RESEARCH PAPER | Engineering | Volume : 3 | Issue : 2 | February 2013 | ISSN - 2249-555X

Querying RDF Data: Methods and Issues

KEYWORDS

Gurdas Singh
Regional Institute of Management & Tech., Mandi Gobindgarh, Punjab.

Dr. Jaiteg Singh
Chitkara Institute of Engineering & Technology, Chandigarh-Patiala National Highway (NH-64) Punjab

ABSTRACT *Semantic data is distributed by nature and spread over the different repositories. Resource Description Framework (RDF) is most popular language to store the semantic data. Information retrieval process requires querying these RDF data repositories. There are many methods which effectively and efficiently estimate, evaluate, optimize and execute the query but with constraints. In this paper we discuss the recent query methods for RDF data found in the literature then common issues related to query methods are drawn.*

1. INTRODUCTION

A semantic web means the meaning of data which is understandable by both people and machine.. Information in semantic web is distributed over websites and managed by different organization locally having machine-readable description of the data and published in a human readable form and Uniform Resource Identifiers (URI) are used to point the resources. Many systems are using the Semantic Web technology to handle their metadata. RDF is an upcoming standard that provides the basis for storage and exchange of this metadata. There is urgent need of use of vocabularies for making assertion about the things.

RDF is a general method to decompose knowledge into small pieces, with some rules about the semantics, or meaning, of those pieces.. In RDF, a statement is sometimes called a triple because it has three parts. The subject of a statement is in fact called the subject. The equivalent of a verb is called the predicate, and the remaining part is called the object. Other terms are also in common use: property instead of predicate, and value instead of object

RDF database, also called a triple store because it stores triples, is generally any repository of RDF statements that supports some form of querying operation. A query operation follows three general sequence i.e. query plan, query optimization and query execution.

This paper is organized as follows: Section 2 discuss the different query method in recent literature for RDF data. Section 3 addresses the common issues of query methods and finally section 4 discuss conclusion.

2. QUERY METHODS

In this section we are discussing some query methods which have been proposed recently. An optimizer presented in [2] which determine the relevant query sources at compile time using synopsis and a query engine is used to reduce the volume of intermediate result at run time. It boosts the query execution time up to two times and volume transfer by three order of magnitude. Most of the query optimizations schemes are based on static analysis. In [3] a pattern trees based approached is used which capture the SPARQL graph patterns as a query execution plans. Besides this many transformation rules for pattern tree are proposed and enumeration and counting problems are used for class of queries. Static optimization techniques, based framework for SPARQL queries is proposed [4]. This framework is aimed to reduce the intermediate result sets of triple patterns by enables selectively estimation a pattern ranking according to intermediate result set size. [5] is proposed which avoids the integration and transformation of RDF data into SQL based data. It introduce the RDF_MATCH, a SQL table function, is used by SQL to query RDF data which combine it to traditional relational data, and finally resultant query is executed using B tree indexes and subject-property materialized views. An algebraic approach is proposed in [6] which study the rewrite rule scheme and classic chase algorithm in SPARQL for the optimization by showing the OPTIONAL operator is the responsible for PSSpace based query evaluation. An algorithm for query optimization proposed [8] which uses structural query term index to identify the relevance Semantic data sources. It uses bottom up approach to estimate the selectivity of each relevant node using reformulation of a conjunctive query and then join these node independently.

A large RDF graph suffers from problems like simple scan become complex, make hard of selectivity of pattern. SIP [8] is proposed to reduce scan costs by which uses scan index to skip the tuples given domain and qualifying tuples are stored in B+ tree to organize the estimation data. A heuristics technique for main memory graph implementation can be used to optimize the SPARQL queries [9]. A set of heuristics is created by selectivity estimation using summary statistic of RDF data which form the set of selectivity estimation of joined triple pattern. An RDF query engine is underlined for query optimization using query performance for these heuristics.

