Information Systems International Conference (ISICO), 2 – 4 December 2013

An Holistic Evaluation of Federated SPARQL Query Engine

Nur Aini Rakhmawati
Digital Enterprise Research Institute, National University of Ireland, Galway

Keywords:

Metadata
Dashboard
Dashboard design
Dashboard visualization
Best practice

ABSTRACT

Federated SPARQL Query Frameworks have been actively developed recently. Such a system consists of federated engines as a mediator and a group of SPARQL Endpoints. The mediator receives a query from a client and distributes it to relevant SPARQL Endpoints. As the core component of federation systems, existing evaluation methodologies usually only test the federated engines regardless of the capacity of the SPARQL Endpoints. Most evaluations are chiefly concerned about the speed of the engines response as the sole indicator for federation performance. Hence, the development of federation systems focuses primarily on source selection and join query optimization in the mediator. Several approaches merge a set of sub-queries into one query to reduce the response time. However, SPARQL Endpoints have a restriction to accept a query request in a certain period of time and return the results per query request. We hence conducted an experiment to evaluate federation frameworks by measuring their response time and data transfer between mediator and the SPARQL Endpoints. Additionally, we introduce a set of new metrics that are more suitable to assess the performance of federation frameworks performance in a range of environments.

Corresponding Author:

Nur Aini Rakhmawati,
Digital Enterprise Research Institute, National University of Ireland, Galway.
Email: nur.aini@deri.org

1. INTRODUCTION

As of March 2013, SPARQL 1.1 [1] has became a standard to query Linked Data[2]. It introduces federation features namely *SERVICE* and *VALUES* keywords to merge data from multiple SPARQL Endpoints. The *SERVICE* keyword allows us to specify the destination of a sub query, whereas *VALUES* can limit the size of intermediate results by filtering the result with the desired value. However, due to the lack of data knowledge location, determining data location while writing a query is a hard task. In order to tackle that issue, a number of federation frameworks provide a mediator to receive queries from the user and predict the query destination based on the data catalogue, ASK SPARQL Query and data indexing.

The characteristics of the federation over SPARQL Endpoints has similar to the characteristics of the federated databases where the data are distributed at multiple locations. As a result, the networking can influence the federated SPARQL query engines. Aside from multiple locations, this system consists of heterogeneous systems in terms of hardware and software. Therefore, it is a big challenge to asses the federation framework performance. As of out today, FedBench [11] is the only benchmark proposed for testing the performance of the federation over SPARQL Endpoints and federation over single RDF repositories. As complementary, it comes with the static dataset from cross domain and life science domain, and static query set. To assess the federation performance, it provides two metrics: Loading Time and Response Time. The loading time refers to the required time to load the RDF data from the RDF Dump. Hence, it is only applicable for Federation over single RDF repositories. Those metrics are dependent on the federation framework environment such as network bandwidth, CPU speed, memory capacity, application etc. Consequently, it is hard to generate a gold standard for federation benchmark. In this work, we introduce a set of independent metrics to evaluate the federation performance by observing the federated engine architecture and the data transmission between the mediator and SPARQL Endpoints.

1 http://www.w3.org/TR/sparql11-query/
2 http://www.w3.org/standards/semanticweb/data

2. RESEARCH METHOD

As the primary component of the federation framework, the existing evaluation methodologies usually only test the federated engines regardless of the capacity and restriction of the SPARQL Endpoints within framework. In general, public SPARQL endpoints prevent us to send query more than one query in one second. Moreover, it usually only returns result no more than 2000 rows. In terms of query execution, it sometimes rejects the expensive query to be executed. Those limitations of SPARQL Endpoint can lead to the uncompleted results, execution time out and other issues. In order to investigate the performance of federation framework, we propose new metrics that are related to the SPARQL Endpoint as part of federation framework.

In summary, our contributions are stated as follows:

- we introduce a set of independent metrics that are more suitable to assess the performance of federation frameworks performance in a range of environments.
- To conduct a comprehensive evaluation, we also propose metrics that are partially dependent to SPARQL Endpoint.
- We investigate several indicators to observe the performance of federation system. Eventually, we can give recommendation to optimize query planning and execution.

The rest of this paper is organized as follows: Section 2 describes the state of art of the existing evaluation of Federation over SPARQL Endpoints framework. We introduce set of metrics that are suitable to asses federated SPARQL Engines in Section 3. Afterwards, we conduct an evaluation and analysis the result in Section 4. Lastly, we conclude our work in Section 5.

