Linked Open Data Publication Strategies: Application in Networking Performance Measurement Data

Renan F. Souza[1], Les Cottrell[2], Bebo White[2], Maria L. Campos[1], Marta Mattoso[1],
[1]Federal University of Rio de Janeiro, Brazil
[2]SLAC National Accelerator Laboratory
renanfs@cos.ufrj.br, cottrell@slac.stanford.edu, bebo@slac.stanford.edu, mluiza@ppgi.ufrj.br, marta@cos.ufrj.br

Abstract

Most of the data published on the web is unstructured or does not follow a standard. This makes it harder to retrieve and interchange information between different data sources. This work uses Linked Open Data (LOD) technologies and applies them in a scenario that deals with a large amount of computer network measurement data. The goal is to make the data more structured, hence easier to be retrieved, analyzed, and more interoperable. We discuss the challenges of processing large amount of data to: transform it into a standard format (RDF); link it to other data sources; and analyze and visualize the transformed data. Moreover, an ontology that aims to minimize the number of triples is proposed and a discussion of how ontologies may impact performance is presented. In addition, both the advantages of having the data in RDF format and the obstacles that the LOD community still faces are analyzed within the use cases on the scenario of the project.

Keywords: Linked Open Data; Semantic Web; LOD Networking Measurement; LOD Publication Strategies; PingER LOD.

1. Introduction

Most of the data on the web today is unstructured and does not adhere to any standard format. As a consequence, information retrieval is inefficient and data exchange is very limited. As people have become dependent on the Web and, especially, on search engines, hence searching technologies need to be enhanced. To illustrate that, David Siegel points out that approximately 25% of people's working hours are taken just for searching the web [1]. Many search attempts return undesired results (usually out of context) or, worse, no results at all. It is believed that searching could be powerfully improved if the web content were semantically linked, using a common open standard to exchange information, enabling interoperable data mashups. This is the main idea behind the Linked Data efforts on the Semantic Web [2]. However, both the technologies and the recognition of their benefits by the web developers' community are still in a very early stage. Furthermore, the web itself contains a large amount of data and, specifically, it is common to deal with big datasets when dealing with semantic web due to the nature1 of RDF (Resource Description Framework) *triplified* data [3].

This work applies semantic web concepts and existing technologies to a specific large dataset of worldwide Internet measurements. The approach used is based on a domain analysis of the scenario which these technologies will be applied to; ontology engineering focusing on an evaluation for reuse and impact on performance when querying large datasets; and a concurrent solution to *triplify* a large amount of data, linking the triples to other existing data sources in the LOD Cloud [4].

Regarding related work, it is known that ontologies should be reused in the context of the semantic web [5]. However, there is not an approach yet to analyze if an ontology should be reused or not. This paper introduces a proposal of such approach and applies part of it to the studied scenario. Additionally, there are *triplification* tools and frameworks that would fit the domain, such as ETL4LOD [6] and Hiroyuki *et. al´* work [7]. Nevertheless, they do not provide solutions to work specifically with large datasets. Thus, the *triplification* process of a huge amount of data would take an impractical amount of time. In subsection 4.2, it will be shown that our solution relies on distributing and parallelizing single tasks, aiming to transform big CSV data into RDF format in a relatively reasonable period of time.

Furthermore, this work is based on a proposed methodology that systematically states the main phases of a LOD publication. This methodology determines the organization of this paper: Section 2 introduces the domain analysis of the real scenario studied; Section 3 shows the ontology engineering of the domain; Section 4 presents the process of *triplification* of the large amount of data; Section 5 shows the advantages of having linked open data in a structured standard format, with graphs and dashboards that support data analysis, and combining original data with other existing data; and Section 6 presents the conclusion, which synthetizes the main advantages brought about by this work to the functionality and usefulness of the dataset.

2. Domain Analysis

Neighbors (*apud* [5]) defines Domain Analysis as "an attempt of identifying objects, operations, and relations among what experts of a determined domain perceive as important". The goal is to model a problem in a way that makes it closer to the reality, increasing the chances of fulfilling the goals of the project. The more information we can gather and understand from the domain, the more realistic the model will be. This also decreases the need for future software changes, since it was developed focused and oriented to the domain of the problem. It is important to note that software changes when dealing with large datasets can be very expensive. The ontology (that is, how the entities in a real-world problem are modeled so a computer can understand,) is ideally based on the domain analysis. The domain analysis can be further separated into (1) domain understanding, (2) data selection, and (3) domain modeling [5].

