



Developing a Benchmark Suite for Semantic Web Data from Existing Workflows

DOI:

[10.5281/zenodo.159568](https://doi.org/10.5281/zenodo.159568)

Document Version

Accepted author manuscript

[Link to publication record in Manchester Research Explorer](#)

Citation for published version (APA):

Troumpoukis, A., Charalambidis, A., Mouchakis, G., Konstantopoulos, S., Siebes, R., de Boer, V., Soiland-Reyes, S., & Digles, D. (2016). *Developing a Benchmark Suite for Semantic Web Data from Existing Workflows*. Paper presented at Workshop on Benchmarking Linked Data, Kobe, Japan. <https://doi.org/10.5281/zenodo.159568>

Citing this paper

Please note that where the full-text provided on Manchester Research Explorer is the Author Accepted Manuscript or Proof version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version.

General rights

Copyright and moral rights for the publications made accessible in the Research Explorer are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Takedown policy

If you believe that this document breaches copyright please refer to the University of Manchester's Takedown Procedures [<http://man.ac.uk/04Y6Bo>] or contact uml.scholarlycommunications@manchester.ac.uk providing relevant details, so we can investigate your claim.



Developing a Benchmark Suite for Semantic Web Data from Existing Workflows

Antonis Troumpoukis¹, Angelos Charalambidis¹, Giannis Mouchakis¹, Stasinos Konstantopoulos¹, Ronald Siebes², Victor de Boer², Stian Soiland-Reyes³, and Daniela Digles⁴

¹ Institute and Informatics and Telecommunications, NCSR ‘Demokritos’, Greece

{antru,acharal,gmouchakis,konstant}@iit.demokritos.gr

² VU University Amsterdam, the Netherlands

{v.de.boer,r.m.siebes}@vu.nl

³ eScience Lab, The University of Manchester, UK

<http://orcid.org/0000-0001-9842-9718>

⁴ Department of Pharmaceutical Chemistry, University of Vienna, Austria

daniela.digles@univie.ac.at

Abstract. This paper presents work in progress towards developing a new benchmark for federated query processing systems. Unlike other popular benchmarks, our queryset is not driven by technical evaluation, but is derived from workflows established by the pharmacology community. The value of this queryset is that it is realistic but at the same time it comprises complex queries that test all features of modern query processing systems.

Keywords: Triple store benchmarking; Pharmacology data; Distributed and federated querying.

1 Introduction

Performance benchmarks allow systems to be evaluated and compared, but designing such a benchmark is subject to considerations that are difficult to satisfy simultaneously. For benchmarking query processing systems in particular, one such consideration is the selection of the data that will be included in the benchmark and the query workload that will be applied to this data. One potential tension is the creation of a *realistic* benchmark that reflects workflows that occur commonly versus the creation of a *generic* and *informative* benchmark that tests as many characteristics of query processing systems as possible. One would expect the former to be useful for selecting what query processing infrastructure to use for specific domains and applications, and the latter to be a generic, multi-dimensional tool for evaluating the technical quality of an infrastructure.

Most well-known benchmarks use natural datasets but define an artificial queryset based on what technical characteristics should be tested [12, 11], while some benchmarks also provide synthetic data [6]. However, considering the complex and multi-dimensional nature of modern database systems, fine-tuning

generic benchmarks to specific applications can prove to be too difficult and too prone to human bias: a database system can be fine-tuned to perform well on a given dataset and query load, in which case measuring performance on an artificial problem is a lot less informative for deciding what infrastructure to use than measuring performance on a natural problem. A promising compromise could be to design benchmarks that are derived from realistic workflows, but preferring among all possible workflows those that measure many different technical aspects and functionalities of the tested systems. The problem is that realistic workflows typically comprise extremely simple queries. Such queries could be used to measure robustness and reactivity on large data volumes, but are not very informative about the ability to optimize complex queries.