Requirement of manual interaction of RDF at some level increase the query execution time. A SPARQL query graph model [10] uses to overcome this problem. This model is implemented using Starburst database management system which is used to store the ongoing query information. A query is rewritten using transformation rule which in turn develop the heuristic to achieve an efficient query execution plan. Formalization of automatic rewrite rules which covers the challenges like to select options which are natural to semantic perspective and accommodate the cross query effect of rules which eliminate effect by other queries and rules [11]. [12] Propose a two level query suggestion model. This model builds query similarity graph using low rank query latent feature space. It uses on line ranking method for query, graph and latent similarity that is relevant to user query. Semantic data has been published in large amount and can be found anywhere in the web. DARQ [13] uses multiple SPARQL services to perform the query while it gives the illusion as it is working on single RDF graph while. It subdivide the query into sub queries then speeds up the query execution by rewrite and optimize it using cost based model. To store RDF data Jena, Seasame etc. infrastructure is used but they are not supported integrated querying of distributed RDF repositories.

A heuristics based algorithm for multi-query optimization

Volume : 3 | Issue : 2 | February 2013 | ISSN - 2249-555X

based on NP-hardness [14]. It creates groups of given queries by discovering the common sub-structure. Then optimization performs using effective cost model to compare the execution plan across the different RDF stores. A distance-based record linkage doesn't use parameterization. A new method is proposed for distance based record linkage which uses weighted mean and OWA operator which uses parameterization for optimization and permits the linking of records by their closeness of distributed database [15]. [16] Describes entity search which uses the Ranking method. A Vertical index groups the same weight properties and then text retrieval performance is measure by query expressivity by ranking the result. An engine for complex queries is based on MapReduce based system [17]. It translate the SPARQL to Pig Latin and a cost model is developed by interleaves the query optimization and execution. Data samples and statistics of previous step are lead to the next step of step of execution. Estimation of result set and skewing of join key done by methods which also resist the skewing of joins. Due to distributed nature of data RDF query requires number of joins from different repositories.

To reduce the number of join and effective resource selection framework FedX [18] is developed which allowing virtual integration of heterogeneous of Linked data cloud. It groups the triple patterns at endpoint by sent request to federation member which evaluate then hence minimize the join requests. A data graph PIG [19] partitioning the data. A group of triple elements is created which share the same structure and develop an index whose size is controlled by means of parameter. A query is boosted by choosing index first which is smaller than data graph. In Electronic Commerce large amount of data requires fast processing of query. RCQ-GA (20) a genetic algorithm is devised that can efficiently evaluation of RDF chain queries by determine their order. It performs consistently even with more complex query.

3. ISSUES

In previous section different query methods are discussed. Main emphasis of these methods is to optimize the query process in order to execute it efficiently. Different domain uses the RDF data which affect the query methods. Some query methods work for particular domain as we have discussed the RCQ-GA genetic algorithm which is for electronic commerce data. Some the methods work for general data. Methods in section 2 falls in second categories hence some common issues are discussed for these methods.

1. As data increases hence join also increases and requires large amount of memory which hinder the scalability in distributed data environment. These case rule out the cost based optimization. Questions of scalability can affect Semantic Web technologies in many areas.
2. Redundancy can occurs when integrate the data and produce the result set hence need to be reduced.
3. Design and development of efficient and precise data discover strategy required.
4. Irrelevance can occur in the graph so relevant sub-graph makes the query efficient.
5. Schema difference in different repositories requires additional processing in query optimization hence increase the cost of execution.
6. Current RDF triples does not support fuzzy queries.
7. RDF data in most cases is static and query is performed on single machine or using neighbor system hence does not deal with real life distributed network problems like latency, congestion etc.

4. CONCLUSION

In this paper we discuss different query methods and issues in RDF data. Query methods are based on index the structure, ranking the result, heuristic set, Genetic algorithm, tree based organization, partitioning the data etc. Some of them deal with distributed environment in multi query execution whereas other deals with static data. Some common issues are also discussed with these query methods like scalability, irrelevancy and extensibility. Apart from this it requires automate learning, however some methods have been proposed in literature, using AI methods.

