

# The Need for Better RDF Archiving Benchmarks

Olivier Pelgrin<sup>1,\*</sup>, Ruben Taelman<sup>2</sup>, Luis Galárraga<sup>3</sup> and Katja Hose<sup>4,1</sup>

<sup>1</sup>Aalborg University, Denmark

<sup>2</sup>Ghent University, Belgium

<sup>3</sup>INRIA, France

<sup>4</sup>TU Wien, Austria

## Abstract

The advancements and popularity of Semantic Web technologies in the last decades have led to an exponential adoption and availability of Web-accessible datasets. While most solutions consider such datasets to be static, they often evolve over time. Hence, efficient archiving solutions are needed to meet the users' and maintainers' needs. While some solutions to these challenges already exist, standardized benchmarks are needed to systematically test the different capabilities of existing solutions and identify their limitations. Unfortunately, the development of new benchmarks has not kept pace with the evolution of RDF archiving systems. In this paper, we therefore identify the current state of the art in RDF archiving benchmarks and discuss to what degree such benchmarks reflect the current needs of real-world use cases and their requirements. Through this empirical assessment, we highlight the need for the development of more advanced and comprehensive benchmarks that align with the evolving landscape of RDF archiving.

## 1. Introduction

The continuous advancement and widespread adoption of Semantic Web technologies have generated a growing demand for robust systems to manage knowledge graphs. This demand is particularly pronounced for RDF, the Semantic Web's most prevalent and accessible data model. Along with the rest of the Web, Semantic Web data is continuously evolving [1, 2, 3]. This has inspired related work on capturing metadata, such as RDF-star [4, 5], and in general raised the need to keep track of the revision history of those datasets for the sake of multiple applications. Examples are version control or historical data analytics, which, in turn, have sparked the development of dedicated techniques and systems for RDF archiving [3].

The availability of widely adopted benchmarks is of crucial importance for the development of RDF archiving systems. Standardized benchmarks enable the impartial evaluation of new indexing and storage techniques, as well as the performance of query engines. Although numerous benchmarks have been designed specifically for evaluating RDF stores [6, 7, 8], the number of benchmarking options for RDF archiving systems remains limited [9].

In this paper, we present an analysis of the current state of RDF archiving benchmarks through an evaluation of their strengths and limitations. We show that despite advancements in

---

*MEPDAW'23: Managing the Evolution and Preservation of the Data Web, November 06, 2023, Athens, Greece*


\*Corresponding author.

✉ olivier@cs.aau.dk (O. Pelgrin); ruben.taelman@ugent.be (R. Taelman); luis.galarraga@inria.fr (L. Galárraga); katja.hose@tuwien.ac.at (K. Hose)

🆔 0000-0002-1025-9687 (O. Pelgrin); 0000-0001-5118-256X (R. Taelman); 0000-0002-0241-5379 (L. Galárraga); 0000-0001-7025-8099 (K. Hose)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

the field, current benchmarks do not sufficiently capture emerging challenges faced by archiving systems. We use this finding to derive a set of requirements, that we believe, are essential for benchmarks to advance research and development of RDF archives.

The remainder of this paper is organized as follows. First, we discuss the current state of RDF archiving research and relevant benchmarks in Section 2. Second, in Section 3, we discuss the shortcomings of current RDF archiving benchmarks and our recommendations and requirements for the future. Finally, Section 4 concludes the paper.

## 2. Related Work

We now provide a brief survey of the available RDF archiving systems as well as of existing languages and SPARQL extensions designed for RDF archives. Furthermore, this section delves into existing benchmarks tailored for assessing the performance of RDF archiving systems.

### 2.1. RDF Archiving

RDF archiving, at its core, consists of storing and querying the entire evolution history of an *RDF graph*. This has proven to be a challenging task due to the additional temporal dimension compared to traditional RDF stores. While the design of efficient indexing and querying systems for RDF archives is still an ongoing effort, multiple approaches have been proposed throughout the years [3]. Existing works can generally be categorized into three main paradigms [1], Independent Copies (IC), Change-based (CB), and Timestamp-based (TB), with some modern approaches proposing the use of a combination of those [10, 11, 12, 13]. Some approaches are now able to scale to much larger RDF archives compared to early proposals [10], however querying capabilities remain limited. Efficient processing of complex archive queries is one of the key areas of development for the future.

In contrast to conventional RDF, the existence of multiple versions within an RDF archive introduces the need for novel query types that can be hardly expressed in standard SPARQL. Some approaches propose the extension of SPARQL to support temporal queries, i.e., by specifying a timestamp or interval in which the query results should hold [14]. Other works attempt to formally categorize the different possible types of queries on RDF archives [1, 9], but do not address the implementation of these categorizations via formal SPARQL extensions.

### 2.2. Benchmarks for RDF Archives

Benchmarks play a crucial role in guiding the development of systems by facilitating their evaluation and enabling comparisons with existing systems in terms of implementation and design. Due to being a relatively new area in RDF data management, we only account for three benchmarks tailored for RDF archiving in the literature: EvoGen [15], BEAR [1], and SPBv [9].

