Answering SPARQL Queries using Views

Gabriela Montoya

LINA– Université de Nantes, France
gabriela.montoya@univ-nantes.fr

Abstract. Views are used to optimize queries and to integrate data in Databases. The data integration schema is composed of terms, they are used to pose queries to the integration system, and to describe sources data. When the data descriptions are SPARQL conjunctive queries, their number and the complexity of answering queries using them may be very high. In order to keep query answering cost low, and increase the usability of views to answer queries, we make the assumption of the simplest form of replication among services, and use triple pattern views as the Linked Data Fragments and Col-graph approaches have recently done. We propose two approaches: the SemLAV approach to integrate heterogeneous data using Local-as-View mappings to describe Linked Data and Deep Web sources, and the FEDRA approach to process queries against federations of SPARQL endpoints with replicated fragments.

1 Scene setting

The Linked Data Cloud includes more than one thousand datasets.[1] However, a larger number of sources is available in the Web [14], and because of their heterogeneity their usage by Semantic Web technologies and in combination with Linked Data is limited. For the Web context, Local-as-View (LAV) data integration paradigm is well suited [1], nevertheless the traditional techniques used to produce query answers, *query rewritings*, may be too expensive in the context of SPARQL queries and numerous sources [21].

Many of the Linked Data datasets provide SPARQL endpoints.[2] These endpoints allow users to explore datasets, and even use federated SPARQL query engines to answer queries using linked data across several datasets, i.e., *federated queries*. Unfortunately as the number and complexity of queries posed to the endpoints increases, their availability decreases [27]. An alternative to improve data availability is to give data consumers a more active role, and use their resources to replicate data, and consequently increase the data availability [18]. As Linked Data consumers are autonomous participants, the replication cannot closely follow the techniques used in distributed databases, and new strategies to select and localize sources are needed.

Data replication has been already used by data consumers that, not being able to rely on SPARQL endpoints with availability limitations, have set up their

[1] According to the 2014 report about the state of Linked Data Cloud available at `http://linkeddatacatalog.dws.informatik.uni-mannheim.de/state/`

[2] 615 SPARQL endpoints are registered at `http://datahub.io/` (July, 2015)

own SPARQL endpoints. These endpoints are mostly mirrors, and their use is restricted to local users. This replication strategy is limited by the user resources, and a user with a modest amount of resources may not set up endpoints for all datasets relevant for all the queries she may want to answer.

Linked Data Fragments (LDF) [27] has been proposed to exploit clients resources to relieve servers resources and improve availability, and Col-graph [15] has been proposed to use clients resources to replicate datasets fragments and improve data quality. We think Linked Data users may also benefit from replicating datasets fragments to execute federated queries; but in order to achieve this, they need to put their own replicated fragments at other users disposal.

Solving the availability limitations of Linked Data using data consumer resources, to replicate data and to make these data available to others, may lead to query processing performance concerns. For example, given popular datasets with many data consumers willing to set up endpoints with datasets subsets, how are these endpoints going to be used to execute a federated query? A very simple solution may be to declare all of these endpoints as part of the federation used by a federated query engine, but this simple solution may incur in very expensive execution times. For example, in [24], we give an example where two replicas of the relevant fragments leads to increase the execution time in two orders of magnitude for federated query engines FedX [26] and ANAPSID [2]. In order to properly exploit the benefits of replicated datasets fragments, we propose a source selection strategy that is aware of fragment replication able to enhance federated query processing engines. This idea of sharing replicated fragments among Linked Data consumers is new, and so there are no existing techniques that can perform a source selection aware of data replication. Even if techniques to detect data overlapping based on data summaries exists [25], applying them to scenarios with data replication will incur in expensive computations that are unnecessary in a replication scenario, will have accuracy limitations that may be overcome by tailored approaches, and will make greedy decisions that choose the best sources for a triple pattern without considering that the triple pattern would be executed in combination with other triple patterns.

In this thesis, our aim is to process SPARQL queries using views in two contexts. First, in the context of data integration, answer queries using views that describe sources as conjunctive SPARQL queries. And second, in the context of SPARQL federations of endpoints, answer queries using queries that describe the data fragments that have been replicated across SPARQL endpoints.

The research objective is to improve the query execution performance. This improvement may be measured in terms of amount of transferred data or answer throughput.