The bio-medical domain is one exception where complex queries occur naturally. Within this domain, the Open PHACTS project has put together the datasets and workflows that aim to answer scientific competency questions that were collected to represent standard use cases for drug discovery. We have used these workflows to derive the queryset of our benchmark, a queryset that is simultaneously complex and frequently used and independently motivated.

2 The Open PHACTS platform and the ‘20 questions’ approach

The Open Pharmacological Concepts Triple Store (Open PHACTS) Discovery Platform is an initiative to integrate publicly available data relevant for both academia and the pharmaceutical industry which provides an easy interface that allows researchers to consult the database without being confronted with the complexity of defining efficient Linked Data queries. For the end-user, the platform offers a set of services which are accessible via a RESTful interface. The choice of the services is based on consulting the domain experts among the Open PHACTS project consortium on which questions are most relevant to them when doing their daily research tasks. Through this process, twenty key questions were identified [1] which combine four important pharmacological concepts: compound, target, pathway and disease. Compounds are usually small molecules which can influence targets by activating or inhibiting them (bioactivity). These targets (often proteins) are important for many functions of organisms and are often part of cellular pathways by interacting with other entities (both molecules and targets). Errors in the function of the targets can lead to diseases, and the aim of a drug discovery process is usually to find compounds which can restore the correct function of the targets.

The Open PHACTS Discovery Platform provides an interpretation of these questions as *workflows* that are authored using visual tools. Workflows retrieve data via API calls. These API calls correspond to SPARQL query templates which are instantiated by the parameters of the API call [8]. The platform executes the resulting instantiated queries at an endpoint that serves relevant data [5, 3]

Dataset	# triples	# subjects	# predicates	# objects
Uniprot	1,131,186,434	235,053,262	122	322,660,114
Gene Ontology	882,958,562	144,881,590	15	140,689,015
ChEMBL	445,732,880	54,923,033	146	118,629,007
OPS Chemical Registry	241,986,722	38,555,884	18	89,882,844
DisGeNET	17,791,631	1,367,616	77	4,891,477
OPS Identity Mappings	14,431,716	5,254,745	71	10,874,931
WikiPathways	11,781,627	871,000	110	1,467,010
DrugBank	5,478,852	330,274	104	1,917,893
ConceptWiki	4,331,760	3,024,393	4	4,319,478
ChEBI	1,012,056	113,446	22	651,682
Total	2,756,732,240	484,375,243	689	695,983,491

Table 1. Dataset statistics

Table 1 lists the datasets needed to execute these workflows and their size. Each of the sources adds a different perspective of data which is needed to answer the questions:

- *UniProt* collects sequence and functional data of proteins, providing commonly used identifiers of proteins through their accession codes.
- *ChEMBL* provides bioactivity data, which is of high importance in many of the questions, where literature is curated to collect activity of molecules against targets (often proteins).
- *DrugBank* provides information on drug molecules, such as the approval status for clinical trials.
- *DisGeNET* associates genes and diseases.
- *WikiPathways* is a collection of cellular pathways which can be edited by the scientific community.
- Two ontologies, the *Gene Ontology* for proteins, and *ChEBI* for compounds provide additional annotations of the respective entities.

Some of the datasets specifically focus on mapping entities from different data sources:

- The OPS Chemical Registry standardizes molecules from the different data sources in Open PHACTS, to provide a single identifier if the structures are identical.
- Similarly, ConceptWiki collects labels for entities of the different datasets, allowing text searches in the Open PHACTS Discovery Platform.
- The Open PHACTS Identity Mapping Service (IMS) is a collection of all different linksets used within the system to match the identifiers of the different data sources.

Query	Question expressed in natural language
Q1	Give me all oxidoreductase inhibitors active <100 nM in human and mouse data.
Q3	Given a target find me all actives against that target, and find and/or predict the polypharmacology of actives.
Q6	For a specific target family, retrieve all compounds in specific assay.
Q7	For a target, give me all active compounds with the relevant assay data.
Q8	Identify all known protein-protein interaction inhibitors.
Q9	For a given compound, give me the interaction profile with targets.
Q15	Which chemical series have been shown to be active against target X?
Q15b	Which new targets have been associated with disease Y?
Q16	Targets in Parkinson’s disease or Alzheimer’s disease are activated by which compounds?
Q18	For pathway X, find compounds that agonize targets assayed in only functional assays with potency < 1 μ M.
Q19	For the targets in a given pathway, retrieve the compounds that are active with more than one target.

