RDF Graph Stores as Convergent Datatypes

James Anderson
Datagraph GmbH
james@dydra.com

ABSTRACT

This report describes a way to represent and operate on an RDF dataset such the it behaves as an instance of a conflict free replicable datatype. In this industry presentation, we describe how we accomplish this for the Dydra RDF graph storage service in a manner compatible with the SPARQL Graph Store HTTP Protocol (GSP). The standard GSP concerns the current store state only. Dydra retains previous store states as active addressable aspects analogous to named graphs in a quad store. It incorporates and addresses arbitrary revisions of target datasets according to `ETag` and `Content-Disposition` specifications in HTTP headers. Appropriate interpretation of these arguments permits to replicate datasets among cooperating participants.

CCS CONCEPTS

• **Information systems → Version management**; **Resource Description Framework (RDF)**; *Cloud based storage*; *Distributed storage*; *Digital libraries and archives.*

KEYWORDS

CvRDT; Graph Store Protocol; revisions; RDF;
temporal data

ACM Reference Format:

James Anderson. 2019. RDF Graph Stores as Convergent Datatypes. In *Companion Proceedings of the 2019 World Wide Web Conference (WWW '19 Companion), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3308560.3316517

1 INTRODUCTION

Dydra is an RDF graph storage service. It operates as a cloud service, a local service or an embedded library. It stores and retrieves the contents of RDF datasets through SPARQL[1], GraphQL[14], TPF[2]and LDP[3] interfaces. It provides uniform access through the SPARQL query language to a variety of data sources, from local RDF quad stores, to local and remote SQL[13] stores, and remote REST[4] data servers. Dydra retains previous store states, in addition to the current state, as active addressable aspects of a dataset analogous to named graphs in a quad store by annotating statements with temporal attributes[5] and permits to address these states as static projections through provision of a `REVISION` argument in SPARQL and Graph Store requests or as streams over MQTT[16] through provision of a `REVISION-WINDOW` argument. Through an appropriate interpretation of revision states, in combination with the appropriate state transfer protocol, it is possible to comprehend repositories as components of a distributed store which behaves as an instance of a conflict-free replicated data type and use them as the storage substrate for distributed collaborative document management.

This document describes how Dydra's schematic repository storage model and Graph Store Protocol extensions combine to provide this capability. The next section describes a schematic graph revision model which fulfils the requirements of a conflict-free replicable data type. Section 3 introduces the Graph Store Protocol and describes extended requests which convey annotations necessary to replicate graph revisions. Section 4 proposes an approach to establish semantically correct states.

2 A MODEL FOR GRAPH REVISIONS AS CRDT

Lamport [19] describes the requirements to order events in a distributed system. He intends to afford causality to the process by deriving a total order for the events in order to achieve "correct" operation, where "correct" intends : all participants agree as to one causality, in which any state depends only on states which "precede" it. As one example, where correctness matters, he discusses how to arbitrate access to a shared resource. For this purpose he proposes, "[i]t just involves making sure that each process learns about all other processes' operations." [19, p.151] For a temporal RDF storage service, in Lamport's terms the shared resource, becomes the "$repositoryState@time >= T_i$". It can achieve a form of *eventual correctness*[6] and derive it retrospectively, rather than àpriori, by relying on asynchronous local clocks and convergent replicable data types. A participant may assert states for times after they have been admitted to the process only and assertions take the form of statement additions and removals with annotations to specify their provenance. Based upon these, replicas will converge globally on a state which reflects causality.

[1]The *SPARQL* Protocol and RDF Query Language (SPARQL) is a query language and protocol for RDF[8]-

[2]"*Triple Pattern Fragments* (TPF), a low-cost interface to triples"[30] within the *Linked Data Fragments* framework[29].

[3]"*Linked Data Platform* (LDP) defines a set of rules for HTTP operations on web resources, some based on RDF, to provide an architecture for read-write Linked Data on the web"[9]

[4]The *Representational State Transfer* (REST) style is an abstraction of the architectural elements within a distributed hypermedia system... in order to focus on the roles of components, the constraints upon their interaction with other components, and their interpretation of significant data elements[12]. Dydra supports Web API endpoints which can be expressed in terms of the Hydra vocabulary[20].

