. The procedure includes defining key performance indicators (KPIs) such as query execution time, scalability across federated endpoints, and efficiency in source selection and query decomposition.

FedShop is deployed to simulate e-commerce environments, where autonomous vendors and rating sites emulate real-world federations of SPARQL endpoints. The elicitation involves executing a set of predefined query templates and instantiating them across federations of varying sizes. Metrics are collected and analyzed to understand the performance of Eccenca's Corporate Memory platform against these benchmarks. Additionally, Eccenca incorporates its proprietary testing suite to conduct further validation. These tests ensure functionality and scalability by analyzing the integration, performance, and exploratory capabilities within the Knowledge Graphs.

Requirements

By integrating FedShop and Eccenca's testing suite, the benchmarking process ensures a robust evaluation framework, facilitating improved scalability and functionality for managing disruptive innovations.

ID	Title	Description	Priority
1-1	Benchmarking Capabilities	Support for FedShop's scalability tests, covering federations from 20 to 200 endpoints. Compatibility with schema-based dataset generators to simulate real-world data distributions.	High
1-2	Testing Suite Integration	Incorporation of Build KG Integration Tests to verify seamless Knowledge Graph construction. Execution of Build KG Performance Tests to measure the system's efficiency under load.	High
1-3	Scalability and Performance	Evaluation of query execution times for increasingly complex federated queries. Use of cmemc Integration Tests to assess end-to-end system reliability and interoperability.	High
1-4	Exploratory Testing	Execution of Explore KG Integration Tests to validate search and data retrieval operations.	High
1-5	Real-World Applicability	Ability to generate actionable insights for large-scale industry use cases, ensuring alignment with SPARQL-ML objectives.	Medium

Table 1: Runtimes (in ms) on Linked TCGA queries with all Virtuoso endpoints. The values inside the brackets show the percentage of the actual query results obtained. (**TO** = Time out after 2.5 hour, **RE** = runtime error).

Qr.	FedX (cold)	FedX (warm)	SPLENDID	ANAPSID	FedX+HiBISCuS	CostFed
L1	TO (7.2 %)	TO (7.2 %)	123735 (2.73 %)	19672 (15.76 %)	TO (7.2 %)	1237000 (100 %)
L2	35 (0 %)	35 (0 %)	45473 (1.8 %)	TO (0 %)	76 (0 %)	454709 (100 %)
L3	27 (0 %)	27 (0 %)	4877696 (100 %)	TO (0 %)	47 (0 %)	4877991 (100 %)
L4	TO (0.08 %)	TO (0.08 %)	7535531 (0 %)	8775598 (0 %)	62595 (48.34 %)	7535200 (100 %)
L5	TO (0 %)	TO (0 %)	RE (0 %)	TO (0 %)	TO (0 %)	RE (0 %)
L6	TO (0 %)	TO (0 %)	RE (0 %)	TO (0 %)	6127090 (0 %)	RE (0 %)
L7	122633 (100 %)	122500 (100 %)	114456 (100 %)	105447 (100 %)	119449 (100 %)	114400 (100 %)
L8	TO (0.01 %)	TO (0.01 %)	TO (0.05 %)	TO (0.05 %)	TO (0.01 %)	TO (0.05 %)

2.2 Linked Cancer Genome Atlas

Linked Cancer Genome Atlas (Linked TCGA): Linked TCGA is the RDF version of the Cancer Genome Atlas[1]. This knowledge base contains cancer patient data generated by the TCGA pilot project, started in 2005 by the National Cancer Institute (NCI) and the National Human Genome Research Institute (NHGRI). Currently, Linked TCGA comprises a total of 20.4 billion triples[2] from 9000 cancer patients and 27 different tumour types. For each cancer patient, Linked TCGA contains expression results for the DNA methylation, Expression Exon, Expression Gene, miRNA, Copy Number Variance, Expression Protein, SNP, and the corresponding clinical data. Storing such a large dataset in a single endpoint is simply not scalable. In this use case we are aiming to show the actual benefit of our proposed solutions when applied to a real practical use case.