In particular, we want to answer the following research questions: *I*) When SPARQL is used as language for LAV global schema, can view loading outperform traditional LAV query rewriting techniques in query answering? *II*) How can the knowledge about the fragment replication be used to safely reduce the number of selected sources by federated query engines while keeping the same answer completeness? *III*) Does considering BGPs instead of just triple patterns at source selection time produce better source selections that lead to less intermediate results?

The main challenges of this work are: *i*) Choosing the order in which views are loaded into the local graph that represents a partial instance of the global schema. *ii*) Keeping the size of transferred data low. *iii*) Keeping low the complexity of the containment computation among replicated fragments.

2 Proposed Approach

We propose two approaches to solve our problem: the SemLAV approach, and the FEDRA approach. SemLAV integrates heterogeneous data from Linked Data and Deep Web, following the Local-as-View (LAV) paradigm [1] to describe the setup, it uses a graph instance that integrates views instances, built during query execution, to answer users queries. FEDRA selects the sources to be contacted to execute each subquery in order to produce the query answer.

2.1 Formal Definition and Properties of the Approach

The SemLAV approach is based on the LAV paradigm, wrappers are used to transform data from Deep Web sources into the global schema. Given a SPARQL query Q on a set M of LAV views, SemLAV selects relevant views for Q and ranks them in order to maximize query results. Next, data collected from selected views are included into a partial instance of the global schema, where Q can be executed whenever new data is included. The SemLAV approach has the following properties: *1*) If the global schema instance includes the data from all the relevant views ranked by SemLAV, the query execution produces the same answer as traditional rewriting-based query processing approaches. *2*) The effectiveness of SemLAV is proportional to the number of covered rewritings. *3*) The view loading and query execution time linearly depends on the number of views loaded in the global schema instance. *4*) Answers may be produced incrementally if all the relevant views do not fit in memory.

The FEDRA approach is based on query containment and equivalence to prune relevant sources to retrieve data from, and a reduction to the set covering problem to produce as few subqueries as possible, sending to the endpoints subqueries composed by as many triple patterns as possible, and hopefully reducing the number of intermediate results to transfer from endpoints to the query engine. Given the set of the endpoints descriptions that constitute the federation, and a query, find a function D that for each query triple pattern returns the set of endpoints that need to be contacted in order to obtain a query answer as complete as possible while intermediate results are reduced. The FEDRA approach has the following properties: *1*) FEDRA source selection used in combination with query engines, produces at least as many answers as the query engines alone. *2*) FEDRA selects as few sources as possible for each triple pattern. *3*) FEDRA source selection aims to reduce the number of transferred tuples from endpoints to query engines.

2.2 Relationship between your approach and state-of-art approaches

Several approaches have been proposed for querying the Web of Data [2, 6, 12, 13, 19]. All these approaches assume that queries are expressed in terms of RDF vocabularies used to describe the data in the RDF sources; thus, their main challenge is to effectively select the sources, and efficiently execute the queries on the data retrieved from the selected sources. In contrast, SemLAV attempts to integrate data sources, and relies on a global schema to describe data sources and to provide a unified interface to the users. As a consequence, in addition to collecting and processing data transferred from the selected sources, SemLAV decides which of these sources need to be contacted first, to quickly answer the query. The Local-As-View paradigm for data integration allows to easily integrate new data sources; further, data sources that publish entities of several concepts in the global schema, can be naturally defined as LAV views. Query rewriters allow to answer queries against sources described using the LAV paradigm, some examples are: MCD-SAT [5], GQR [17], Bucket Algorithm [20], and MiniCon [11].

Recently, DAW, a source selection duplication aware approach for Linked Data has been recently proposed [25]. DAW relies on indexes with data summaries, that allow to measure the overlapping among sources and properly reduce the number of selected sources. Nevertheless, keeping up-to-date endpoints data summaries may be very expensive, and they cannot guarantee the correct overlapping detection. Moreover, DAW is a triple pattern wise approach, and data localities are not used to choose sources that may reduce the number of transferred tuples from endpoints to the query engines. FEDRA endpoints description are perfect data summaries, with no accuracy issues with a size independent of the data size, and that requires less updates that data summaries. FEDRA does take into account the data localities to produce as few subqueries as possible and reduce the size of intermediate results.