2.1 Domain Understanding

The dataset used in this work is generated by the PingER (Ping End-to-end Reporting) project[2], which monitors performance of Internet links around the world [8]. It was developed by a team of collaborators from Universities and National Laboratories in North America, Europe, Pakistan and Malaysia. PingER project is mainly concentrated at the SLAC National Accelerator Laboratory, which is operated by the Stanford University for the U.S. Department of Energy Office of Science.

Since 1998, PingER has stored data about the quality of Internet links both hourly and daily, measuring more than 10 different metrics. The project describes measurements from around 80 monitor nodes to

[1] RDF triples have a very granular nature. Granularity indicates how detailed an information is. See http://wisegeek.com/what-is-granularity.htm

[2] http://www-iepm.slac.stanford.edu/pinger/

This work was supported in part by the Department of Energy contract DE-AC02-76SF0051.

over 800 monitored nodes (more than 8000 pairs of nodes – not all monitor nodes monitor all monitored nodes), in more than 160 countries.

The result is a non-trivial amount of data that is available for analysis. This data is stored in millions of flat CSV files, which are organized using meaningful file names. As such, it can, at best, be described as "semi-structured" data. PingER has not, traditionally, been stored in an existing database management system. Access to this data is accomplished via the Pingtable[3] application.

Despite this file structure organization that makes it possible to access the right data, it is far from a standard database management system (DBMS) with well-known features and benefits [9]. Incorporating such features would significantly improve the retrieval and management of the data thereby making it much easier to build more complex structured queries to retrieve very specific data and support more informative graphs, reports, and dashboards.

Furthermore, to interoperate and interchange PingER data with other existing data sources is not a very simple task. Most traditional DBMSs do not provide this functionality satisfactorily. Since the data could be highly useful and applied to many different situations such as economical, geographical, and seasonal events [8], it would therefore be desirable to make the data easily interoperable to any other kind of data source. If PingER data were available in a standard and widely used format, there could be more joint exploration of the many possibilities that the data could offer, enabling combination with other kinds of sources, and bringing more diversity to its usage and analysis. Besides, by utilizing a standard common and open format, more people would consume the data, enabling possible specific uses that have not been anticipated.

Therefore, the target problem of this project, which will be hereafter called PingER Linked Open Data (LOD), can be stated as follows: PingER data is not stored in a conveniently accessible way, hence the ease of production of reports, smart visualizations, and dashboards could be significantly improved, especially when the data involves different data sources. Moreover, the data could be published in an open standard format to enable wider consumption. The project draws on the Semantic Web and LOD strategies and techniques [10] to publish the data according to community and World Wide Web Consortium recommendations[4].

2.2 Data Selection

Once we have defined the problem, the approach to the solution begins with selecting part of the huge amount of PingER data that will be considered to be converted and stored in RDF format. The selected data was:

- Regarding the network nodes scope: measurement between all pairs of nodes considered by PingER;
- Regarding the geographic level of detail: PingER stores not only node to node data, but also site to site, country to country, region to region, etc. Although PingER LOD considers only node to node data, the geographic hierarchy of the nodes is very well defined so one can easily aggregate the data by country, continent, etc.;
- Regarding the time level of detail: the smallest time grain considered by PingER is one *hour*, however, the smallest grain considered by PingER LOD is one *day*. In this initial analysis, we determined that processing the entire PingER dataset dating back to 1998 would require a prohibitive amount of computation time. Instead, the last 60 days of data is continuously being inserted into the PingER LOD triples store, beginning in August 2013.
- Regarding the types of network metrics: Pingtable utilizes 16 metrics, but PingER LOD considers only the 11 most important ones: Mean Opinion Score, Directivity, Average Round Trip Time, Conditional Loss Probability, Duplicate Packets, Inter Packet Delay Variation, Minimum Round Trip Delay, Packet Loss, TCP Throughput, Unreachability, and Zero Packet Loss Frequency. These metrics are defined elsewhere[5].
- Regarding the network packet size: Pingtable is designed to accommodate packet sizes of 100 and 1000 bytes. It was determined that considering both sizes would take approximately double of the time to load and just one of them would be sufficient to satisfy the initial goals of the project. Thus, PingER LOD contemplates only packet sizes of 100 bytes.

