Optimizing Traversal Queries of Sensor Data Using a Rule-Based Reachability Approach

Bryan-Elliott Tam[1,*], Ruben Taelman[1], Julián Rojas Meléndez[1] and Pieter Colpaert[1]

[1]IDLab, Department of Electronics and Information Systems, Ghent University – imec

Abstract

Link Traversal queries face challenges in completeness and long execution time due to the size of the web. Reachability criteria define completeness by restricting the links followed by engines. However, the number of links to dereference remains the bottleneck of the approach. Web environments often have structures exploitable by query engines to prune irrelevant sources. Current criteria rely on using information from the query definition and predefined predicate. However, it is difficult to use them to traverse environments where logical expressions indicate the location of resources. We propose to use a rule-based reachability criterion that captures logical statements expressed in hypermedia descriptions within linked data documents to prune irrelevant sources. In this poster paper, we show how the Comunica link traversal engine is modified to take hints from a hypermedia control vocabulary, to prune irrelevant sources. Our preliminary findings show that by using this strategy, the query engine can significantly reduce the number of HTTP requests and the query execution time without sacrificing the completeness of results. Our work shows that the investigation of hypermedia controls in link pruning of traversal queries is a worthy effort for optimizing web queries of unindexed decentralized databases.

Keywords

Linked data, Link Traversal Query Processing, Fragmented database, Descentralized environments

1. Introduction

The increasing amount of available Linked Data on the Web [1] prompts the need for efficient query interfaces. During a typical query execution in a SPARQL endpoint, the endpoint takes the whole query load and delivers the results to the client. This paradigm can lead to high workloads, which are partly responsible for the historically low availability of SPARQL endpoints [2]. Researchers and practitioners have made efforts to introduce alternative Linked Data publication methods that enable client's participation in the query execution process [3]. The goal of those methods is to lower server-side workloads while keeping fast query execution to the client [4]. The TREE hypermedia specification is an effort in that direction [5, 6], that introduces the concept of domain-oriented fragmentation of large RDF datasets. For example, in the case of periodic measurements of sensor data, a fragmentation can be made on the publication date of each data entity. A fragment can be considered an RDF document published in a server. TREE aims to describes dataset fragmentation in ways that enable clients to easily fetch query-relevant subsets. The data within a fragment are bound by constraints expressed through hypermedia descriptions [7]. Each fragment contains relations to other pages, and those relations contain the constraints of the data of every reachable fragment. In this paper, we refer to those constraints as domain-specific expressions. They can be expressions such as $?t > \text{2022-01-09T00:00:00.000000} \implies$ ex:afterFirstSeptember given that $?t$ is the date of publication of sensor data and the implication pertains to the location of the data respecting the constraint. In English, the expression means "the data produced by the sensors after the first of September are stored at `ex:afterFirstSeptember`." Because of the hyperlinked nature of the documents network, clients must traverse them to find the relevant data

Posters, Demos, and Industry Tracks at ISWC 2024, November 13–15, 2024, Baltimore, USA

*Corresponding author.

bryanelliott.tam@ugent.be (B. Tam); ruben.taelman@ugent.be (R. Taelman); JulianAndres.RojasMelendez@UGent.be (J. Rojas Meléndez); pieter.colpaert@ugent.be (P. Colpaert)

https://www.rubensworks.net (R. Taelman); https://julianrojas.org (J. Rojas Meléndez); https://pietercolpaert.be (P. Colpaert)

0000-0003-3467-9755 (B. Tam); 0000-0001-5118-256X (R. Taelman); 0000-0002-6645-1264 (J. Rojas Meléndez); 0000-0001-6917-2167 (P. Colpaert)

to answer their queries. We propose to use Link Traversal Query Processing (LTQP) [8] as a query mechanism to perform those queries.

