

An analysis of RDF view maintenance using Jena

Syed. M. Shehram Shah*, John Wilson**, Vijdan Khalique*, Hira Noman*
*Department of Software Engineering, Mehran University of Engineering & Technology, Pakistan, **Department of Computer & Information Sciences, University of Strathclyde, United Kingdom
shehram.shah@faculty.muets.edu.pk, john.n.wilson@strath.ac.uk,
vijdan.khalique@faculty.muets.edu.pk, hira.noman@faculty.muets.edu.pk

Abstract

Resource Description Framework is a next generation technique to create web content. This has given rise to the need to develop efficient and effective techniques to manage high volume RDF structures. This paper deals with Semantic Web technologies and presents an analysis of JENA based updation of RDF structures. The view maintenance of RDF structures (varying sizes), i.e. updating RDF structures using views is performed through JENA constructs and performance of insertion and deletion operations is measured. After analysis, Insert operation time was observed to increase proportionally however time remained the same for delete operations performed on the RDF data.

Keywords: Resource Distribution Framework, JENA, Semantic Web, View Maintenance

1. Introduction

The amount of data we generate has increased exponentially in the last few decades attributed mostly to the integration of information and communication technologies in our everyday activities. The internet is growing every minute leading to an enormous number of web pages and other documents being added to it. Managing this huge amounts of data presents several challenges. It is important to note that since a large part of this data was designed for human consumption, much of it is not completely interpretable by machines. A very limited amount of the web content (rendering information) that is intended for end user presentation is understandable by computers. This issue is worsened by the fact that the internet contains data of a polymorphic nature while not having effective methods for the sharing, managing and organizing of data. To resolve this issue, scientists introduced the Semantic Web.

The Semantic Web extends current Web technologies with an aim to transform the web's functioning, specifically it aims for the integration of data on the internet for sharing of information. The Semantic web provides a framework to create web content which is intelligible by machines. The World Wide Web Consortium (W3C) recommended a universal data format to be used for information interchange on the internet called the Resource Description Framework (RDF) [1-2]. RDF makes data more meaningful for computer by using structured data. It solves existing problems with data representation by keeping metadata along with the main data containing information about the use and intended meaning of the data thus making the web more intelligent. In this regard, the means of updating RDF structures are an important point of study. Currently, this practice requires the regenerating of the underlying RDF structure after a modification has been made. This is

quite expensive in terms of computer resources used as well as the time taken. The creation of RDF knowledge bases has been of keen interest of researches in the field of medicine, physics and chemistry as it allows for easy sharing and exchange of information among different entities [3-6], examples are the DBpedia and Freebase [7] and Wikipedia, whose knowledge base is RDF based. Therefore, it is pertinent that methods that provide reliable and efficient RDF structure maintenance be produced.

2. Related Work

As discussed before, RDF structures form the core of Semantic Web technology. Within an RDF structure, triples i.e. subject, predicate and the object represent data. Information about the entities are contained within the subjects and the objects whereas the relationship between them is contained in the predicate. URLs (Universal Resource Identifiers) or XML tags can be used to specify each component of a RDF triple [8]. The subject and object within an RDF structure point to a Web resource where its attributes are described as properties. Every subject and object has some kind of relationship between them and therefore, it can be said that an RDF structure describes the properties between two entities. These concepts have helped researchers develop methods of using RDF structures for content management.