2.3 Data Modeling and Data Quality

For semantic web projects, data modeling is especially important and should be discussed in specific detail.

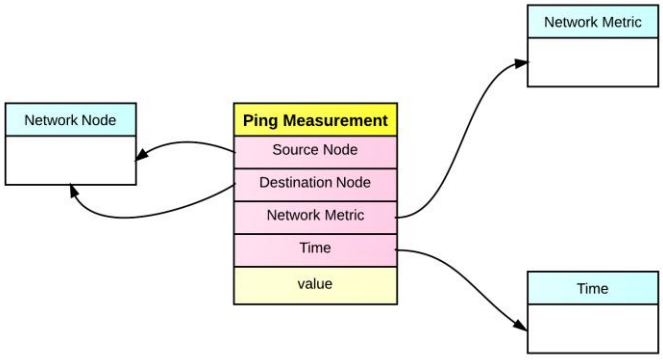

Figure 1 – Simple conceptual model of a PingER measurement, based on a star schema

For the domain analysis, though, a simple conceptual model was designed based on the PingER data characteristics: each measurement is basically defined by a ping sent from a source (or monitor) node to a destination (or monitored) node, sent in a determined time, and related to a specific network metric, as shown in Figure 1.

Additionally, the data is accumulated in a historical database over the years of the project and can be used to support decision making. Data with these characteristics can be modeled using a well-known and studied data model: star schema [11], upon which the designed conceptual model was based.

We assumed that the `Measurement` entity would be the most critical one, i.e., the main component of the dataset. Therefore, it should be considered more carefully than the other entities of the domain.

In addition to the conceptual model of the main characteristics of the domain, a glossary was written[6] to help to understand the concepts of the domain and to serve as basis to build a controlled vocabulary [12].

[3] http://www-wanmon.slac.stanford.edu/cgi-wrap/pingtable.pl
[4] http://www.w3.org/standards/semanticweb/

[5] See http://www.slac.stanford.edu/comp/net/wan-mon/tutorial.html
[6] https://confluence.slac.stanford.edu/display/IEPM/PingER+LOD+Glossary

The quality of the data is also supposed to be, at least superficially, analyzed before the *triplification*. For the PingER LOD scenario, the data set is well controlled and complete. Missing data occurs when there was not any measurement for a given set of parameters. The data that would be published in LOD format was previously analyzed and treated, making its mining and retrieval easier.

3. Ontology Engineering

Upon completing the data model and sufficiently understanding the domain, we were able to begin the construction of the ontology applicable to the domain.

An ontology is used to formalize the knowledge of the domain, utilizing well-defined terms and the relationships between each of them. It should be understandable both to humans and to computers, in a way that machines can process and infer information through it [13]. In addition, ontologies establish levels of interoperability among semantic web applications [14], adding meanings, representation and expressivity over the layers of the current web [15].

The reuse concept is very important in ontology engineering. There are concepts and terms that are common over a wide range of domains and have been previously modeled. In such cases "reuse" is possible via a common vocabulary. More common examples include geographic and time concepts. In addition to those more common concepts, it is possible that there exists a working model that fits the specific concepts of the domain to be modeled. Therefore, it is important to search for existing solutions for the domain and carefully analyze them before building your own model. Just like the Domain Analysis, Ontology Engineering can save a lot of extra effort, eventually avoiding future changes, especially when dealing with a huge amount of data. Moreover, reusing common ontologies supports the idea of standardization, facilitating the interoperability within the LOD community. However, the reutilization (or not) must be carefully investigated, especially when it contemplates critical entities. We proposed a way to evaluate whether or not an ontology should be reused: analyzing its (a) semantic expressivity, (b) completeness in relation to the domain, and (c) impacts on query performance.

An ontology is (a) semantically expressive if it is able to clearly formalize – in a language which is understandable both to humans and machines – the knowledge of the reality of the domain, which was gathered in the domain analysis phase. An ontology is (b) complete in relation to the domain if it can totally represent the knowledge gathered in the domain analysis phase. An ontology (c) has low impacts on query performance when the model used to describe the knowledge considers using the minimum number of triples to represent a statement of the domain.

