

# Towards an Open Extensible Framework for Empirical Benchmarking of Data Management Solutions: LITMUS

Harsh Thakkar<sup>(✉)</sup>

Enterprise Information Systems Lab, University of Bonn, Bonn, Germany  
[hthakkar@uni-bonn.de](mailto:hthakkar@uni-bonn.de)

**Abstract.** Developments in the context of Open, Big, and Linked Data have led to an enormous growth of structured data on the Web. To keep up with the pace of efficient consumption and management of the data at this rate, many Data Management Solutions There exists many efforts for benchmarking these domain specific DMSs, however, (*i*) reproducing these third party benchmarks is an extremely tedious task, and (*ii*) there is a lack of a common framework which enables and advocates the extensibility and re-usability of the benchmarks. We propose LITMUS, one such framework for benchmarking data management solutions. LITMUS will go beyond classical storage benchmarking frameworks by allowing for analysing the performance of DMSs across query languages. In this early stage doctoral work, we present the LITMUS concept as well as the considerations that led to its preliminary architecture, and progress reported so far in its realisation.

## 1 Introduction

Vast amounts of structured (following Linked Data principles) and un/semi-structured data is constantly being made available on the Web, often in an open manner<sup>1</sup>, and within organisations. This rapid growth of data, available across organisations, has affected the data management layer of modern applications.

Consequently, organisations are increasingly facing the need to find data management tools suited for the specific tasks at the core of their information management. Choosing the best<sup>2</sup> data management solution is nonetheless challenging due to the limited comparability and compatibility of existing evaluation results and benchmarks. With regard to the limited domain expertise of the end user, the need for standardised frameworks to benchmark and analyse the existing diverse data management platforms is consequently of paramount importance.

Despite the growing interest and use in both research and the industry communities, currently the creators of benchmarks for Data Management Solutions (DMS) [1,4] do not offer a common suite/platform for performing cross-domain

<sup>1</sup> With *open* we follow the Open Data Definition (<http://opendefinition.org/>).

<sup>2</sup> We refer to best in terms of fitness for use.

benchmarks (i.e., one-to-one performance comparison of RDF, Graph, or Relational engines). In addition, there is no significant baseline to compare these cross-domain DMSs against each other. Moreover, reproducing benchmarks is a non-trivial problem owing to reasons such as non-standardised setup configurations, lack of publicly available resources (such as scripts, libraries or packages) and lack of transparent evaluation policies. Results in areas such as named entity recognition and linking [25] as well as question answering [23, 24] have, however, shown that the provision of standardised interfaces and measures can contribute to the improvement of the performance of software solutions.

In this early stage doctoral work we propose LITMUS, a generic approach for benchmarking DMSs. LITMUS aims to provide support to organisations aspiring to use Linked Data management technologies in a wide spectrum of applications and magnitudes. LITMUS will provide a realistic performance evaluation platform covering a plethora of heterogeneous technologies (see Sect. 4) for storage and query benchmarking. To put the reader into the context of this work, and to highlight the objectives of LITMUS, we present the following **user scenario**:

“The WDAqua research project<sup>3</sup> aims at building a data-driven question answering platform by using Web data, available in various formats, e.g., RDF, CSV, SQL, or Graph. Harsh, a researcher within the project, is responsible for ensuring efficient data management (storage and retrieval) for this project. There are a large number of DMSs, each deliberately tailored to handling specific formats of data and queries, which need to be benchmarked to select the best solution for the project’s needs. However, benchmarking of DMSs is non-trivial: it takes large amounts of human effort in designing, administering, evaluating, and analysing the diverse systems involved. Additionally, for the research project, a large set of factors, e.g., query typology, indexing speed, index size, query response time, and dataset size, need to be considered to ensure reproducibility and generality of the observed experimental results. *Harsh wants to automate the whole benchmarking process, allowing easy integration, evaluation on custom stress loads, and fast analysis of the results. He would also expect the framework to be flexible to integrate new DMSs to the plethora of existing systems and benchmark them against a baseline*”.

