Consistency criteria for a Read/Write Web of Linked Data

Luis-Daniel Ibáñez

LINA, University of Nantes
luis.ibanez@univ-nantes.fr

Abstract. The Linked Data initiative has made possible the publishing and interlinking of a tremendous amount of data, this links can be followed allowing the gathering of related concepts stored in different servers, enabling the answering of powerful queries and richer end-user applications. Current Linked Data is Read-only and many researchers, including its creator Tim Berners-Lee, stand for its evolution to Read-/Write to enable man-machine and machine-machine collaboration in the process of interlinking, cleaning and querying the Linked Data. However, when multiple agents can read/write on the same distributed data, the issue of *consistency* arise. The subject of this thesis is to determine which consistency criterion is the more adequate for the Web of Linked Data and propose tractable algorithms to maintain it.

1 Problem Statement

The quest of this thesis is stated as follows: To determine which consistency criterion is the most adequate for a Read/Write Web of Linked Data and to propose algorithms to maintain it. Such criterion must be strong enough to give guarantees to the consumers, programmers and users when updating or querying Linked Data, and tractable enough to be maintainable under the Web of Linked Data conditions: A steadily growing and projected very high number of *autonomous* participants with a wide-range of dynamics, that together hold a very large volume of data.

2 Relevancy

The Linked Data initiative [4,14] has led to the publication and interlinking of billions of pieces of data in the Resource Description Format, transforming the traditional *Web of Documents* into the *Web of Linked Data*. In the Linked Data ideal, data consumers, *i.e.*, developers and their applications make use of the links between pieces of data to discover and query on related data stored in remote servers, augmenting their added-value and enriching user experience.

However, the current Web of Linked Data is Read-Only, limiting its potential. The evolution to a *Read/Write* Web of Linked Data will have the following benefits:

– Enable truly REST-ful resource oriented Web-APIs [8].
– Break the data silos and empower the users to regain control of their data. Applications can access and modify the data they have permission to. Combined with the Linked Data principles, this also allows networking across platforms and the use of the Web as infrastructure for applications [3].
– Data could be cleaned and evolve with the collaborative intervention of human and machine agents. The knowledge stored in different communities or even by different individuals or applications could *co-evolve* [11].

The Read/Write Web of Linked Data can be seen as network of participants that can update each other's data or copy, modify and exchange updates autonomously. If this exchange is not properly managed, data may diverge uncontrollably, or in a non-deterministic way. This makes impossible the assertion of guarantees on the results of the queries and updates, severely undermining the interaction between the participants [32]. Therefore, to realize the vision of a Read/Write Web of Linked Data, a suitable *consistency* criterion is needed.

3 Related Work

Consistency is a problem transversal to several research communities. In Computer-Supported Cooperative Work is studied in the context of cooperative edition, while in Distributed Systems, Databases, and Semantic Web, often appears when studying the general *replication* problem: A set of *replicas* on which operations are issued by clients (human or agents) and communicate by exchanging messages [33]. When all messages are exchanged, the system must comply with a set of assertions: the consistency criterion.

3.1 Consistency in Computer-Supported Cooperative Work

The main study about consistency in this community was made in the context of Real-Time Editing Systems [28], where the Causality-Convergence-Intention (CCI) model was proposed. The formalisation of all three is studied in Distributed Systems.

Another very important model developed in this community is the *copy-modify-merge* paradigm [10], well known to developers thanks to its implementation in *Version Control Systems*. For the Web of Linked Data characteristics, the most related are *Distributed* Version Control Systems (DVCS), whose prime example is Git[1].

Both models are oriented to text and documents rather than to data.

3.2 Replication and Consistency in Distributed Systems

Replication algorithms and consistency criteria in Distributed Systems can be divided in two main families [25,22]. The first one is the *Pessimistic* category,

[1] `http://git-scm.org`

the goal is to attain a *Strong* consistency, *i.e.*, clients will have the illusion that there is only one replica, fully protected from the effects of failures and concurrent updates. The main consistency criteria is *linearisability* [15].

However, the fundamental CAP Theorem [5] states than in the presence of partitions, whether it be by communication disconnection or by *off-line operations*, is it not possible to have strong consistency without sacrificing high availability. Indeed, the protocols to guarantee strong consistency need to block the system, therefore, the family of *Optimistic* replication algorithms [25] was developed. Optimistic protocols focus on *weak consistency* criteria, where replicas are allowed to diverge during some time (the time of the partition) but remain available, causing the output of some reads (queries) to be different at different replicas during this time window. The main criterion to maintain in this family is *Eventual Consistency* [25]

3.3 Replication and Consistency in Databases

In Distributed Databases, the classification of strategies is done with respect to which replicas can execute updates (master-slaves vs multi-master scheme) and how updates are propagated (eagerly or lazily) [33,24]. Nevertheless, this induces two types of consistency [24]: *Transactional* consistency, which refers to the maintenance of global integrity constraints when concurrent updates occur, and *Mutual* consistency, which refers to data items having identical values at all replicas.