The MOMENT ontology, presented in the following subsection, was carefully studied, but for the scope of this paper, only an analysis of its impacts on query performance is further explained. Additionally, the Geonames Ontology [16] was reused to describe geographic concepts and the W3C Time ontology [17] was reused to describe time concepts in the PingER domain.

3.1 MOMENT Ontology

Considering the reuse idea, there are not many ontologies publically available for the network measurement domain, more specifically, that would fit the PingER domain. However, we searched and found a very useful ontology, which seemed to be at least close to being complete in relation to the PingER domain. For this reason, we started to further investigate it. The project "Monitoring and Measurement in the Next Generation Technologies" (MOMENT) [18], which produced a Measurement Ontology[7] for IP (MOI) traffic, that is an European Telecommunications Standards Institute (ETSI) Group Specification [19].

Figure 2 - Proposal of MOMENT ontology for the PingER domain

The MOMENT ontology is complex and generic in the way it contemplates the main characteristics referring to network measurement. This generality of the ontology enables it to be adapted to many different network measurement scenarios, including the PingER domain. Figure 2 represents a proposal of how PingER domain would be modeled only using the MOMENT ontology8.

However, since the ontology is so generic, the ontology fails in representing PingER reality as well as it would be if it were developed specifically to the domain. Additionally, the ontology presents some characteristics that make it harder or slower to process a large amount of data.

When measuring performance in semantic web applications, it is important to observe the number of triples in the triple database, especially when dealing with critical entities (like the `Measurement` entity, as seen in Section 2). It is still a challenge in the semantic web community to deal with huge amount of triples[9], hence it should be minimized. The ontology may significantly interfere with the above goal, and thus critical entities need to be carefully modeled within the ontology.

In the MOMENT Ontology case, instances of subclasses of the class `MeasurementData` have an attribute that defines its data value. Instances of `Measurement` do not have the measured data directly linked to them, but linked to an instance of a subclass of `MeasurementData`. An example in triples:

```
(measurement1,
 hasMeasurementData,  packetlossmeasurement1).
(packetlossmeasurement1,
PacketLossMeasurementValue,
numeric value) .
```

Where `measurement1` and `packetlossmeasurement1` are instances. It was stated before that the measurement entity is critical;

[7] To see MOMENT OWL files, go to https://svn.fp7-moment.eu/svn/moment/public/Ontology/
[8] https://confluence.slac.stanford.edu/display/IEPM/MOMENT+Ontology
[9] http://www.w3.org/wiki/LargeTripleStores

hence the model should consider minimizing its instances. In numbers, for each PingER measurement, at least 2 triples are needed to describe its value. A common specific scenario in the PingER project considers that each network measurement is minimally given by a pair of monitor and monitored nodes (approximately 8000 pairs), that occurred in the last 60 days, and that measured a determined metric (11 are considered). A quick calculation gives a number greater than 5 million (8000*60*11) of measurements. If 2 triples are needed to describe each value observed, there will be at least 10 million triples just for this simple scenario. A proposal to decrease the number of triples would be to simply link the data property of the observed value *directly* to the instance of the measurement.

Moreover, another analogous situation refers to subclasses of `NodeInformation`. The representation of the MOMENT ontology (Figure 2) shows that an instance of `NetworkNode` links to instances of subclasses of `NodeInformation`, which link to their data property to describe the value. In triples:

```
(networknode1, hasMeasurementData, nodeip1).
(nodeip1, hasNodeIPValue, string that contains
information about the node IP).
```

Where `networknode1` and `nodeip1` are instances of classes. Intuitively, the number of triples decreases when an instance of `NetworkNode` directly links to its specific information through a specific relationship (data property). For example:

```
(networknode1, hasNodeIP, string that contains
the node IP).
```

In addition to the idea of minimizing the number of triples, there is the fact that a ping measurement is defined (together with other parameters) by a single entity that represents a pair of nodes; not necessarily 2 entities, one for the source and the other for the destination. Instead of having 2 links for each measurement (one for each node), if we could model the measurement entity with only one link to define a pair of nodes, we would cut the number of measurement triples in half. Thus, it would be a significant reduction in the number of triples.