Elicitation Procedure

Our main requirements for this use case came from the evaluation we performed in LargeRDFBench [8] and our planned extensions based on Linked TCGA data. The LargeRDFBench includes portions of Linked TCGA (0.7 billion triples), and in this use case, we aim to demonstrate the scalability and effectiveness of our proposed federated SPARQL query optimization solutions. By extending the dataset to include up to 5 billion triples and distributing it across 10 triplestores, we will compare the performance of our solution with state-of-the-art federation engines such as FedX [10], CostFed [9], Odyssey [6], SPLENDID [4], and ANAPSID [1]. This elicitation process will allow us to address key challenges and shortcomings in existing systems, particularly concerning large-scale datasets like Linked TCGA.

Our evaluation highlighted that current SPARQL query federation systems struggle with handling large data queries effectively, often failing to guarantee completeness and correctness. For instance, incomplete results or timeout issues arise due to flaws in query planning strategies and join techniques. These limitations underscore the necessity of developing a robust and efficient federated SPARQL query engine. A key innovation is the integration of **Deep Reinforcement Learning (DRL)**-based techniques for federated query optimization. These methods optimize source selection, join-ordering, and query execution plans. DRL ensures adaptive optimization by training policies to minimize execution time, reduce intermediate results, and improve throughput.

The following requirements have been derived based on these insights and the unique characteristics of the

[1] http://cancergenome.nih.gov/
[2] http://tcga.deri.ie/

Linked TCGA use case.

Requirements

Table 2: Requirements for Federated SPARQL Query Optimization for Linked TCGA Use Case

ID	Title	Description	Priority
2-1	Scalable Federation Engine	The proposed engine should efficiently execute federated SPARQL queries over large datasets.	High
2-2	Completeness and Correctness	Ensure complete and accurate results for all federated queries on Linked TCGA.	High
2-3	SPARQL Query Embeddings	Represent the SPARQL Query in a continuous vector space	High
2-4	Optimized Query Planning	Use DRL-based methods to generate efficient query execution plans for large data.	High
2-5	Efficient Source Selection	Develop a hybrid join-aware source selection mechanism to minimize irrelevant sources.	High
2-6	Parallel Query Execution	Enable parallelism in join and union operations to improve runtime efficiency.	High
2-7	Incremental Results Presentation	Provide initial results quickly and stream remaining results as they become available.	Medium
2-8	Data Distribution	Distribute Linked TCGA data among 10 triplestores efficiently to support scalable querying.	High
2-9	Comprehensive Benchmarking	Extend LargeRDFBench with Linked TCGA data and queries to evaluate and compare engines.	High
2-10	Advanced SPARQL Feature Support	Ensure full compatibility with SPARQL 1.1 for complex query support.	High

Benchmark Data (Data and Test Queries)

LargeRDFBench [8] is a billion-triple benchmark for SPARQL query federation, encompassing real data and queries derived from bio-medical use cases, including TCGA data. LargeRDFBench currently includes a subset of Linked TCGA, comprising 306 patient records distributed evenly across three cancer types: Cervical (CESC), Lung squamous carcinoma (LUSC), and Cutaneous melanoma (SKCM). The selection of these patients was conducted in collaboration with domain experts to ensure relevance and representativeness.

The data is hosted across three SPARQL endpoints: the first endpoint contains all DNA methylation data, the second contains all Expression Exon data, and the third hosts all remaining data. Consequently, the dataset has been divided into three subsets: Linked TCGA-M (methylation data), Linked TCGA-E (exon data), and Linked TCGA-A (all remaining data).