[5]Dydra implements a transaction-time temporal RDF store[1][28] with a particular form of what is characterised as a "timestamp-based approach"[10].

[6] The approach relaxes the requirement, that a local clock be adjusted to reflect remote timestamps[19, p.560].

ACM ISBN 978-1-4503-6675-5/19/05.
https://doi.org/10.1145/3308560.3316517

The treatments of RDF graphs as CRDT follow from Shapiro's *Strong Eventual Consistency* (SEC) model and its application to set and graph data types. Within this model, if an object supports commutative, associative and idempotent operators, it qualifies as a Convergent Replicated Data Type (CvRDT) [26, p.6][27]. Initial work by Weiss[31] and Nedelec[23] applied this approach to collaboratively edited text documents to demonstrate that is is possible to satisfy CRDT ordering requirements by replicating identifiers which are generated locally by incrementally uniquely modifying existing identifiers. Bieniusa [4] and Deftu [7] described more effective implementations of Shapiro's Observed-Removed-(OR)-Set. An RDF dataset is a set of statements which model a graph [6]. The standard HTTP Graph Store access protocols[24] permit to add to and remove statements from graph. For an RDF dataset which follows just the standard abstract semantics, these operations do not fulfil all of the requirements to qualify as a CvRDT. The operations insertion and deletion are idempotent and associative, but they are not commutative. Aslan's C-Set [2], Ibanez's SU-Set [18][17] and Zarzour's B-Set[32] extend the OR-Set and apply it to distributed semantic stores by following the example of local identifier generation and associating some combination of a presence indicator and provenance record with each operand statement in order to satisfy the SEC requirements and allow operations to converge on a shared state.

As a revisioned RDF dataset extends the standard semantics to distinguish versions, it is instead possible to order operations according to the rules for asynchronous clocks and, with that ordering, to render the operations commutative. This is possible, if one ensures that each modification is labeled in such a manner that a means exists for all participants to establish the same subsequent state given the same eventual knowledge of events. This, in turn is possible if the labels are sufficient to place all operations in a space from which all participants can establish the same total order for events in order to produce the same projection for any given location in the space. This is accomplished by establishing revision identifiers which follow Lamport's rules for distributed clocks: each revision is identified by combinations of

$$localTimestamp \times participantId \times (ADD + REMOVE) \qquad (1)$$

and ordering them by first comparing the time value, then the mode number and then the operation, for which ADD precedes REMOVE. As the store is revisioned, all states are retained and the behavior is that of a Last-WriterWins(LWW)-element-Set [25, p. 24], but without the tombstone overhead.

3 GRAPH STORE PROTOCOL EXTENSIONS

The core of the Dydra RDF service is a revisioned quad store, which is accessed through the W3C RDF Graph Store (GSP) and SPARQL protocols. As an extension to standard RDF semantics [6], the store implements the versioning pattern, as described above ("timestamp/participant/operation"), In order to permit operations which express the terms described above, the information must be conveyed in Graph Store Protocol requests. This means any extensions must be compatible with the GSP/HTTP protocol at the protocol and the application level.

The standard GSP update methods are

```
PATCH /jhacker/test/propagate HTTP/1.1
Content-Type: multipart/related; boundary=PATCH
Etag: d745f480-2661-11e9-9eb9-010203040506
Content-Disposition: replicate=d3graph

--PATCH
X-HTTP-Method-Override: DELETE
ContentType: application/n-quads

<http://example.org/5f3bc330-2052-11e9-82ae-010203040506>\
 <http://example.org/x>\
  ``963''^^<http://www.w3.org/2001/XMLSchema#integer> .
--PATCH
X-HTTP-Method-Override: POST
ContentType: application/n-quads

<http://example.org/5f3bc330-2052-11e9-82ae-010203040506>\
 <http://example.org/y>\
  ``291''^^<http://www.w3.org/2001/XMLSchema#integer> .
--PATCH--
```

Figure 1: HTTP PATCH request

- **DELETE**: delete the request content from the the graph.
- **POST**: add the request content to the content of the graph.
- **PUT**: replace the content of the graph with the request content.

We extend these to

- use multipart/related content as a sequence of sub-GSP operations in a **PATCH** request to modify store content,
- use an `ETag` header to specify revision identifiers, and
- use a `Content-Disposition` request header to define, subscribe to and route **PATCH** requests.

