PRES LUNAM
Ecole Doctorale STIM
Sciences et Technologies de l'Information et Mathématiques

Spécialité : Informatique
Laboratoire : LINA
Equipe : GDD

Implementing the Synchronize in *Publish-Synchronize-Search*, a fresh and efficient paradigm to query the Web of Linked Data

Luis-Daniel Ibáñez
Mél : luis.ibanez@univ-nantes.fr

Abstract: The Linked Data initiative aims to transform the Web of interlinked documents into a Web of interlinked data in order to combine different knowledge bases and ask more powerful queries than text-based ones. However, current approaches to execute this queries suffer of problems of data up-to-dateness or scalability and performance. Linked Data participants are starting to provide streams of the changes they do in their datasets, opening the door to an alternative execution strategy: *Publish-Synchronize-Search*. Unfortunately, data synchronization among autonomous participants raises issues about consistency when concurrent updates occur. In this work, we propose the design of a Conflict-Free Replicated Data Type (CRDT) for the basic structure of Linked Data: RDF-Graph updated with SPARQL Update operations, to achieve eventual consistency guarantees with a low overhead in time, space and communication.

Keywords: *Semantic Web, Linked Data, Data Consistency, Query Execution*
Collaborations : ANR KolFlow

1 Introduction

The advent of the World Wide Web (WWW) as a massive tool for communication has made available a tremendous amount of information. The WWW is based on HTML, a text-oriented format that needs to be pre-processed to extract data that can be used to answer queries. The most popular search engines use the crawl-and-index strategy, i.e., to fetch the web pages into local servers by following the hyper-links in them, and then computing their relevance with respect to a set of keywords, and their ranking based on the number of inbound links. In practice, this approach has been proved very efficient to answer simple text-based queries, but many issues still need to be considered:

- How to ease the extraction of the data and make it easily reusable?
- How to discover *relevant* data for the query from many available sources?
- How to integrate data from large numbers of possibly unknown sources.

To solve this issues, the Linked Data [1] approach was proposed. Under this perspective the links are no more between text documents, but between data, and their nature is explicitly stated. In order to achieve this, the Resource Description Framework (RDF) [2, 3] data model was developed. RDF formalizes the use of Unique Resources Identifiers (URIs) to identify things and the relations they have as *RDF-triples* of the form *(subject,predicate,object)*. For example, assuming that we have the domain www.example.org (as we have with an HTML web site) the text asserting "Paris is the capital of France" can be expressed as:

```
(www.example.org/Paris , www.example.org/IsCapital , www.example.org/France)
```

Therefore, instead of having a collection of web pages we will have a set of RDF-triples, called RDF-Graphs. To perform queries on RDF-Graphs, the SPARQL [4, 5] and SPARQL Update [6] query languages were developed, with much more expressive power than text-based queries. For example, the query "What is the capital of France" in our example can be expressed as:

```
SELECT ?x WHERE {?x www.example.org/IsCapital www.example.org/France . }
```

With only minor changes we can ask a much more complex question, e.g. "List the capitals of all European countries":

```
SELECT ?x WHERE {?x www.example.org/IsCapital ?y .
                 ?y www.example.org/LocatedIn www.example.org/Europe .}
```

State of the art approaches to perform queries on distributed Linked Data sources [7, 8] suffer of execution time problems or of freshness problems, i.e., they can return answers that are not up-to-date with respect to the original sources. Nevertheless, we observe that many sites that adhere to the Linked Data approach have started to publish their changes to tackle the problem of freshness. The idea is that any interested client can consume these changes as soon as they are available. This opens the door to a new paradigm to query the Web of Linked Data: *Publish-Synchronize-Search*. The *Publish* phase refers to the data made available in real-time by the Linked Data participants, the *Synchronize* phase means to, after doing a first download of the required data to answer the query, to feed from the streams of changes published by the original sources to keep synchrony with them, and finally *Search* over the resulting set with fully up-to-date data.

However, if the synchronization is not properly managed, inconsistencies may arise. For example, if two different sources A, B contain the same RDF-triple (S, P, O), when we synchronize for the first time, we will have one copy of it, but if thereafter A modifies the triple as (S, P, O') and B as (S, P, O'') the final result of the synchronization will depend on the order in which the modifications were received. Figure 1 illustrates this issue.