Linked Data Fragments (LDF) [27], a new publishing strategy that reduces the data publisher resources usage, and proposes a more active role for the data consumers. Like LDF, our approach seeks to use data consumers computational resources, and hopefully it may contribute to improve data availability of data publishers. Contrarily to LDF, FEDRA approach seeks to share the query execution among data consumers, and does not intend that each client communicates with the data publisher, and creates a cache of her queries datasets.

FedX [26] and ANAPSID [2] are state-of-art query engines for federations of SPARQL endpoints. Both, FedX and ANAPSID, send queries for triple patterns that should be evaluated in more than one endpoint, individually to the endpoint; and group into exclusive groups (FedX) or star-shaped groups (ANAPSID) the triple patterns that can be solely evaluated using one endpoint. We have extended FedX and ANAPSID query engines with FEDRA source selection strategy, and study FEDRA impact during query execution.

3 Implementation of the Proposed Approach

We use SPARQL conjunctive queries as views that describe sources contents. As in the Bucket Algorithm [20], relevant sources are selected per query subgoal, i.e., triple pattern. And the selected sources are ranked according to the number of query subgoals that are covered by their views. Loading first the views that cover more subgoals aims to produce answers as soon as possible. We expect to observe that using LAV paradigm to answer SPARQL queries allows to integrate heterogeneous data sources, and that the SemLAV approach produce more answers sooner than traditional query rewriting-based approaches. We will use variable substitution [10] to detect containment among fragments, and containment to detect equivalence of fragments [11]. Endpoints that have replicated equivalent fragments are considered as equivalent sources for that fragment. Fragment containment will be used to select the set of non-redundant relevant fragments for each triple pattern. Then, a second selection will be done at BGP level to reduce the number of subqueries to be sent to the endpoints, and in consequence, the number of transferred tuples. To further prune the selected sources, a set covering heuristic [16] will be used to determine the minimal set of endpoints that can be used to execute the subqueries. The approach implementation may be done stand-alone, having as input a query without SERVICE clauses, it produces a query with SERVICE clauses that delegate subqueries execution to remote endpoints; or existing federated query engines may be extended with FEDRA source selection strategy, to be used before the engines optimizations. We expect to observe that query engines enhanced with FEDRA incur in less intermediate results, and produce answers sooner.

The SemLAV approach has been formalized [21], and empirical tests have been performed [21, 9]. Some limitations of SemLAV implementation are that it lacks of strategies to overcome the memory limitations, and that views are loaded sequentially because the data store used did not allow parallel loading of data. The FEDRA source selection strategy for federations with replicated fragments has been proposed in [24], and some empirical tests have been performed [24]. Currently FedX and ANAPSID query engines have been extended to include the FEDRA source selection strategy. A limitation of the FedX extension is that it may increase the number of intermediate results in some cases, when one endpoint is selected for triple patterns that do not share variables.

4 Empirical Evaluation Methodology

The approaches will be used to execute queries in different setups, and they will be compared with alternative approaches.

For the SemLAV approach, the Berlin Benchmark [7] dataset generator will be used to generate a ten million triples dataset, and queries and views based on the ones proposed in [8] will be used. To stress the scalability of SemLAV and query rewriting, 476 views are used. SemLAV performance is compared to the performance of three query rewriters: GQR [17], MCD-SAT [5], and MiniCon [11]. Query execution performance will be measured in terms of answer size, query execution time, amount of memory used, and answer throughput.

Our research hypothesis is that the SemLAV approach will produce a higher answer throughput than traditional query rewriting-based approaches, for LAV data integration in the context of Linked Data and the Deep Web. Currently, the SemLAV evaluation has been performed, and results are available at [21].

For the FEDRA approach, we will study two federated query engines: FedX and ANAPSID. As baseline we will consider the federated query engines source selection strategy. We will consider real and synthetic datasets of varying sizes from 72 thousand to 10 millions triples. Query generators will be used to produce more than 10,000 queries for each dataset. And we will use each dataset to setup a federation of ten endpoints, and simulate the execution of federated queries. The federations will contain endpoints with heterogeneous replicated fragments, and a sample of 100 random queries will be taken for each federation. For each query, the number of selected sources per triple pattern, source selection time, execution time, answer completeness, and number of transferred tuples will be measured. The obtained results will be statistically analyzed using the Wilcoxon signed rank test for paired non-uniform data.

