Statistics Based Optimization Technique for Query Processing

Rubin Thottupurathu Jose, Sojan Lal Poulose

Abstract*: Semantic web consists of the data in the structure manner and query searching methods can access these structured data to provide effective search result. The query recommendation in the semantic web relevance is needed to be improved based on the user input query. Many existing methods are used to improve the query recommendation efficiency using the optimization technique such as Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO). These methods involve in the use of many features which are selected from the user query. This in-turn increases the cost of a query in the semantic web. In this research, the query optimization was carried out by using the statistics method. The statistics based optimization method requires fewer features such as triple pattern and node priority etc., for finding the relevant results. The LUBM dataset contains the semantic queries and this dataset is used to measure the efficiency of the proposed Statistical based optimization method. The SPARQL queries are used to plot the query graph and triple scores are extracted from the graph. The cost value of the triple scores is measured and given as input to the proposed statistics method. The execution time of the statistics based optimization method for the query is 35 ms while the existing method has 48 ms.

***Index Terms*: LUBM dataset, Semantic web query, SPARQL, statistics based optimization and triple scores.**

I. INTRODUCTION

The Recourse Description Framework (RDF) [1], the Web Ontology Language (OWL) [2] and the Linked Data Principles are proposed for the World Wide Web (W3C) community to manage the data and share them in the structured and semantic related way. The Semantic Web tool and RDF data are used to describe the web data and make them publicly available to the user [3]. If the publisher provides the live query access to the databases, the query search default choice is SPARQL endpoint [4]. Federated SPARQL query engine answer SPARQL queries based on the SPARQL endpoints. Query processing is the process of decreasing the time for the answer to the user without affecting the performance. Query cost can be minimized by selecting the relevant features of the query for the answer [5, 2]. Finding the optimal web services in the available web services is known as Web service optimization problem in the community of service computing. [6].

The RDF nodes are developed using the data which is distributed in the graph over the storage nodes for the scaling process that requires more query processing and memory [7]. RDF represents the web data in triple (subject-predicate-object) model. SPARQL has a triple pattern of main component that makes it easy to match RDF triples, varying triple patterns are filtered using the Boolean conditions [8]. Many methods are developed to improve the efficiency of the query at low cost by optimizing the query [9 – 10]. The existing method uses the optimization technique for finding the relevant queries in semantic web. The optimization techniques such as PSO uses many features to identify the relevant query. The number of features tends to increases the query cost and memory storage. The proposed statistics method involves the use of statistics based optimization method in query optimization. The statistics method involves in the use of few features like triplet score. The statistics based optimization method helps to reduce the query cost of the semantic web.

The organization of the paper is given as follows, Literature survey is presented in section II, the proposed method is explained in section III, experimental result and discussion are given in section IV and the conclusion of this research is made in section V.

II. LITERATURE SURVEY

The recent researches involves in query optimization in the semantic web were analyzed in this section.

Franck Michel, et al. [11] developed the SPARQL interface for the heterogeneous databases for providing the relationship between the Semantic Web and NoSQL worlds. The two-phase method is developed that eliminates the needs of the translation between the each and every database. The first step involves in converting the SPARQL query into pivot abstract query and secondly, the abstract query is converted into query language of large database, consider the specific database capability and constraints. The effectiveness of the method is high in the query optimization. The translation of the SPARQL query into the effective query is difficult due to the address data sources.

Giacomo, et al. [12] focused on the specific part in the sematic data integration known as Ontology-Based Data Access (OBDA). The OBDA is the advance technique for the semantic data integration, the global technique is provided in the manner of an ontology. The process in streaming data is measured as the capability of OBDA method to react to edit in the instance of ontology.

Revised Manuscript Received on August 02, 2019.

Rubin Thottupurathu Jose, School of Computer Sciences, M G University, Kottayam, Kerala, India.

Dr Sojan Lal Poulose, Principal, Mar-Baselious Institute of Technology and Science, Kothamangalam, Kerala, India.

The efficiency of the proposed method is low and query optimization technique can be applied to improve the performance.

Peng, et al. [13] established the technique to process the SPARQL queries in the large RDF graph in the distributed scenario. The method such as "Partial evaluation and assembly" architecture is applied in the SPARQL queries. Answering the query is equivalent to the finding the subgraph matches in the distributed graph, which introduces the partial answer based on match in each part of RDF graph. The two methods are proposed namely centralized and distributed assembly for query search. The method is analyzed experimentally and theoretically. The query cost value is high and need to be reduced.

Pham and Boncz [14] investigated the RDF data to store more compactly and efficiently to execute SPARQL queries. The efficient emergent schema method in the RDF storage is developed and the query operator method is used to analyze the scans and joins in the system. In all these techniques, RDF schema techniques are allowed to process with relational database techniques in the rich physical database in the manner of options and efficiency, without in needed of schema structure definition. The cost value of optimization technique is high and effective technique is need to optimize the query cost. Ibragimov, et al. [15] developed and tested Materialized Rdf Views with Entailment and incompleteness (MARVEL) in the semantic web. The method consists of view selection algorithm that is based on a syntax of view definition, RDFspecific cost model, and a technique for rewriting SPARQL queries based on the materialized RDF views. The investigation of the method shows that the MARVEL technique can increase the query response time. The algorithm has to be improved to incrementally maintain the materialized path in the presence of updates.

