International Journal of Internet and Distributed Systems, **2013, 1, 1-8**
http://dx.doi.org/10.4236/ijids.2013.11001 Published Online February 2013 (http://www.scirp.org/journal/ijids)

How Interlinks Influence Federated over SPARQL Endpoints

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

Received January 15, 2013; revised February 17, 2013; accepted February 25, 2013

ABSTRACT

As the Web of Data grows, the number of available SPARQL endpoints increases. SPARQL endpoints conceptually represent RPC-style, coarse-grained data access mechanisms. Nevertheless, through the potential interlinking of the contained entities, SPARQL endpoints should be able to over distinct advantages over plain Web APIs. To our knowledge, to date, there has been no study conducted that gauges the impact of the link on SPARQL query execution, especially in a federated set-up. In this paper, we investigate how the existence and types of typed links influences the execution characteristics of different SPARQL federation frameworks. In order to measure the query performance, we propose a combined cost model based on a statistic analysis of the query performance metrics that involves parameters such as type of link, the data catalogues and cache, number of links, and number of distinct subjects. As result, we show that number of distinct subject and number of links have significant impact on Federation over SPARQL Endpoints performance whereas type of link does not have significantly influence in federation query performance.

Keywords: SPARQL; Federation over SPARQL Endpoints

1. Introduction

Nowadays many data have been published in RDF format and connected each other by a link. This condition encourages people to integrate data across dataset to yield more valuable information. The easiest way to integrate data is employing links between dataset. Those links can navigate us to data which has the same identity or has relation each other. According to Linked Open Data (LOD) Cloud statistics [1], more than 50% of dataset in LOD cloud have more than 1000 out going links. It indicates that a dataset publisher consider to put effort to generate links. However, to the best of our knowledge, there have been no study conducted to investigate the benefit of link in linked data.

The higher number of links may cause usage of bandwidth increases, but in the other hand, the data result can be more retrieved easily. Moreover, the duration of gathering data may take longer than usual. Besides the number of link, the type of link may influence the performance of query. The identity link such owl:sameAs could answer more than relation link such rdf:seeAlso. Thus, we observe carefully the impact of number of link as well as type of link on query performance. Further, we also identify the other factors that could have significant impact on Federation over SPARQL Endpoint performance. For instance, an entity may have more than one link which navigate to several different dataset. In this case, the number of datasets involved could be one factor to be considered. The more number of dataset involved, the more number of request delivered is.

To submit a query, there are three type query interfaces to access data: SPARQL Endpoint, native repository and HTTP request [2]. Since the SPARQL Endpoint offers flexibility in term of formulating query, we focus on the Federation over SPARQL Endpoints query performance. In addition, it is also motivated by the beyond emerging of SPARQL 1.1[1] which will support federation query service. The federation features allows us to write SPARQL query easier to gather data from various SPARQL Endpoint.

To summarize, the primary contributions of our work are the following:

- To the best of our knowledge, no study has probed the impact of link on federation SPARQL query. Our observation shows the effect of the link on federation query through the experimental as well as statistical way.
- We also propose cost and benefit model in relation with several of our observation key performance factors.
- We conduct an investigation of the significant performance factor in the Federation over SPARQL Endpoints.

[1]http://www.w3.org/TR/sparql11-query/

This paper is structured as follows: We review related works in the Section 2. Section 3 gives an overview how to write SPARQL query to integrate data from multiple data sources. We investigate the cost and benefit model and its variables, followed by our approach to construct a cost model in Section 4. To build our cost model, we run experiment in Section 5. We also validate our cost equation. Eventually, we conclude our work in Section 6.