EvoGen [15] is a benchmark based on the LUBM [8] data generator extended to support evolving RDF scenarios. The benchmark data can be configured on the desired number of versions and the magnitude of changes. The querying workload is derived from the 14 LUBM queries and includes variations of materialization, delta, and mixed queries. Due to the nature of the LUBM queries, support for RDFS reasoning is needed to resolve the complete result sets.

BEAR [1] is a benchmark for RDF archives consisting of three different RDF archives. Those different flavours, namely *BEAR-A*, *BEAR-B* and *BEAR-C*, are extracted from real-world datasets,

and are characterised by their various sizes and change behaviour. BEAR comes with predefined query workloads, based on single triple pattern queries for both BEAR-A and BEAR-B, while for BEAR-C, a set of 10 full SPARQL queries are proposed.

SPBv [9] is a benchmark for RDF archives that consists of a data generator based on the Semantic Publishing Benchmark (SPB) [16] from the Linked Data Benchmark Council (LDBC) [7]. The number of versions and the size of the data can be configured, as well as the number of generated queries. The generated data comes as full versions, changesets, or both. The query workload consists of SPARQL queries where versions are represented as named graphs.

### 3. Benchmarking RDF Archives

In this section, we examine the qualities and features that a benchmark for RDF archiving should strive to possess. We propose that benchmarks for RDF Archives should strive for three main overarching qualities, namely *reproducibility*, *realism*, and *configurability*. *Reproducibility* represents the ease at which the benchmark results can be shared and reproduced by others. *Realism* is about how the benchmark setting, both in the choice of dataset and query loads, models or emulates the real world. *Configurability* represents the ability of the benchmark to propose workloads of various sizes, relevant for a wide range of system configurations and use cases. We further detail our recommendations of a concrete implementation of the aforementioned qualities by first detailing the choice of data. We then will discuss the design of query workloads, and finally, we discuss whether existing benchmarks fulfill those requirements.

#### 3.1. Dataset

The choice of data is an important aspect when designing a benchmark. Current benchmarks use either a configurable generator for synthetic data [9, 15], or directly provide data based on existing real world datasets [1]. As discussed by Duan et al. [6], many data generators produce data that is not necessarily representative of real-world RDF datasets. However, they also demonstrate the possibility to make generators truer to the real world by taking into account their proposed *coherence* metric in the generation process. We are although not aware of any other generator-based benchmark for RDF archiving taking advantage of this metric.

Most importantly, a benchmark should cover different, realistic, scaling options. In the RDF archiving world, the scaling options do not only cover different data sizes, but also the history's size, i.e. the number of versions and the magnitude of changes within each version. Generator-based benchmarks should provide users with all the necessary scaling parameters, while real-world-based benchmarks should offer different datasets scaling along those axes.

#### 3.2. Query Workload

Early RDF archiving systems could be adequately tested with single triple pattern queries, but contemporary archiving benchmarks should prioritize comprehensive SPARQL query workloads. We believe that efficient support for full SPARQL represents a major challenge that RDF archiving systems currently need to solve. Consequently, in order to fulfill our *realism* requirement, benchmarks should provide comprehensive assessment of those capabilities, guiding the development of existing and new systems. Benchmark query workloads should be carefully

designed to align with real-world use cases. Following recommendations from the LDBC [7], a "choke-point" approach to the design of the benchmark should be considered through a comprehensive evaluation of real-world RDF archive usages.

Finally, the lack of an accepted standard to formulate archiving queries into SPARQL is a major brake for the design of benchmark queries. Addressing this issue necessitates a dedicated standardization effort, drawing inspiration from the RDF stream community, and the RSP-QL standardization [17]. This would require a broader study of the overlap between RDF stream processing and RDF archiving, notably on the relation between temporal graphs and archives.

**Table 1**

Comparison table of existing RDF Archiving benchmarks.

|             | Dataset    | Reproducibility | Realism (data) | Realism (queries) | Configurability |
|-------------|------------|-----------------|----------------|-------------------|-----------------|
| EvoGen [15] | Synthetic  | -/+             | -              | +                 | +               |
| BEAR [1]    | Real-world | +               | +              | -                 | -               |
| SPBv [9]    | Synthetic  | -/+             | -              | +                 | +               |

### 3.3. Comparison of Existing RDF Archiving Benchmarks

Table 1 summarizes the characteristics of the existing RDF archiving benchmarks. Among the available benchmarks, two of them rely on synthetic data generated through a data generator. Generator-based systems fulfill the *configurability* criteria easily due to their nature, but may fall short of also proving their *realism*, while their *reproducibility* is dependent on the sharing of the exact parameters and random seed. Both EvoGen [15] and SPBv [9] provide SPARQL queries of varied nature, but only focus on the generation of one restrictive type of datasets, which have not been evaluated realism, e.g., via the coherence metric [6]. BEAR [1] on the other hand provides datasets of various sizes, based on real-world data. This increases the *reproducibility* and *relevance* of the benchmark compared to generator-based ones. The number of scalability options is however limited, but BEAR still offers five different alternative datasets. However, 10 full SPARQL queries are only provided for one of the datasets, the others being limited to single triple pattern queries. As discussed in Section 3.2, this limits BEAR’s *realism*, and makes the evaluation of SPARQL-capable archiving systems quite limited.