Where these requests identify no revision, a revision identifier is generated which combines the current participant-time position of the service node with the operation identifier. If a revision identifier is provided, that is asserted as the participant-time position. In order to provide for the case where an operation comprises both additions and deletions, the **PATCH** operation is also supported in a manner which accepts multipart/related request content to perform the composite operation within a single store transaction. Such a request would take the form illustrated in figure 1[7]. It replaces the 'x' property of a node by sequencing a **DELETE** and a **POST** operation and positions the result revision by providing an `ETag`. The `Content-Disposition` identifies an exchange which indexes WebSocket[11] connections to participants over which the **PATCH** requests are sent in order to replicate state.

4 APPLICATION-LEVEL SEMANTIC INTEGRITY

The CRDT approach guarantees eventual consistency between trusted nodes at the RDF graph level. It does not address malicious or untrusted participants at the application level and it does not provide means to address application level semantic constraints, e.g. double spend race conditions. We propose to mitigate malicious operations by recording each parent revision in order to capture provenance chains and suppressing any states asserted by such participants based on the location information encoded in the revision identifier. Where the revisions from a specific location are suppressed, a temporal projection is then computed as if they were

[7]The example depicts N-Quads[5], with the lines are wrapped for presentation. Where structural anonymous nodes[21, p. 425] are required, Turtle[3] can be used to distinguish identifiers with document scope and dynamic extent from those with indefinite extent and a scope which comprises the surface shared among the replicas[15].

absent, their effect is eliminated from any projection and their support is removed from any dependant revision. Correctness is then established by electing the states which derive respectively from the earliest independent parent revision.

REFERENCES

[1] James Anderson and Arto Bendiken. 2016. Transaction-Time Queries in Dydra. In *Joint Proceedings of the 2nd Workshop on Managing the Evolution and Preservation of the Data Web (MEPDaW 2016) and the 3rd Workshop on Linked Data Quality (LDQ 2016) co-located with 13th European Semantic Web Conference (ESWC 2016), Heraklion, Crete, Greece, May 30th, 2016. (CEUR Workshop Proceedings)*, Jeremy Debattista, Jürgen Umbrich, Javier D. Fernández, Anisa Rula, Amrapali Zaveri, Magnus Knuth, and Dimitris Kontokostas (Eds.), Vol. 1585. CEUR-WS.org, 11–19. http://ceur-ws.org/Vol-1585/mepdaw2016_paper_02.pdf

[2] Khaled Aslan, Pascal Molli, Hala Skaf-Molli, and Stéphane Weiss. 2011. C-set: a commutative replicated data type for semantic stores. In *RED: Fourth International Workshop on REsource Discovery*.

[3] David Becket, Tim Berners-Lee, Eric Prud'hommeaux, and Gavin Carothers. 2014. *RDF 1.1 Turtle*. W3C Recommendation. World Wide Web Consortium (W3C). http://www.w3.org/TR/turtle/

[4] Annette Bieniusa, Marek Zawirski, Nuno Preguiça, Marc Shapiro, Carlos Baquero, Valter Balegas, and Sérgio Duarte. 2012. An optimized conflict-free replicated set. *arXiv preprint arXiv:1210.3368* (2012). https://arxiv.org/pdf/1210.3368.pdf

[5] Gavin Carothers. 2014. *RDF 1.1 N-Quads*. W3C Recommendation. World Wide Web Consortium (W3C). http://www.w3.org/TR/turtle/

[6] Richard Cyganiak, David Wood, and Markus Lanthaler. 2014. *RDF 1.1 Concepts and Abstract Syntax*. W3C Recommendation. World Wide Web Consortium (W3C). http://www.w3.org/TR/rdf11-concepts/

[7] Andrei Deftu and Jan Griebsch. 2013. A scalable conflict-free replicated set data type. In *Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on*. IEEE, 186–195.