2. Related Work

Decentralized data is nature of Linked Data infrastructure. Crawling data in the single repository could not be cheap solution in Linked Data as it requires much disk space to store data and high system specification to process a query [3]. To overcome this problem, several Linked Data system that is similar to distributed database have been developed recently. This system can be broken down into Link Traversal, Federation over single repositories and Federation over SPARQL Endpoints. Link Traversal [4] discovers related data by following the HTTP Uniform Resource Identifier (URI). The completeness is big issue in the Link Traversal system, therefore it is not suitable for large scale system. Federation over single repositories and SPARQL Endpoints use a mediator to deliver an incoming query to multiple data sources and aggregate all the retrieved result. Accessing data in the Federation over single repositories relies on native API of the repository. To date, only a few of repository systems provides this API. As stated in LOD Cloud statistic, 68.14% of data sources provides SPARQL Endpoint. Therefore, in this work we only take into account Federation over SPARQL Endpoints. SPARQL Endpoint conceptually represents RPC-style, coarse-grained data access mechanisms to execute SPARQL Protocol and RDF Query Language (SPARQL)[2] query that becomes a standard query for Resource Description Framework (RDF)[3] data since 2008. There exists research addressed to build federation over SPARQL Endpoint, namely:

Sesame Sail[4], one of Sesame part in conjunction with Alibaba[5], allows multiple datasets to be virtually combined into a single dataset. The performance of federation to execute complex query is poor since it sends query to all datasource.

FedX [5] is addressed to deal with Federation SAIL performance in federated query. To define relevant source, it delivers ASK query before query processing. It only submits sub query to the source that answers TRUE value, in order to reduce the cost of communication. It also applies exclusive group to cluster related sub queries that have same query destination.

Splendid [6] is also extended from Sesame which employs VoID[6] as data catalogue. Based on statistic in the VoID, it calculates the cardinality function to detect the relevant source for a sub query. Apart from cardinality estimation, it sends ASK query if sub query destination can not defined by cardinality estimation. Once the source selection is done, it builds sub queries and join order for optimization.

DARQ [7] is an extension of ARQ[7], a well known SPARQL query engine processor. Similar to Splendid, it employs Service Description[8] as its data catalogue to specify the destination of sub query. The Service Description contains data description and statistical information has to declare in advance during setup phase.

To construct our cost model, we run our query set on Federation Sail, FedX and Splendid since they are extension of Sesame Framework.

3. Data Integration in SPARQL Query

We discover three ways to integrate data in Federation over SPARQL Endpoints by distinguishing the availability of link among datasets.

3.1. No Link

The availability of link between two datasets allows us to integrate their data, but there are some possibilities to gather data among dataset without any links. First, we can use UNION operator. In this way, the query result of one datasource treats independently from other result, the query processor only combines them before passing it to user. Thus, this scheme is usually suitable to collect all data that having the same behaviour but not identical. **Query 1** presents how to collect medicines for certain disease from Drugbank[9], Dailymed[10] and Diseasome[11] SPARQL Endpoints.

The other alternative of data integration without employing any links is object comparison. With regard to sameness of data, we can compare several non URI object of each predicate among dataset by defining them in FILTER. The easiest way is only compare the *rdfs*:*label* among data, however it may be inaccurate since the same label does not mean the same data. Due to case sensitive of SPARQL query, we must add REGEX in FILTER condition. Thus, the cost query is more expensive. Consider an example, **Query 2** aims to find drug in Sider[12] SPARQL Endpoint which is similar to Acetaminophen in Drugbank SPARQL Endpoint. If the REGEX is removed,

[2]http://www.w3.org/TR/rdf-sparql-query/
[3]http://www.w3.org/RDF/
[4]http://wiki.aduna-software.org/confluence/display/SESDOC/Federation
[5]http://www.openrdf.org/alibaba.jsp
[6]http://www.w3.org/TR/void/
[7]http://jena.apache.org/documentation/query/index.html
[8]http://darq.sourceforge.net/#Service_Descriptions
[9]http://www4.wiwiss.fu-berlin.de/drugbank/
[10]http://www4.wiwiss.fu-berlin.de/dailymed/
[11]http://www4.wiwiss.fu-berlin.de/diseasome/
[12]http://www4.wiwiss.fu-berlin.de/sider/


```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX drugbank:
<http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/>
PREFIX diseasome-disease:
<http://www4.wiwiss.fu-berlin.de/diseasome/resource/diseases/>
PREFIX dailymed:
<http://www4.wiwiss.fu-berlin.de/dailymed/resource/dailymed/>

SELECT ?diseasename ?drugname WHERE {
 {
       ?drug a a drugbank:drugs    .
        ?drug rdfs:label ?drugname .
       ?drug drugbank:possibleDiseaseTarget diseasome-disease:1055 .
       diseasome-disease:1055 rdfs:label ?diseasename .
}
UNION
{
      ?drug a dailymed:drugs .
      ?drug rdfs:label ?drugname .
       ?drug drugbank:possibleDiseaseTarget diseasome-disease:1055 .
      diseasome-disease:1055 rdfs:label ?diseasename .
}
}}
```