LTQP starts by dereferencing a set of user-provided URLs [8]. From these dereferenced documents, links to other documents are dereferenced recursively and inserted in an internal data store. LDQL [9] is a theoretical query language to define the traversal of LTQP queries. However, LDQL is centered around nested regular expressions, thus, is not made to express the traversal of links based on domain-specific expressions such as time relations. The subweb specifications language (SWSL) [10], allows data providers to define traversal paths concerning the information they publish. Thus, given that the query engine trusts the data publisher it can adapt its traversal to follow the paths given by the specification. Akin to LDQL, it is difficult with the SWSL to express traversal using domain-specific expressions, because its syntax is centered around the matching of triple patterns and not reasoning rules or evaluation of literals. Furthermore, SWSL does not propose a mechanism for using the query or input from the user to impact the source selection process, unlike LDQL. Given those limitations, we propose to return to the more abstract concept of reachability criteria [11], to define a mechanism of traversal centered around rules.

In this paper, we propose to use a boolean solver as the main link pruning mechanism for a reachability criterion to traverse TREE documents. The logical operators are defined by the TREE specification. [1] As a concrete use case, we consider the publication of (historical) sensor data. An example query is presented in Figure 1 along with the triples representing the link between two documents expressed using the TREE specification.

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX saref: <https://saref.etsi.org/core/>
PREFIX dahcc: <https://dahcc.idlab.ugent.be/
    Ontology/Sensors/>

SELECT * WHERE {
  ?s saref:hasTimestamp ?t;
    saref:hasValue ?result;
    saref:measurementMadeBy ?sensor.
  ?sensor dahcc:analyseStateOf ?stateOf;
    saref:measuresProperty {:property}.
  FILTER(?t="2022-01-03T10:57:54.000000"^^xsd:
      dateTime)
}
```

```
@prefix tree: <https://w3id.org/tree#> .
@prefix xsd: <http://www.w3.org/2001/
    XMLSchema#> .
@prefix ex: <https://example.be/> .
@prefix saref: <https://saref.etsi.org/
    core/> .