[8] Lee Feigenbaum, Gregory Williams, Kendall Clark, and Torres Elias. 2013. *SPARQL 1.1 Protocol*. W3C Recommendation. World Wide Web Consortium (W3C). http://www.w3.org/TR/sparql11-protocol/

[9] Lee Feigenbaum, Gregory Williams, Kendall Clark, and Torres Elias. 2015. *Linked Data Platform 1.0*. W3C Recommendation. World Wide Web Consortium (W3C). http://www.w3.org/TR/ldp/

[10] Javier D. Fernández, Jürgen Umbrich, Axel Polleres, and Magnus Knuth. 2019. Evaluating query and storage strategies for RDF archives. *Semantic Web* 10, 2 (2019), 247–291. https://doi.org/10.3233/SW-180309

[11] Ian Fette and Alexey Melnikov. 2011. The WebSocket Protocol. *RFC* 6455 (2011), 1–71. https://doi.org/10.17487/RFC6455

[12] Roy Thomas Fielding. 2000. *Architectural styles and the design of network-based software architectures. 2000*. Ph.D. Dissertation. University of California, Irvine. https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf

[13] James R Groff, Paul N Weinberg, et al. 2002. *SQL: the complete reference*. Vol. 2. McGraw-Hill/Osborne.

[14] Olaf Hartig and Jorge Pérez. 2017. An Initial Analysis of Facebook's GraphQL Language. In *Proceedings of the 11th Alberto Mendelzon International Workshop on Foundations of Data Management and the Web, Montevideo, Uruguay, June 7-9, 2017. (CEUR Workshop Proceedings)*, Juan L. Reutter and Divesh Srivastava (Eds.), Vol. 1912. CEUR-WS.org. http://ceur-ws.org/Vol-1912/paper11.pdf

[15] Patrick J Hayes. 2009. BLOGIC or Now What's in a Link. In *Keynote presented at the 8th International Semantic Web Conference, October, Washington, DC http://videolectures. net/iswc09_hayes_blogic*. http://videolectures.net/iswc09_hayes_blogic/

[16] Urs Hunkeler, Hong Linh Truong, and Andy J. Stanford-Clark. 2008. MQTT-S - A publish/subscribe protocol for Wireless Sensor Networks. In *Proceedings of the Third International Conference on COMmunication System softWAre and MiddlewaRE (COMSWARE 2008), January 5-10, 2008, Bangalore, India*, Sunghyun Choi, Jim Kurose, and Krithi Ramamritham (Eds.). IEEE, 791–798. https://doi.org/10.1109/COMSWA.2008.4554519

[17] Luis Daniel Ibáñez, Hala Skaf-Molli, Pascal Molli, and Olivier Corby. 2013. Live linked data: synchronising semantic stores with commutative replicated data types. *IJMSO* 8, 2 (2013), 119–133. https://www.researchgate.net/profile/Pascal_Molli2/publication/264419928_Live_Linked_Data_Synchronizing_Semantic_Stores_with_Commutative_Replicated_Data_Types/links/53e8d65d0cf25d674ea87057/Live-Linked-Data-Synchronizing-Semantic-Stores-with-Commutative-Replicated-Data-Types.pdf

[18] Luis Daniel Ibáñez, Hala Skaf-Molli, Pascal Molli, and Olivier Corby. 2014. Col-Graph: Towards Writable and Scalable Linked Open Data, See [22], 325–340. https://doi.org/10.1007/978-3-319-11964-9_21

[19] Leslie Lamport. 1978. Time, clocks, and the ordering of events in a distributed system. *Commun. ACM* 21, 7 (1978), 558–565.

[20] Markus Lanthaler and Christian Guetl. 2013. Hydra: A Vocabulary for Hypermedia-Driven Web APIs. In *Proceedings of the WWW2013 Workshop on Linked Data on the Web, Rio de Janeiro, Brazil, 14 May, 2013 (CEUR Workshop Proceedings)*, Christian Bizer, Tom Heath, Tim Berners-Lee, Michael Hausenblas, and Sören Auer (Eds.), Vol. 996. CEUR-WS.org. http://ceur-ws.org/Vol-996/papers/ldow2013-paper-03.pdf

[21] Alejandro Mallea, Marcelo Arenas, Aidan Hogan, and Axel Polleres. 2011. On Blank Nodes. In *The Semantic Web - ISWC 2011 - 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part I (Lecture Notes in Computer Science)*, Lora Aroyo, Chris Welty, Harith Alani, Jamie Taylor, Abraham Bernstein, Lalana Kagal, Natasha Fridman Noy, and Eva Blomqvist (Eds.), Vol. 7031. Springer, 421–437. https://doi.org/10.1007/978-3-642-25073-6_27