Query 1. Example of SPARQL Query to collect data from multiple sources using UNION.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX sider: <http://www4.wiwiss.fu-berlin.de/sider/resource/sider/>
PREFIX drugbank-drug:
<http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugs/>
select *
{
drugbank-drug:DB00316    rdfs:label ?drugname    .
?sider a    sider:drugs .
?sider rdfs:label ?sidername .
FILTER REGEX(str(?sidername),str(?drugname),"i")
}
```

Query 2. Example of SPARQL Query to collect data from multiple sources using REGEX.

the query will yield empty result because each label has different case.

3.2. Reusing Identifier

To find related data, we can reuse URI identifier from other party. In this situation, the datasets do not have link between them directly, but both of them pose the same URI object from other dataset that can join them. The comparing indirect link is much better than comparing non URI object in term of accuracy issue. For instance, we pick one query from FedBench [2] (**Query 3**). This query finds the relation of drug in Drugbank and KEGG via *drugbank*:*casRegistryNumber* which is an URI identifier of BioRDF-Cas dataset.

3.3. Link

Typically, a query in federation utilizes link to gather data across dataset. This link can be generated manually and automatically by a tool such SILK [8] and Limes [9]. Those tools produce a set of links from two dataset as defined in the link specification. Having links both identity and relationship make data can be integrated in straight way. By employing *diseasome*:*possibleDrug*, **Query 1** can be altered by **Query 4**.

By having a link between two datasets, the query cost can be cheaper while number of distinct outgoing datasets is not too high. The high number of distinct outgoing datasets leads the number of requests to other dataset increases. Consequently, the query mediator needs longer time to process a query. For better explanation, given an example **Query 5** which purposes to gather all drug that are as same as drug in Drugbank via owl:sameAs. According to Drugbank dataset, each owl:sameAs in entity *drugbank*:*drugs* could have four distinct outgoing datasets such as *DBpedia*, *purl.org/net/tcm/tcm.lifescience.ntu.edu.tw/id/medicine*, *www*4.*wiwiss.fu-berlin.de/sider* and *data.linkedct.org*. If we assume all subjects have exactly four distinct datasets, the query mediator will send four requests for each subject. Therefore, the highest number of request is four times number of subjects.

```
PREFIX drugbank:
<http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/>
PREFIX drugbank-cat:
<http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugcategory/>
PREFIX kegg: <http://chem2bio2rdf.org/kegg/resource/>
PREFIX purl: <http://purl.org/dc/elements/1.1/>
SELECT ?drug ?title WHERE {
?drug drugbank:drugCategory drugbank-cat:micronutrient> .
?drug drugbank:casRegistryNumber ?id .
?keggDrug a kegg:Drug .
?keggDrug kegg:xRef ?id .
?keggDrug purl:title ?title .
 }
```

Query 3. Example of SPARQL Query to collect data from multiple sources using Reusing Identifier.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX diseasome:
<http://www4.wiwiss.fu-berlin.de/diseasome/resource/diseasome/>
PREFIX diseasome-disease:
<http://www4.wiwiss.fu-berlin.de/diseasome/resource/diseases/>
SELECT ?diseasename ?drugname    {
diseasome-disease:1055 diseasome:possibleDrug ?drug .
?drug rdfs:label ?drugname .
diseasome-disease:1055 rdfs:label ?diseasename .
}
```

Query 4. Example of Federation SPARQL Query by using link.

```
PREFIX drugbank:
<http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
Select * {
?drug a drugbank:drugs .
?drug owl:sameAs ?other .
?other rdfs:label ?name .
}
```

Query 5. Example of Federation Sparql Query by using owl:sameAs.

4. Cost and Benefit Model

As we described in Section 3, the unavailability of link causes user has to map data from one data source to other source. Consequently, consumer puts totally effort in data integration. If the publisher generates link to connect its data to other dataset, the data integration cost is shared between consumer and publisher. In order to be measurable, we develop cost model for consumer as outcome of federation query performance metric.