Table 2. Multi-domain drug-discovery questions expressed in natural language

3 Queries

3.1 Deriving Queries from Workflows

Most of the multi-domain questions described in Section 2 have been answered by developing visual scientific workflows [3] that consecutively request relevant data using the Open PHACTS APIs [5] and then fuse results from different datasets into a unified answer. Besides from the fact that workflows provide a user friendly way for composing different API calls, the workflow engine can also provide more complex data manipulation across different datasets and therefore provide an expressive mechanism to answer complex questions.

Fortunately, most of the questions require only basic data processing and as a result the developed workflows use operations that can be simulated by traditional relational algebra operators. Therefore, the majority of the questions can be also expressed as a single (typically more complex) SPARQL query that joins multiple datasets together. However, not every question can be translated into a SPARQL query for reasons that will become apparent later in this section.

Our benchmark proposes a set of SPARQL queries that can express the drug-discovery questions enumerated in Table 2. Notice that some questions are missing from Table 2 and the reason is three-fold. First, there are questions that do not have a corresponding workflow mainly because they could not be answered from the original datasets. These are the target of currently ongoing work on including new data on patent and pathway interactions), that are needed to answer most of the missing questions. Second, there are workflows that can answer multiple questions simultaneously. For example, the workflow that answers questions Q7 answers also question Q17 to some extent. Lastly, some questions require information that is produced dynamically and not from

Query	Number of		Complexity	SPARQL Features	Resultset size
	datasets	patterns			
Q1	1		8	U,F	331,600
Q3	3		15	V,B,D	6,628
Q6	4		11	Opt	3,148,566
Q7	4		16	B	2,589
Q8	2		9	F	21,881
Q9	3		12	V,B,F	252
Q15	3		12	V,U,G	242
Q15b	1		6		164
Q16	3		11	V,F	6,386,715
Q18	3		15	B,F,Opt	18,298
Q19	4		16	F,G,H	5,660

Table 3. Queryset characteristics. The SPARQL features are encoded as follows: F: filter, V: values, B: bind, D: distinct, Opt: optional, G: group by, H: having, Ord: order, L: limit, U: union.

a materialized dataset. For example, most of the API calls access implicitly the IMS (Identity Mapping Service) and some depend on the similarity and structural search service. For the purposes of the benchmark we have materialized the IMS to a dataset (*OPS Identity Mappings* in Table 1) that contains resource mappings related with the `skos:exactMatch`, `skos:relatedMatch` and other similar predicates. On the other hand we have excluded the workflows that make use of the similarity and structural search service.

3.2 Query Characteristics

The selected queries correspond to realistic questions posed by drug discovery scientists and thus form a base on evaluating different triple store system. Since the queries typically require multiple datasets to compute an answer it is natural to consider using the selected queryset to evaluate federated SPARQL querying systems. This does not exclude the benchmarking of triple stores that have loaded the datasets in different graphs in the same system. However, we focus our attention to federated SPARQL querying system and we discuss the suitability of the proposed query set for such benchmarking.

Typically, the main components of a federated SPARQL querying system consists of:

- the *Source Selection* phase where the system decides which data sources must be involved in the given query.
- the *Query Planning* phase where the system decomposes the initial query into a set of simpler query fragments, each of which will be send to a specific data source. In the query planning phase the federator may also decide the optimal order and type of the operations that must be performed to the intermediate results returned from the data sources.
- the *Query Execution* phase where the system executes the plans.

All those phases of a federator are crucial to the efficiency of the query processing and evaluation and therefore are subject to benchmarking.