LITMUS will not only satisfy the requirement for automating the tedious benchmarking process, but will also offer: (1) an efficient way for replicating existing benchmarks (e.g., BSBM [4] or WAT-DIV [1]); (2) a wide set of performance evaluation metrics/indicators tailored specifically for the DMS being evaluated; and (3) quick analytical insights on performance comparison of benchmarked DMSs wrt various intrinsic factors (such as query length, query structure, etc.) employing visualisation via custom charts, graphs and tabular data.

The remainder of this article is organised as follows: Sect. 2 summarises the state of the art in benchmarking efforts, and their shortcomings, Sect. 3 sheds light on the foci, challenges, objectives and planned outcomes of LITMUS, Sect. 4 describes the conceptual architecture of LITMUS, its target audience, and Sect. 5 concludes with the work progress and future agenda.

---

<sup>3</sup> WDAqua ITN – (<http://wdaqua.eu>).

## 2 State of the Art

Benchmarking is widely used for evaluating data stores (DMSs). Benchmarks exist for a variety of levels of abstraction from simple data models to graphs and triple stores, to entire enterprise information systems. We describe the current state of the art in benchmarking, in particular for: (a) Relational databases, (b) Graph databases, (c) RDF stores, and (d) cross-domain benchmarking efforts. We identify the scope and shortcomings of existing benchmarking efforts, to determine the gaps that LITMUS needs to take into consideration.

1. In *Relational* DMSs, the benchmarks of the Transaction Processing Performance Council (**TPC**) [14] are well established. TPC uses discrete metrics for measuring the performance of the relational DMS. The online transaction processing benchmarks **TPC-C** and **TPC-E** use a transactions per minute metric. The analytics **TPC-H** and decision support **TPC-DS** benchmarks use the queries per hour and cost per performance metrics, respectively.
2. For *Graph* DMSs, there exist benchmarks, some of which are in their early stages (such as **HPC Scalable Graph Analysis Benchmark** [6], **Graph 500** [13], **XGDBench** [5]) that deal with graph suitability transformations and graph analysis. However, they do not succeed to define standards for graph modelling and query languages.
3. Benchmarking *RDF* DMSs. The substantial increase in the number of applications that use *RDF* data has encouraged the need for large-scale benchmarking efforts on all aspects of the Linked Data life cycle, mostly focusing on query processing [15]. RDF DMS benchmarks use real (i.e., DBpedia or Wikidata) and synthetic (i.e., Berlin SPARQL Benchmark or WAT-DIV) datasets to evaluate DMS performance over custom stress-loads and setup environments.<sup>4</sup> DBpedia SPARQL Benchmark (**DBPSB**) [12] assesses RDF DMSs performance over DBpedia by creating a query workload derived from the DBpedia query logs. The aim of the Lehigh University Benchmark (**LUBM** [8]) is to evaluate the performance of Semantic Web triple stores over a large synthetic dataset that complies to a university domain ontology. The Waterloo SPARQL Diversity TEST Suite (**WatDiv** [1]) provides data and query generators to enable benchmarking of RDF DMSs against a varying query structure (also complexity) to understand correlation of query typology with the variance in DMS performance. **SP2Bench** [21], one of the most commonly used synthetic data based benchmarks, uses the schema of the DBLP bibliographic dataset<sup>5</sup> to generate arbitrarily large datasets.
4. Benchmarking *Cross-domain* DMSs. There are only a few efforts that benchmark cross-domain DMS so far. The Berlin SPARQL Benchmark (**BSBM** [4]) is a synthetic data benchmark, based on an e-commerce use cases built around a set of products offered by different vendors. It provides the dataset and queries for both RDF and Relational DMS benchmarking. **Pandora**<sup>6</sup>,

---

<sup>4</sup> <https://www.w3.org/wiki/RdfStoreBenchmarking>.

<sup>5</sup> <http://dblp.uni-trier.de/db/>.

<sup>6</sup> <http://pandora.ldc.usb.ve/>.

uses the Berlin SPARQL Benchmark data to benchmark RDF stores against relational stores (Jena-TDB, MonetDB, GH-RDF-3X, PostgreSQL, 4Store). **Graphium** [7] is a similar study benchmarking RDF stores against Graph stores (Neo4J, Sparksee/DEX, HypergraphDB, RDF-3X) on graph datasets including a 10M triple graph data generated using the Berlin SPARQL Benchmark data generator. More recently, the **LDBC** [2] focused on combining industry-strength benchmarks for graph and RDF data management systems. The LDBC introduces a new choke-point analysis methodology for developing benchmark workloads, which tries to combine user input with feedback from system experts.