Mutual consistency can be weak or strong, and is equivalent to the weak and strong consistency in Distributed Systems. Transactional consistency can be seen as strong consistency with the extra constraint of integrity (therefore, harder to maintain).

Another related database subject related to our work is *Materialized View Maintenance* [7]. A *target* database stores a snapshot of the result of a query (the view) on one or more *source* databases, when updates happen at the sources the snapshot stored at the target may become inconsistent with the new data at the sources, *i.e.*, the evaluation of the view at the sources is not equal to what is stored. To regain consistency, the target needs to choose, based on a cost model, if it re-evaluates the view, or if integrates the incoming updates (that are assumed to be available somewhere, or arriving through a stream). The design of such integration or *maintenance* algorithms under various constraints is the main issue of this research area.

3.4 Replication in Semantic Web

The first work on update exchange for Semantic Stores appeared in [2]. An ontology was proposed to add semantics to the exchange of diff files. Replication for Semantic Stores has been treated in RDFGrowth [31] where an update exchange algorithm is proposed assuming *monotonic* updates; and in [30], that discusses an adaptation of the popular rsync algorithm to RDF. A full collaborative semantic platform based on these ideas is presented in [29].

However, neither of these works presents any consistency criterion. In fact, there is a gap in the Linked Data and Semantic Web literature concerning the study of criteria specific to them. For example, [14] and [13] do not treat the issue, while [1] refers to the distributed systems and databases criteria discussed in sections 3.2 and 3.3.

There also exist adaptations of DVCS to data encoded in RDF [6,26]. However, their main focus is to bring to the Web their versioning capabilities; there is no study of a general criterion for the Web of Linked Data. Nevertheless, DVCS comply with the postulates of Eventual Consistency, but adding cooperative editing functionalities.

4 Research Questions

1. Which consistency criterion is the most appropriate for a Read/Write Web of Linked Data?
2. Is there a scalable algorithm to maintain such criterion while respecting the autonomy of the participants and without compromising their availability?
3. How to handle conflicting updates with respect to ontology constraints, *i.e.*, when two members disagree semantically?

5 Hypotheses

1. The appropriate consistency criterion lies on the weak category.
2. An optimistic scalable algorithm to maintain such criteria while respecting the autonomy and without compromising the availability exists. Its complexity in time and space is at most polynomial in each of the following parameters: Number of sites, number of updates, size of the datasets, and at most linear in the number of messages exchanged (communication complexity)
3. Semantic agreement at the whole Web of Linked Data relates to transactional consistency and is not attainable in a scalable way.
4. The use of provenance, together with a consistency criterion, help users to solve semantic conflicts locally while still having weaker consistency guarantees at the global level.

6 Approach

From our study of the state of the art we can draw the conclusion that all research communities agree on the CAP theorem result - Strong consistency is not attainable in a scalable and available way - and that the criterion of choice for applications where scalability and availability are a must is *Eventual Consistency*. Therefore, our first approach is to test *Eventual Consistency* and design an algorithm to maintain it on the Web of Linked Data.

We use the recently developed formalism of Conflict-Free Replicated Data Types (CRDTs) [27] for its simple, yet powerful, theoretical foundation and multiple cases of success in industrial and collaborative scenarios [9,23,**?**]. CRDTs also have low footprint compared to DVCS. We claim that participants interested in versioning may put them on top of their stack, while others without the resources or the interest should have the choice of not doing so.

We design a CRDT for the RDF GraphStore type with the SPARQL 1.1 Update Operations, thus, following the W3C recommendations. It will be the first time that CRDTs will be applied to the Web and Linked Data world.

However, Eventual consistency requires two strong assumptions about the network: (i) all updates eventually reach all participants, which implies (ii) the network is connected.

Therefore, as a second approach, we develop a criterion strictly stronger than Eventual consistency based on the View Maintenance notion, and an algorithm to maintain it based on annotated data [20,12]. These tools were developed in the context of Collaborative Data Sharing Systems (CDSS) [21], however, CDSS do not scale beyond few hundreds of participants as their consistency criterion is closer to transactional consistency on heterogeneous databases. Our innovation here is the realization that for the Web of Linked Data we look for the reverse goal, weaker consistency to boost scalability, but we still can adapt the same tools.

Both strategies adapt well to the inclusion of provenance expressed as semirings [20], allowing us to test our fourth hypotheses.