4.1. Cost Variables

1) Bandwidth Usage (*BU*)

In order to measure networking cost, we only calculate total of uplink and downlink bandwidth during query execution between framework and SPARQL Endpoints. We ignore the bandwidth usage from user to framework.

2) Number of Requests (*RQ*)

The bandwidth usage depends on amount of data transmission. Therefore, it can not present number of request from framework to SPARQL Endpoint. A query may have more than one request to complete the result. In our experiment, we measure number of requests that refers to number of submission of sub query to each SPARQL endpoint.

3) Response Time (*T*)

How responsive system to respond a query need to be evaluated in federated query. The response time is defined as how long it takes time from a query generated to result retrieved.

4.2. Benefit Variables

By categorizing the availability of link of data integration in Federated over SPARQL Endpoints, we define benefit and cost that arise as result of the existence of link as well as the type of link. Hartig [4] proposed query execution time as cost and number of result as benefit in Link transversal. In the Link Traversal environment, the data knowledge is hard to know in advance and the data could be change dynamically. Contrast to Link Traversal environment, the data in the Federation over SPARQL Endpoints can be observed before query execution. Therefore, we consider query completeness and soundness as benefit instead of number of result. The high number of result does not mean better result because that might be redundant result or invalid result. The query completeness is defined as number of true answer that is stored in dataset, whereas the query soundness refers to number of retrieved of true answer. We adopt completeness and soundness metric combination from LUBM [10]. Let S_q be Soundness of query result, C_q be Completeness of Query and β be weight between C_q and S_q, then F_{measure} of query completeness and soundness F_q is defined as follow:

$$F_q = \frac{\left(\beta^2 + 1\right) C_q \cdot S_q}{\beta^2 C_q + S_q}$$

4.3. Multiple Regression Model

Multiple Regression model is common way to present relation of cost with its parameters. The cost acts as dependent variable that its value is depend on certain independent variables. There are two kinds of regression models : linear and non linear.

1) Linear Regression Model

Given y as dependent variable, x_1, x_2, ... x_n as n independent variable and c_0, c_1, … c_{n-1} as coefficient of regression, the multiple regression linear is

$$y = c_0 + c_1 x_1 + c_2 x_2 + \cdots + c_n x_n$$

In this model, dependent variable has linear correlation with each independent variables. Further, the good linear model should have the coefficient of determinant (R^2) close to one. R^2 represents the correlation between actual and predicted dependent variable which is described by following formula:

$$R^2 = \left(\frac{1}{K} \frac{\sum_{i=0}^{n} \left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)}{\delta_x \delta_y} \right)^2$$

where K is the number of samples, $\bar{x}$ and $\bar{y}$ are mean of x and y respectively, δ_x is the standard deviation of x, and δ_y is the standard deviation of y.

2) Non Linear Regression Model

If the majority of independent variables could not fulfil linear correlation and the R^2 close to zero, we must transform it to non linear model. There are many form of non linear regression models such exponential, power, polynomial, trigonometric, etc. Non linear regression model is more complicated than linear regression model because the function is built from trial and error. In this paper, we endeavor to build our formula in exponential and power model. Let C_n be $\ln(c_n)$, the exponential model can be written in the following equation:

$$y = C_0 C_1^{x_1} C_2^{x_2} \cdots C_n^{x_n}$$

where as the power model is explained in the following formula:

$$y = C_0 x_1^{C_1} x_2^{C_2} \cdots x_n^{C_n}$$

The coefficient of regression is estimated statistically from sample experiment of known independent and dependent variables. To obtain those coefficients, we define following independent variables:

1) Type of Link (*TL*)

[11] distinguishes the link into three categories: Relationship Link, Identity Link and Vocabulary Link. We only take account Relationship and Identity link in our model since we do not integrate data with its vocabulary. To observe the effect of link type, we define TL as one of cost parameter which equals 0 and 1 for Relationship Links and Identity Links respectively.

2) Data Catalogue and Cache Benefit (*DC*)

By having a catalogue and cache, federation framework can reduce the request of the existence of data. Thus, we consider this as important factor in federation query performance. Through our observation, applying caching scheme during execution make framework perform better after second running. The second better performance is framework that have a data catalogue. Hence, we define the level of range of DC value in **Table 1**.