<> tree:relation [
    a  tree:GreaterThanOrEqualToRelation ;
    tree:node <nextNode> ;
    tree:value "2022-01-03T09:47:59"^^xsd:
        dateTime ;
    tree:path saref:hasTimestamp
] .
```

Figure 1: On the left, is a SPARQL query to get sensor measurements and information about the sensor. On the right, is the hypermedia description of the location and constraint of the next fragment located in `ex:nextNode`. The constraint describes publication times ($?t$) where $?t >=$ 2022-01-03T09:47:59.000000.

2. A Rule-Based Reachability Criterion

Most research on LTQP is centered around query execution in Linked Open Data environments. Given the pseudo-infinite number of documents on the Web, traversing over all documents is practically infeasible. To define completeness, different reachability criteria [11] were introduced to allow the discrimination of links. Recently, an alternative direction was designed where the query engine uses the structure from the data publisher to guide itself towards relevant data sources [12, 13].

We define our approach as a rule-based reachability criterion. Our approach builds upon the concept of structural assumptions [12] to exploit the structural properties of TREE annotated datasets. We therefore interpret the hypermedia descriptions of constraints in TREE fragments as boolean expressions E ($?t >=$ 2022-01-03T09:47:59.000000 in Figure 1). Upon discovery of a document, the query engine gathers the relevant triples to form the boolean expression of the constraint on the data of reachable fragments. After the parsing of the expression, the filter expression F of the SPARQL query is *pushed*

[1] https://treecg.github.io/specification/

down into the engine's source selection component. The source selection component can be formalized as a reachability criterion [2]

$$c(i) \rightarrow \{\text{true}, \text{false}\} \tag{1}$$

where when it returns true the target IRI i *must* be dereferenced. Finally, the two boolean expressions are evaluated to determine their satisfiability. It can be formalized as determining if

$$c(i) = \exists x | (F(x) \wedge E_i) = \text{true} \tag{2}$$

hold true given x is the variable targeted by E_i and i is the link towards the next fragment (`<nextNode>` from "`<> tree:node <nextNode>`" in Figure 1). A variable targetted by E is defined by an RDF object where the predicate as a value `?target` from the triple defining the fragmentation path in the form ""`?s tree:path ?target`" (`saref:hasTimestamp` in Figure 1). Upon satisfaction the IRI targeting the next fragment is added to the link queue otherwise the IRI is pruned. The process is schematized in Figure 2.

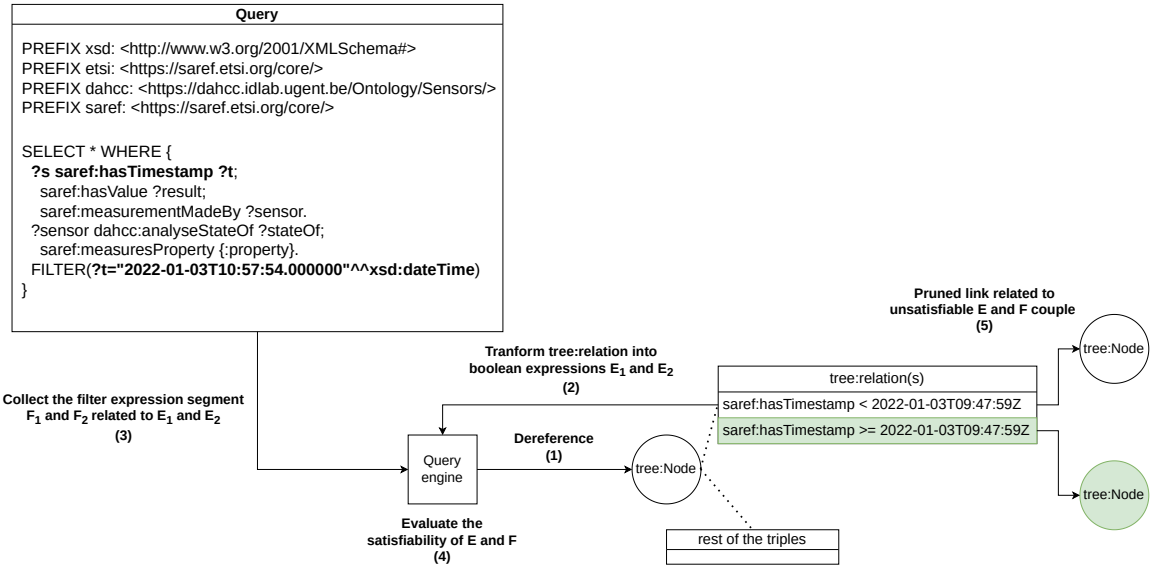

Figure 2: A schematization of our rule-based reachability criteria with a TREE document. First a TREE node is dereferenced, then the TREE relations are transformed into boolean expressions E, followed by the construction of F from the filter expression related to the path of E (the variable t related to `saref:hasTimestamp`), then the satisfiability $E \wedge F$ is determined and finally links to non-query relevant data are pruned.

2.1. Preliminary Results

We implemented our approach using the query engine Comunica [14]. For evaluation, we executed four queries similar to the one in Figure 1. [3] They were executed over the DAHCC participant 31 dataset [15] (487 MB) with a timeout of two minutes. We fragmented the dataset according to the TREE specification. We use a B-tree topology with a depth of 1 using 100 and 1000 nodes (n).

The queries were executed using two configurations. In the first configuration, we use a predicate-based reachability criterion where the engine follows each link of the fragmented dataset. [4] For the

[2] We use a simplified formalization to illustrate the source selection mechanism and to not introduce unnecessary concepts for the aim of this poster paper.

