Efficient Source Selection and Benchmarking for SPARQL Endpoint Query Federation

Cover Page · July 2017

CITATIONS
0

READS
21

1 author:

Muhammad Saleem
University of Leipzig
46 PUBLICATIONS **291** CITATIONS

SEE PROFILE

Efficient Source Selection and Benchmarking for SPARQL Endpoint Query Federation

Muhammad Saleem

Universität Leipzig, IFI/AKSW, PO 100920, D-04009 Leipzig

1 Motivation and Objectives

Due to the decentralised architecture of the Web of Data, several datasets contain complementary data. Running complex queries on this compendium thus often requires accessing data from different data sources within one query. The abundance of datasets and the need for running complex query has thus motivated a considerable body of work on SPARQL query federation systems, the dedicated means to access data distributed over the Web of Data. This thesis addresses two key areas of federated SPARQL query processing: (1) efficient source selection, and (2) comprehensive SPARQL benchmarks to test and ranked federated SPARQL engines as well as triple stores.

Efficient Source Selection: Efficient source selection is one of the most important optimization steps in federated SPARQL query processing. To ensure that a recall of 100% is achieved, most SPARQL query federation approaches (e.g, SPLENDID [1], DAW [4], FedX [7] etc.) perform *triple pattern-wise source selection* (TPWSS). The goal of the TPWSS is to identify the set of relevant (also called capable) sources against individual triple patterns of a query. However, it is possible that a relevant source does not *contribute* to the final result set of the complete query. This is because the results from a particular data source can be excluded after performing *joins* with the results of other triple patterns contained in the same query. An overestimation of such sources increases the network traffic, result in irrelevant intermediate results, and can significantly affect the overall query processing time.

Previous works have focused on generating optimized query execution plans for fast result retrieval. However, devising join-aware source selection approaches has not received much attention. Similarly, only little attention has been paid to the effect of duplicated data on federated querying. This thesis presents HiBISCuS [5] and TBSS[1], novel, join-aware, hypergraph-based source selection approaches, and DAW, a duplicate-aware source selection approach to federated querying over the Web of Data. Each of these approaches can be combined directly with existing SPARQL query federation engines to achieve the same recall while querying fewer data sources. We integrate the three (HiBISCuS, DAW, and TBSS) source selections approaches with CostFed to form a complete SPARQL query federation suit named QUETSAL[2]. Furthermore, we present TopFed [6], a Cancer Genome

[1] Part of CostFed federation engine: `https://github.com/AKSW/costfed`

[2] QEUTSAL federation suit: `http://aksw.org/Projects/QUETSAL.html`

Atlas (TCGA[3]) tailored federated query processing engine that exploits the data distribution to perform intelligent source selection while querying over large TCGA SPARQL endpoints. Finally, we address the issue of rights managements and privacy while accessing sensitive resources. To this end, we present SAFE : a global source selection approach that enables decentralised, policy-aware access to sensitive clinical information represented as distributed RDF Data Cubes.
Comprehensive SPARQL Benchmarks: Benchmarking is indispensable when aiming to assess technologies with respect to their suitability for given tasks. While several benchmarks have been developed to evaluate federated SPARQL engines and triple stores, they mostly provide a one-fits-all solution to the benchmarking problem. This approach to benchmarking is however unsuitable to evaluate the performance of a triple store for a given application with particular requirements. The fitness of current SPARQL query federation approaches for real applications is difficult to evaluate with current benchmarks as current benchmarks are either synthetic or too small in size and complexity and mostly focused on a single performance criterion, i.e., the overall query runtime. Thus, they cannot provide a fine-grained evaluation of the systems. We address these drawbacks by presenting FEASIBLE [3], an automatic approach for the generation of benchmarks out of real query logs and LargeRDFBench [2], a billion-triple benchmark for SPARQL query federation which encompasses real data as well as real queries pertaining to real bio-medical use cases.

2 Approach

HiBISCus is a novel join-aware labelled-hypergraph-based approach for efficient source selection in SPARQL endpoints federation. The SPARQL query is first represented as labelled-hypergraph where each subject, predicate, and object of the triple patterns is a node. The hyperedge connects the subject node with the predicate and object nodes. The source pruning algorithm operates on each join node and makes use of URI authorities[4] to only select those sources that contribute the final result set of any given query.

