Devising Affordable and Functional Linked Data Archives

Ruben Verborgh
Ghent University – imec
IDLab
Ghent, Belgium

Miel Vander Sande
Ghent University – imec
IDLab
Ghent, Belgium

Harihar Shankar
Los Alamos National
Laboratory
Los Alamos, USA

Luda Balakireva
Los Alamos National
Laboratory
Los Alamos, USA

Herbert Van de Sompel
Los Alamos National
Laboratory
Los Alamos, USA

ABSTRACT

Linked Data has become an integral part of the Web. Like any other web resource, Linked Data changes over time. Typically, only the most recent version of a Linked Data set can be accessed via Subject-URIs and queried by means of SPARQL. Sometimes, select archived versions are made available for bulk download. This archive access approach is cheap for the publisher but, unfortunately, very expensive for consumers. The entire data dump must be downloaded and ingested into infrastructure that supports subject-URI and/or SPARQL access. Comparing data across different archived versions is even harder. To address this publisher–consumer imbalance, we propose a solution for publication of archived Linked Data that is affordable for publishers and functional for consumers. It consists of two components: a static storage approach for archived Linked Data that exposes a lightweight RDF interface, and the subsequent extension of that interface to versioned data.

The Linked Data Fragments (LDF) conceptual framework [2] allows for an analysis of different existing and possible new interfaces to publish RDF data on the Web. Based on insights gained from this framework, we previously designed an interface called Triple Pattern Fragments (TPF) [2], which provides access to data by means of `?s ?p ?o` Query-URIs. In contrast to SPARQL endpoints, TPF servers cannot evaluate SPARQL queries, making the maximal per-request processing cost much more limited. To obtain answers to complex SPARQL queries, clients need to execute SPARQL queries locally, only using the server to retrieve triple pattern data. While this makes query evaluation slower and more bandwidth-intensive, the total server cost remains lower. Additionally, because request patterns are more limited, responses are more likely to be cached.

Since TPF is a regular HTTP interface, it can be augmented with support for datetime negotiation as defined in the Memento protocol [1]. This allows clients to use the accept-datetime HTTP request header to ask for the responses to `?s ?p ?o` Query-URIs and Subject-URIs as they were at a given time in the past. The TPF server replies with the temporally best version, using the `memento-datetime` HTTP response headers to indicate the archival datetime of the returned representations. As a result, if a client wants to obtain the result of a SPARQL query as it was at a previous point in time, it needs to break down the SPARQL query into the necessary `?s ?p ?o` Query-URIs and simply request each using the accept-datetime HTTP request header.

In the Linked Data Archive, each temporal version of a Linked Data set is stored according to the HDT (Header Dictionary Triples) format for binary representation of RDF data. HDT files are static, highly compressed, and provide fast triple pattern lookups and estimate result counts. The latter two features are essential for the TPF interface, as they allow breaking SPARQL queries down into multiple `?s ?p ?o` Query-URIs. The TPF/HDT combination is attractive for a publisher of archived Linked Data because of its static nature, minimal storage requirement, and constrained query support. It provides attractive functionality for a consumer of the archive through datetime negotiation with Subject-URIs and `?s ?p ?o` Query-URIs. Resolving temporal SPARQL queries is possible yet more expensive for the client. But it is far cheaper than downloading one or more data dumps and uploading them into infrastructure that natively supports SPARQL queries.

In this presentation, we included a brief refresher of the Memento protocol as it applies to Linked data, and we covered Linked Data Fragments, Triple Pattern Fragments, and the HDT storage format in more detail. We introduced the DBpedia Archive that contains each version of the DBpedia dataset and, in total, consists of over 5 billion RDF triples. To demonstrate the vast potential of the solution, we showed how queries can be executed live over multiple datasets, using examples from the digital libraries domain that were impossible over live Web data before.

REFERENCES

[1] H. Van de Sompel, M. Nelson, and R. Sanderson. HTTP framework for time-based access to resource states – Memento. Technical Report RFC 7089, Internet Engineering Task Force, 2013. https://tools.ietf.org/rfc/rfc7089.

[2] R. Verborgh, M. Vander Sande, O. Hartig, J. Van Herwegen, L. De Vocht, B. De Meester, G. Haesendonck, and P. Colpaert. Triple Pattern Fragments: a low-cost knowledge graph interface for the Web. *Journal of Web Semantics*, 37–38:184–206, Mar. 2016.