Naturally, different characteristics of a query will stress different phases of the query processing system. The proposed queries vary on complexity, on the number of datasets that are involved and on the SPARQL features needed. Table 3 collects the different characteristics of each query.

All the queries access between one and four datasets to compute the result. This is relevant for testing if the source selection phase prunes efficiently the irrelevant datasets. However, most of the predicates used in the queries exist in only one dataset and therefore in most cases they uniquely identify the associated dataset if the source selection exploits such relations [10]. On the other hand, triples with common predicates exist in every dataset (e.g. `rdf:type`) but will not be joinable with any other dataset except the one that will be found.

Another characteristic of the proposed queries is that they naturally need a large number of triple patterns in order to retrieve the required information. This is commonly encountered in SPARQL queries in contrast to SQL queries. Traditional join optimization techniques derived from databases cannot cope efficiently with a large number of joined relations. Therefore, the large number of triple pattern will challenge the join optimization phase of a federated system. Triple patterns are also joined in multiple ways most commonly as stars and chain of stars. Apart from the number of joined triple patterns, another factor of optimization is the optimization of other SPARQL operators such as left outer join, union and grouping and ordering. Typically, query planners do not consider reordering other operators than non inner-join, despite the fact that can produce more efficient plans. This inability often leads to query plans that force these operators to be executed on the side of the federator than on the data stores often needing larger results sets to be transferred over the network. The proposed queries use some of those operators, presented in Table 3.

There are cases though where the transfer of large result sets over the network is inevitable and there is no valid query plan that can avoid that due to the form of the query. In that case, the handling of the large result sets by the execution engine of the federator may vastly differ among systems. In the proposed query set there are queries that need to produce large result sets challenging the implementation and execution techniques used by the execution engine.

Also relevant to evaluating execution engines is their behaviour when a large number of remote endpoints must be accessed. In our queryset, between one and four (out of a total of ten) endpoints must be accessed. This is one point where the current queries do not stress query processors enough, but it should be noted that the Open PHACTS queries were authored having in mind the query processing state of the art. Further work will identify use cases (including queries pulling data from many different sources) that are relevant to the domain and challenging for query processing systems.

In order to demonstrate the queries that are derived from the workflows consider the query Q19 that is depicted in Listing 1. The specific query makes use of four datasets, including the OPS Identity Mappings. The datasets are

organized in logical graphs and the joins between the entities of different datasets are linked through the `http://ims.openphacts.org` graph. The query consists of 16 triple patterns. Notice that the triple pattern joins in various ways. For example, triple pattern in line 13 and line 14 are subject-subject joined (i.e. star), while the triple patterns in line 12 and line 13 are subject-object joined (i.e. chain). Moreover, there exist object-object joins, as in patterns in line 15 and line 21. The specific query also makes use of the grouping operator and produces an aggregate count over the tuples of the group. Moreover, it filters the tuple based on the aggregated value.

4 Workload Execution and Measurements

4.1 Workload Generation

In order to help users to perform several experiments of our benchmark, we provide an engine to generate a workload to be posed to the data store in question. The benchmark engine is based on the driver provided by the FedBench suite [12].

The engine can be configured in order to support various experiment scenarios. An experiment can be configured with the use of simple configuration files for benchmark settings (query sets, number of runs per query, execution timeout). Apart from the configuration of this benchmark, we have also included configuration files for other existing benchmarks, such as FedBench [12] and BigRDFBench [11], and for various federation engines that support the Sesame API, such as SemaGrow [2], FedX [13] and SPLENDID [4] federators. Every federation engine requires its own configuration. Also, SPLENDID and SemaGrow use additional metadata that are generated by extracting statistics directly from the actual data.

The driver can connect to the specific federation engine via the Sesame API ⁵, and all the federation engines can access the data sources via the SPARQL protocol. At each step of the experiment, all queries from the given queryset are executed one time and in subsequent runs, and then this step is repeated a desired number of times. This process allows us to distinguish between the performance of cold runs and hot runs, and therefore to exclude (if wanted) the effect of cold starts in our measurements. This distinction may be useful since in many situations the execution time of the first stage of the experiment is much larger than the following stages, usually due to caching and metadata loading.