3) Number of Links (*NL*)

The higher number of link is, the higher the number of request is because the framework has to ask to SPARQL Endpoint as amount of link.

4) Number of Distinct Subject (*DS*)

The number of distinct subject which is involved in a query is also considered as key performance factor. The larger number of distinct subject is, the more time needed to execute a query is.

5) Number of Distinct Outgoing Datasets (*DD*)

Without data catalogue, a framework queries blindly to each dataset. Thus, the number of outgoing dataset might not influence performance. The number of outgoing dataset has a significant impact on framework with catalogue because the number of request is limited by the number of outgoing dataset.

For each model (linear and non linear), we create $2^5 - 6$ combination of independent variables that influence the value of cost. The independent variable can be eliminated and added in the model. Note that, the model must have at least two independent variables. Finally, the model with highest R^2 is chosen as our model.

Once we obtain the best model, we decide the significance of independent variable by calculating P_{value} of independent variable using T Test of null hypothesis. If the P_{value} is smaller than 0.05, we accept the independent variable has significant impact on federation query performance.

Table 1. Data catalogue and cache benefit value.

Framework Features	Value
Data Catalogue and Cache	1
Cache	0.75
Data Catalogue	0.5
Data Catalogue and Cache	0.25

5. Experiment

5.1. Environment

1) Dataset

As proof of concept, we run queries in four datasets, namely Sider, Diseasesome, Dailymed and Drugbank. Those datasets are chosen because there exist links among of them and comprises identity and relationship type as illustrated in **Figure 1**. For more details of statistic of related links in dataset, refer to **Table 2**. To provide

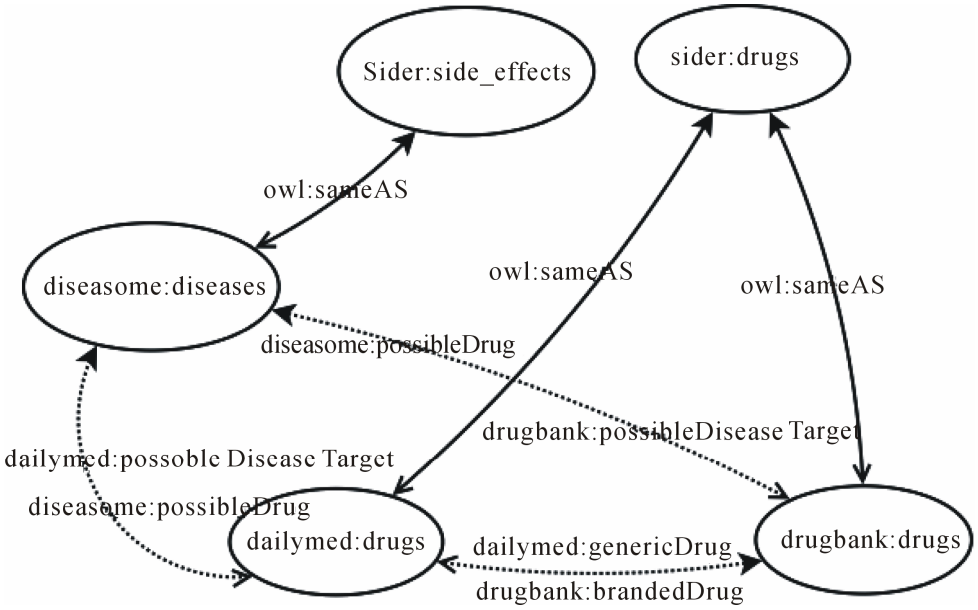

Figure 1. The Relation of drugbank, disease, dailymed and sider dataset.

Table 2. Dataset statistic.

Dataset	Distinct Outgoing Link	Distinct Outgoing Dataset	Triples	Links
Dailymed	3	8	164,276	39,635
Drugbank	14	18	766,920	56,958
Disease	6	10	91,182	31,750
Sider	2	7	193,249	20,294

SPARQL Endpoints, we install four Fuseki[13] as Endpoint service for each dataset.