3. RELATED WORKS

Federation over SPARQL Endpoints is still relatively far from maturity, only few works have been focusing on Federation Benchmark. [5, 9, 3, 2] created their own test bench to evaluate the federation system. As a notable state of the art of federation benchmark, FedBench[11] has been applied to compare the existing of federated engines [6, 12, 13, 8]. [13] conducts a large scale experiment over life science datasets. The datasets are set up in two strategies: local federation and hybrid federation. Apart from the response time, [8] also calculates minimum, maximum and average number of relevant sources. In order to tackle some of FedBench limitations, [7] provides additional variables and dimensions such as data partitioning and network latency. To measure the federation frameworks performance, it computes the Endpoint Selection Time, Execution Time and Answer Completeness.

4. FEDERATION FRAMEWORK PERFORMANCE

Aforementioned existing evaluations only focused on the speed of the mediator response as the sole indicator for measuring federation performance. Response time and Endpoint Selection time depend on the environment. A benchmark should produce the same result in a range of environments (small and large scale systems) to provide a good standard. Therefore, we propose following metrics that can be suitable to evaluate federation performance:

4.1. Independent Metrics

There are three types of data metric units arising during query execution, namely rows, requests and bytes. Those data metric units can present the cost of communication between federated engine and SPARQL Endpoints. Based on data transactions between the federated engine and SPARQL Endpoints, we identify independent metrics as follows :

- **Number of requests (RQ)** refers to the number of SPARQL queries (ASK, SELECT, CONSTRUCT, DESCRIBE) delivered by a federated engine to SPARQL Endpoints.
- **Number of ASKs (ASK)** Several approaches deliver ASK SPARQL query to find a suited source for certain sub query[12,4]. The basic idea of this strategy is discovering the relevant source with the minimum communication cost as the ASK SPARQL query only returns a boolean value.
- **Size of intermediate results (IR)** A query can be answered by only one SPARQL Endpoint, but the result is commonly retrieved from multiple SPARQL Endpoints. To compute the size of intermediate results, we count the total number of rows received by the federated engine during query execution.
- **Maximum Triples (MAX)** is defined as the maximum size of intermediate results obtained in the runtime per query request.
- **Amount of data sent and received (DSR)**
  Apart from rows calculation, we also consider to quantify the data transmission in bytes unit. The current of federated engines only estimated the join cardinality based on the number of triples, subjects, objects and predicates. In fact, the number of rows cannot reflect the actual of the amount

of data transmitted. Different triples have different sizes in bytes. The literal object could contain more characters than URI object. There are diversified query forms transmitted from mediator to SPARQL Endpoints to accomplish a query execution. Consequently, the mediator sends different amount of data according to number of characters used to formulate a SPARQL Query.

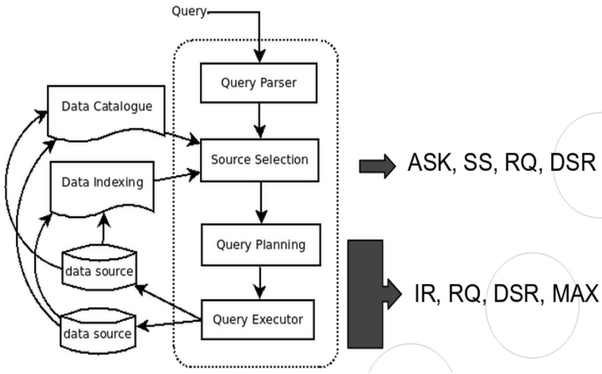

Figure 1. The Relation between Independent Metrics and Federated Engine Components

Besides calculating data transmission, we measure **Number of Selected Sources (SS)**. SS presents the number of SPARQL endpoints involved to accomplish one query execution. A query can be answered by either partial or all sources. The effectiveness of source selection can be shown from the number of SPARQL Endpoints accessed. Since it is hard to distinguish which query that is part of the source selection process or not, we ignore ASK SPARQL query which is delivered to the sources.

Above independent metrics are influenced by the strategy employed in the federated engine components. According to our federated engine architecture observation [1], we find a relation among independent metrics and federated engine components as depicted in Figure 1. Obviously, ASK and SS values rely on source selection strategy. The query optimization approach at Query planning and Execution affect IR and MAX values. As data transmission occurs during source selection and query execution, DSR and RQ are influenced by source selection, query planning and execution.