REFERENCE 1. Thomas B. Passin, "Explorer's Guide to the Semantic Web", MANNING Greenwich, 2004. | 2. Fabian Prasser, Alfons Kemper and Klaus A. Kuhn, "Efficient Distributed Query Processing for Autonomous RDF Databases", EDBT 2012, Berlin, Germany, 2012. | 3. Andrés Letelier, Jorge Pérez, Reinhard Pichler and Sebastian Skritek, "Static Analysis and Optimization of Semantic Web Queries", PODS'12, Arizona, USA, 2012. | 4. Abraham Bernstein, Christoph Kiefer and Markus Stocker "OptARQ: A SPARQL Optimization Approach based on Triple Pattern Selectivity Estimation", Technical Report No. ifi-2007. | 5. C Hong, E. I., Das, S., E Adon, G., And Srinivasan, J., "An efficient SQL-based RDF querying scheme", In VLDB (2005). | 6. S Chmidt, M., M Eier, M., And Lausen, G., "Foundations of SPARQL Query Optimization", In ICDT (2010). | 7. Y. Li and J. Heflin., "Using reformulation trees to optimize queries over distributed heteroge-neous sources", In ISWC, pages 502–517, 2010. | 8. N Eumann, T., And Weikum, G., "Scalable join processing on very large rdf graphs", In SIGMOD, pp. 627–640, 2009. | 9. M. Stoker, A. Seaborne, A. Bernstein, C. Keifer, and D. Reynolds. SPARQL Basic Graph Pattern Optimizatin Using Selectivity Estimation. In WWW, 2008. | 10. H Artig, O., And Heese, R., "The SPARQL Query Graph Model for Query Optimization", In ESWC (2007). | 11. Zhuowei Bao, Benny Kimelfeld and Yunyao, "Automatic Suggestion of Query-Rewrite Rules for Enterprise Search", Li, SIGIR USA, 2012. | 12. Hao Ma, Haixuan Yang, Irwin King and Michael R. Lyu, "Learning Latent Semantic Relations from Clickthrough Data for Query Suggestion", CIKM'08 , 2008. | 13. Bastian Quilitz and Ulf Leser, "Querying Distributed RDF Data Sources with SPARQL", ESWC'08 Proceedings of the 5th European semantic web conference on The semantic web: research and applications, Pages 524-538, Springer-Verlag Berlin, Heidelberg, 2008. | 14. Wangchao Le, Anastasios Kementsietsidis, Songyun Duan and Feifei Li, "Scalable Multi-Query Optimization for SPARQL", Data Engineering (ICDE), 2012 IEEE 28th International Conference, 2012. | 15. Peter Teu and Gunther Lackner, "RDF Data Analysis with Activation Patterns", In 10th International Conference on Knowledge Management and Knowledge Technologies, 2010. | 16. Roi Blanco , Peter Mika , and Sebastiano Vigna, "Effective and Efficient Entity Search in RDF data", SWC'11 Proceedings of the 10th international conference on The semantic web, springer-Verlag Berlin, Heidelberg, 2011. | 17. Spyros Kotoulas, Jacopo Urbani, Peter Boncz and Peter Mika, "Robust Runtime Optimization andSkew-Resistant Execution of Analytical SPARQL Queries on Pig", International Semantic Web Conference.2012. | 18. Andreas Schwarte, Peter Haase, Katja Hose, Ralf Schenkel and Michael Schmidt, "FedX: Optimization Techniques for FederatedQuery Processing on Linked Data" | 19. Thanh Tran and Guunter Ladwig, "Structure Index for RDF Data", Workshop on Semantic Data Management, 2010. | 20. Alexander Hogenboom, Viorel Milea, Flavius Frasincar, and Uzay Kaymak, "RCQ-GA: RDF Chain Query Optimization using Genetic Algorithms", |