The World Wide Web Consortium developed a query language for RDF structure data retrieval called SPARQL (SPARQL Protocol and RDF Query Language) [8] which is designed to work with RDF triples. This language was designed to specifically work with RDF data and operation is based predicates in graph traversal mode. Several applications called as RDF engines allow for RDF structures to be easily used and maintained have been develop on SPARQL on the concept of RDMBS (Relational Database Management Systems) also sometimes. The authors in [7] developed RDF: 3X (RDF Triple Express) which is an RDF engine use to manage and query RDF collections. The application designed is light weight and is based on the RISC paradigm and involves algorithms for the querying, manipulation and processing of RDF content. Another approach is that provides data management capabilities in a variety of formats is a database engine called the Virtuoso Universal Server [10]. This provides data integration services, supports SPARQL and is said to fit well in terms of Sematic Web as the external sources. A couple of examples of ‘triple stores’ are the 3Store [11], 4Store [12]. A more detailed list of similar application can be found on [13], these are recognized by the W3C. Another way of updating RDF structures is to use SPARQL/UPDATE (SPARUL), which is an updated version of SPARQL [14]. It uses add and delete operations to modify the RDF structure while having a syntax similar to SPARQL. However, no editing mechanism is present in SPARUL.

Another approach for modifying RDF structures is to use views. This is a flexible and powerful way used by applications to access web information and perform manipulations as desired. The creation of views on RDF data was proposed by [15]. They used a declarative language called RQL (RDF Query Language) for querying RDF graphs. View maintenance then requires that after the data source has been modified, the queries that make up the view are updated accordingly. They have performed maintenance and updating of RDF views using the operations of insertion, deletion and modification.

3. Methodology

A prototype Java application, based on the Jena extension, has been developed to perform insertion and deletion operations on RDF structures. The intension is to perform update

operation and analyze the impact of the size of data being updated. Jena is an open source framework for developing Semantic Web applications. [16] and allows for the creation of Semantic Web based applications using libraries. The application built in Jena have the capability to interact with Semantic Web applications as well as manipulate content on the Semantic Web. The created application is designed to read three RDF files which are stored in to instances of three separate models. This is followed by the execution of insert and delete operations on the models while the time taken for each operation is recorded as shown in Figure 1.

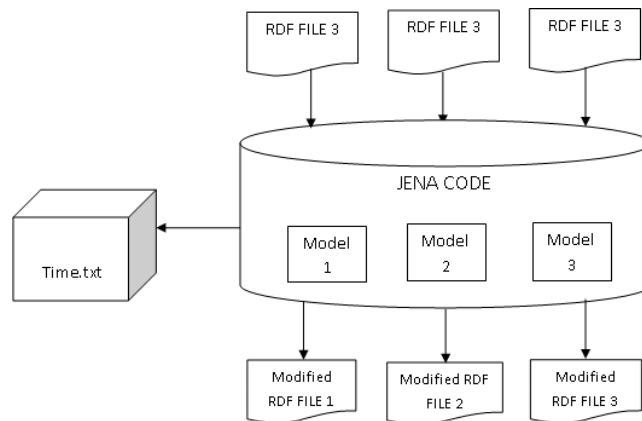


Figure 1. Block diagram of application

4. Data

The test data used for the application uses the vCard Ontology which was developed by the Semantic Web Interest Group [17] as a W3C working draft for describing organizations and people [18]. Within the vCard ontology, people and organizations can be represented by a comprehensive set of classes, subclasses, properties and modifiers including bio-data, work as well as home information. It provides a simple and easily readable structure for input and thus has been used in this research. A VCARD specification (RFC6350) has been mapped on to the RDF structure. New resources and property values were created by expanding the ontology for the experiments having sizes of 1 MB, 5 MB and 10 MB. Since we focus on the time taken for RDF structure updating for varying sizes, the content of the taken into consideration.

5. Experiments and Discussion

We have performed five different experiments for analyzing Jena base updating of RDF structures. Each experiment is performed on files of three different sizes, 1 MB, 5 MB and 10 MB with insertion and deletion operations being performed in each while time was recorded for each operation.

5.1. Experiment 1

One statement was separately inserted and deleted in to each of the three models while time was recorded during the performance of the operations. Table 1 shows the results of this experiment and a pictorial representation the form of a scatter plot is shown in Figure 2.