7 Evaluation Plan

Both approaches will undergo a thorough complexity analysis to identify their worst case complexities and test of our hypotheses that the algorithm to maintain consistency is polynomial in the key parameters: number of participants, number of updates and size of the datasets.

With the worst cases established, we plan to analyze the average cases by implementing them and testing them with a high number of participants and current Linked Data dynamics and size [19].

Comparisons will be done with respect to doing nothing and between the two approaches. The questions to answer are: how much is the overhead we need to pay for having each of these levels of consistency? How much more expensive is our view-maintenance based criterion with respect to the eventual consistency provided by our CRDT? Is it affordable at all cases? If not, in which ones it is?

The planned experimental measures are:

- Disk usage (Maximum and average).
- Execution Time of the update exchange protocol at a given participant.
- Time of convergence to the criteria of all the network (still open if its better to measure this in number of times that the algorithm was executed at each participant instead of wall time)
- Number of messages exchanged before convergence.

8 Preliminary Results

SU-Set, a CRDT for RDF-Graph and SPARQL 1.1 Update operations was first sketched in [16] and subsequently specified and analyzed in [17]. The worst case complexities were showed to be indeed polynomial (the communication complexity being constant), confirming our hypotheses.

The view-maintenance based criterion was presented in [18]. The algorithm to maintain it also showed to be polynomial in the key factors except for space in terms of number of updates when the network has a high density: an integer coefficient required by the algorithm to be stored as part of the data annotation may attain values in the order of factorial of the number of participants if the network forms a complete graph.

The experimentation to estimate if this is a concern in the average case, and to estimate the performance in all average cases, is ongoing work. We also work on the full characterization of the parameters that affect our solutions (*e.g.* we already found that network topology is one of them).

9 Conclusion and Reflections

Shifting the Web of Linked Data paradigm from Read-Only to Read/Write will benefit its data quality and general usability. However, when humans and machines have permission to write and exchange updates, consistency criteria are needed to state some guarantees on the evolution of the knowledge it stores and on the queries posed on it.

The main quest of this thesis is to find the most appropriate consistency criteria for the Linked Data Web and to design and analyze the algorithms to maintain them. Such criteria must be strong enough to be useful, but tractable enough to be maintained considering the characteristics of the projected Web of Linked Data: Very high number of autonomous participants that together represent a very high volume of data, a potentially high number of clients (human and machine) performing queries on it, and highly dynamic in nature.

We test the Eventual Consistency criteria from Distributed Systems by designing a Conflict-Free Replicated Data Type for the RDF-Graph type with SPARQL Update operations and found it complied with the imposed requirement of polynomial complexity. We also proposed a second criterion, stronger than Eventual Consistency, based on the notion of View Maintenance and an algorithm to maintain it founded on the concept of semi-ring annotated data. This second criterion complies with the polynomial complexity requirements except for when the network is dense. Experimentation to estimate the performance on average cases is ongoing work.

We believe that in the end our second criteria will be proven affordable for two reasons: first, the Linked Data Web will tend to be a socially-organized network more than random or self-organized and such networks are far from being complete; second, if the value of some of these coefficients is high, implementations may switch to BigInt arithmetics, meaning that the effective cost in storage is still affordable except in the most extreme cases.

Acknowledgements: This thesis is principally supervised by Prof. Pascal Molli (University of Nantes), and co-supervised by Associate Prof. Hala Skaf-Molli (University of Nantes) and Olivier Corby (Researcher at INRIA Sophia-Antipolis Mediterranée). Funding is provided by the French National Research Agency (ANR) through the KolFlow project (code: ANR-10-CONTINT-025).