Efforts have so far been focused on benchmarking single-domain (RDF-vs-RDF stores, Graph-vs-Graph stores, etc.) DMSs, despite the need for integrating cross-domain DMSs and automating the benchmarking process. LITMUS aims at addressing these shortcomings and serve as an open, extensible platform allowing easy integration, benchmarking and performance comparison of diverse DMSs. To the best of our knowledge, no such extensible and reusable framework exists, which enables the exploration and analysis of a wide spectrum of DMSs.

### 3 Problem Statement and Contributions

The following generic research question acts as a guiding force to our efforts: *How can diverse cross-domain DMSs be benchmarked in an established standard environment*<sup>7</sup>? We hypothesise that: Devising a **generic** data and query translation mechanism together with a defined set of key performance indicators (KPIs) will enable the comparison of diverse cross-domain DMSs

#### 3.1 Challenges to Be Addressed

The aim of the doctoral work is to validate the proposed hypothesis by developing such a benchmarking platform. In doing so we identify three key challenges (sub-research questions) which need to be addressed, namely:

- **C1 Data conversion:** This challenge mandates the development of a generic data conversion mechanism for converting the RDF data to a format interpretable by the corresponding DMSs (i.e., RDF, pure graphs, or SQL). The goal of this task is to efficiently represent RDF data in multiple formats, keeping the end user as secluded as possible from the underlying technicalities of the conversion. This leads us to our first research question: **RQ1: What are the methods to convert RDF into proprietary data formats?**
- **C2 Query translation:** Cross-domain benchmarking of DMSs demand that queries be represented in all languages and formats supported by the respective tools. Query languages differ in their structure and expressivity.

---

<sup>7</sup> By established standard environment we mean that all benchmarks will run under the same conditions and are not affected by external factors (e.g. different memory allocation by the OS).

For instance, complex path queries (in SPARQL, in particular Kleene stars) cannot be expressed in an equivalent SQL query [26]. Thus, there is a need to develop an intermediate mechanism to translate the queries from one form to the other (e.g., from SPARQL to Gremlin, SQL, etc.). This requires an exhaustive study of the query languages' specifications. The main challenge is to identify the correct mappings between different languages, preserving the semantics of the original query. Thus our second research question is: **RQ2: What are the semantic preserving methods/approaches for translating SPARQL queries to a graph query language<sup>8</sup> such as Gremlin?**.

- **C3 Performance indicators:** The performance of a DMS can be assessed with respect to a wide variety of indicators (referred to as performance metrics or key performance indicators (KPIs)). Dealing with the diverse characteristics of the DMSs, it is necessary to explore a range of performance indicators in contrast to traditional ones, namely precision, recall, index size, storage size, number of triples, number of unique instances, query response time, etc. The work by LDBC [2] presents a related study on this topic. We would like to dig deeper into this and other works, compare and analyse the strengths and limitations of the KPIs, ultimately select a set of KPIs to be considered for evaluation of these DMSs. Thus, **RQ3: What are the strengths and the limitations of the existing KPIs, and to what extent do they reflect the performance of a DMS.**

### 3.2 Focus of the LITMUS Framework

The focus of LITMUS is to bridge the gaps in adopting, deploying and scaling the consumption of Linked Data. LITMUS thrives on simplifying the use, assessment and the performance analysis of a wide spectrum of cross-domain DMSs. In particular, the LITMUS framework will:

- **F1** enable a common platform for benchmarking and comparing a plethora of cross-domain DMSs, and reproducing existing third-party benchmarks;
- **F2** create (*i*) interoperable machine-readable evaluation reports and (*ii*) scientific studies on the correlation of a variety of factors (such as query typology, data structures used for indexing, etc.) with respect to the performance of DMSs;
- **F3** recommend particular DMSs and benchmarks based on a set of requirements predefined by the user.

### 3.3 Planned Outcomes

The planned artifacts resulting from the LITMUS project can be classified into two categories, namely **(A1)** scientific findings and **(A2)** software.

---

<sup>8</sup> We emphasise on graph query language in this question as there exists sufficient work addressing SPARQL-SQL (relational query language) translation problem.