LargeRDFBench comprises a total of 32 queries specifically designed for evaluating *SPARQL endpoint federation approaches*. These queries are categorized into three types: 14 simple queries (S1-S14) adapted from FedBench (CD1-CD7 and LS1-LS7), 10 complex queries (C1-C10), and 8 large data queries (L1-L8). The large data queries are particularly relevant to the TCGA use case and were created with input from domain experts to simulate realistic challenges in federated query processing. These queries test federation engines' ability to process large intermediate results (often in the hundreds of thousands) or generate large result sets (with a minimum of 80,459 results) while involving a substantial number of endpoint requests. As a result, the processing time for these large queries often exceeds one hour.

In this use case, we extend LargeRDFBench to include Linked TCGA data up to 5 billion triples, distributed across 10 triplestores. This extension allows us to benchmark and evaluate the performance of our proposed federated SPARQL query optimization solutions against state-of-the-art engines such as FedX [10], CostFed [9], Odyssey [6], SPLENDID [4], and ANAPSID [1]. The benchmark will focus on key aspects such as scalable query execution, completeness and correctness of results, efficient source selection, parallel query execution, and support for large-scale real-world datasets.

Our main goal is to demonstrate the practical benefits of our proposed solutions in addressing the challenges of federated SPARQL query processing over large-scale datasets, as exemplified by the Linked TCGA use case.

2.3 LinkedGeoData and DBpedia

LinkedGeoData [12] and DBpedia [2, 5] are large-scale RDF datasets (Knowledge Graphs) which have different usage patterns. LinkedGeoData, being the RDF version of the OpenStreetMap data [7], contains geospatial data which have a very specific usage pattern. DBpedia, on the other hand, represents an RDF version of Wikipedia, which in turn has a very broad usage.

This use case focuses on evaluating the performance improvements of SPARQL query optimization for the LinkedGeoData and DBpedia datasets. The primary goal is to demonstrate significant enhancements in SPARQL query response times through the use of a deep reinforcement learning (DRL) agent integrated into the Virtuoso RDF Quad Store. This evaluation will leverage the extensive query logs collected over the past decade from the canonical DBpedia SPARQL endpoint and the LinkedGeoData dataset. By analyzing these logs to identify commonly queried data patterns, the DRL agent will be trained to optimize join orders and improve overall query execution efficiency. The developed solution will be tested using real-world workloads from these datasets, highlighting its effectiveness in addressing scalability challenges and advancing query performance for end-users.

Elicitation Procedure

The LinkedGeoData and DBpedia datasets are widely used and continue to experience a steady increase in daily query activity. Ensuring their availability and optimizing query performance are critical to meeting user demands. As the host of the canonical DBpedia SPARQL endpoint and the LOD Cloud Cluster cache of Linked Data datasets, including LinkedGeoData, for over a decade, OpenLink has amassed extensive query logs. These logs will be analyzed to identify the most frequently used queries and recurring query patterns. The insights gained will serve as a foundation for training DRL models to optimize query execution plans, improve join ordering, and enhance overall query performance, addressing scalability challenges effectively.

In order to specify the use case scenario for the LinkedGeoData dataset, by analyzing the query logs from the current deployment, we will identify common usage patterns. Additionally, we will use existing

geospatial benchmarks, e.g. GeoBench [11] and existing example queries from the LinkedGeoData project[3]. These approaches will allow us to define a set of SPARQL queries (or query templates) which mimic usual usage patterns of the dataset, and which we can use to benchmark the performance of the original and the ML-enabled deployment of the LinkedGeoData dataset. The ML-enabled deployment refers to the use of a Virtuoso instance which implements the ML-based improvements to the SPARQL query optimization layer.

Similarly, we will analyze the query logs from the current deployment of DBpedia in order to identify common usage patterns and the most accessed parts of the dataset. With this, we will develop a set of SPARQL queries which mimic the real-world usage patterns of DBpedia, and use them to benchmark the DBpedia ML-enabled deployment, as well.

Requirements

The previously mentioned insights led to the requirements presented below.