4.2. Independent Metrics

As part of federation framework, a SPARQL endpoint can induce the performance of the overall federation framework. However, most of the current evaluation approaches disregard the existing of SPARQL Endpoint. Hence, the query optimization in the federation engine only focus on how to get the results shortly without considering SPARQL Endpoint capability. For instance, FedX [12] proposed the bound join strategy to merge a number of intermediate result variables in one query. In theory and ideal condition, it can reduce the number of requests and size of intermediate results. However, it consumes much bandwidth to receive the result from SPARQL Endpoint. Moreover, the federated engine could communicate more than one request to a SPARQL Endpoint in a period of time. Consequently, SPARQL Endpoint workload tends to be high. In order to keep the sustainability of SPARQL Endpoint server, a SPARQL Endpoint sometimes rejects an expensive query and only returns limited number of triples. Ultimately, the query answer could be incomplete. Based on the observation of the capability of SPARQL Endpoint, we generate semi-independent metrics from independent metrics :

- **Request Workload (RW)** Generally, public SPARQL Endpoint does not allow us to send many requests in one interval of time. Hence, we define request workload as Number of Requests is divided by Response Time and Number of Sources Selected.
- **Average Intermediate Results (AIR)** One SPARQL query can return zero or many rows. In order to decrease bandwidth usage, it usually limits the answer size in a certain number of rows. AIR can indicate the capacity of SPARQL Endpoint to answer a query.
- **Average Data Received (ADR)** is defined as the amount of data received by the federated engine in bytes per query request. The high value of ADR implies a costly communication between the Federated engine and SPARQL Endpoints.

5. EXPERIMENT AND RESULT

In order to investigate the performance of the existing federation frameworks from different metrics, we performed a comprehensive experiment on Ubuntu Linux 64 bits. We set up three SPARQL Endpoint servers and the federated engine in one machine. We divided Dailymed[3] dataset based on its classes: Drugs, Ingredients and Organizations and stored them in different SPARQL Endpoints. We run 16 queries defined in [10] on FedX [12], Splendid [4] and DARQ [9] federation engines. Each query was executed for three times. DARQ failed at Query 4 and 5 since these queries contain unbound predicate, while FedX can not perform at Query 6 because of evaluation time out.

All frameworks have a source selection strategy, therefore number of source selected results in all queries were the same (Figure 7). In general, the fastest response time was done by FedX (Figure 2), but FedX communication is costly (Figure 3). As depicted in Figure 4, all frameworks generally produced the same size of intermediate results. FedX does not have the data catalogue to predict the data location, consequently, the number of requests and ASK SPARQL of FedX is higher than other federated engines results (Figure 5 and 6). Only Splendid and FedX generate ASK query during execution. As seen in Figure 8 and 11, the lowest value of ADR and AIR were achieved by FedX, since it employs the bound join strategy. As a result, FedX Request Workload is lower than DARQ and SPLENDID requests workload (Figure 10). The speed of mediator can be main indicator of federation performance, but other metrics should be considered to assess the federation performance. For instance, although FedX can answer queries quickly, it produces too many requests than can lead to the poor performance of SPARQL Endpoint. The communication cost can not be only described by the average size of intermediate results, but the average data received should be taken account. As shown in Figure 8 and 11, the value of AIR generated by SPLENDID is higher than the value of AIR generated by DARQ, but SPLENDID produced less amount of data received.

6. CONCLUSION AND FUTURE WORKS

We have presented an holistic evaluation of the existing federation SPARQL Engines by introducing two types of metrics: independent metrics and semi-independent metrics. The independent metrics are not influenced by the evaluation environment such as networking and application. These metrics includes Number of Requests, Number of ASKs, Size of Intermediate Results, Amount of Data Sent and Received and Number of Selected Sources. Most of those metrics are obtained from data transmission between the federated engined and SPARQL Endpoints. As part of the federation framework, we proposed three metrics that are associated with SPARQL Endpoint capacity, namely Request Workload, Average Answer Size and Average Data Received.

On the one hand, the speed of the federated engine to answer a query can be main indicator of federation performance, but on the other hand, it sometimes leads to expensive communication such as a high number of requests delivered in a certain interval of time and high data transmission. In real case, such condition can significantly impact on SPARQL Endpoint performance. Eventually, it will also affect the whole federation system. Hence, the federated engine developer should consider an approach to minimize the number of Requests in a periode of time such as applying window size in query execution strategy.