**A1 Scientific findings:**

- An in-depth analysis of the (research challenges, ref. Sect. 3.1) (i) various RDF data representation formats and their conversion complexity, addressing challenge (**C1**); (ii) query language expressivity and supported features striving to address the language barrier (**C2**). These studies will provide us with deep insights about the functionality of various query languages, RDF data formats, their strengths and limitations.
- An exhaustive exploratory study on the selection of performance measures for evaluating cross-domain DMSs, addressing challenge (**C3**)

**A2 Software (i.e., algorithms, scripts, tools):**

- A novel data converter of RDF data to multiple data formats (such as CSV, JSON, SQL, etc.), providing compatible data as input to the cross-domain DMSs (i.e., the software implementation of outcome **A1.(i)**, Sect. 3.3).
- A novel query translator for the automatic conversion of SPARQL to DMS-specific query language (e.g., Gremlinator, ref Sect. 4, etc.), enabling compatible query input for cross-domain DMSs (i.e., the software implementation of outcome **A1.(ii)**, Sect. 3.3)
- An open, extensible benchmarking platform, for cross-domain DMS performance evaluation and easy replication of existing benchmarks.

## 4 Research Approach and Initial Results

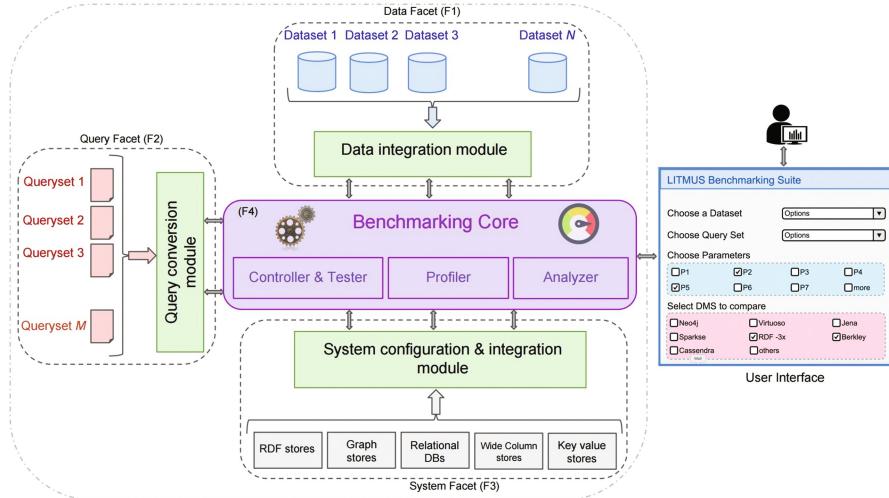
Here, we present the conceptual architecture of LITMUS. It comprises of four major facets: Data Facet (F1), Query Facet (F2), System Facet (F3), and Benchmarking Core (F4) (ref. Fig. 1). The role of each facet is as follows:

**Data Facet F1:** The Data Facet consists of the (i) Dataset(s) and the (ii) Data Integration Module. Datasets chosen for benchmarking can be real datasets such as DBpedia<sup>9</sup>, Wikidata<sup>10</sup>, synthetic datasets such as the Berlin SPARQL Benchmarking (BSBM) [4], Waterloo SPARQL Diversity Test Suite (WatDiv) [1], or hybrid datasets comprising both real and synthetic data. The *Data Integration Module* is responsible for (a) making data available to the system in the requested formats (such as N-Triples, Graphs, CSV, SQL) by carrying out appropriate data conversion and mapping tasks (cf. Challenge **C1**), and (b) loading the desired format of data to the respective DMSs selected for the benchmark.

**Query Facet F2:** The Query Facet comprises of the (i) Queryset(s), and the (ii) Query Conversion Module. The *Queryset* refers to the set of query input files. The *Query Conversion Module* will be one of the key components addressing the language barrier (Challenge **C2**). It is responsible for converting the input SPARQL queries to the respective DMSs' query languages (such as Gremlin, SQL, etc.). The conversion will be performed by developing an intermediate

<sup>9</sup> <http://wiki.dbpedia.org/>.

<sup>10</sup> <https://www.wikidata.org/>.