References

1. Abiteboul, S., Manolescu, I., Rigaux, P., Rousset, M.C., Senellart, P.: Web Data Management. Cambridge University Press, 1st edn. (February 2012)
2. Berners-Lee, T., Connolly, D.: Delta: an ontology for the distribution of differences between rdf graphs. `http://www.w3.org/DesignIssues/Diff` (2004)
3. Berners-Lee, T., O'Hara, K.: The read-write linked data web. Philosophical Transactions of the Royal Society (A 371) (February 2013)
4. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. International Journal of Semantic Web Information Systems 5(3), 1–22 (2009)
5. Brewer, E.: Cap twelve years later: How the "rules" have changed. IEEE Computer 45(2) (2012)
6. Cassidy, S., Ballantine, J.: Version control for rdf triple stores. In: Proceedings of the Second International Conference on Software and Data Technologies (ICSOFT) (2007)
7. Chirkova, R., Yang, J.: Materialized views. Foundations and Trends in Databases 4(4) (2011)
8. Coppens, S., Verborgh, R., Sande, M.V., Deursen, D.V., Mannens, E., de Walle, R.V.: A truly read-write web for machines as the next generation web? In: Proceedings of the SW2012 workshop: What will the Semantic Web look like 10 years from now? (2012)
9. Deftu, A., Griebsch, J.: A scalable conflict-free replicated set data type. In: IEEE 33rd International Conference on Distributed Computing Systems (ICDCS) (2013)
10. Dourish, P.: The parting of the ways: Divergence, data management and collaborative work. In: Proceedings of the fourth European Conference on Computer-Supported Cooperative Work (ECSCW) (1995)
11. Engelbart, D., Lehtman, H.: Working together. Byte 13(13) (1988)
12. Green, T.J., Ives, Z.G., Tannen, V.: Reconcilable differences. Theory of Computer Systems 49(2) (2011)
13. Groppe, S.: Data Management and Query Processing in Semantic Web Databases. Springer (2011)
14. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Morgan and Claypool (2011)
15. Herlihy, M., Wing, J.: Linearizability: A correctness condition for concurrent objects. ACM Transactions on Programming Languages and Systems (TOPLAS) 12(3) (1990)
16. Ibáñez, L.D., Skaf-Molli, H., Molli, P., Corby, O.: Synchronizing semantic stores with commutative replicated data types. In: Proceedings of the first Semantic Web Collaborative Spaces Workshop (SWCS@WWW'12) (2012)
17. Ibáñez, L.D., Skaf-Molli, H., Molli, P., Corby, O.: Live linked data: Synchronizing semantic stores with commutative replicated data types. International Journal of Metadata, Semantics and Ontologies 8(2) (2013)

18. Ibáñez, L.D., Skaf-Molli, H., Molli, P., Corby, O.: Making linked data writable with provenance semi-rings. Tech. rep., Université de Nantes (2014)
19. Käfer, T., Abdelrahman, A., Umbrich, J., O'Byrne, P., Hogan, A.: Observing linked data dynamics. In: The Semantic Web: Semantics and Big Data, 10th International Conference (ESWC) (2013)
20. Karvounarakis, G., Green, T.J.: Semiring-annotated data: Queries and provenance. SIGMOD Record 41(3) (2012)
21. Karvounarakis, G., Green, T.J., Ives, Z.G., Tannen, V.: Collaborative data sharing via update exchange and provenance. ACM Transactions on Database Systems (TODS) 38(3) (August 2013)
22. Kemme, B., Ramalingam, G., Schiper, A., Shapiro, M., Vaswani, K.: Consistency in distributed systems. Dagstuhl Reports 3(2), 92–126 (2013)
23. Nédelec, B., Molli, P., Mostéfaoui, A., Desmontils, E.: Lseq: an adaptive structure for sequences in distributed collaborative editing. In: ACM Symposium on Document Engineering (DocEng) (2013)
24. Özsu, M.T., Valduriez, P.: Principles of Distributed Database Systems. Springer (2011)
25. Saito, Y., Shapiro, M.: Optimistic replication. ACM Comput. Survey 37(1), 42–81 (2005)
26. Sande, M.V., Colpaert, P., Verborgh, R., Coppens, S., Mannens, E., de Walle, R.V.: R&wbase:git for triples. In: Proceedings of the WWW2013 Workshop on Linked Data on the Web (LDOW) (2013)
27. Shapiro, M., Preguiça, N., Baquero, C., Zawirski, M.: Conflict-free replicated data types. In: International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS). pp. 386–400 (2011)
28. Sun, C., Jia, X., Zhang, Y., Yang, Y., Chen, D.: Achieving convergence, causality preservation, and intention preservation in real-time cooperative editing systems. ACM Transactions on Computer-Human Interaction 5(1), 63–108 (1998)
29. Tummarello, G., Morbidoni, C.: The dbin platform: A complete environment for semantic web communities. Journal of Web Semantics 6(4) (2008)
30. Tummarello, G., Morbidoni, C., Bachmann-Gmür, R., Erling, O.: Rdfsync: Efficient remote synchronization of rdf models. In: 6th International and 2nd Asian Semantic Web Conference (ISWC + ASWC). pp. 537–551 (2007)
31. Tummarello, G., Morbidoni, C., Petersson, J., Puliti, P., Piazza, F.: 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)
32. Umbrich, J., Karnstedt, M., Parreira, J.X., Polleres, A., Hauswirth, M.: Linked data and live querying for enabling support platforms for web dataspaces. In: Third International Workshop on Data Engineering Meets the Semantic Web (DESWEB) (2012)
33. Wiesmann, M., Pedone, F., Schiper, A., Kemme, B., Alonso, G.: Understanding replication in databases and distributed systems. In: Proceedings of the 20th International Conference on Distributed Computing Systems (ICDCS) (2000)