2) Query Set

The query set should cover wide range of all parameters but it should be in general form, in order to compare among queries performances fairly. Our query set comprises 90 queries which is not included operators and operands since we only consider the usability of link to improve federated query performance. With respect to data integration category in Section 3, we set up query to cover all categories. But we could not find any query pattern for Reusing Identifier category in our dataset.

3) Federation Framework

To evaluate the query performance, we choose three frameworks, namely FedX, Sesame Sail and Splendid. All of them are built on top of Sesame framework. FedX and Sesame Sail represent framework without data catalogue whereas Splendid represents framework having data catalogue. The frameworks and SPARQL Endpoints are installed in a Linux virtual machine.

5.2. Result

In total, we should have 240 results but 28 queries are failed execution because the query execution time exceeds the time out duration (one hour) or the federation query framework does not support such query. For example, Splendid could not execute no link query pattern because it binds the same address while comparing the literal value. Although we increase the time out limit to 3 hours, only two no link queries can be processed successfully by Sesame Sail and FedX. Hence, we exclude all the no link query result to build our model. Based on no link query result, the existence of link can boost federation query performance.

The average of $F_{measure}$ of completeness and soundness is 9.67. It implies that all independent parameters do not influence the query completeness and soundness result. All the framework accomplish to execute query even though the performance is poor.

The linear regression model is our first fitting cost model. **Table 3** depicts low coefficient of determination (R^2 adjustment) in the linear regression model. Obviously, the independent and dependent variables have little linear correlation. As described in **Table 4**, the better result is obtained from Exponential Regression Model which the R^2 adjustment is nearly 50% or above. According to our null hypothesis, the Exponential Regression model contains only *DD* and *DS* as its significant variables.

Table 5 shows that the value of R^2 Adjustment of power model equation surpasses the value of exponential model in all cost. Therefore, we choose this model as our cost model. Number of distinct subjects (*DS*) significantly contribute in all cost. On the other hand, the type of link is not significantly related with query federation performance because the framework treats all the type in the same way.

With respect to the power regression calculation, we obtain the value of each coefficient parameters that will be inserted to the cost formula. As written in formula **Figure 2**, it can be noted that not all the parameters is included in the formula.

$$T = 136.67 DS^{0.18} DD^{0.17} NL^{0.35} DC^{-0.14}$$
$$RQ = 5.62 DS^{0.063} NL^{0.47} DC^{-0.82}$$
$$BU = 4.539 DS^{0.11} NL^{0.68} DC^{-0.42}$$

Figure 2. Multiple power regression model for federation over SPARQL endpoints cost model.

Table 3. R^2 adjustment linear model.

Cost	Independent Variable	R^2 Adjustment	Significant Variable
T	*DS DC*	2.1%	*DS*
RQ	*DS DD NL DC*	24.52%	*DD NL*
BU	*DS DD NL*	48.54%	*DS NL*

Table 4. R^2 adjustment exponential model.

Cost	Independent Variable	R^2 Adjustment	Significant Variable
T	*DS DD NL DC*	54.76%	*DS DD*
RQ	*DS DD NL DC*	47.5%	*DS DD*
BU	*DS DD NL DC*	58.2%	*DS DD*

Table 5. R^2 adjustment power model.

Cost	Independent Variable	R^2 Adjustment	Significant Variable
T	*DS NL DC*	71.58%	*DS NL*
RQ	*DS DD NL DC*	61%	*NL*
BU	*DS DD NL DC*	79.19%	*DS NL*

[13]http://jena.apache.org/documentation/serving_data/

5.3. Cross Validation

Once the cost formula is created, we conduct new evaluation as cross validation. The new evaluation composing new 10 queries is also executed on three federation frameworks. Eventually, we calculate the relative error produced by estimation equations for each query as shown in the following formula:

$$RE = \frac{|V' - V|}{V}$$

where RE is relative error, V' is estimated value and V is actual value.

The result of validation can be found at **Figures 3**, **4** and **5**. 50% of estimated response time value has error smaller than 50%. Bandwidth usage validation result shows that more than 57.14% of validation value has RE less than 30%. The most promising result is the number of request which 71.42% of estimated value is less than 30% RE. In general, the RE tends to be high when number of links and number of distinct subjects are high. As illustrated in **Figure 1**, we only have 11 links that connect two entities in our dataset. Given that, we only can generate 9 queries for building our model and 2 queries for validation dealing with the high of number of link and distinct subject. This number is too small comparing to total of query. Hence, our model is not suitable for high number of links and number of distinct subjects.