**Fig. 1.** The architectural overview of the LITMUS benchmarking framework [22].

language/logic representation of the input query. The aim of this module is to allow efficient conversion of a wide variety of SPARQL queries (such as path, star-shaped, and snowflake queries) to other query languages, ultimately breaking the language barrier.

**System Facet F3:** The System Facet consists of (i) DMSs and (ii) DMS Configuration and Integration module. The *DMS Configuration and Integration* module is responsible for (i) providing easy integration, via wrapper(s) or as a plug-in, of the DMSs, and (ii) monitoring and configuring the integrated DMSs for the benchmark. On top of this, this module makes use of *Docker containers*<sup>11</sup> to ensure a fair allocation of resources and to provide the necessary segregation required for conducting realistic benchmarks.

**Benchmarking Core F4:** The Benchmarking Core is the heart of the LITMUS framework, consisting of three modules: (i) Controller and Tester, (ii) Profiler, and (iii) Analyser. The *Controller and Tester* is responsible for executing the respective scripts for loading data, fetching the queries to their corresponding DMSs, validating the specified system configurations, and finally, executing the benchmark on the selected setting. The *Profiler* is responsible for: (a) generating and loading various profiles (stress loads, query variations, etc.) for conducting the benchmark tests and (b) storing the custom benchmark results. The *Analyser* is responsible for collecting the benchmark results from the *Profiler* and generates performance reports. It will perform correlation analysis between the parameters specified by the user. The final results (reports) will then presented to the end user in a suitable visualisation.

<sup>11</sup> <https://www.docker.com/>.

**Initial results.** We currently focus on curating the necessary benchmarking infrastructure for RDF and Graph DMSs. Thereafter, having achieved this milestone, we will cultivate the support for Relational DMSs. The preliminary results, can be clubbed together according to the planned outcomes (discussed in Sect. 3.3), addressing the research challenges and technological developments (ref. Sect. 3.1) of the framework, as follows:

## 1. Research challenges

(i) *Query translation*: We are currently focused on addressing the query translation challenge [RQ2] (**C2**, Sect. 3.1) developing a novel SPARQL (defacto RDF query language) to Gremlin (graph traversal (query language)), “**Gremlinator**”. We choose Gremlin over other graph query languages (such as Cypher), owing to Gremlin’s wide-spread popularity, coverage of graph DMSs and its strong support for both OLTP-based as well as OLAP-based graph processors. We are studying the underlying semantics and complexity of both the query languages for proposing a novel transformation function, mapping SPARQL algebra [3, 17] to Gremlin traversals [18–20] ensuring soundness and completeness. Consequently devising a query engine for SPARQL queries to be able to exploit the benefits of existing graph database engines, e.g., neighbourhood indexes, transaction management, and built-in graph-based tasks.

(ii) *Data conversion*: Our next milestone is to address the data conversion challenge [RQ1] (**C1** ref. Section 3.1). We start by first converting RDF to Graphs. Here, our goal is to propose a novel mechanism for generating Graphs from RDF data, theoretically transforming any RDF dataset to a pure Graph format. The related work in this topic includes efforts such as [9–11, 16] who advocate the generation of property graphs using reification. We would like to study these and other works in detail and develop a generic RDF data converter as our ultimate goal.

## 2. Implementation

This framework will be made available as open source software for encouraging research, open discussions and possible extensions to the idea. The source code, scripts, and other relevant modules are open-sourced at the Github organisation<sup>12</sup>. We are working on the query facet (**F2**) developing the query conversion module along with the continuous (incremental) development of the benchmarking core (**F4**) (ref. Fig. 1). We have developed bash-scripts and DMS dockers for the easy integration of DMSs, as a part of the System facet (**F3**). The overall development progress of the overall framework is around 25%.

