Access Logs Don't Lie: Towards Traffic Analytics for Linked Data Publishers

Luca Costabello[1], Pierre-Yves Vandenbussche[1], Gofran Shukair[1],
Corine Deliot[2], and Neil Wilson[2]

[1] Fujitsu Ireland Ltd., Ireland
firstname.lastname@ie.fujitsu.com
[2] British Library, United Kingdom
firstname.lastname@bl.uk

Abstract. Considerable investment in RDF publishing has recently led to the birth of the Web of Data. But is this investment worth it? Are publishers aware of how their linked datasets traffic looks like? We propose an access analytics platform for linked datasets. The system mines traffic insights from the logs of registered RDF publishers and extracts Linked Data-specific metrics not available in traditional web analytics tools. We present a demo instance showing one month (December 2014) of real traffic to the British National Bibliography RDF dataset.

1 Introduction

We believe Linked Data publishers have limited awareness of how datasets are accessed by visitors. While some works describe specific access metrics for linked datasets [1,2], no comprehensive analytics tool for Linked Data publishers has ever been proposed, and in most cases publishers have no choice but to manually browse through records stored in server access logs. Applications for analysing traditional websites traffic exist, but none takes into account the specificities of Linked Data: Google Analytics[1] and other popular web analytics platforms[2] (e.g. Open Web Analytics, PIWIK[3]) are not designed for linked datasets. For example, existing systems do not offer insights on SPARQL queries, or properly interpret `303` URIs. Besides, to the best of our knowledge, there are no tools that detect Linked Data visitors sessions, or that help identifying workload peaks of SPARQL endpoints.

This has two consequences: first, publishers struggle to justify Linked Data investment with management. Second, they miss out technical benefits: For instance, limited awareness of traffic spikes prevents predicting peaks during real-world events, and hinders the identification of visitors that overload triplestores with repeated SPARQL queries.

[1] http://analytics.google.com

[2] https://en.wikipedia.org/wiki/List_of_web_analytics_software

[3] http://piwik.org — http://www.openwebanalytics.com

2 Our Contribution

We present an hosted analytics platform for linked datasets. The system mines the logs of registered Linked Data publishers and extracts traffic insights. The analytics system is designed for RDF data stores with or without SPARQL engine, and supports load-balancing scenarios. The online demo[4] shows one month of traffic insights of the The British National Bibliography (BNB) dataset[5]. The system can easily accommodate any Linked Data publisher and only requires the modification of the log parser to meet publisher's log syntax.

The system offers Linked Data-specific features which are currently not supported by classic web analytics tools (e.g. SPARQL-specific statistics). We do not track clients, thus preserving visitors privacy. The system supports Linked Data HTTP dereferencing with HTTP `303` patterns, and filters out search engines and robots activity. It also detects linked data visitor sessions with an unsupervised learning algorithm. To better identify workload peaks of a SPARQL endpoint, supervised learning is adopted to label SPARQL queries as *heavy* or *light*, according to SPARQL syntactic features.

System Overview. Our traffic analytics platform is organised in the following components (Figure 1):

Extract-Transform-Load (ETL) Unit. On a daily basis, for registered publishers, the *Log Ingestion* sub-component fetches and parses access logs from one or more linked dataset servers (see Figure 2 for an example). Records are filtered to remove robots and search engine crawlers noise.

Metrics Extraction Unit. Extracts traffic metrics from access logs.

Data Warehouse and MOLAP Unit. Traffic metrics are stored in a data warehouse equipped with an SQL-compliant MOLAP[6] unit that answers queries with sub-second latency.

Web user interface. The front end queries the RESTful APIs exposed by the MOLAP Unit, and generates a web UI that shows traffic metrics filtered by date, user agent type, and access protocol (Figure 3). The user interface runs on Node.js, and charts are based on amCharts[7].

Metrics. We support three groups of traffic metrics:

Content Metrics. How many times RDF resources have been accessed. We support Linked Data dual access protocol; this means that the system counts how many times an RDF resource is dereferenced with HTTP operations, but also how many times its URI is included in SPARQL queries[8]. Unlike existing tools, we support `303` URIs[9], thus counting each HTTP `303` pattern as a single

[4] `http://52.49.205.156/analytics/`

[5] Released as Linked Open Data in July 2011, the dataset offers SPARQL and HTTP access to almost 100 million statements about books and serials. It is available at `http://bnb.data.bl.uk`

[6] Multidimensional Online Analytical Processing

[7] `https://www.amcharts.com`

[8] This is a lower bound estimation. Access logs do not contain SPARQL result sets.

[9] `https://www.w3.org/TR/cooluris`

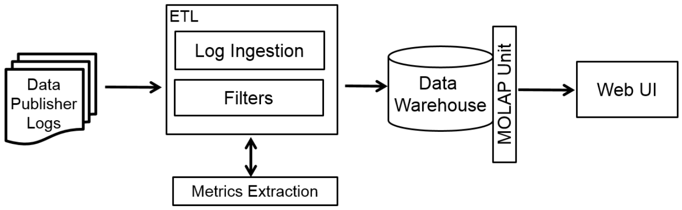
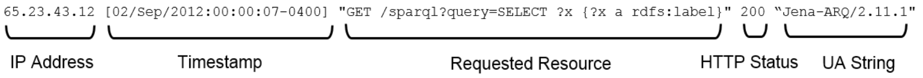