Table 1. Insertion and Deletion times for Experiment 1

Size of Data (MB)	Insertion Operation Time (ns)	Insertion Operation Time (sec)	Deletion Operation Time (ns)	Deletion Operation Time (sec)
1.0	46515	0.046515	18046	0.018046
5.0	17805	0.017805	17113	0.017113
10.0	19326	0.019326	17662	0.017662

The time taken for Insert operations is represented by a blue line while delete operations are represented by a red line. Table 1 and Figure 2 show that the time taken for the insert and deletion operations of a statement is largest for the first model even though it has the smallest size among the three.

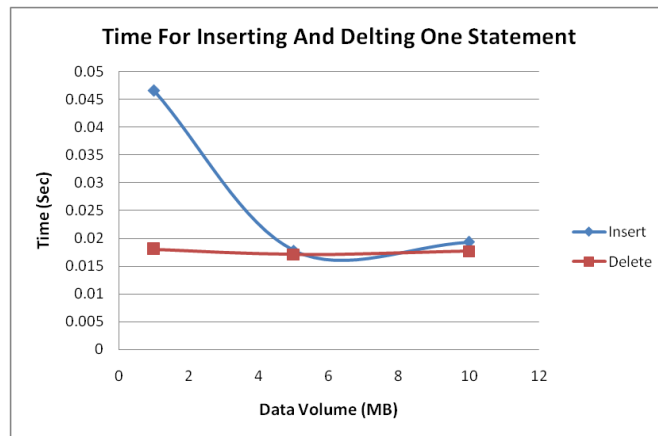


Figure 2. Time taken for Insertion and Deletion of 1 statement

5.2. Experiment 2

In order to explore this further, another experiment was conducted in a similar manner to the first one (insertion and deletion), however, the number of 100 statements was increased to 1000 so as to observe the effect of increasing the number of statements. The results obtained were similar to first experiment in that the first model took the longest time for the operations to be completed. Therefore, providing the conclusion that the insertion and deletion operation on the first model take longer than the other two models and the file size does not have a significant performance impact.

The outcome of the second experiment is similar to the first one. The only difference is the number of statements. Time was calculated for inserting and deleting 1000 statements to see if the number of statements had any influence on the update operations. However they were

similar to the results obtained earlier. It always takes longer to insert or delete operations on the first model and the size of file does not have a significant impact on performance. This time is significantly higher than the other two models with sizes of 5 MB and 10 MB for insert operations. However, the time taken for delete operations by the first model is just slightly higher than that of the 5 MB model, the time taken by the 10 MB model is a little larger than the 5 MB model. This indicates that the size of the model does not affect performance much when delete operations are being performed.

5.3. Experiment 3

This experiment was conducted to further validate the observation that modification of the first model is the slowest regardless of the size of data being inserted. To verify this, equally sized models were created and the test of inserting and deleting 100 statement was performed while the time taken for the operations was recorded as shown in Table 2. The results support the previous observations as the time taken was highest for model 1 as before, supporting the observation that RDF size doesn't have a large effect on the operations.

Table 2. Insertion and Deletion times for Experiment 3

Model Number	Insertion Operation Time (ns)	Insertion Operation Time (sec)	Deletion Operation Time (ns)	Deletion Operation Time (sec)
1	95470744	95.470744	2189576	2.189576
2	1967537	1.967537	1920509	1.920509
3	1983792	1.983792	1979855	1.979855

5.4. Experiment 4

The size of the first and third model were interchanged in Experiment 4 i.e. the size of Model 1 was set to 10 MB and that of Model 3 to 1 MB. 100 statements were inserted and deleted on all the models in the same sequence (from Model 1 to Model 3) while time was being recorded. The results are given in Table 3 and Figure 3. As before, the first model has shown to take the longest for operation execution.