## 5 Evaluation Plan and Conclusion

This doctoral work is dimensioned for three years, out of which the first year is dedicated for intense literature review. We identified the challenges and shortcomings of existing works summarised in Sect. 2 through 3. The literature review

---

<sup>12</sup> LITMUS Benchmark Suite: (<https://github.com/LITMUS-Benchmark-Suite/>).

confirms the absence of a cross-domain benchmarking platform. We first start by addressing the research challenges identified in Sect. 3.1, proposing the solutions (formally), implementing the solution (i.e., the components described in the architecture) and thereafter repeating this methodology for the planned architecture. We plan to devote a time period of six months for addressing each research challenge (i.e., **C1**, **C2** and **C3**) and the last six months for the integration, evaluation and testing of the overall framework. More than providing visualisation of DMS performance comparison and analysis scripts, what LITMUS will provide is a common open and extensible ground for independent evaluation and comparison of a given approach with respect of the state-of-the-art. This promotes and enhances not only reproducibility of the benchmarking results but also generality and experimental transparency.

**Evaluation.** We plan to evaluate our hypothesis by validating each research challenge/question defined in Sect. 3.1. The evaluation of the challenges **C1** and **C2** will be done by formally proving that the conversion/translation process is sound, complete and preserves the semantics (of the data and query). Furthermore, we will also evaluate the time complexity of the implemented converter and translator ensuring that a scalable solutions is possible for both **C1** and **C2**. We will evaluate challenge **C3**, by the means of empirical study. In this we will analyse and compare various KPIs using a wide variety of DMSs and datasets. Finally, for the whole platform, we plan to do an evaluation taking in consideration all the three components and define user scenarios (similar the one described in Sect. 1). These scenarios will be validated keeping in mind the existing benchmarks, thus proving its validity and strengths.

**Acknowledgements.** Supervised by Prof. Dr. Sören Auer and Prof. Dr. Maria-Ester Vidal. I would like to express gratitude to Prof. Dr. Jens Lehman, Dr. Christoph Lange, and Dr. Andreas Both for their quality insights on LITMUS. This work is supported by the H2020 WDAqua ITN (GA: 642795).

## References

1. Aluç, G., Hartig, O., Özsü, M.T., Daudjee, K.: Diversified stress testing of RDF data management systems. In: Mika, P., Tudorache, T., Bernstein, A., Welty, C., Knoblock, C., Vrandečić, D., Groth, P., Noy, N., Janowicz, K., Goble, C. (eds.) ISWC 2014. LNCS, vol. 8796, pp. 197–212. Springer, Cham (2014). doi:[10.1007/978-3-319-11964-9\\_13](https://doi.org/10.1007/978-3-319-11964-9_13)
2. Angles, R., Boncz, P.A., Larriba-Pey, J., et al.: The linked data benchmark council: A graph and RDF industry benchmarking effort. SIGMOD Rec. **43**(1), 27–31 (2014)
3. Angles, R., Gutierrez, C.: The expressive power of SPARQL. In: Sheth, A., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) ISWC 2008. LNCS, vol. 5318, pp. 114–129. Springer, Heidelberg (2008). doi:[10.1007/978-3-540-88564-1\\_8](https://doi.org/10.1007/978-3-540-88564-1_8)
4. Bizer, C., Schultz, A.: The berlin SPARQL benchmark. Int. J. Semant. Web Inf. Syst. **5**(2), 1–24 (2009)