## 4. Conclusion

In this paper, we presented the current state in RDF archiving systems and benchmarks. We have proposed a set of requirements that benchmarks should have in order to contribute to the advancement of the field. We showed that among the only three available benchmarks for RDF archiving systems, none of them proposes a satisfactory set of features. This ranges from a general lack of realism w.r.t. the real world, lack of SPARQL support, or concerns with reproducibility. We see several areas open for future work. First, precisely defining the semantics and syntax of SPARQL archive queries would benefit greatly to the wider RDF community. This would open the door for standardized support across various RDF stores and research systems. Secondly, benchmarks relevant to the modern challenges faced by RDF archiving applications and systems are needed to guide and evaluate efforts in that area. We believe that this is paramount to current development efforts of fully-fledged RDF archiving systems.

## Acknowledgments

This research is partially funded by the Poul Due Jensen Foundation and the Independent Research Fund Denmark (DRF) under grant agreement no. DRF-8048-00051B, the TAILOR Network (EU Horizon 2020 research and innovation program under GA 952215), and the Research Foundation – Flanders (FWO) (1274521N).

## References

- [1] J. D. Fernández, J. Umbrich, A. Polleres, M. Knuth, Evaluating query and storage strategies for RDF archives, *Semantic Web Journal* 10 (2019) 247–291.
- [2] K. Hose, Knowledge Graph (R)Evolution and the Web of Data, in: *MEPDAW@ISWC, 2021*, pp. 1–7.
- [3] O. Pelgrin, L. Galárraga, K. Hose, Towards fully-fledged archiving for RDF datasets, *Semantic Web Journal* 12 (2021) 903–925.
- [4] O. Hartig, Foundations of RDF\* and SPARQL\* (An Alternative Approach to Statement-Level Metadata in RDF), in: *AMW, 2017*.
- [5] G. Abuoda, C. Aebeloe, D. Dell’Aglío, A. Keen, K. Hose, StarBench: Benchmarking RDF-star Triplestores, in: *QuWeDa@ISWC, 2023*.
- [6] S. Duan, A. Kementsietsidis, K. Srinivas, O. Udrea, Apples and oranges: a comparison of RDF benchmarks and real RDF datasets, in: *SIGMOD, 2011*, pp. 145–156.
- [7] P. A. Boncz, I. Fundulaki, A. Gubichev, J. L. Larriba-Pey, T. Neumann, The linked data benchmark council project, *Datenbank-Spektrum* 13 (2013) 121–129.
- [8] Y. Guo, Z. Pan, J. Heflin, LUBM: A benchmark for OWL knowledge base systems, *J. Web Semant.* 3 (2005) 158–182.
- [9] V. Papakonstantinou, G. Flouris, I. Fundulaki, K. Stefanidis, Y. Roussakis, Spbv: Benchmarking linked data archiving systems, in: *BLINK@ISWC, volume 1932, 2017*.
- [10] O. Pelgrin, R. Taelman, L. Galárraga, K. Hose, Scaling Large RDF Archives To Very Long Histories, in: *ICSC, 2023*, pp. 41–48.
- [11] R. Taelman, M. Vander Sande, J. Van Herwegen, E. Mannens, R. Verborgh, Triple Storage for Random-Access Versioned Querying of RDF Archives, *J. Web Semant.* (2018).
- [12] N. Arndt, P. Naumann, N. Radtke, M. Martin, E. Marx, Decentralized collaborative knowledge management using git, *J. Web Semant.* 54 (2019) 29–47.
- [13] O. Pelgrin, R. Taelman, L. Galárraga, K. Hose, GLENDA: Querying RDF Archives with full SPARQL, in: *ESWC, 2023*.
- [14] F. Grandi, T-SPARQL: A sql2-like temporal query language for RDF, in: *ADBIS, 2010*, pp. 21–30.
- [15] M. Meimaris, G. Papastefanatos, The EvoGen Benchmark Suite for Evolving RDF Data, in: *MEPDAW@ESWC, 2016*, pp. 20–35.
- [16] V. Kotsev, N. Minadakis, V. Papakonstantinou, O. Erling, I. Fundulaki, A. Kiryakov, Benchmarking RDF query engines: The LDBC semantic publishing benchmark, in: *BLINK@ISWC, 2016*.
- [17] D. Dell’Aglío, E. D. Valle, J. Calbimonte, Ó. Corcho, RSP-QL semantics: A unifying query model to explain heterogeneity of RDF stream processing systems, *Int. J. Semantic Web Inf. Syst.* 10 (2014) 17–44.