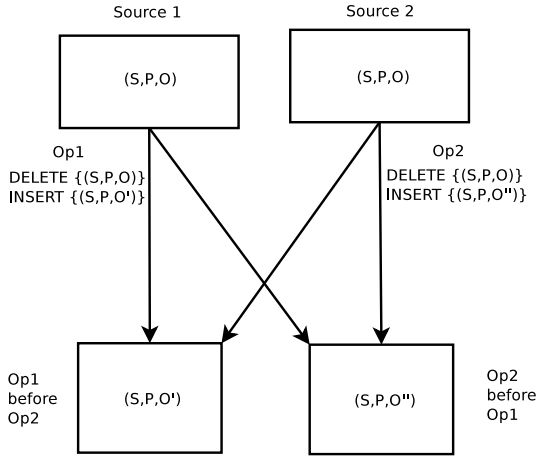

Figure 1: Inconsistencies in the synchronization phase. Starting with a triple (S,P,O) present in both sources, concurrent editions of this triple will produce different results depending on the order of application

In this work we propose the use of Conflict-Free Replicated Data Types (CRDTs) [9], an emerging formalism from the distributed systems field to give guarantees of *Eventual Consistency* [10] when different RDF-Graphs synchronize with each other in an autonomous way.

2 Problem Formalization

The RDF-Graphs that conform the web of Linked Data are managed by humans, who decide from which other RDF-Graph they are going to synchronize. Given the inherently social nature of this associations, we model them as a Social Network [11]:

Definition 1. A *Social Network* is a directed graph (N, L) where N is a finite set of elements called *actors* and L is a relation on the members of N.

We also state the required definitions from the RDF specification [3]:

Definition 2. Let URI is the set of URI References, $Literal$ the set of literal unicode strings and $Blank$ the set of blank nodes. An *RDF triple* is a 3-tuple (s, p, o) where $(s \in URI \vee s \in Blank) \wedge p \in URI \wedge (o \in URI \vee o \in Blank \vee o \in Literal)$. s is the subject of the triple, p is the predicate and o is the object where

For an *RDF triple* t, $t.s$ denotes the subject, $t.p$ the predicate and $t.o$ the object of t.

Definition 3. An *RDF Graph* is a set of RDF triples.

Then, we define the relation "follows changes of" to capture the fact of one RDF-Graph consuming the updates of anothers, and a corresponding social network that we call "Live Linked data":

Definition 4. Given two RDF-Graphs S_1 and S_2, we say that they have the tie *Follows changes of* $(S_1, S_2)_{follows-changes}$ iff every SPARQL Update operation applied by S_2 is eventually also applied by S_1. We shall call S_1 the *follower* and S_2 the *followed*.

Definition 5. The social network (N, L) where N is a set of RDF-Graphs and L the relation *Follows changes of* is called *Live Linked Data*(LLD).

The eventual consistency problem on this context is stated as follows: if LLD is connected, when all RDF-Graphs in N stop generating update operations and finish to apply the ones from the RDF-Graphs they are following, then, all RDF-Graphs must have the same triples.

3 Proposed approach

If the application of all SPARQL Update operations over any RDF-Graph commutes, then there is no possibility of having inconsistencies, i.e., the RDF-Graph type would be Conflict-Free. SPARQL Update operations do not commute, as we illustrated in figure 1 , but we can rewrite them into another type with operations that does commute while preserving the original semantics. In [12] we presented such a rewriting under the name of SU-Set. We do not show the full specification due to space constraints, but we sketch the general idea: As SPARQL Update basic operations can be expressed as set union and difference operations, we adapted an existing CRDT for the set type called ObservedRemove-Set [9], based on tagging each inserted element uniquely, to be able to delete only the elements we have observed. We also made one optimization for the special case of SPARQL Update operations: the unique tag is not generated for each element, but for each insert operation, and attached to each of the inserted RDF-triples.

In a subsequent work [13], we conducted a detailed performance analysis of SU-Set, using as reference the statistics of one week of DBpedia Live [14], the framework to publish changesets implemented by DBpedia, the largest participant of the Linked Data initiative. We concluded that:

- SU-Set is optimal in terms of rounds of communication needed to converge. No additional communication is needed after the operation is received and executed.
- In terms of space, the extra storage needed for an unique identifier for each triple is negligible.
- There is no overhead in the query operations caused by the unique identifier.
- The only parameter where SU-Set is more expensive is in communication, i.e., the size of the messages sent, notably for the deletions. However, considering the dinamics of DBPedia Live, with almost 12 times more triples inserted than deletions, SU-Set indtroduces an overhead of only 5%.