ID	Title	Description	Priority
3-1	LGD Facet Count Query	Evaluation time for this type of queries from GeoBench for different query parameters should be interactive	High
3-2	LGD Instance Query	Evaluation time for this type of queries from GeoBench for different query parameters should be interactive	High
3-3	LGD Instance Aggregation Query	Evaluation time for this type of queries from GeoBench for different query parameters should be interactive	High
3-4	LGD Example Queries	Evaluation time for these types of example queries should be interactive	Medium
3-5	DBPedia Typical Queries	Evaluation time for these types of queries should be reasonable	High
3-6	Query Log Analysis	LGD and DBPedia query logs analysis will be performed in order to identify common usage patterns, queries and query patterns	High

Benchmark Data (Data and Test Queries)

This use case will use the data available in the LinkedGeoData and DBpedia datasets. It will use separate sets of SPARQL queries for benchmarking the two datasets, as outlined above.

3 Alignment of Functional Requirements to Work Packages

Section 2 presented the use case specific requirements. Some of them need to be fulfilled within different SPARQL-ML components and others are use case specific. In the following table, we map the requirements to the corresponding work package.

[3] http://linkedgeodata.org/docs/examples/osm-queries.html

Task ID	Description	Use Case Requirement ID
WP2 - Knowledge Graphs Creation, Storage and Integration		
T2.1	SPARQL Benchmark Curation with Real-world Industry Queries	1-1 to 1-6
T2.2	Integrate Tentris into CMEM Architecture	1-1 to 1-6
T2.3	Integrate Tentris into CMEM Data-Integration	1-1 to 1-6
T2.4	Test and Evaluate Overall Usability	1-1 to 1-6
WP3 - Machine Learning for SPARQL Query Optimization in Triplestores		
T3.1	SPARQL Query Embeddings	2-3, 2-10
T3.2	Deep Reinforcement Learning for SPARQL	2-1 to 2-4
T3.3	Benchmarking the Proposed DRL System	1-1 to 1-5, 2-1 to 2-5, 2-9, 3-1 to 3-6
WP4 - Machine Learning for Federated SPARQL Query Optimization Over Multiple Endpoints		
T4.1	Join-Aware Source Selection	2-4, 2-1, 2-2
T4.2	ML-Based Optimized Query Plan Generation and Implementation	2-1 to 2-4
T4.3	Benchmarking the Proposed DRL-based Federation Engine	1-1 to 1-5, 2-1 to 2-5, 2-9, 3-1 to 3-6
WP5 - Use Cases		
T5.1	Benchmarking for Disruptive Innovation in Data Management Use Case	1-1 to 1-6
T5.2	Linked Cancer Genome Atlas Use Case	2-1 to 2-10
T5.3	LinkedGeoData and DBpedia Use Case	3-1 to 3-6

4 Conclusion

In this deliverable, we outlined the use case specifications relevant to the SPARQL-ML project. We provided an overview of the project's use cases, along with the elicitation process, functional requirements, and relevant data sources. A summary of these specifications is provided in the table below.

Use Case	Linked TCGA	LinkedGeoData & DBpedia	Business Process Automation (BPA)
Description	• Linked TCGA is the RDF version of the Cancer Genome Atlas (TCGA) data. Currently it has over 20 billion triples. • Querying such large-scale biomedical data requires intelligent query optimization techniques to achieve low-latency results while ensuring high recall and precision. • The goal of this use case is to distribute the Linked TCGA dataset across multiple triplestores and leverage deep reinforcement learning (DRL) techniques to optimize query execution in both centralized (single triplestore) and federated SPARQL settings. – **WP3**: Developing **DRL-based query optimizers** for individual triplestores. – **WP4**: Implementing **ML-driven federated query processing** to efficiently retrieve and integrate TCGA data from multiple SPARQL endpoints.	• LinkedGeoData uses the information collected by the OpenStreetMap project and makes it available as an RDF Knowledge Graph • DBpedia dataset contains structured content from the information created in the Wikipedia project and publishes it as an RDF Linked Data Knowledge Graph • Both datasets contain more than a billion triples, which can be challenging hosting on a single server SPARQL endpoint	• This approach addresses the growing complexity of data management and integration, allowing Eccenca to stay at the forefront of disruptive innovations in the field. At the core of this benchmarking process is the FedShop benchmark, a realistic e-commerce scenario that simulates the challenges of large-scale federated query processing. • The FedShop environment involves autonomous vendors, rating sites, and product data sources, each represented by SPARQL endpoints. By testing query performance under these conditions, Eccenca can assess how well its platform handles complex, federated queries across multiple data sources, simulating real-world use cases. The benchmarking strategy combines FedShop with Eccenca's proprietary testing suite, which includes a range of integration and performance tests.