While HiBISCuS can significantly remove irrelevant sources, it fails to prune those sources which share the same URI authority. For example, all the `Bio2RDF` sources contains the same URI authority. We address the drawback of HiBISCuS by proposing TBSS, a novel *Trie-Based Source Selection approach*. By moving away from authorities, TBSS is flexible enough to distinguish between URIs from different datasets that come from the same namespace.

DAW is the first, to the best of our knowledge, duplicate-aware federated queryin approach over the Web of Data. DAW combines min-wise independent permutations with selectivity values to estimate the number of duplicate-free results. This estimation is used to first rank triple pattern-wise sources, based on their contribution, and to skip sources that contribute with little or no new results. DAW can be directly combined with existing index-assisted federated

[3] Linked TCGA: `http://tcga.deri.ie/`
[4] URI authority: `https://tools.ietf.org/html/rfc3986#section-3.2`

query processing systems, in order to improve the query execution. We combined HiBISCuS, TBSS, and DAW with CostFed into QUETSAL, a complete SPARQL query federation suite.

SAFE is the first, to the best of our knowledge, query federation engine that enables policy-based access to sensitive statistical datasets represented as RDF Data Cubes. The work is motivated in particular by the needs of three clinical organizations who wish to develop a platform for collaboratively analyzing clinical data that spans multiple clinical sites. Clinical data – even in aggregated form – is of a highly sensitive nature, and thus federated querying methods must take access policies into account. SAFE is developed as an extension on top of the FedX federation engine to support two main features: (1) optimization tailored for federating querying of RDF Data Cubes; and (2) source-selection on the level of named graphs that allows for integration with an existing access-control layer.

TopFed is a Linked TCGA tailored federated SPARQL query processing engine. It makes use of the intelligent data distribution combined with efficient source selection. The TopFed source selection algorithm considers the metadata about the data distribution with the type of the joins among query triples patterns. it is the first approach, to the best of our knowledge, the takes the data distribution into account during the source selection.

LargeRDFBench is the first, to the best of our knowledge, billion-triple benchmark for federated SPARQL query engines based on real data and real queries. It comprises three different types of queries (namely simple, complex, and large data) of varying complexities in terms of the number of triple patterns, the result set sizes, the join and triple pattern selectivities, the join vertex degree, the use of different SPARQL constructs etc.

Finally FEASIBLE is a customized benchmark generation framework, able to generate benchmarks from a set of queries (in particular from query logs). FEASIBLE assumes that it is given a set of queries well as the number of queries (e.g., 25) to be included into the benchmark as input. The approach first represent the input queries as features vectors and plot them in a multi-dimensional space. Then, the approach generate N number of clusters and select a single most representative query from each cluster. The goal is computes a sample of the selected subset that reflects the distribution of the queries in the input set of queries.

3 Major Results

Overall, our source selection results suggest that the join-aware TPWSS (as implemented by HiBISCuS, TBSS, SAFE, and TopFed) is the superior paradigm when performing source selection for SPARQL endpoint federation. Our benchmark evaluation clearly indicates that while current federation engines can deal with simple and complex queries, they are currently not up to the challenge of dealing with real queries that involve processing large intermediate result sets or lead to large result sets. In addition, one-fits-all solutions do not work when

benchmarking towards a given use case and benchmarks must look at all types of SPARQL constructs. In the following, we provide more concrete results.

HiBISCuS overall relevant sources overestimation is 11.8% (4 times less than FedX (the fastest engine at the time of writing this thesis) and SPLENDID) on FedBench. TBSS is the first source selection approach to remain under 5% overall relevant sources overestimation. DAW has successfully improved the query execution time of the existing federation engines up to 16%. QUETSAL's and CostFed's result show that they have reduced the number of sources selected (without losing recall), the source selection time as well as the overall query runtime up to 121 times comparing to state-of-the-art federation engines. TopFed overall query execution time is half to that of FedX on TCGA queries and endpoints. SAFE results shows that it has significantly outperformed FedX (3 times on average) in all queries in the context of the presented use-cases. LargeRDFBench evaluation shows that the ranking of federation engines based on benchmarks (i.e., FedBench) with simple queries differs significantly from their ranking on more complex queries. FEASIBLE's compared for the first time, to the best of our knowledge, existing triple stores using all the four – SELECT, CONSTRUCT, DESCRIBE, ASK – forms of the SPARQL and its composite error is 54.9% smaller than DBpedia SPARQL benchmark.