In particular, we have the following research hypotheses: *1*) The FEDRA approach will select significantly less sources than federated query engines like FedX and ANAPSID. *2*) The FEDRA approach enhances federated query like FedX and ANAPSID, and reduces the number of intermediate results during query execution. *3*) The reductions that the FEDRA approach achieves in terms of number of selected sources and intermediate results, do not reduce the number of obtained answers.

Experiments with six federations and extended version of the query engines have already been performed, and they support our research hypotheses [24].

5 Lessons Learned, Open Issues and Future Directions

Query rewriting approaches are too stressed by the large number of triple patterns in SPARQL queries and the high number of sources in the Web, these characteristics prevents query rewriters to offer a practical query evaluation strategy for Linked Data. SemLAV uses query rewriters most basic information, *buckets*, to select relevant views, and ranks sources in a way that when k views have been loaded, they cover the maximal number of rewriting that can be covered with k views. The SemLAV approach implementation can be improved by loading views in parallel, and considering memory limits.

In federations of SPARQL endpoints with replicated fragments, federated query engines need to be enhanced with a source selection approach like FEDRA that allows to select sources for each triple pattern in a way that data transferred from endpoints to query engines is reduced. The strategy used by FEDRA gets excellent results when used with ANAPSID, but results are less good when used with FedX. ANAPSID does not send subqueries with Cartesian products to the endpoints, but FedX may do it when triple patterns that do not share variables need to be executed in the same endpoint. Cartesian products may significantly increase the number of transferred tuples when compared to the evaluation of each triple pattern individually. These limitations may be overcome with a stand-alone implementation of the FEDRA approach that transforms a plain query into a query with query decomposition and source localization represented as SERVICE clauses that avoids Cartesian products. Unfortunately, such implementation has faced with query engines limited support for SERVICE clauses. In this direction, we are currently working in a version that do rewrite the query using SERVICE clauses, and use some heuristics to overcome current federated query engines limitations during execution.

Other extensions of FEDRA include the usage of cost functions, and the consideration of divergent fragments. Cost functions may be used to select the endpoints that satisfy the user criteria, e.g., in [23] public endpoint usage was reduced. Removing the assumption that all the endpoints fragments descriptions are up-to-date, endpoints may offer data with different levels of divergence with respect the current dataset, in the same direction as in [22].

Acknowledgments. This thesis is supervised by Pascal Molli and Hala Skaf-Molli. It has received contributions from Maria-Esther Vidal.

References

1. S. Abiteboul, I. Manolescu, P. Rigaux, M.-C. Rousset, and P. Senellart. *Web Data Management*. Cambridge University Press, New York, NY, USA, 2011.
2. M. Acosta, M. Vidal, T. Lampo, J. Castillo, and E. Ruckhaus. ANAPSID: An Adaptive Query Processing Engine for SPARQL Endpoints. In Aroyo et al. [4], pages 18–34.
3. G. Antoniou, M. Grobelnik, E. P. B. Simperl, B. Parsia, D. Plexousakis, P. D. Leenheer, and J. Z. Pan, editors. *The Semantic Web: Research and Applications - 8th Extended Semantic Web Conference, ESWC 2011, Heraklion, Crete, Greece, May 29-June 2, 2011, Proceedings, Part I*, volume 6643 of *Lecture Notes in Computer Science*. Springer, 2011.
4. L. Aroyo et al., editors. *ISWC 2011, Part I*, volume 7031 of *LNCS*. Springer, 2011.
5. Y. Arvelo, B. Bonet, and M.-E. Vidal. Compilation of query-rewriting problems into tractable fragments of propositional logic. In *AAAI*, pages 225–230. AAAI Press, 2006.
6. C. Basca and A. Bernstein. Avalanche: Putting the Spirit of the Web back into Semantic Web Querying. In A. Polleres and H. Chen, editors, *ISWC Posters&Demos*, volume 658 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2010.
7. C. Bizer and A. Schultz. The berlin sparql benchmark. *Int. J. Semantic Web Inf. Syst.*, 5(2):1–24, 2009.
8. R. Castillo-Espinola. *Indexing RDF data using materialized SPARQL queries*. PhD thesis, Humboldt-Universität zu Berlin, 2012.
9. P. Folz, G. Montoya, H. Skaf-Molli, P. Molli, and M.-E. Vidal. SemLAV: Querying Deep Web and Linked Open Data with SPARQL. In *ESWC: Extended Semantic Web Conference*, volume 476 of *The Semantic Web: ESWC 2014 Satellite Events*, pages 332 – 337, Anissaras/Hersonissou, Greece, May 2014.