Data Specification	• **LargeRDFBench** as the primary benchmark. • **Custom TCGA benchmark** for domain-specific queries.	• We will use the existing data from the LinkedGeoData and DBpedia datasets • We will identify existing or develop new sets of SPARQL queries which mimic typical use-case scenarios for using both LinkedGeoData and DBpedia datasets, in order to create a benchmark to test the performance improvement from the 3DFed architecture	• At this core are RDF Knowledge Graphs (KGs), which are structured data models represented in standard RDF formats like Turtle or JSON-LD. These graphs consist of entities (e.g., products, vendors, ratings) and their relationships (e.g., "sold by," "rated by") that are queried using the SPARQL query language. • The benchmarking process uses FedShop, an e-commerce simulation where data is distributed across multiple SPARQL endpoints.
Mapping Interface	• RDF data formats: **RDF/XML, Turtle**.	• RDF/XML • Turtle • XML	• RDF data formats: **RDF/XML, Turtle**.

SPARQL-ML Related Metrics	**For WP3 (Single Triple-store Optimization)** • Query execution time reduction using **DRL-based join ordering**. • Efficient **query embeddings** (RDF2Vec, Dice embeddings). • Comparison with **Blazegraph v2.1.4** as a baseline. **For WP4 (Federated Query Optimization)** • **Source selection accuracy**. • **Query execution time** across multiple endpoints. • **Reduction in network traffic**. • Comparison with **DARQ** as a baseline.	• Improvement in average query execution times in SPARQL • It will be based on common use-case scenarios for both datasets	• Query execution time: The time taken for a SPARQL query to execute from submission to retrieval of the results. • Query scalability: The system's ability to maintain performance as the volume of data and number of federated endpoints grows. • Result accuracy: The correctness of the query results returned by the system, ensuring that all relevant data is retrieved and no false positives or negatives occur. • Integration Latency: The time taken for the system to integrate new data sources into the CMEM platform and begin using them in queries.

Expected Results	• Efficient execution of **large-scale Linked TCGA queries**. • **Optimized federated query processing** with reduced network overhead. • **Parallel execution and pipelining** for improved query throughput.	• Significant performance improvement of SPARQL queries when accessing the data • Will be measured on both LinkedGeoData and DBpedia datasets	• The expected results of Eccenca's benchmarking process will validate the scalability, efficiency, and robustness of the Corporate Memory platform in managing large-scale federated queries across distributed RDF Knowledge Graphs. • The platform should consistently meet or exceed key performance benchmarks, demonstrating its ability to handle the challenges of modern data integration and SPARQL query optimization in a dynamic, real-world environment.
Expected Impact of SPARQL-ML	**ML-based query optimization** for large RDF datasets. **Federated SPARQL processing** using DRL. **Scalable data distribution** across triplestores. **Enhanced performance** for biomedical Linked Data applications.	• Automatic and dynamic data distribution across a cluster of SPARQL endpoint servers • Faster data access for the largely popular LinkedGeoData and DBpedia datasets	• The expected impact of Eccenca's benchmarking process will be far-reaching, benefiting not only the performance and scalability of the CMEM platform but also empowering businesses with the tools to handle large-scale data environments efficiently. • By optimizing query performance, enhancing data integration, and ensuring system scalability, Eccenca will enable organizations to streamline their operations, make data-driven decisions faster, and innovate more effectively, all while establishing itself as a leader in the field of semantic technologies and data integration