Still within the `NodeInformation` case, there is an issue regarding SPARQL query [20] processing. To retrieve information of the name of a node (i.e., the `NodeName`) of an instance of `NetworkNode` that is the source node of a measurement (a very common operation in the domain), the SPARQL query is more complex, both for understanding and for the query processer to execute it. In triples, part of the SPARQL query would be as following:

```
?Measurement :hasMeasurementData ?SourceIP .
?SourceIP :SourceIPValue ?SourceIPValue .
#string of the Source IP value
?NetworkNode :hasMeasurementData ?NodeIP .
?NodeIP :NodeIPValue ?SourceIPValue . #filters
by source IP
?NodeIP :hasMeasurementData ?NodeName .
?NodeName :NodeNameValue ?NodeName .
```

Thus, not only in terms of number of triples, but also in terms of complexity of the query to be executed, it would be more appropriate if a `NetworkNode` were directly linked to its information.

Therefore, although the MOMENT ontology, due to its generic nature, can fully describe the PingER domain, it was determined that it does not adequately fit the PingER case for multiple reasons, some of them stated in this section. The most important one is that this ontology does not aim to minimize the number of triples. Thus, it was decided not to reuse the ontology exactly as it is, but, instead, to reuse its concepts and ideas as basis to build an ontology more adequate and specialized for the PingER domain.

3.2 Proposal for PingER Ontology

Finally, after gathering information about the domain and analyzing the potentially suitable existing ontologies, we built an ontology specifically to the PingER domain to satisfy the conceptual model presented in Subsection 2.3. Figure 3 shows an overview of the model. This ontology reutilizes and is based on the concepts introduced by the MOMENT Ontology, adapting them to the necessities of the PingER domain. In addition, as stated previously, it also utilizes the Geonames and W3C Time ontologies.

Comparing our proposal to the one presented by the original MOMENT ontology (Figure 2), we observe that it is more specific hence more semantically applicable to the PingER domain, tending to optimize query performance and minimize the number of triples. Mainly, it efficiently supports the domain requirements, including publishing data in 5-star LOD standards [21].

A further explanation on the ontology is being provided[10].

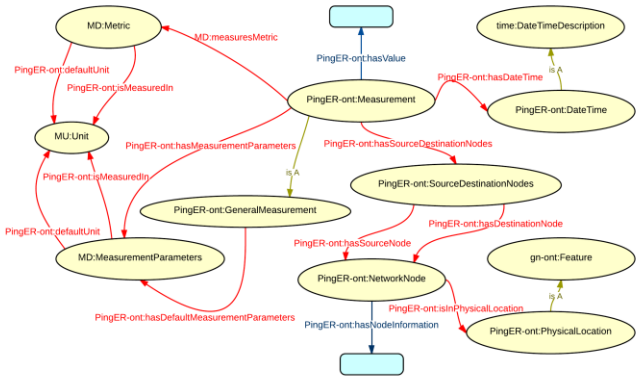

Figure 3 - Proposal for PingER Ontology

4. Triplification and Linkage

Having the ontology, the next phase in the process of publishing LOD is *triplification*. *Triplification* is the process of transforming (or generating) data in RDF triples format, instantiating individuals based on a defined OWL ontology [17]. It is potentially a very complex process, especially when dealing with large datasets.

Since the beginning of the process, it is important to consider linking the generated data to other existing data sources on the LOD cloud, always having in mind to publish following the 5-star LOD rules [21].

4.1 Triple Store Choice

The choice of the triple store may be critical when dealing with a big number of triples, because it is one of the most influencing factors of the queries execution time.

The first step to publish LOD data is to choose a good Triple Store. There are many available[11] and they basically provide the same functionalities. However, they differ in query processing time (which is critical for large databases) and extra features, such as semantic reasoning. W3C has recently provided a centralized collection of studies and benchmarking data that can be useful in making the most appropriate choice [22].

For the PingER project, Open RDF Sesame Native 2.7.2 [23] is currently being used, but it has shown not to work satisfactorily when

[10] https://confluence.slac.stanford.edu/display/IEPM/Ontology
[11] http://en.wikipedia.org/wiki/Triplestore#List_of_implementations

executing complex queries on over 50 million triples. Experiments are being conducted in order to migrate to Open Link Virtuoso [24].

4.2 ETL Triplification

The next step after establishing the RDF Triple Store is to populate the database with triples, that is, the *triplification* process itself.

If there is no digitized data at all, the process of generating data should be adapted in order to publish or export it in RDF format. Otherwise, if the data exists in any other format, a well-known process, especially in data warehousing studies, is required: Extraction, Transform, and Load (ETL) [25].