4 Related Work

Previous efforts on synchronization of RDF-Graphs [15, 16] do not consider any consistency criteria. In Distributed Databases [17], the synchronization problems are studied in the context of data replication, specifically in the multi-master (as all replicas are allowed to write) and lazy propagated (as the updates are committed locally before knowing if they arrived to the other replicas). Under this constraints, the Mutual Consistency studied in this field is the same as the Eventual Consistency we use. However, the approaches to achieve it are based on re-schedulers of operations on arrival. Re-schedulers are complex to program and introduce a high overhead in rounds of communication . To the best of our knowledge, this is the first use of CRDTs in this context.

5 Conclusion and future work

SU-Set is a CRDT for RDF-Graphs updated with SPARQL Update operations that guarantees eventual consistency with very low overhead when RDF-Graphs synchronize with each other. This will allow the correct implementation of *Publish-Synchronize-Search* (PSS) as an alternative paradigm to execute queries in the Web of Linked Data. Future work includes the comparison of PSS with federated queries approaches and warehouses and study the implications that the Live Linked Data network can have on the discovery of relevant sources to answer a query.

References

[1] Tom Heath and Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, 2011.

[2] Eric Miller. An introduction to the resource description framework. *Bulletin of the American Society for Information Science and Technology*, 25(1):15–19, 2005.

[3] W3C. *Resource Description Framework (RDF): Concepts and Abstract Syntax*, February 2004.

[4] W3C. *SPARQL 1.1 Query Language*, November 2012.

[5] Jorge Pérez, Marcelo Arenas, and Claudio Gutiérrez. Semantics and complexity of sparql. *ACM Transactions on Database Systems*, 34(3), August 2009.

[6] W3C. *SPARQL 1.1 Update*, January 2012.

[7] Olaf Hartig and Andreas Langegger. A database perspective on consuming linked data on the web. *Datenbank-Spektrum*, 10(2):57–66, 2010.

[8] Peter Haase, Tobias Mathäß, and Michael Ziller. An evaluation of approaches to federated query processing over linked data. In *Proceedings of the 6th International Conference on Semantic Systems*, pages 51–59. ACM, 2010.

[9] Marc Shapiro, Nuno M. Preguiça, Carlos Baquero, and Marek Zawirski. Conflict-free replicated data types. In *Stabilization, Safety, and Security of Distributed Systems - 13th International Symposium, SSS 2011*, pages 386–400, 2011.

[10] Yasushi Saito and Marc Shapiro. Optimistic replication. *ACM Computer Surveys*, 37(1):42–81, 2005.

[11] Stanley Wasserman and Katherine Faust. *Social network analysis: Methods and applications*. Cambridge University Press, 1994.

[12] Luis Daniel Ibáñez, Hala Skaf-Molli, Pascal Molli, and Olivier Corby. Synchronizing semantic stores with commutative replicated data type. In *Semantic Web Collaborative Spaces Workshop, at the 21st International World Wide Web Conference (WWW)*, 2012.

[13] Luis Daniel Ibáñez, Hala Skaf-Molli, Pascal Molli, and Olivier Corby. Live linked data: Synchronizing semantic stores with commutative replicated data type. *International Journal of Metadata, Semantics and Ontologies (to Appear)*, 2013.

[14] Mohamed Morsey, Jens Lehmann, Sören Auer, Claus Stadler, and Sebastian Hellmann. Dbpedia and the live extraction of structured data from wikipedia. *Program: electronic library and information systems*, 46(2):157–181, 2012.

[15] Giovanni Tummarello, Christian Morbidoni, Joackin Petersson, Paolo Puliti, and Francesco Piazza. Rdfgrowth, a p2p annotation exchange algorithm for scalable semantic web applications. In *Proceedings of the MobiQuitous'04 Workshop on Peer-to-Peer Knowledge Management (P2PKM 2004)*, 2004.

[16] Giovanni Tummarello, Christian Morbidoni, Reto Bachmann-Gmür, and Orri Erling. Rdfsync: Efficient remote synchronization of rdf models. In *6th International and 2nd Asian Semantic Web Conference (ISWC + ASWC)*, pages 537–551, 2007.

[17] M. Tämer Ozsu and Patrick Valduriez. *Principles of Distributed Database Systems*. Springer, 3rd edition, 2011.