Fig. 1: Architecture of the analytics platform for Linked Data publishers



```
65.23.43.12 [02/Sep/2012:00:00:07-0400] "GET /sparql?query=SELECT ?x {?x a rdfs:label}" 200 “Jena-ARQ/2.11.1"
```

Fig. 2: Linked Dataset access record (Apache Commons Logfile Format[10])

request. We also provide aggregates by *family* of RDF resource: *instances* (URIs accessed either in HTTP operations or included in SPARQL queries), *classes* (URIs used as RDFS/OWL classes in SPARQL queries, objects of `rdf:type`), *properties* (URIs used as predicates in SPARQL queries), *graphs* (URIs used as graphs in SPARQL queries - `FROM`/`FROM NAMED`, `USING`/`USING NAMED`, `GRAPH`).

Audience Metrics. Besides traditional information about visitors (e.g. location, network provider, user agent type), these measures include details of visitor sessions (duration, size, depth, bounce rate), which we identify with unsupervised hierarchical agglomerative clustering (HAC) proposed by [3].

Protocol Metrics. Information about the data access protocols used by visitors. It includes a breakdown of requests by protocol (HTTP lookups vs SPARQL queries), and various SPARQL-specific metrics: the count of malformed queries, queries by verb, the count of *light* and *heavy* SPARQL queries (obtained with an off-the-shelf supervised binary classifier trained on a super set of SPARQL syntactic features listed in [4]).

3 Conclusions and Future Perspectives

Our analytics platform relieves Linked Data publishers from time-consuming log mining, and unlike other popular web analytics platforms, supports linked data-specific traffic metrics. Traffic patterns knowledge helps gauging the popularity of a dataset: for example, awareness of decreasing user retention might prompt for better promotion (e.g. hackatons, spreading the word on community mailing lists, etc.). Likewise, if portions of a dataset are never accessed, perhaps better data documentation is required.

Note that the extracted metrics should be considered as a lower-bound estimation: because we do not track visitors, we have a partial view on the communi-

[10] `https://httpd.apache.org/docs/trunk/logs.html#common`

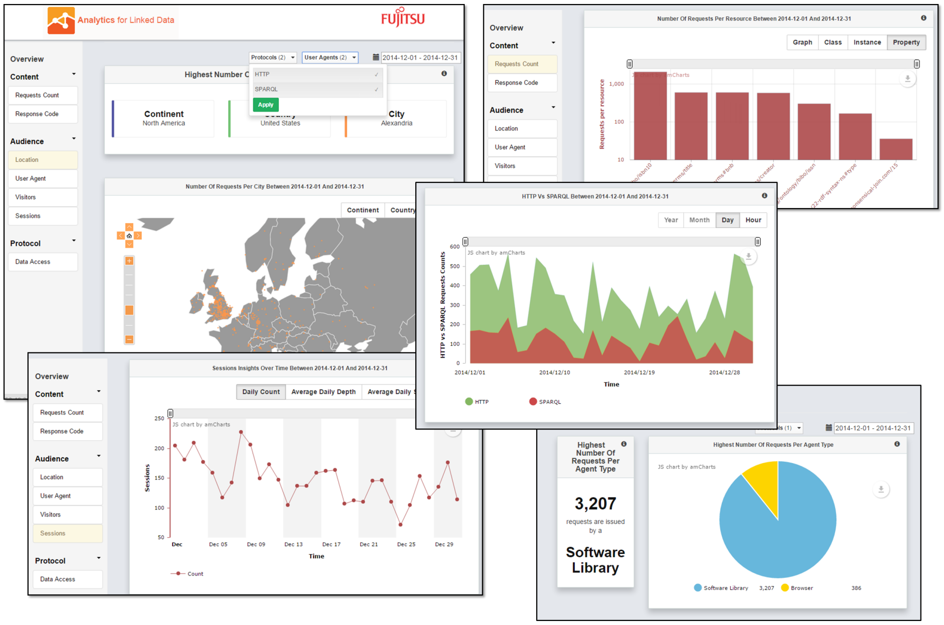


Fig. 3: Screenshots from the web UI

cation with the data store, and we cannot circumvent intermediate components between visitors and datasets (e.g. caches, proxy servers, or NAT). Besides, visitors might fake user agent strings or HTTP referrer, thus leading to client identification mistakes.

We will add new metrics in future extensions, such as finer-grained SPARQL insights (e.g. useful to fine-tune SPARQL engine caches). Users suggest upgrading the web interface with secondary dimensions capabilities, to improve reporting. Real time monitoring is also part of the future work roadmap.

Acknowledgments. This work has been supported by the TOMOE project funded by Fujitsu Laboratories Limited in collaboration with Insight Centre for Data Analytics at National University of Ireland Galway.

References

1. D. Fasel and D. Zumstein. A fuzzy data warehouse approach for web analytics. In *Procs of WSKS*. Springer, 2009.
2. K. Möller, M. Hausenblas, R. Cyganiak, and S. Handschuh. Learning from linked open data usage: Patterns & metrics. 2010.
3. G. C. Murray, J. Lin, and A. Chowdhury. Identification of user sessions with hierarchical agglomerative clustering. *ASIS&T*, 43(1):1–9, 2006.
4. F. Picalausa and S. Vansummeren. What are real SPARQL queries like? In *Procs of SWIM*, page 7. ACM, 2011.