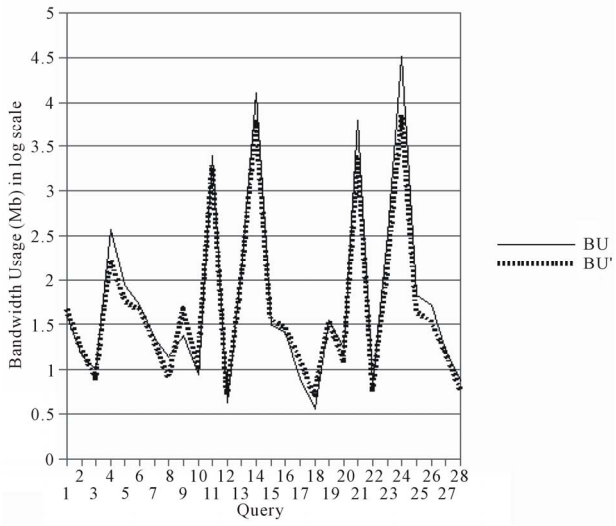

Figure 3. Estimated and actual bandwidth usage.

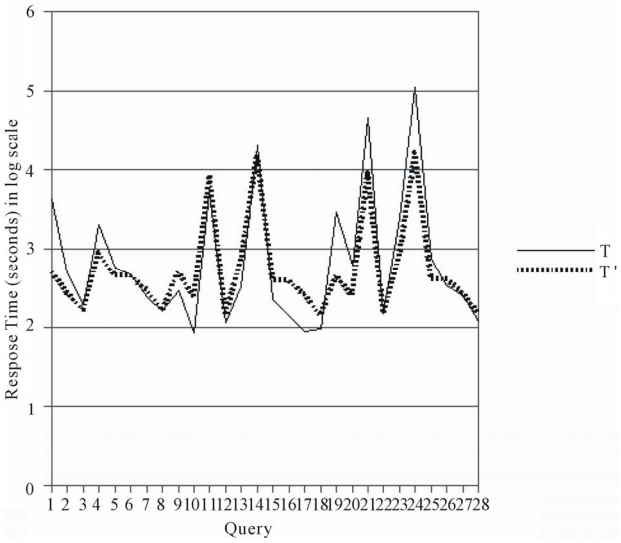

Figure 4. Estimated and actual response time.

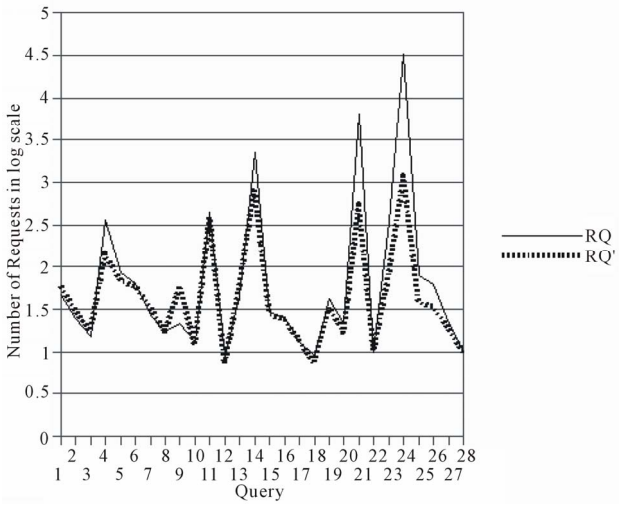

Figure 5. Estimated and actual number of requests.

6. Conclusions

We presented an investigation of impact of the existence of link in Federation over SPARQL Endpoints performance. In order to calculate the query performance, we proposed a cost model in multiple linear and non linear regression form. In addition, we also formalized benefit model by measuring completeness and soundness value. We defined type of link, number of outgoing datasets, number of links, number of distinct subjects and data catalogue benefit as the independent variables to build our cost and benefit model. Those independent variables can determine the estimation of response time, bandwidth usage and number of request as metric of as a result, F_{Measure} of soundness and completeness value closes to 1 which indicates all independent variables do not influence the query completeness and soundness result. Based on coefficient of determination calculation, not all parameters can be inserted in the cost model. Moreover, the power regression model is more fitted for building the cost model than linear regression and exponential models. Hence, we construct cost model based on power regression model. With respect to the significance of parameter, number of distinct subjects and number of links have significant impact on Federation over SPARQL

Endpoints performance.