[22] Peter Mika, Tania Tudorache, Abraham Bernstein, Chris Welty, Craig A. Knoblock, Denny Vrandecic, Paul T. Groth, Natasha F. Noy, Krzysztof Janowicz, and Carole A. Goble (Eds.). 2014. *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I.* Lecture Notes in Computer Science, Vol. 8796. Springer. https://doi.org/10.1007/978-3-319-11964-9

[23] Brice Nédelec, Pascal Molli, Achour Mostefaoui, and Emmanuel Desmontils. 2013. LSEQ: an adaptive structure for sequences in distributed collaborative editing. In *Proceedings of the 2013 ACM symposium on Document engineering*. ACM, 37–46. https://hal.archives-ouvertes.fr/hal-00921633

[24] Chimezie Ogbuji. 2013. *SPARQL 1.1 Graph Store HTTP Protocol*. W3C Recommendation. World Wide Web Consortium (W3C). http://www.w3.org/TR/2013/REC-sparql11-http-rdf-update-20130321/

[25] Marc Shapiro, Nuno Preguiça, Carlos Baquero, and Marek Zawirski. 2011. *A comprehensive study of convergent and commutative replicated data types*. Ph.D. Dissertation. Inria–Centre Paris-Rocquencourt; INRIA. https://hal.inria.fr/inria-00555588/document

[26] Marc Shapiro, Nuno Preguiça, Carlos Baquero, and Marek Zawirski. 2011. *Conflict-free Replicated Data Types*. Ph.D. Dissertation. INRIA. https://hal.inria.fr/inria-00609399/document

[27] Marc Shapiro, Nuno M. Preguiça, Carlos Baquero, and Marek Zawirski. 2011. Conflict-Free Replicated Data Types. https://hal.inria.fr/inria-00609399v1/document. In *Stabilization, Safety, and Security of Distributed Systems - 13th International Symposium, SSS 2011, Grenoble, France, October 10-12, 2011. Proceedings (Lecture Notes in Computer Science)*, Xavier Défago, Franck Petit, and Vincent Villain (Eds.), Vol. 6976. Springer, 386–400. https://doi.org/10.1007/978-3-642-24550-3_29

[28] Richard T. Snodgrass and Ilsoo Ahn. 1985. A Taxonomy of Time in Databases. In *Proceedings of the 1985 ACM SIGMOD International Conference on Management of Data, Austin, Texas, USA, May 28-31, 1985.*, Shamkant B. Navathe (Ed.). ACM Press, 236–246. https://doi.org/10.1145/318898.318921

[29] Ruben Verborgh, Olaf Hartig, Ben De Meester, Gerald Haesendonck, Laurens De Vocht, Miel Vander Sande, Richard Cyganiak, Pieter Colpaert, Erik Mannens, and Rik Van de Walle. 2014. Querying Datasets on the Web with High Availability, See [22], 180–196. https://doi.org/10.1007/978-3-319-11964-9_12

[30] Ruben Verborgh, Miel Vander Sande, Olaf Hartig, Joachim Van Herwegen, Laurens De Vocht, Ben De Meester, Gerald Haesendonck, and Pieter Colpaert. 2016. Triple Pattern Fragments: A low-cost knowledge graph interface for the Web. *J. Web Semant.* 37-38 (2016), 184–206. https://doi.org/10.1016/j.websem.2016.03.003

[31] Stephane Weiss, Pascal Urso, and Pascal Molli. 2010. Logoot-undo: Distributed collaborative editing system on p2p networks. *IEEE Transactions on Parallel and Distributed Systems* 21, 8 (2010), 1162–1174.

[32] Hafed Zarzour and Mokhtar Sellami. 2012. B-Set: A synchronization method for distributed semantic stores. In *2012 IEEE International Conference on Complex Systems (ICCS)*. IEEE, 1–6. https://www.researchgate.net/profile/Hafed_Zarzour/publication/261275016_B-Set_A_synchronization_method_for_distributed_semantic_stores/links/5559176608ae6943a876afb0/B-Set-A-synchronization-method-for-distributed-semantic-stores.pdf