4 Evaluation Methods

We tried our best to use standard benchmarks and performance metrics during the evaluation. To this end, HiBISCuS, TBSS, and QUETSAL were evaluated using FedBench, a well-known federation benchmark. However, this benchmark is not able to full-fill the purpose of DAW, SAFE, and TopFed. To this end, a use-case specific benchmarks were designed. Each of the benchmark dataset were loaded into a virtuoso SPARQL endpoints and was installed on a separate machine. Each of the query were run at 5 times and the average (across 5 run) results were presented. We also used standard significance test (ie., wilcoxon signed rank test) where necessary. To test the federation engines, we used the following metrics: (1) the total number of triple pattern-wise (TPW) sources selected during the source selection, (2) the total number of SPARQL ASK requests submitted to perform (1), (3) the completeness (recall) and correctness (precision) of the query result set retrieved, (4) the average source selection time, (5) the average query execution time, and (6) the number of endpoint requests to test the federation engines. In addition, we also show the results of the data sources index/data summaries generation time and index compression ratio (i.e., index to dataset ratio). To test the triple stores, we used the (a) number of query mixes per hour (QMpH) and (b) number of queries per seconds (QpS).

5 Significance, Open Issues, and Future Directions

The tremendous growth in the LOD opened many research challenges. One of the most important challenge to efficiently query these large compendium of

distributed and linked datasets. The runtime optimization of federated SPARQL query engines is of central importance to ensure the usability of the Web of Data in real-world applications. The efficient selection of sources (SPARQL endpoints in our case) as well as the generation of optimized query plans belong to the most important optimization steps in this respect. Similarly, more comprehensive SPARQL benchmarks are important to position the systems for the problem at hand, pinpoint limitations, and improve current federation systems as well as to develop better systems.

As future vision of federated SPARQL queries, running federated queries over large datasets (e.g., Linked TCGA) already result in millions of results with queries runtimes in hours. In such cases, many users are already not be interested in the complete results. Rather, a subset of results in a reasonable amount of query runtime is more important for the query executor. Existing SPARQL query federation engines focused on the problem of generating optimized query execution plans. None, to the best of our knowledge, has taken in to account the time efficient partial results retrieval which might be important for a particular use case. Similarly, Top-K results retrieval and query personalization based on user profile and location is interesting research direction in federated SPARQL query processing. Optimization based on query tuning (using queries history) has a great potential to improve state-of-the-art work. Collecting provenance information (e.g., how many results are contributed by each data source) and estimating the query runtime before execution might be useful information for the end users. Data distribution-aware federated, non to the best of our knowledge, query engines have the great potential to outperform state-of-the-art engines. Data de-duplication at runtime from federated data sources is interested topic that can be investigated. Finally, caching intermediate results and using them during the query planning is important research direction, yet to be investigated in federated SPARQL querying.

References

1. O. Görlitz and S. Staab. SPLENDID: Sparql endpoint federation exploiting void descriptions. In *COLD*, 2011.
2. M. Saleem, A. Hasnain, and A.-C. N. Ngomo. Largerdfbench: a billion triples benchmark for sparql endpoint federation. *Under minor revision JWS*, 2017.
3. M. Saleem, Q. Mehmood, and A.-C. Ngonga Ngomo. FEASIBLE: A featured-based SPARQL benchmark generation framework. In *ISWC*, 2015. (to appear).
4. M. Saleem, A.-C. Ngonga, J. Xavier Parreira, H. Deus, and M. Hauswirth. DAW: Duplicate-aware federated query processing over the web of data. In *ISWC*, 2013.
5. M. Saleem and A.-C. Ngonga Ngomo. HiBISCuS: Hypergraph-based source selection for sparql endpoint federation. In *ESWC*, 2014.
6. M. Saleem, S. Sampath, A.-C. N. Ngomo, A. Iqbal, J. S. Almeida, S. Decker, and H. Deus. TopFed: TCGA tailored federated query processing and linking to LOD. *JBMS*, 2014.
7. A. Schwarte, P. Haase, K. Hose, R. Schenkel, and M. Schmidt. FedX: Optimization techniques for federated query processing on linked data. In *ISWC*, 2011.