For the PingER LOD specific case, the ETL *Triplification* process was further divided into two independent processes (hence easily parallelizable): general data ETL and network measurement ETL.

PingER utilizes general concepts data about countries, cities, universities, and time. Most of this data is already available in RDF format in the LOD cloud. In the PingER case, this data amount is not critical when compared to network measurement. Thus, no complex process was needed to extract RDF data from the LOD cloud, transform it to link to existing databases and to adapt to PingER's ontology, and then load it into the triple store.

It is important to note that these common concepts (such as country, city, university, etc.) were extensively linked to DBpedia [26], Freebase [27], and Geonames [28] to provide context, fortifying the links between PingER LOD data and other RDF databases. We will see in the next section that this enables a wide variety of query mashups.

In addition to those general concepts data, the most important part in this process is to *triplify* PingER specific network measurement data. It is also the most complex part, since it deals with a large amount of data, requiring more elaborated and specific strategies to optimize the loading process. The solution was to parallelize the process across distributed computers. In summary, this phase extracts data from many CSV files, then transforms them into RDF format, and loads the resulting triples into the RDF database.

As stated in Section 2, PingER LOD considers 11 network metrics (e.g., Round Trip Time) and 3 different time aggregation types (i.e., measurements in relation to a specific day or month or year). A single ETL process transforms data in relation to a single metric and a single time aggregation. Each of these processes is independent. Thus, there are 33 processes that can be executed in parallel. They are launched following an automatic scheduling policy of choosing the most suitable computer among many others within SLAC's computing infrastructure.

Specifically, each of these processes is parallelized as following.

a) Beginning with a list of the monitor nodes, a HTTP GET is executed to download a single CSV file that contains data about a specified network metric, in a specified period of time, pinging from the a specific monitor node to all nodes that it monitors.
b) These GETs happen in parallel: the monitor nodes are grouped into M/T nodes. Where M is the number of monitor nodes and T is the number of threads that download the necessary CSV. During the elaboration of this solution, M = 80 and T = 20.
c) For this scenario, each thread will be responsible for `M/T` monitor nodes. Each of the threads executes one HTTP GET for each monitor node to download the necessary CSV file. Hence there will be `T` threads downloading simultaneously. It is done four times, until the `M` CSV files have been downloaded. Once the threads are joined, the next step begins.
d) The next step is to mine each of the `M` CSV files downloaded. This also happens in parallel, in a way that there are `M` threads running. Each of these threads launches `N(m)` more threads, where `N(m)` is the number of the monitored nodes that the node `m` monitors. Hence, there are `M*N(m)` threads running concurrently, for all `m` in the domain. On average[12], `N(m)` is less than 100, except for the monitor node at SLAC, which monitors more than 700 nodes.
e) Finally, this thread *triplifies* a very specific measurement: from a determined monitor node, to a determined monitored node, measuring a specific network metric, within a specific period of time. This triple is appended to an NTriples file. After *triplifying* the whole CSV file, it uploads the NTriples into the RDF database. The process stops when the `M` CSV files are converted into NTriples files.

All this parallelization was essential to reduce the ETL *Triplification* time. The biggest data case happens when *triplifying* daily data, for the last 60 days. It now takes around 10 hours to finish a single process for 'daily' time aggregation. It was observed that this same process would take more than 30 days to run, if executed sequentially instead of concurrently.

Moreover, it was also experimentally observed that loading the triples into the RDF database as they were being generated (i.e., millions of writings into the database) was making the process considerably slower. Thus, it was chosen to save all the triples in a separate NTriples file and then load it all at once into the repository, thereby decreasing the loading time significantly.

PingER LOD contains today more than 60M triples, linked to many other existing RDF databases on the LOD cloud. This number tends to increase as long as those processes run in an automatic, distributed, and concurrent way, within SLAC's computing infrastructure.

5. Public Access and Applications

After *triplifying* the data, LOD good practices suggests making it publicly available (i.e., publishing). The standard way to accomplish this is to establish a SPARQL Endpoint to the RDF data [29].

For PingER LOD, there is one available[13] in which users can run SPARQL queries to retrieve structured data from the RDF database. In addition to the publicly open access, the entire database dump is available in RDF format, as well as the OWL files for the ontology.