[3] The implementation, the queries and the evaluation are available at the following links: https://github.com/constraintAutomaton/comunica-feature-link-traversal/tree/feature/time-filtering-tree-sparqlee-implementation, https://github.com/TREEcg/TREE-Guided-Link-Traversal-Query-Processing-Evaluation/tree/main

[4] The query engine will always follow `ex:nextNode` from expressions following the schema "`ex:currentNode tree:relation [tree:node ex:nextNode]`" regardless of the constraints.

n	Query	Time-predicate (ms)	Time-rule (ms)	HTTP-request-rule	Res-rule
100	Q1	x	8,892	3	0
100	Q2	x	3,541	3	1
100	Q3	x	59,274	8	8,166
1000	Q1	x	1,171	3	0
1000	Q2	x	734	3	1
1000	Q3	x	39,987	51	8,166

Table 1
The predicate-based (-predicate) reachability criterion is not able to execute the queries. The rule-based (-rule) criterion performs better in term of execution time (Time) with a larger number of fragments even when performing more HTTP requests (HTTP-request). Q4 is not depicted because the instances were not able to terminate before the timeout.

second one, we use our rule-based reachability criterion approach. As shown in Table 1 no queries could be answered within the timeout by following every fragment. A possible explanation is the high number of HTTP requests performed [8] leading to non-relevant data sources. With our rule-based reachability criterion, the queries executed over the 1000 nodes fragmentation perform better than the ones with 100 nodes. The query execution time has a percentage of reduction of 86% with Q1 and 79% with Q2 compared to the fragmentation with 100 nodes. With Q3 we see that the percentage of reduction is 33%, this lowering of performance gain might be caused by the increase by a factor of 6 in HTTP requests. This raises an interesting observation because we do not observe a reduction in execution time with a reduction in HTTP requests. Previous research has proposed that inefficient query plans might be the bottleneck of some queries in structured environments [12, 16]. However, our results seem to show that the size of the internal triple store might have a bigger impact on performance than noted in previous studies. As large-scale link traversal over the web will result in the acquisition of a large number of triples, a future interesting research direction would be to find ways to remove triples that are certain to not lead to a query result from the internal triple store. The query Q4 was not able to be answered, with any setup, because the query requires a larger number of fragments than the other to be processed.

3. Conclusion

This paper reported on preliminary tests to add guided link traversal support into the Comunica querying engine using a rule-based reachability approach. A similar approach could be performed with other SPARQL query engines supporting Link Traversal Query Processing. Our preliminary results show that our rule-based reachability criterion can significantly reduce the execution time of queries aligned with hypermedia description constraints compared to predicate-based reachability opening the possibility for faster and more versatile traversal-based query execution over fragmented RDF documents. Our experiment also highlights that the size of the internal data store might have more impact on performance than noted in previous studies. In future work, we will perform more exhaustive evaluations of other types of domain-oriented fragmentation strategies such as string and geospatial evaluations, and investigate how to generalize our approach to support more expressive online reasoning for online source selection during traversal queries. Furthermore, we also showed there might still be room for optimization by researching ways for pruning useless triples from the internal triple store during the link traversal process.

References

[1] I. Ermilov, M. Martin, J. Lehmann, S. Auer, Linked open data statistics: Collection and exploitation, in: P. Klinov, D. Mouromtsev (Eds.), Knowledge Engineering and the Semantic Web, Springer, Berlin, Heidelberg, 2013, pp. 242–249.