References

[1] Maribel Acosta, Maria-Esther Vidal, Tomas Lampo, Julio Castillo, and Edna Ruckhaus. ANAPSID: An Adaptive Query Processing Engine for SPARQL Endpoints. In Lora Aroyo, Chris Welty, Harith Alani, Jamie Taylor, Abraham Bernstein, Lalana Kagal, Natasha Noy, and Eva Blomqvist, editors, *The Semantic Web – ISWC 2011*, volume 7031 of *Lecture Notes in Computer Science*, pages 18–34. Springer Berlin Heidelberg, 2011.

[2] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A Nucleus for a Web of Open Data. In *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference*, ISWC'07/ASWC'07, page 722–735, Berlin, Heidelberg, 2007. Springer-Verlag.

[3] Minh-Hoang Dang, Julien Aimonier-Davat, Pascal Molli, Olaf Hartig, Hala Skaf-Molli, and Yotlan Le Crom. Fedshop: A benchmark for testing the scalability of sparql federation engines. In *International Semantic Web Conference*, pages 285–301. Springer, 2023.

[4] Olaf Görlitz and Steffen Staab. SPLENDID: SPARQL Endpoint Federation Exploiting VoID Descriptions. In *O. Hartig, A. Harth, and J. F. Sequeda, editors, 2nd International Workshop on Consuming Linked Data (COLD 2011) in CEUR Workshop Proceedings*, volume 782, October 2011.

[5] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, and Christian Bizer. DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web Journal*, 6, 01 2014.

[6] Gabriela Montoya, Hala Skaf-Molli, and Katja Hose. The odyssey approach for optimizing federated sparql queries. In *The Semantic Web – ISWC 2017: 16th International Semantic Web Conference, Vienna, Austria, October 21–25, 2017, Proceedings, Part I*, page 471–489, Berlin, Heidelberg, 2017. Springer-Verlag.

[7] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org. https://www.openstreetmap.org, Accessed on: 15.11.2024.

[8] Muhammad Saleem, Ali Hasnain, and Axel-Cyrille Ngonga Ngomo. Largerdfbench: a billion triples benchmark for sparql endpoint federation. *Journal of Web Semantics*, 48:85–125, 2018.

[9] Muhammad Saleem, Alexander Potocki, Tommaso Soru, Olaf Hartig, and Axel-Cyrille Ngonga Ngomo. Costfed: Cost-based query optimization for sparql endpoint federation. *Procedia Computer Science*, 137:163–174, 2018. Proceedings of the 14th International Conference on Semantic Systems 10th – 13th of September 2018 Vienna, Austria.

[10] Andreas Schwarte, Peter Haase, Katja Hose, Ralf Schenkel, and Michael Schmidt. FedX: Optimization Techniques for Federated Query Processing on Linked Data. In Lora Aroyo, Chris Welty, Harith Alani, Jamie Taylor, Abraham Bernstein, Lalana Kagal, Natasha Noy, and Eva Blomqvist, editors, *The Semantic Web – ISWC 2011*, volume 7031 of *Lecture Notes in Computer Science*, pages 601–616. Springer Berlin Heidelberg, 2011.

[11] Mirko Spasić. Design of Geospatial Benchmarking System and Performance Evaluation of Virtuoso and PostGIS. In Milan Zdravković, Miroslav Trajanović, and Zora Konjović, editors, *Proceedings of ICIST 2015 - 5th International Conference on Information Society and Technology*, volume 1, pages 154–159. Society for Information Systems and Computer Networks, 2015.

[12] Claus Stadler, Jens Lehmann, Konrad Höffner, and Sören Auer. LinkedGeoData: A Core for a Web of Spatial Open Data. *Semantic Web Journal*, 3:333–354, 01 2012.