Federated engine mainly applies the cardinality estimation based on how many tuples or rows selected which is more suitable for single RDF query optimization. Since the federation framework encounters network communication issue, a weighting function should be assigned to the object with a string value. The literal value especially string, consumes more bandwidth than URI value.

3 http://wifo5-03.informatik.uni-mannheim.de/dailymed/

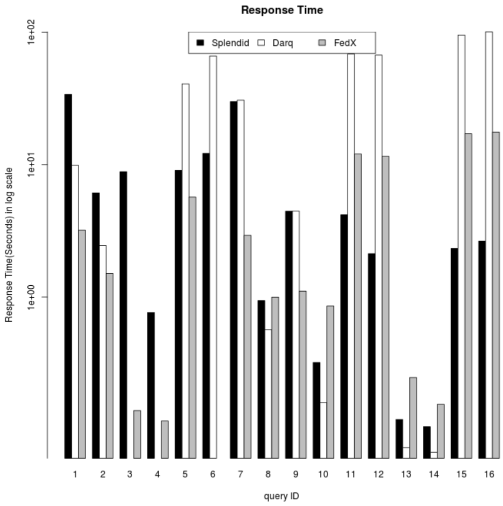

Figure 2. Response Time

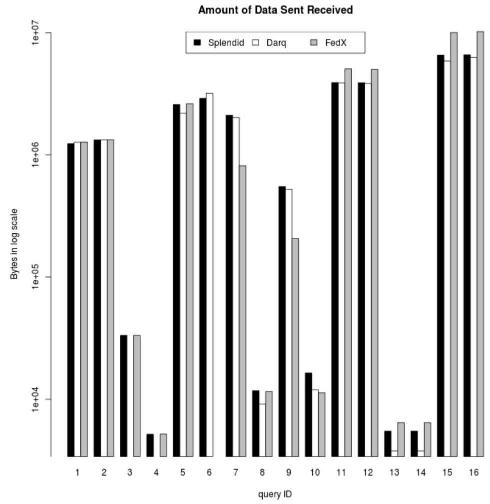

Figure 3. Amount of Data Sent Received

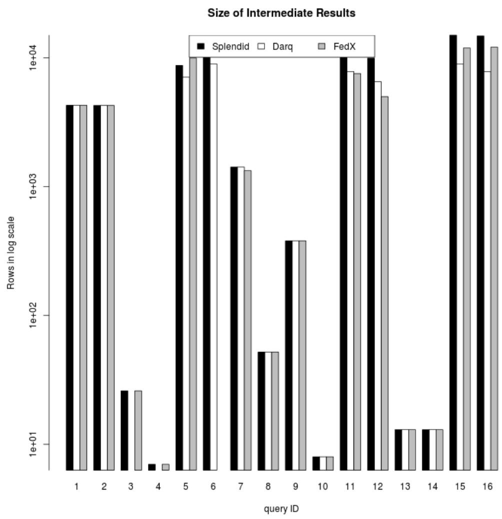

Figure 4. Size of Intermediate Results

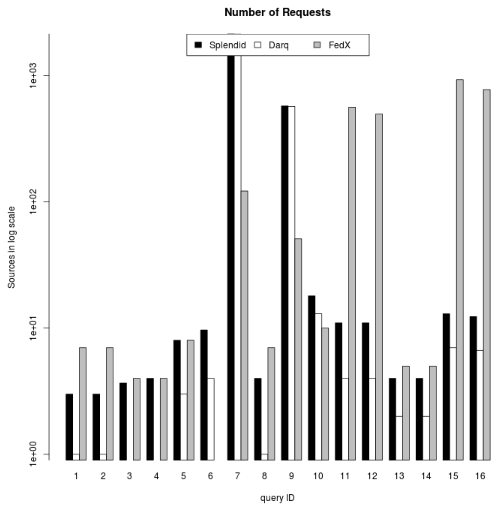

Figure 5. Number of Requests

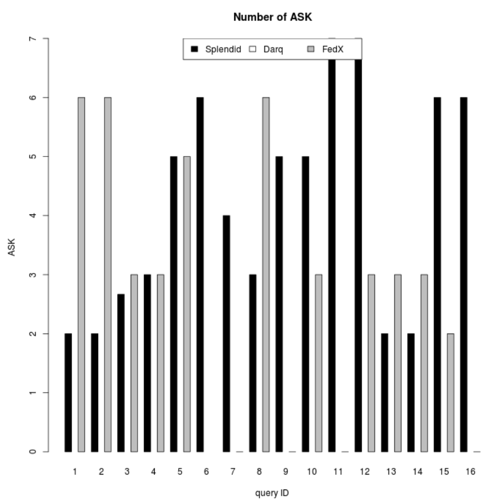

Figure 6. Number of ASKs

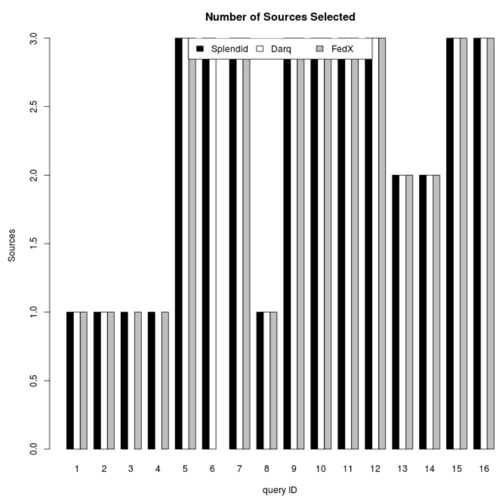

Figure 7. Number of Sources Selected

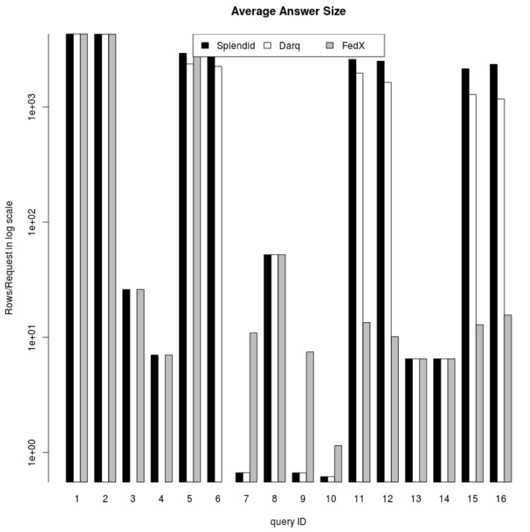

Figure 8. Average Size of Intermediate Results

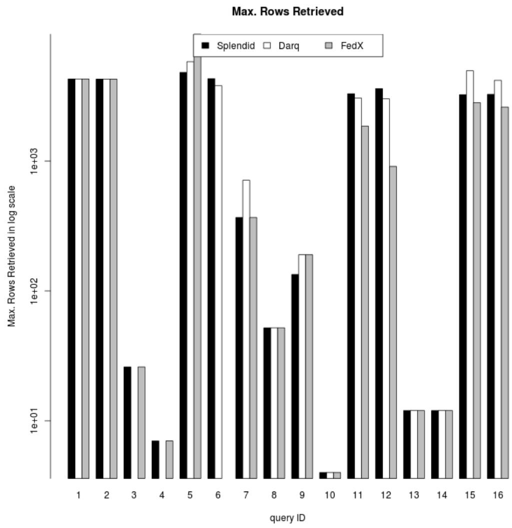

Figure 9. Maximum Rows Retrieved

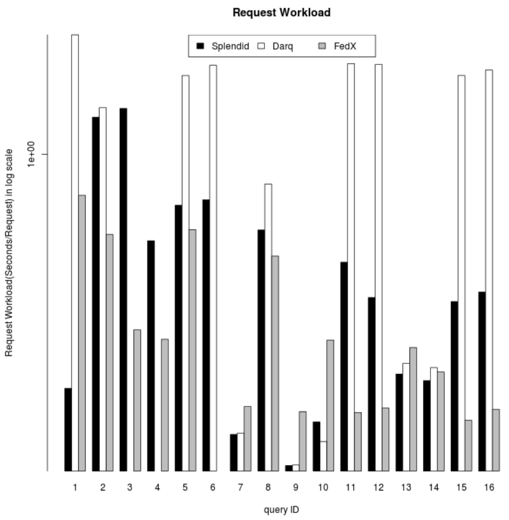

Figure 10. Requests Workload

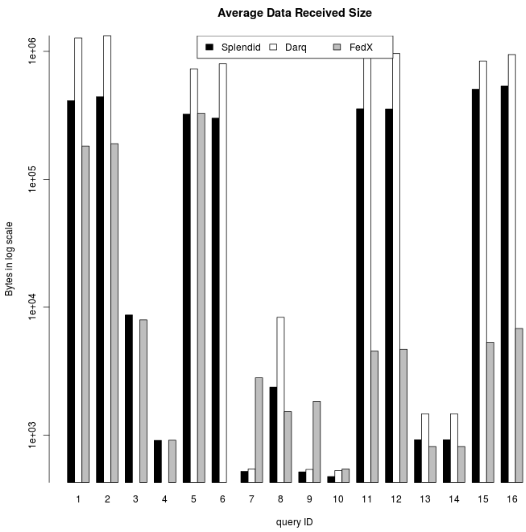

Figure 11. Average Data Received

REFERENCES

[1] N. Aini Rakhmawati, J. Umbrich, M. Karnstedt, A. Hasnain, and M. Hausenblas. Querying over Federated SPARQL Endpoints —A State of the Art Survey. ArXiv e-prints, June 2013.

[2] Ziya Akar, Tayfun Gkmen Hala, Erdem Eser Ekinci, and Ouz Dikenelli. Querying the web of interlinked datasets using void descriptions. In Linked Data on the Web (LDOW2012), 2012.