By analysing the failure of no link query pattern, we found that joining data across different dataset without using a link need large resource such as bandwitdth. This failure is caused by the complexity of no link query pattern and high number of object comparisons. Thus, the existence of link can boost federation query performance. Further, we proved that the type of link does not influence the Federation over SPARQL Endpoints performance since the federation framework treats link like other RDF predicate. The type of link could have more impact for analysing the coverage of link in term of answering query.

7. Acknowledgment

We acknowledge Dr. Michael Hausenblas and Dr. Marcel Karnstedt for supervising author during this research.

REFERENCES

[1] R. C. Christian Bizer and A. Jentzsch, "State of the Lod Cloud," Vol. 9, 2011.

[2] M. Schmidt, O. Grlitz, P. Haase, G. Ladwig, A. Schwarte, and T. Tran, "Fedbench: A Benchmark Suite for Federated Semantic Data Query Processing," In: L. Aroyo, C. Welty, H. Alani, J. Taylor, A. Bernstein, L. Kagal, N. F. Noy and E. Blomqvist, Eds., *International Semantic Web Conference* (1), *Lecture Notes in Computer Science*, Vol. 7031, Springer, Heidelberg, 2011, pp. 585-600.

[3] J. Umbrich, M. Karnstedt, A. Hogan and J. Parreira, "Hybird Sparql Queries: Fresh vs. Fast Results," In: P. Cudr-Mauroux, J. Heflin, E. Sirin, T. Tudorache, J. Euzenat, M. Hauswirth, J. Parreira, J. Hendler, G. Schreiber, A. Bernstein and E. Blomqvst, Eds., *The Semantic Web ISWC* 2012, *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 2012, pp. 608-624. doi:10.1007/978-3-642-35176-1_38

[4] O. Hartig, "Zero-Knowledge Query Planning for an Iterator Implementation of Link Traversal Based Query Execution," *Proceedings of the* 8*th Extended Semantic Web Conference on the Semantic Web*: *Research and Application*, *ESWC* 2011, Berlin, Heidelberg, Springer-Verlag, 2011, pp. 154-169.

[5] A. Schwarte, P. Haase, K. Hoose, R. Schenkel and M. Schmidt, "Fedx: A Federation Layer for Distributed Query Processing on Linked Open Data," *ESWC*, 2011.

[6] O. Görlitz and S. Staab, "SPLENDID: SPARQL Endpoint Federation Exploiting VOID Descriptions," *Proceedings of the* 2*nd Internation Workshop on Consuming Linked Data*, Bonn, 23 October 2011.

[7] B. Quilitz and U. Leser, "Querying Distributed RDF Data Sources with Sparql," *Proceedings of the* 5*th European Semantic Web Conference on the Semantic Web*: *Research and Applications*, *ESWC*'08, Berlin, Springer-Verlag, Heidelberg, 2008, pp. 524-538.

[8] A. Jentzsch, R. Isele and C. Bizer, "Silk—Generating RDF Links While Publishing or Consuming Linked Data," *International Semantic Web Conference* (*ISWC*-2010), Shanghai, 2010.

[9] A.-C. Ngonga Ngomo and S. Auer, "Limes—A Time-Efficient Approcah for Large-Scale Link Discovery on the Web of Data," *Proceeedings of IJCAI*, Vol. 15, 2011, pp. 2312-2317.

[10] Y. Guo, Z. Pan and J. Heflin, "Lubm: A Benchmark for Owl Knowledge Base Systems," *Web Semantic*: *Science*, *Services and Agents on the World Wide Web*. *International Semantic Web Conference*, Vol. 3, No. 2-3, 2005, pp. 158-182.

[11] C. Bizer, T. Heath, K. Idehen and T. Berners-Lee, "Linked Data: Evolving the Web into a Global Data Space," Morgan & Calypool Publishers, San Rafael, 2008.