[2] C. Buil-Aranda, A. Hogan, J. Umbrich, P.-Y. Vandenbussche, Sparql web-querying infrastructure: Ready for action?, in: Proceedings 12th ISWC, ISWC '13, Springer-Verlag, Berlin, Heidelberg, 2013, p. 277–293. URL: https://doi.org/10.1007/978-3-642-41338-4_18. doi:10.1007/978-3-642-41338-4_18.

[3] R. Verborgh, M. V. Sande, O. Hartig, J. V. Herwegen, L. D. Vocht, B. D. Meester, G. Haesendonck, P. Colpaert, Triple pattern fragments: A low-cost knowledge graph interface for the web, J. Web Semant. 37-38 (2016) 184–206.

[4] A. Azzam, C. Aebeloe, G. Montoya, I. Keles, A. Polleres, K. Hose, Wisekg: Balanced access to web knowledge graphs, in: Proceedings of the Web Conference 2021, WWW '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 1422–1434. URL: https://doi.org/10.1145/3442381.3449911. doi:10.1145/3442381.3449911.

[5] P. Colpaert, Building materializable querying interfaces with the tree hypermedia specification (2022). URL: https://treecg.github.io/paper-materializable-interfaces/.

[6] D. Van Lancker, P. Colpaert, H. Delva, B. Van de Vyvere, J. Rojas Meléndez, R. Dedecker, P. Michiels, R. Buyle, A. De Craene, R. Verborgh, Publishing base registries as linked data event streams, Proceedings 21th ICWE (2021). URL: https://raw.githubusercontent.com/ddvlanck/Publishing-Base-Registries-As-LDES/master/Linked-Data-Event-Streams.pdf.

[7] R. T. Fielding, Architectural Styles and the Design of Network-based Software Architectures, Ph.D. thesis, University of California,Irvine, 2000.

[8] O. Hartig, M. T. Özsu, Walking without a map: Ranking-based traversal for querying linked data, in: P. Groth, E. Simperl, A. Gray, M. Sabou, M. Krötzsch, F. Lecue, F. Flöck, Y. Gil (Eds.), The Semantic Web – ISWC 2016, Springer International Publishing, Cham, 2016, pp. 305–324.

[9] O. Hartig, J. Pérez, Ldql: A query language for the web of linked data, Journal of Web Semantics 41 (2016) 9–29. URL: http://dx.doi.org/10.1016/j.websem.2016.10.001. doi:10.1016/j.websem.2016.10.001.

[10] B. Bogaerts, B. Ketsman, Y. Zeboudj, H. Aamer, R. Taelman, R. Verborgh, Link traversal with distributed subweb specifications, in: S. Moschoyiannis, R. Peñaloza, J. Vanthienen, A. Soylu, D. Roman (Eds.), Proceedings of the 5th International Joint Conference on Rules and Reasoning, volume 12851 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 62–79. URL: https://www.bartbogaerts.eu/articles/2021/005-RuleML-GuidedLink-SubwebSpec/SubwebSpecifications.pdf. doi:10.1007/978-3-030-91167-6_5.

[11] O. Hartig, J.-C. Freytag, Foundations of traversal based query execution over linked data, in: Conference on Hypertext and Social Media, HT '12, ACM, New York, NY, USA, 2012, p. 43–52. URL: https://doi.org/10.1145/2309996.2310005. doi:10.1145/2309996.2310005.

[12] R. Taelman, R. Verborgh, Link traversal query processing over decentralized environments with structural assumptions, in: Proceedings of the 22nd International Semantic Web Conference, 2023. URL: https://comunica.github.io/Article-ISWC2023-SolidQuery/.

[13] R. Verborgh, R. Taelman, Guided link-traversal-based query processing, 2020. URL: https://arxiv.org/abs/2005.02239. doi:10.48550/ARXIV.2005.02239.

[14] R. Taelman, J. Van Herwegen, M. Vander Sande, R. Verborgh, Comunica: a modular sparql query engine for the web, in: Proceedings 17th ISWC, 2018. URL: https://comunica.github.io/Article-ISWC2018-Resource/.

[15] Bram, S., De Brouwer, M., Stojchevska, M. , Van Der Donckt, J. , Nelis, J. , Ruyssinck, J., van der Herten, J. , Casier, K. , Van Ooteghem, J. , Crombez, P. , De Turck, F. , Van Hoecke, S. and Ongenae, F., Data Analytics For Health and Connected Care: Ontology, Knowledge Graph and Applications, in: Published in the proceedings of the sixteenth EAI Pervasive Healthcare conference, Springer, 2022. URL: https://dahcc.idlab.ugent.be.

[16] Eschauzier, Ruben and Taelman, Ruben and Verborgh, Ruben, How does the link queue evolve during traversal-based query processing?, in: 7th Workshop on Storing, Querying and Benchmarking Knowledge Graphs (QuWeDa) at ISWC 2023, 2023, pp. 26–33.