[3] Cosmin Basca and Abraham Bernstein. Avalanche: Putting the Spirit of the Web back into Semantic Web Querying. 2010.

[4] Olaf G ̈rlitz and Steffen Staab. SPLENDID: SPARQL Endpoint Federation Exploiting VOID Descriptions. In Proceedings of the 2nd International Workshop on Consuming Linked Data, Bonn, Germany, 2011.

[5] Shinji Kikuchi, Shelly Sachdeva, Subhash Bhalla, Steven Lynden, Isao Kojima, Akiyoshi Matono, and Yusuke Tanimura. Adaptive integration of distributed semantic web data. Databases in Networked Information Systems, 5999:174–193, 2010.

[6] Gabriela Montoya, Maria-Esther Vidal, and Maribel Acosta. Defender: a decomposer for queries against federations of endpoints. 9th Extended Semantic Web
Conference (ESWC) Demo Session, Mai 2012.

[7] Gabriela Montoya, Maria-Esther Vidal, and Maribel Acosta. A heuristic-based approach for planning federated sparql queries. In Proceedings of the 3rd Consuming Linked Data Workshop (COLD 2012), 2012. ́

[8] Gabriela Montoya, Maria-Esther Vidal, Oscar Corcho, Edna Ruckhaus, and Carlos Buil Aranda. Benchmarking

federated sparql query engines: Are existing testbeds enough? In International Semantic Web Conference (2), pages 313–324, 2012.

[9] Bastian Quilitz and Ulf Leser. Querying distributed rdf data sources with sparql. In Proceedings of the 5th European semantic web conference on The semantic web: research and applications, ESWC'08, pages 524–538, Berlin, Heidelberg, 2008. Springer-Verlag.

