Paths towards the Sustainable Consumption of Semantic Data on the Web

Aidan Hogan and Claudio Gutierrez

Department of Computer Science, Universidad de Chile

Abstract. Based on recent results, we argue that the right method for Web clients to access relevant information from Linked Datasets has not yet been found. We propose that something is needed between (i) Linked Data dereferencing, which is simple and reliable but too vaguely defined; (ii) data dumps, which are simple and reliable but too coarse-grained, and (iii) SPARQL querying, which is powerful and fine-grained but too unreliable. We argue that new protocols and query languages need to be investigated and define eight desiderata that an access method should meet in order to be considered sustainable for a mature *Web of Data*.

A reader familiar with the Linked Data literature will already know about the hundreds of Linked Datasets published on the Web: they are mentioned in the introduction to almost every research paper written in the area. They will likewise be familiar with the ubiquitous LOD Cloud, often cited as the realisation of a new emerging "Web of Data". They may even have read about hundreds of public SPARQL endpoints on the Web waiting to be queried or to be federated.[1]

What the literature may not mention is that the LOD cloud has not been updated in 2.5 years.[2] It may not mention that, in total, one new LOD dataset has been added to the official DATAHUB catalogue in the past twelve months.[3] It may not mention that many legacy Linked Datasets are not accessible for a variety of reasons [4, §4.3] or that half the endpoints listed in DATAHUB are offline or that only one-third are available more than 99% of the time [1].[4]

We take these observations as warning signs: signs that it is time for the community to look critically at current Linked Data practices and research directions. In this short paper, we focus (for now) on core data access methods.

Review of access methods: Linked Data clients currently have three main options for accessing data: (i) dereferencing, where a client performs lookups on individual URIs; (ii) data dumps, where a client downloads an archive of the Linked Dataset; and (iii) SPARQL endpoints, where a client issues (potentially complex) SPARQL queries. But as we now discuss – and as we have previously argued elsewhere [6] – each of these access methods has proven flawed in different ways.

With respect to dereferencing, in previous work we showed that 13 of the 25 largest reported Linked Datasets in DATAHUB provided no ($\frac{11}{25}$) or fewer than 1,000

This works was supported by the Millennium Nucleus Center for Semantic Web Research under Grant NC120004.

[1] ... papers/talks prepared by the current authors included.

[2] http://lod-cloud.net/; 2014/04/11.

[3] http://datahub.io/group/activity/lodcloud/0; as of 2014/04/11.

[4] We note that SPARQL and Linked Data are separate notions: one can exist without the other. Here we critique SPARQL only as a public Linked Data access method.

($\frac{2}{25}$) triples for the BTC '11 crawl due to accessibility issues [4, §4.3]. In other work, we showed that publishers often provide incomplete, ad hoc information in dereferenced documents: e.g., we found that on average, publishers include ∼83.6% of local triples where a URI appears as subject in its resp. dereferenced document, but only ∼55.2% of local triples where it appears as object, and that only ∼32.8% of local dereferenceable resources are assigned a human readable label [3].

Downloading the data-dump is coarse and involves accessing a lot of irrelevant data. The problem can be mitigated by compression: in previous works, we proposed HDT, which can compress data-dumps by a factor of 15× while offering lookup functionality over the archive, thus tackling problems with bandwidth and helping clients to extract relevant data offline [2]. However, the full dataset still needs to be transferred, which is wasteful for small requests, and updates are difficult to mirror.

For SPARQL endpoints, evaluating even a SPARQL 1.0 query is PSPACE-complete [5] and query-planning costs become less reliable as queries grow more complex. Relatedly, we previously showed that many public SPARQL endpoints suffer from various issues, harming their usability: for example, we found that of the 427 public endpoints surveyed, only 32% had an availability (i.e., "uptime") falling into 99–100% and that only 13.3% could return more than 100,000 results upon request (due to the popular use of result-size thresholds) [1].

Each access method has, in practice, exhibited issues that undermine its sustainability; furthermore, these three access methods cannot be readily combined.