Table 3. Insertion and Deletion times for Experiment 4

Size of Data (MB)	Insertion Operation Time (ns)	Insertion Operation Time (sec)	Deletion Operation Time (ns)	Deletion Operation Time (sec)
10.0	6349189	6.349189	1207012	1.207012
5.0	2557257	2.557257	1190055	1.190055
1.0	2476046	2.476046	973642	0.973642

Experiments 1 to 4 indicate that the time taken to perform operations on the first model is always larger than the two other models. Moreover, this was found to be effected very little by model size. The reason for this is the way a Java program works. Since JVM loads the classes and other static library blocks when an initial piece of code is run, the first run always takes the longest time thus resulting in the large time taken

to update the first model. The successive execution of the same code would always be faster than the first run. Since Java compiles the code in to machine language if the same code is run 10,000 times, this would also result a slight variation in performance. As observed, the time taken for the deletion operations also follows this trend.

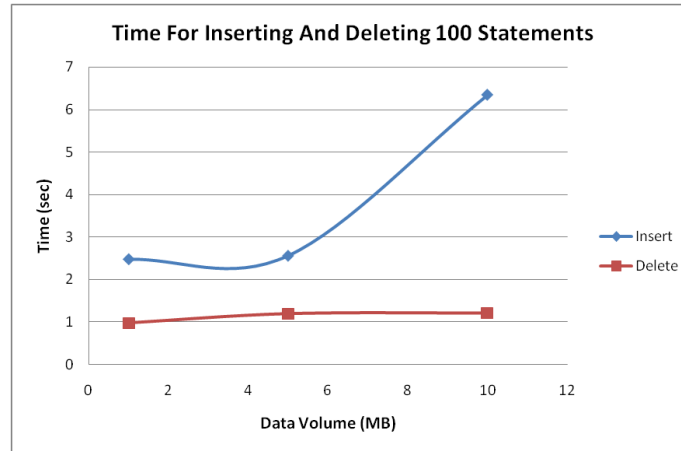


Figure 4. Time taken for Insertion and Deletion of 100 statements
 (Note: Operations performed on 10 MB model first)

5.5. Experiment 5

Lastly, to isolate the optimization performed by JVM and to assess the relation of updating operations and file size we conducted an experiment using four models. We performed ‘warming up’ of the JVM by first performing insertion and deletion operation a test model followed by the 100 statement updating and deleting operations similar to the first experiment. The results are given in Table 4 and Figure 5.

Table 4. Insertion and Deletion times for Experiment 5

Data Volume (MB)	Insert (ns)	Insert (sec)	Delete (ns)	Delete (sec)
1	1802277	1.802277	1997721	1.997721
5	1835297	1.835297	2061084	2.061084
10	1832620	1.83262	2299811	2.299811

As can be observed from Table 3 and Figure 5, running the operations on a test model before performing the operations has eliminated the long time needed for the first code run. Moreover, the times for the operation in experiment 5 are very similar and no significant difference can be observed. Thus providing the conclusion that update operations on RDF structures are not affected significantly by its size as observed from this experiment.

6. Conclusion

We have analyzed the updating of RDF structures using Jena. RDF structures provide an easy and reliable way to access and extract web based content formed using the Semantic

Web model. Experiments conducted have taken in to consideration the updating of RDF structures by means of insert and delete operations. Models of three different sizes were used for the experiments and the time was recorded for each operation. It was observed that first runs of any code are the slowest in any updating task as the underlying entities are loaded in to JVM. Overall, it was also found that delete operations are much faster as compared to update operations. A reason for this is that insert statements may involves the creation of new subject, object and predicate having a combination of existing and new components and values within in the triple of the model. Moreover, update operations are not significantly affected by RDF size.