[10] Nur Aini Rakhmawati and Michael Hausenblas. On the impact of data distribution in federated sparql queries. In Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on, pages 255 –260, sept. 2012.

[11] Michael Schmidt, Olaf Grlitz, Peter Haase, Gnter Ladwig, Andreas Schwarte, and Thanh Tran. Fedbench: A benchmark suite for federated semantic data query processing. In Lora Aroyo, Chris Welty, Harith Alani, Jamie Taylor, Abraham Bernstein, Lalana Kagal, Natasha Fridman Noy, and Eva Blomqvist, editors, International Semantic Web Conference (1), volume 7031 of Lecture Notes in Computer Science, pages 585–600. Springer, 2011.

[12] Andreas Schwarte, Peter Haase, Katja Hoose, Ralf Schenkel, and Michael Schmidt. Fedx: A federation layer for distributed query processing on linked open data. In ESWC, 2011.

[13] Andreas Schwarte, Peter Haase, Michael Schmidt, Katja Hose, and Ralf Schenkel. An experience report of large scale federations. CoRR, abs/1210.5403, 2012.

BIBLIOGRAPHY OF AUTHOR

Nur Aini Rakhmawati is a PhD student at the Linked Data Research Centre at DERI. She holds a BSc degree in Informatics Engineering at Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia and received her Msc in Electronic Engineering at National Taiwan University of Science and Technology. She was part of the EC FP7 Support Action LATC where she contributed
to the development of the 24/7 Interlinking Engine backend and recently she has joined the EC FP7 BIG project. In her PhD work she focused on Federation over SPARQL Endpoints and the characteristics of the underlying datasets.