5. Dayarathna, M., Suzumura, T.: XGDBench: A benchmarking platform for graph stores in exascale clouds. In: CloudCom. IEEE Computer Society (2012)
6. Dominguez-Sal, D., Urbón-Bayes, P., Giménez-Vañó, A., Gómez-Villamor, S., Martínez-Bazán, N., Larriba-Pey, J.L.: Survey of graph database performance on the HPC scalable graph analysis benchmark. In: Shen, H.T., Pei, J., Özsu, M.T., Zou, L., Lu, J., Ling, T.-W., Yu, G., Zhuang, Y., Shao, J. (eds.) WAIM 2010. LNCS, vol. 6185, pp. 37–48. Springer, Heidelberg (2010). doi:[10.1007/978-3-642-16720-1\\_4](https://doi.org/10.1007/978-3-642-16720-1_4)
7. Flores, A., Palma, G., Vidal, M.-E., et al.: GRAPHIUM: Visualizing performance of graph and RDF engines on linked data. In: Proceedings of the 2013th International Conference on Posters & Demonstrations Track-Volume, vol. 1035 (2013). CEUR-WS.org
8. Guo, Y., Pan, Z., Heflin, J.: LUBM: A benchmark for OWL knowledge base systems. *Web Semant.* **3**(2–3), 158–182 (2005)
9. Hartig, O.: Reconciliation of RDF\* and property graphs. CoRR, abs/1409.3288 (2014)
10. Hernández, D., Hogan, A., Krötzsch, M.: Reifying RDF: What works well with wikidata? In: Proceedings of the 11th International Workshop on Scalable Semantic Web Knowledge Base Systems co-located (ISWC 2015), Bethlehem, PA, USA (2015)
11. Hernández, D., Hogan, A., Riveros, C., Rojas, C., Zerega, E.: Querying wikidata: Comparing SPARQL, relational and graph databases. In: Groth, P., Simperl, E., Gray, A., Sabou, M., Krötzsch, M., Lecue, F., Flöck, F., Gil, Y. (eds.) ISWC 2016. LNCS, vol. 9982, pp. 88–103. Springer, Cham (2016). doi:[10.1007/978-3-319-46547-0\\_10](https://doi.org/10.1007/978-3-319-46547-0_10)
12. Morsey, M., Lehmann, J., Auer, S., Ngonga Ngomo, A.-C.: DBpedia SPARQL benchmark – Performance assessment with real queries on real data. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011. LNCS, vol. 7031, pp. 454–469. Springer, Heidelberg (2011). doi:[10.1007/978-3-642-25073-6\\_29](https://doi.org/10.1007/978-3-642-25073-6_29)
13. Murphy, R.C., Wheeler, K.B., Barrett, B.W., Ang, J.A.: Introducing the GRAPH 500. Cray User's Group (CUG) (2010)
14. Nambiar, R., Wakou, N., Carman, F., Majdalany, M.: Transaction processing performance council (TPC): State of the council 2010. In: Nambiar, R., Poess, M. (eds.) TPCTC 2010. LNCS, vol. 6417, pp. 1–9. Springer, Heidelberg (2011). doi:[10.1007/978-3-642-18206-8\\_1](https://doi.org/10.1007/978-3-642-18206-8_1)
15. Ngomo, A.-C.N., Röder, M.: HOBBIT: Holistic benchmarking for big linked data. ERCIM News 2016 (2016)
16. Nguyen, V., Leeka, J., Bodenreider, O., et al.: A formal graph model for RDF and its implementation. CoRR, abs/1606.00480 (2016)
17. Pérez, J., Arenas, M., Gutierrez, C.: Semantics and complexity of SPARQL. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 30–43. Springer, Heidelberg (2006). doi:[10.1007/11926078\\_3](https://doi.org/10.1007/11926078_3)
18. Rodriguez, M.A.: The gremlin graph traversal machine and language (invited talk). In: Proceedings of the 15th Symposium on Database Programming Languages, Pittsburgh, PA, USA, 25–30 October 2015 (2015)
19. Rodriguez, M.A., Neubauer, P.: The graph traversal pattern. In: Graph Data Management: Techniques and Applications (2011)
20. Rodriguez, M.A., Neubauer, P.: A path algebra for multi-relational graphs. In: Proceedings of the 27th International Conference on Data Engineering Workshops, ICDE 2011 (2011)

21. Schmidt, M., Hornung, T., Meier, M., et al.: SP<sup>2</sup>Bench: A SPARQL performance benchmark. In: de Virgilio, R., Giunchiglia, F., Tanca, L. (eds.) Semantic Web Information Management, pp. 371–393. Springer, Heidelberg (2009)
22. Thakkar, H., Dubey, M., Sejdiu, G., et al.: LITMUS: An open extensible framework for benchmarking RDF data management solutions. CoRR, abs/1608.02800 (2016)
23. Tsatsaronis, G., Balikas, G., Malakasiotis, P., et al.: An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. BMC Bioinform. **16**, 138 (2015)
24. Unger, C., Forascu, C., Lopez, V., et al.: Question answering over linked data (QALD-5). In: Working Notes of CLEF 2015, Toulouse, France (2015)
25. Usbeck, R., Röder, M., Ngomo, A.N., et al.: GERBIL: General entity annotator benchmarking framework. In: Proceedings of the 24th International Conference on World Wide Web, WWW 2015 (2015)
26. Zhang, X., Van den Bussche, J.: On the power of SPARQL in expressing navigational queries. Comput. J. **58**(11), 2841–2851 (2015)