Desiderata: To help define a path forward, we propose a list of eight desiderata for *sustainable* data-access methods, divided into four different goals, as follows:

Standardised: The first two criteria refer to the agreement that exists between client(s) and server(s):

1. ACCESSIBLE: a software agent can access data through a uniform protocol without location-specific logic. This holds for SPARQL and for dereferencing, but not for dumps (which vary in formats, compression, access, etc.).
2. WELL-DEFINED: given a query Q and a dataset D, both client and server can precisely agree on what the response $R(Q, D)$ should be. This holds for SPARQL, but not for dumps (which may vary in their completeness) or dereferencing (where dereferenced content varies from server to server).

Bandwidth conservation: The second two criteria aim to minimise wasting bandwidth in transferring irrelevant data to a client:

3. GRANULAR: the query language allows the client to specify sufficient information in Q to avoid transferring irrelevant data. This is true for SPARQL, but not for dereferencing (e.g., to get the capitals of countries, full descriptions for each country must be dereferenced) or dumps.
4. PAGINATION: a large response $R(Q, D)$ can be served in chunks until the client is satisfied. This is not true for dereferencing or dumps and is costly in SPARQL (which relies on `ORDER BY` or vendor-specific heuristics).

Server efficiency: The third two criteria aim to make the access method sustainable for the server to host:

5. CACHEABLE: common requests are amenable to caching techniques/answerable by direct lookup; previously computed responses can be easily re-used. This is true for dereferencing and dumps but is prohibitive for SPARQL.

	Accessible	Well-defined	Granular	Pagination	Cacheable	Costable	Tranparent	Robust
Dereferencing	✓				✓	✓	✓	✓
Dumps					✓	✓		✓
SPARQL endpoints	✓	✓	✓	∼				

Table 1: Desiderata (not) met by current data access methods

6. Costable: the server can efficiently and accurately predict the processing/-transport cost of serving $R(Q, D)$, fostering quality of service. This is true for dereferencing and dumps, but not for general SPARQL queries.

Client usability: The final two criteria refer to the needs of clients:

7. Transparent: the client can determine if a dataset D is relevant for their needs and if a service is sufficiently reliable to serve a given purpose. This is (arguably) true for dereferencing since a client knows the topic of a page; however, a client will not know *a priori* what content (or quality of service) is provided behind a SPARQL endpoint or a dump.
8. Robust: the access method can gracefully handle any type of valid request from multitudinous clients; if exceptions occur, they are clearly identified and accounted for by quality of service. This is true for dereferencing and dumps but not for SPARQL (which may, e.g., fail or silently return a partial response).

Table 1 summarises these desiderata for the three state-of-the-art Linked Data access methods: clearly the right combination of protocol and query language has not yet been found. New proposals for access methods that satisfy more of these desiderata – perhaps like "Linked Data Fragments" [7] – need to be investigated in order to meet the expectations of future applications and increased traffic. Otherwise, if we stick with current trends, the Web of Data may continue to stagnate in its current local maximum: a perpetual experimental phase; a nice idea.

References

1. C. B. Aranda, A. Hogan, J. Umbrich, and P.-Y. Vandenbussche. SPARQL Web-Querying Infrastructure: Ready for Action? In *ISWC*, pages 277–293, 2013.
2. J. Fernández, M. Martínez-Prieto, C. Gutiérrez, A. Polleres, and M. Arias. Binary RDF representation for publication and exchange (HDT). *JWS*, 19:22–41, 2013.
3. A. Hogan, J. Umbrich, A. Harth, R. Cyganiak, A. Polleres, and S. Decker. An empirical survey of Linked Data conformance. *JWS*, 14:14–44, 2012.
4. T. Käfer, J. Umbrich, A. Hogan, and A. Polleres. Towards a Dynamic Linked Data Observatory. In *LDOW*, 2012.
5. J. Pérez, M. Arenas, and C. Gutierrez. Semantics and complexity of SPARQL. *ACM Trans. Database Syst.*, 34(3), 2009.
6. J. Umbrich, C. Gutierrez, A. Hogan, M. Karnstedt, and J. X. Parreira. Eight fallacies when querying the Web of Data. In *DESWEB*, 2013.
7. R. Verborgh, M. Vander Sande, P. Colpaert, S. Coppens, E. Mannens, and R. Van de Walle. Web-scale querying through Linked Data Fragments. In *LDOW*, 2014.