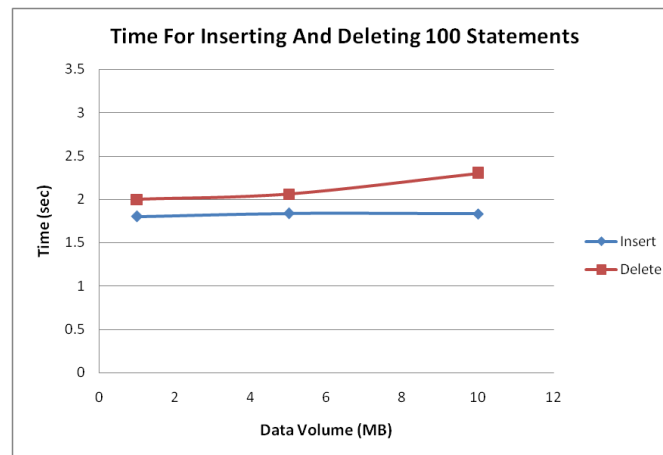


Figure 5. Time taken for Insertion and Deletion of 100 statements

References

- [LS99] Lassila, O and Swick, R.: Resource Description Framework (RDF) Model and Syntax Specification, <http://www.w3.org/TR/REC-rdf-syntax/>. [Accessed 30 August 2016].
- [W3C04] Daly J., Fergue M.C., Hirakawa Y.: World Wide Web Consortium Issues RDF and OWL Recommendations, <https://www.w3.org/2004/01/sws-pressrelease> [Accessed 30 August 2016].
- [AC17] Chougule, A., Jha, V.K., Mukhopadhyay, D.: Adaptive Ontology Construction Method for Crop Pest Management. In: Proceedings of the International Conference on Data Engineering and Communication Technology, Advances in Intelligent Systems and Computing. pp. 665–674 (2017).
- [SC15] Chaudhary, S., Bhise, M., Banerjee, A., Goyal, A., Moradiya, C., Technology, C.: Agro Advisory System for Cotton Crop. In: 7th International Conference on Communication Systems and Networks (COMSNETS) (2015).
- [KS16] Sangeetha, K., Santhiya, S.: Hospital Record Search using RDF based Information Retrieval. *Int. J. Softw. Eng. Its Appl.* 10, 109–116 (2016).
- [KPC16] Chand, K.P.: Dynamic Ontology Based Model for Text Classification. *Int. J. Appl. Eng. Res.* 11, 4917–4921 (2016).
- [NT08] Neumann, T., Weikum, G.: RDF-3X: a RISC-style engine for RDF. *Proc. VLDB Endow.* 1, 647–659 (2008).
- [RDFConcepts] Klyne G. and Carroll J. J., Resource Description Framework (RDF): Concepts and Abstract Syntax, <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/#section-Graph-URIref>

- [SPARQL] Prud'hommeaux E. and Seaborne A., SPARQL Query Language for RDF, <http://www.w3.org/TR/rdf-sparql-query/>
- [EO09] Erling, O., Mikhailov, I.: RDF Support in the Virtuoso DBMS. In: Networked Knowledge- Networked Media. pp. 7–24. Springer (2009).
- [3S] 3Store, Semantic Web Standards, <https://www.w3.org/2001/sw/wiki/3Store>
- [4S] 4Store, Semantic Web Standards, <https://www.w3.org/2001/sw/wiki/34Store>
- [TS] Category:Triple Store, Semantic Web Standards, https://www.w3.org/2001/sw/wiki/Category:Triple_Store
- [SA08] Seaborne A, SPARQL/Update: A language for updating RDF, <http://www.w3.org/Submission/2008/04/>
- [VR02] Volz, R., Oberle, D., Studer, R.: Towards views in the semantic web. In: 2nd Int'l Workshop on Databases, Documents and Information Fusion (DBFUSION02) (2002).
- [Jena] Apache Jena-What is Jena?, https://jena.apache.org/about_jena/about.html
- [W3CSWIG06] W3C Semantic Web Interest Group, <http://www.w3.org/2001/sw/interest> [Accessed: July. 02, 2015].
- [W3CVCARD RDF 13] vCard Ontology - for describing People and Organizations, <https://www.w3.org/TR/vcard-rdf/>