10. C. Gutierrez, C. A. Hurtado, A. O. Mendelzon, and J. Pérez. Foundations of Semantic Web databases. *J. Comput. Syst. Sci.*, 77(3):520–541, 2011.
11. A. Y. Halevy. Answering queries using views: A survey. *VLDB J.*, 10(4):270–294, 2001.
12. A. Harth, K. Hose, M. Karnstedt, A. Polleres, K.-U. Sattler, and J. Umbrich. Data summaries for on-demand queries over linked data. In M. Rappa, P. Jones, J. Freire, and S. Chakrabarti, editors, *WWW*, pages 411–420. ACM, 2010.
13. O. Hartig. Zero-knowledge query planning for an iterator implementation of link traversal based query execution. In Antoniou et al. [3], pages 154–169.
14. B. He, M. Patel, Z. Zhang, and K. C.-C. Chang. Accessing the Deep Web. *Commun. ACM*, 50(5):94–101, 2007.
15. L. D. Ibáñez, H. Skaf-Molli, P. Molli, and O. Corby. Col-Graph: Towards Writable and Scalable Linked Open Data. In P. Mika et al., editors, *ISWC 2014, Part I*, volume 8796 of *LNCS*, pages 325–340. Springer, 2014.
16. D. S. Johnson. Approximation Algorithms for Combinatorial Problems. In A. V. Aho et al., editors, *ACM Symposium on Theory of Computing*, pages 38–49. ACM, 1973.
17. G. Konstantinidis and J. L. Ambite. Scalable query rewriting: a graph-based approach. In T. K. Sellis, R. J. Miller, A. Kementsietsidis, and Y. Velegrakis, editors, *SIGMOD Conference*, pages 97–108. ACM, 2011.
18. D. Kossmann. The state of the art in distributed query processing. *ACM Computer Survey*, 32(4):422–469, 2000.
19. G. Ladwig and T. Tran. Sihjoin: Querying remote and local linked data. In Antoniou et al. [3], pages 139–153.
20. A. Y. Levy, A. Rajaraman, and J. J. Ordille. Querying heterogeneous information sources using source descriptions. In T. M. Vijayaraman, A. P. Buchmann, C. Mohan, and N. L. Sarda, editors, *VLDB*, pages 251–262. Morgan Kaufmann, 1996.
21. G. Montoya, L. D. Ibáñez, H. Skaf-Molli, P. Molli, and M.-E. Vidal. SemLAV: Local-As-View Mediation for SPARQL Queries. *Transactions on Large-Scale Data- and Knowledge-Centered Systems XIII*, pages 33–58, 2014.
22. G. Montoya, H. Skaf-Molli, P. Molli, and M.-E. Vidal. Fedra: Query Processing for SPARQL Federations with Divergence. Technical report, Université de Nantes, May 2014.
23. G. Montoya, H. Skaf-Molli, P. Molli, and M.-E. Vidal. Efficient Query Processing for SPARQL Federations with Replicated Fragments. Jan. 2015.
24. G. Montoya, H. Skaf-Molli, P. Molli, and M.-E. Vidal. Federated SPARQL Queries Processing with Replicated Fragments. In *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference*, Bethlehem, United States, Oct. 2015.
25. M. Saleem, A.-C. N. Ngomo, J. X. Parreira, H. F. Deus, and M. Hauswirth. DAW: Duplicate-AWare Federated Query Processing over the Web of Data. In H. Alani et al., editors, *ISWC 2013, Part I*, volume 8218 of *LNCS*, pages 574–590. Springer, 2013.
26. A. Schwarte, P. Haase, K. Hose, R. Schenkel, and M. Schmidt. FedX: Optimization Techniques for Federated Query Processing on Linked Data. In Aroyo et al. [4], pages 601–616.
27. R. Verborgh, M. V. Sande, P. Colpaert, S. Coppens, E. Mannens, and R. V. de Walle. Web-Scale Querying through Linked Data Fragments. In C. Bizer et al., editors, *WWW Workshop on LDOW 2014*, volume 1184 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2014.