The output of the experiment is written on a CSV file which contain information about each query. An example from the output file is the following:

```
Query;run1;run2;run3;run4;run5;run6;avg;numResults;minRes;maxRes;  
SQ1;793;148;117;114;124;118;236;1159;1159;1159;  
SQ2;707;472;383;360;447;480;475;333;333;333;
```

⁵ cf. <http://www.openrdf.com>


```

1 PREFIX dc: <http://purl.org/dc/elements/1.1/>
2 PREFIX dcterms: <http://purl.org/dc/terms/>
3 PREFIX wp: <http://vocabularies.wikipathways.org/wp#>
4 PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
5 PREFIX chembl: <http://rdf.ebi.ac.uk/terms/chembl#>
6 PREFIX cheminf: <http://semanticscience.org/resource/>
7
8 SELECT ?smiles (COUNT(DISTINCT ?chembl_target_uri) AS ?count)
9 WHERE {
10   GRAPH <http://www.wikipathways.org> {
11     ?rev dc:identifier <http://identifiers.org/wikipathways/WP382> .
12     ?rev dc:title ?title .
13     ?gene_product_internal dcterms:isPartOf ?rev .
14     ?gene_product_internal rdf:type ?type .
15     ?gene_product_internal dc:identifier ?gene_product .
16     FILTER (?type = wp:GeneProduct || ?type = wp:Protein).
17     FILTER (!REGEX(?gene_product, "/DataNode/noIdentifier")).
18   }
19
20   GRAPH <http://ims.openphacts.org/> {
21     ?item skos:relatedMatch ?gene_product.
22   }
23
24   GRAPH <http://www.ebi.ac.uk/chembl> {
25     ?targetComp chembl:targetCmptXref ?item .
26     ?target chembl:hasTargetComponent ?targetComp .
27     ?target dcterms:title ?target_name_chembl .
28     ?target chembl:organismName ?target_organism .
29     ?assay chembl:hasTarget ?target .
30     ?assay chembl:hasActivity ?act .
31     ?act chembl:hasMolecule ?compound .
32     ?act chembl:pChembl ?pChembl.
33     FILTER (?pChembl > 5).
34   }
35
36   GRAPH <http://ims.openphacts.org/> {
37     ?ocrs_compound skos:exactMatch ?compound.
38   }
39
40   GRAPH <http://ops.rsc.org> {
41     ?ocrs_compound cheminf:CHEMINF_000018 ?smiles .
42   }
43 }
44 GROUP BY ?smiles
45 HAVING COUNT(DISTINCT ?chembl_target_uri) > 1

```

Listing 1. SPARQL query that answers Q19

Dataset	Number of triples	Number of nodes
Uniprot	1,131,186,434	4
Goa	882,958,562	4
ChEMBL	445,732,880	1
OPS Chemical Registry	241,986,722	1
Disgenet	17,791,631	1
OPS Identity Mappings	14,431,716	1
WikiPathways	11,781,627	1
Drugbank	5,478,852	1
ConceptWiki	4,331,760	1
ChEBI	1,012,056	1
Total	2,756,732,240	15

Table 4. The number of 4store nodes used to serve each dataset.

In this example, we have executed a workload which executes SQ2 after SQ1 six times. For each query, we display the query execution time for each run, the average execution time for all runs, and the minimum, the maximum and the average number of results that were returned by the federation engine.

4.2 Datasource endpoints

In order to provide an easy way to redistribute in multiple platforms we have packaged all the components of the benchmark as Docker images [9]. This enables us to provide a highly configurable benchmarking environment that contains all data source endpoints, the benchmarking engine and the federation systems in separate Docker containers. As a result, the components can be either deployed in the same or in separate physical machines.