Use cases of the structured data are available through easy interfaces to support writing SPARQL Queries and to plot graphs in order to show some of the advantages of using LOD techniques. Specifically, three cases were developed (they are available on the project website's Visualizations tab[14]).

- Multiple network metrics analysis: This case utilizes PingER data only. It exemplifies how LOD aids even when not using mashups (crossing PingER data with other data sources). It highlights the advantage of having well-structured data with a schema, in a very expressive format: RDF triples. It also explores the use of complex SPARQL queries that are able to capture precisely what is being searched. A single query can retrieve network measurements using any possible combination of parameters. After running the query, a graph is plotted showing multiple network metrics simultaneously for the specified parameters. Before the project, the task of combining multiple metrics in a single data sheet to build a graph was not simple.

[12] http://www-wanmon.slac.stanford.edu/cgi-wrap/dbprac.pl?monalias=all
[13] http://pingerlod.slac.stanford.edu/sparql
[14] pingerlod.slac.stanford.edu/

- Crossing network metrics with university metrics: Since most of the nodes considered by PingER are educational institutions, it is intriguing to speculate on the relation between network quality and indicators of "university quality." This case illustrates a mashup of PingER data with DBpedia data about universities. To measure "university quality", some metrics were considered: number of students, number of undergraduate and graduate students, faculty size, and endowment. After executing the "mashed query", a map is drawn plotting circles on the universities PingER monitors. Size of the circles represents the value of the university metric (e.g., the bigger the circle, the greater the number of students), the filling color of the circles represents the value of the network metric (e.g., the whiter the color, the greater the value of throughput from a PingER monitor to that university), and the stroke color represents the type of universities (e.g., gold color represents private universities). Thus, using this graph, one could visually verify that well-funded universities have better network connectivity.

- Crossing network metrics with percentage of GDP that countries invest in research and development: this illustrates another mashup with PingER data with another RDF data source, in this case, World Bank Data. PingER data gives network measurements to many countries on the globe and World Bank gives many different interesting indicators[15], including countries' Research and Development (R&D) expenditure (% of GDP). Similarly, a map is drawn and circles are plotted on countries. The bigger the circle, the more the country invested in R&D. The whiter the color, the greater the value of the network metric. Moreover, an evolution within the years (since 1998) of the countries' investment in R&D as well as their network quality can be easily visualized on the map.

It is important to observe that the last case does not add RDF data about economy to PingER database. Instead, an application programmatically builds a mashed up query joining World Bank economy data to PingER network measurement data.

A SPARQL federated [30], which runs over the distributed databases on the LOD cloud, was built to try to execute the mashed up query. However, it was observed that the query was taking more than 1 hour to execute, whereas the application that simulates the federated query takes less than 1 minute.

6. Conclusion

This work presented a methodology to publish Linked Open Data. It illustrated an application on a real scenario that deals with big datasets about internet quality measurement, i.e., the PingER project, operated by SLAC National Accelerator Laboratory and other universities around the world.

The approach of this work relied on:

- Domain analysis, focusing on understanding the domain and selecting which data is useful for *triplification*;

- Ontology engineering, which evaluates ontology reuse according to its (a) semantic expressivity, (b) completeness in relation to the domain, and (c) impacts on query performance. It was stated that ontologies should be carefully designed so it will not increase the number of triples, making it easier to process in big data domains. This paper showed a proposal of an ontology that aims to decrease the number of triples;

- Distributing and parallelizing ETL process to *triplify* big data, linking to other data sources in the LOD cloud;

- Making the data available in a standard and structured format to provide ease of access.

As a result, it produced new and different applications for the PingER data, some of them not anticipated previously. This is possible since the data can now be linked to different databases, with a variety of information providing more contexts. More importantly, it opened PingER data to the community in a standard way, providing easy public access and interoperability.

Finally, the semantic web still has characteristics of a developing technology, although it has a clear potential to continue to evolve and find broader usage beyond the initial proposals by Berners-Lee [14]. To reach this potential, it will be necessary to produce more applications that consume semantic web technologies and concepts, generating more significant scientific knowledge and evolution regarding this area. Thus, the involved concepts would be extended, specific demands of the society would be identified, and people would be motivated to utilize these powerful and interesting technologies.

References

[1] David Siegel, *Pull: The Power of the Semantic Web to Transform Your Business*.: Portfolio, 2009.

