Evaluation of SPARQL queries using relational databases *

Jiří Dokulil

Department of Software Engineering,
Faculty of Mathematics and Physics,
Charles University in Prague, Czech Republic
Jiri.Dokulil@mff.cuni.cz

Abstract. Basic storage and querying of RDF data using a relational database can be done in a very simple manner. Such approach can run into trouble when used on large and complex data. This paper presents such data and several sample queries together with analysis of their performance. It also describes two possible ways of improving the performance based on this analysis.

1 Introduction

The RDF [2] is a key part of the Semantic Web. It defines the format and semantics of such data but does not provide query capabilities. Several query languages have been created or modified to support RDF querying. One of the languages is SPARQL [5]. We have created an experimental implementation of SPARQL.

We used a straightforward way of storing RDF data in a relational database. This allowed us to evaluate SPARQL queries by translating them to SQL queries. Although it worked nicely for small or simple RDF data, we suspected that evaluation times may turn bad with large and complex data. We have been able to obtain such data [4].

2 Experiments, RDF indexes and statistics

The data is complex. The RDFS schema consists of 226 classes and 1898 properties and contains 26 million tripes.

Simple test queries showed that no SPARQL feature creates a bottleneck of the system. But when the features were combined in complex queries the performance decreased greatly. After examining the execution plans of the queries, we came up with a possible explanation for this undesirable behavior. The SQL optimizer makes wrong assumptions about the size of the intermediate data produced during the evaluation of the query.

* This research was supported in part by the National programme of research (Information society project 1ET100300419).

Major weakness of our system is the fact that for every triple used in the query one table join is added to the result SQL query. This means that a more complex SPARQL query is evaluated using many joins that are usually expensive because large sets are being joined.

We designed a so called *RDF indexes* to overcome this problem. An RDF index is a pre-evaluated SPARQL query and the result of the evaluation is stored inside the database to speed up evaluation of similar queries. Experiments have shown that the RDF indexes improve the performance of the query evaluation.

The Oracle database contains a general way of collecting statistics about the data they contain. But the RDF data have several characteristics that can be used to gather more precise statistics. For instance the system can store precise number of triples for each predicate. This information could be used to help the optimizer build better execution plans via Oracle SQL hints. We plan to implement this function in the future.

3 Conclusion

Large and complex RDF data are not yet widely available. Although large data such as WordNet or DBLP libraries are available [6], their structure is simple. WordNet is commonly used to test performance of RDF databases, for instance in [3]. We feel that the real data of the Semantic Web will be more complex.

The complex data we used in out tests helped us develop two methods of improving the query performance. The methods are in very early stage of development and should be compared to other systems like Sesame [1]. We used much more complex data and encountered problems that would not show up if tests were run on simple data even if the data was very large. This led us to development of two methods of fighting the problems. We implemented one of the methods and measurements confirmed that it has the desired impact on query evaluation performance.

References

1. Broekstra J., Kampman A., Harmelen F. (2002): *Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema*, in Proceedings of the First International Semantic Web Conference, Italy, 2002, 54-68
2. Carroll J. J., Klyne G. (2004): *Resource Description Framework: Concepts and Abstract Syntax*, W3C Recommendation, 10 February 2004
3. Chong E. I., Das S., Eadon G., Srinivasan J. (2005): *An Efficient SQL-based RDF Querying Scheme*, in Proc. of the 31st International Conference on Very Large Data Bases, Trondheim, Norway, August 30 - September 2, 2005, 1216-1227
4. Dokulil J. (2006): *Transforming Data from DataPile Structure into RDF*, in Proceedings of the Dateso 2006 Workshop, Desna, Czech Republic, 2006, 54-62 http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS//Vol-176/paper8.pdf
5. Prud'hommeaux E., Seaborne A. (2005): SPARQL Query Language for RDF, W3C Working Draft, 23 November 2005
6. http://www.semanticweb.org/library/