Data can be served from any public or local endpoint, as configured by the experimenter. But we also provide Docker images so that the experimenter can conveniently deploy them locally. Specifically, we have prepared a collection of Docker images that have Debian 8.5 and 4store 1.1.5 pre-loaded. Each image also executes the commands needed to download from a public location one dataset dump, carry out any necessary pre-processing (e.g., convert from RDF/XML to N-TRIPLES), and bulk-load the dump in 4store. Most of the images are single-node, but some (those serving larger datasets) are multi-node distributed 4store instances [7] deployed using Docker Swarm. Table 4 lists the number of nodes recommended for each dataset, although this can be easily re-configured by the experimenter.

5 Related Benchmarks

FedBench [12] is a popular suite for benchmarking federated SPARQL query processing systems. It is comprised from three data collections, two of them using real datasets and focused on domain-specific queries and one collection that contains synthetic data. The first, named *Cross-Domain* collection refer to

datasets of general interest and federate 6 datasets including DBpedia, Geonames and LinkedMDB; the second, called *Life-Science* considers queries that combine data from datasets from the drug domain, such as ChEBI, Drugbank and KEGG. The queries proposed are considered to be typical scenarios for combining those datasets and are selected in such a way as to measure basic query characteristics of a federation engine, but are not produced from a real workflow of the domain. However, the complexity of queries is low using mainly inner joins of triple patterns.

BigRDFBench [11] extends FedBench by introducing additional large-scale real datasets to the federation and by proposing more complex queries that make use of various SPARQL operators. The benchmark splits the queries into two collections. The *Complex* collection which contain queries of increased complexity and the *Big Data* collection which contain queries that require processing of large intermediate results. The total federation consists of 13 datasets that contain in total one billion triples. Moreover, the proposed queries are complex having on average 10 triple patterns each, involving three different datasets.

6 Conclusion

We presented work in progress towards developing a new benchmark for federated query processing systems. The benchmark engine and the queries are available as open source⁶ while all datasets are also publicly available. Unlike other popular benchmarks, our queryset is not driven by technical evaluation, but is derived from workflows established by the pharmacology community. The value of this queryset is that it is realistic but at the same time it comprises complex queries that test all features of modern query processing systems.

Our next steps will be to use the new benchmark to test state-of-the-art federated query processing systems and to analyse the results from the perspective of the discussion on the characteristics of our queries (Section 3.2). This analysis and a comparison against the results obtained over the FedBench and LargeRDFBench benchmarks will help us understand what new insights can be gained by this new benchmark and how these can drive research in federated query processing.

Furthermore, we are planning to increase the size of the data and the queryset, following and transferring the latest results obtained by the Open PHACTS Foundation on using data on patent and pathway interactions to answer the questions that were not addressed by the originally released workflows. This could address the weak point observed in our queryset, that there are no queries that require data from a large number of different endpoints; if not, the queryset will be complemented with demanding queries that answer specific research questions, besides the generic, commonly recurring questions that comprise the twenty Open PHACTS questions.

⁶ Cf. <https://github.com/semagrow/kobe>

Acknowledgements

The work described here has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 644564. For more details, please visit <https://www.big-data-europe.eu>

We wish to acknowledge the Open PHACTS Foundation, the charitable organisation responsible for the Open PHACTS Discovery Platform, without which this work would not have been possible.