[2] Linked Data - Connect Distributed Data Across the Web. [Online]. http://linkeddata.org/

[3] World Wide Web Consortium. (2014) Resource Description Framework (RDF). [Online]. http://www.w3.org/RDF/

[4] Richard Cyganiak and Anja Jentzsch. (2012) Linking Open Data Cloud Diagram. [Online]. http://lod-cloud.net/

[5] Giancarlo Guizzard, "Uma abordagem metodológica de desenvolvimento para e com reúso, baseada em ontologias formais de domínio," 2000.

[6] Federal University of Rio de Janeiro. (2012) Extract, Transform, and Load For Linked Open Data. [Online]. http://greco.ppgi.ufrj.br/lodbr/index.php/principal/etl4lod-2/

[7] Hiroyuki Inoue, Toshiyuki Amagasa, and Hiroyuki Kitagawa, "An ETL Framework for Online Analytical Processing of Linked Open Data," in *Web-Age Information Management*. Beidaihe, China: Springer Berlin Heidelberg, 2013, pp. 111-117.

[8] Les Cottrell and W. Matthews, "The PingER Project: Active Internet Performance Monitoring for the HENP Community," *IEEE Communications Magazine*, 2000.

[9] Shamkant Navathe and Ramez Elmasri, *Fundamentals of Database Systems*.: Addison-Wesley, 2010.

[10] Tom Heath and Christian Bizer, "Linked Data: Evolving the Web into a Global Data Space," in *Synthesis Lectures on the Semantic Web: Theory and Technology*.: Morgan & Claypool, 2011, pp. 1-136.

[11] Learn Data Modeling. Star Schema: General Information. [Online]. http://www.learndatamodeling.com/star.php#.UogBUsTUDIF

[12] Fred Leise, Karl Fast, and Mike Steckel. (2012) What is a Controlled Vocabulary? [Online]. http://boxesandarrows.com/what-is-a-controlled-vocabulary/

[13] Thomas R Grubber, *A translation approach to portable ontology specifications.*, 1993.

[14] Tim Berners-Lee, Nigel Shadbolt, and Wendy Hall, "The Semantic Web revisited," 2006.

[15] Dragan Djuric, Dragan Gaševic, and Vladan Devedžic, "The

[15] http://data.worldbank.org/indicator

Tao of Modeling Spaces," ETH Zurich, 2006.

[16] Geonames. (2013) GeoNames Ontology. [Online]. http://www.geonames.org/ontology/documentation.html

[17] World Wide Web Consortium. (2006) Time Ontology. [Online]. http://www.w3.org/TR/owl-time/

[18] Sathya Rao, "Monitoring and measurement in the next generation technologies," 2010.

[19] European Telecommunications Standards Institute, "Measurement Ontology for IP traffic (MOI); Requirements for IP traffic measurement ontologies development," 2012.

[20] World Wide Web Consortium. (2008) SPARQL Query Language for RDF. [Online]. http://www.w3.org/TR/rdf-sparql-query/

[21] EC FP7 Support Action LOD-Around-The-Clock. (2012) 5 Stars Data. [Online]. http://5stardata.info/

[22] World Wide Web Consortium. (2013) RDF Store Benchmarking. [Online]. http://www.w3.org/wiki/RdfStoreBenchmarking

[23] Open RDF. (2013) Sesame 2.7 Developer documentation. [Online]. http://www.openrdf.org/documentation.jsp

[24] Open Link Software. (2014) Open Link Virtuoso Universal Server. [Online]. http://virtuoso.openlinksw.com/

[25] Ralph Kimball and Joe Caserta, *The Data Warehouse ETL Toolkit.*: Wiley India Pvt. Limited, 2004.

[26] DBpedia. (2014) DBpedia. [Online]. http://dbpedia.org/About

[27] Freebase. (2014) Freebase. [Online]. http://www.freebase.com/

[28] Geonames. (2014) About GeoNames. [Online]. http://www.geonames.org/about.html

[29] Semanitc Web. (2012) SPARQL Endpoint. [Online]. http://semanticweb.org/wiki/SPARQL_endpoint

[30] World Wide Web Consortium. (2013) SPARQL 1.1 Federated Query. [Online]. http://www.w3.org/TR/sparql11-federated-query/

[31] World Wide Web Consortium. (2004) OWL Web Ontology Language Reference. [Online]. http://www.w3.org/TR/owl-ref/