References

- [1] Azzaoui, K., Jacoby, E., Senger, S., Cuadrado Rodríguez, E., Loza, M., Zdravil, B., Pinto, M., Williams, A.J., de la Torre, V., Mestres, J., Pastor, M., Taboureau, O., Rarey, M., Chichester, C., Pettifer, S., Blomberg, N., Harland, L., Williams-Jones, B., Ecker, G.F.: Scientific competency questions as the basis for semantically enriched open pharmacological space development. *Drug Discovery Today* 18(17–18), 843–852 (2013), <http://www.sciencedirect.com/science/article/pii/S1359644613001542>
- [2] Charalambidis, A., Troumpoukis, A., Konstantopoulos, S.: SemaGrow: Optimizing federated SPARQL queries. In: Proceedings of the 11th International Conference on Semantic Systems (SEMANTiCS 2015), Vienna, Austria, 16–17 September 2015 (2015)
- [3] Chichester, C., Digles, D., Siebes, R., Loizou, A., Groth, P., Harland, L.: Drug discovery FAQs: Workflows for answering multidomain drug discovery questions. *Drug Discovery Today* 20(4), 399–405 (2015), <http://www.sciencedirect.com/science/article/pii/S1359644614004437>
- [4] Görnitz, O., Staab, S.: SPLENDID: SPARQL endpoint federation exploiting VOID descriptions. In: Proceedings of the 2nd International Workshop on Consuming Linked Data (COLD 2011), Bonn, Germany, October 23, 2011. CEUR Workshop Proceedings, vol. 782 (2011)
- [5] Groth, P., Loizou, A., Gray, A.J., Goble, C., Harland, L., Pettifer, S.: API-centric linked data integration: The Open PHACTS Discovery Platform case study. *Web Semantics: Science, Services and Agents on the World Wide Web* 29, 12–18 (2014), <http://www.sciencedirect.com/science/article/pii/S1570826814000195>
- [6] Guo, Y., Pan, Z., Heflin, J.: LUBM: A benchmark for OWL knowledge base systems. *Journal of Web Semantics* 3(2-3), 158–182 (2005), <http://dx.doi.org/10.1016/j.websem.2005.06.005>
- [7] Harris, S., Lamb, N., Shadbolt, N.: 4store: The design and implementation of a clustered RDF store. In: 5th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS2009), held at the 8th International Semantic Web Conference (ISWC 2009), Washington, DC USA, 25–29 Oct 2009. CEUR Workshop Proceedings, vol. 517 (2009)
- [8] Loizou, A., Angles, R., Groth, P.: On the formulation of performant SPARQL queries. *Web Semantics: Science, Services and Agents on the World Wide Web* 31, 1–26 (Mar 2015)

- [9] Merkel, D.: Docker: Lightweight Linux containers for consistent development and deployment. *Linux Journal* 2014(239) (2014), <http://dl.acm.org/citation.cfm?id=2600239.2600241>
- [10] Ozkan, E.C., Saleem, M., Dogdu, E., Ngonga Ngomo, A.C.: UPSP: unique predicate-based source selection for SPARQL endpoint federation. In: Demidova, E., Dietze, S., Szymanski, J., Breslin, J.G. (eds.) *Proceedings of the 3rd International Workshop on Dataset Profiling and Federated Search for Linked Data (PROFILES 2016)* co-located with the 13th ESWC 2016 Conference, Anissaras, Greece, May 30, 2016. *CEUR Workshop Proceedings*, vol. 1597. CEUR-WS.org (2016), http://ceur-ws.org/Vol-1597/PROFILES2016_paper4.pdf
- [11] Saleem, M., Hasnain, A., Ngonga Ngomo, A.C.: BigRDFBench: A billion triples benchmark for SPARQL endpoint federation <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.667.3600>
- [12] Schmidt, M., Görlitz, O., Haase, P., Ladwig, G., Schwarte, A., Tran, T.: FedBench: A benchmark suite for federated semantic data query processing. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) *Proceedings of the 10th International Semantic Web Conference (ISWC 2011)*, Bonn, Germany, October 23-27, 2011, Part I. pp. 585–600. Springer, Berlin/Heidelberg (2011), http://dx.doi.org/10.1007/978-3-642-25073-6_37
- [13] Schwarte, A., Haase, P., Hose, K., Schenkel, R., Schmidt, M.: FedX: A federation layer for distributed query processing on Linked Open Data. In: *Proceedings of the 8th Extended Semantic Web Conference (ESWC 2011)*, Heraklion, Crete, Greece, May 29 – June 2, 2011. *Lecture Notes in Computer Science*, vol. 6644, pp. 481–486. Springer (2011)