III. PROPOSED METHOD

The query optimization technique involves the use of many number of features to provide the recommendation. This method tends to increase the query cost and required more time to process the query recommendation. The proposed statistics based optimization method involves in the use of less number of feature for the query processing. The process of the proposed stastical based optimization method is explained in this section. The query graph is plotted based on the value of the SPARQL queries in the LUBM datasets. The triplet score is measured from the graph and given as input to the proposed statistics method. The recommended queries are provided by the proposed method in the low execution time. The block diagram of the proposed statistical-based optimization method is shown in Fig. 1.

A. LUBM datasets

The Lehigh University Benchmark (LUBM) [16] is the popular dataset available for the semantic web. The LUBM consists of the ontology briefing the universities with data generation and 14 queries. The test data consists of generated queries instance of ontology; the generation of data is repeatable and present in the arbitrary size. The 14 queries are present to test over the data. The triples values are extracted to identify the relevant queries for the system.

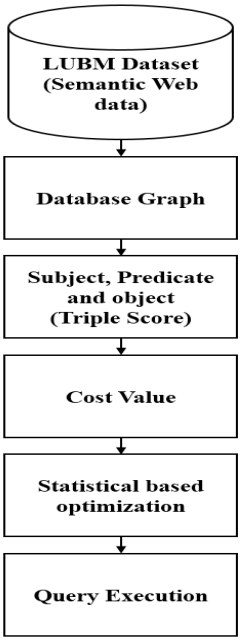

Fig. 1. The block diagram of statistics based optimization

B. SPARQL and Query Graph

An RDF dataset was present in the graph where subjects and objects were considered as vertices and predicates as labeled edges.

An RDF graph is expressed as $G = \{V, E, \Sigma\}$, where V is a set of vertices that denotes all subjects and objects in RDF data; $E \subseteq V \times V$ is a directed edges multiset that present to all triples in RDF data; Σ is a set of edge labels. For each edge $e \in E$, represents its corresponding property. Similarly, a SPARQL query is denoted in the query graph Q .

C. Triplet Score

The original technique allows to store and process the query of RDF data with the SQL systems, but in that case the SQL query answers for only "regular" triples that suitable for the relational tables. The objective of the method is to provide the 100% relevant answer to the SPARQL queries and minimize the execution time of the query.

RDF systems stores the triple tables T values in the multiple order of Subject (S), Predicate (P) and Object (O), among which typically TP SO ("column-wise"), TSP O ("row-wise") or even all permutations. The RDF system storage can be improved by changing the order of the TP SO representation. The triple values that do not fit in the PSO table is T_{pso} . The TP SO value is replaced by the smaller T_{PSO} table and a set of relational tables [17].

Triple data of relational storage is applied and though these prior approaches investigate an explicit and human controlled map to a relational schema. The relational RDF method has high performance, remained vulnerable for the SPARQL queries that don't involve on star patterns.

D. Cost Value

Consider a query graph Q with n vertices $v_1,...,v_n$ and a partitions $P=\{P_{v_1},...,P_{v_n}\}$ in the local partial matches set Ω, the join cost is measured in the Eq. (1).

$$Cost(\Omega)=O\left(\prod_{i=n}^{i=1}(|Pvi|+1)\right) \quad (1)$$

Where $|Pv_i|$ is the count of local partial matches in Pv_i and 1 is introduced to avoid zero element in the product. Assume that each pair of local partial matches are joinable that can qualify the worst-case performance. More sophisticated cost value can be applied, this is difficult to identify the low cost.

E. Statistics Based Optimization

Joints are the most expensive operator in the method due to the large portion of the data is need to shuffle in the graph. The joints order is important to limit this problem and this speedup the calculation. The effective technique for sorting the joints is to use the statistics in the input graph. The Join tree decides the order in which the operations are need to be performed. The statistics that are applied are simple and efficient in practice, are the total number of triples and number of distinct subjects for each predicate. The Join tree node priority value is measured using the following criteria:

- Triple patterns consist of literals that are scored with the highest priority. The availability of a literal is a high constraint that limit the resulting tuples. Therefore, it is a good method to down the order of the nodes.
- A triple value of which the underlying data contains many tuples that are calculated proportionally. For example, the triple with the more number of tuples have a low priority and this is considering as tree root. These values are turned depend on the number of distinct subject for that predicate.
- The node priority of the data belongs to the Property table that is measured while considering all its triple patterns. However, triple pattern presence is highly weighted.

Along with the statistical method, Spark SQL's Catalyst Optimization method is used with its internal heuristics to improve the query performance further. The trees are not continuously changed, but Spark intervenes for optimized physical plans, it has the concrete location for the data on the cluster. In particular, the optimizer chooses the joins types to process the query, for example if one relations are small, a broadcast join will be performed.

IV. Experimental Result

The sematic query recommendation involves in high cost and this method doesn't provide much effective query for the user. Then, optimization techniques were applied to decrease the cost value and increase the relevant recommendation. The different optimization techniques were used for the query optimization in RDF distribution and these methods are involving in high-cost value. In this research, Statistics based optimization method is applied to minimize the cost value of query. The eclipse java 1.8 apache jena was used for experimental simulation with 3.2 GHz and i5 processor. The proposed statistical-based optimization method and existing methods are simulated in the same environment and compared with each other. The execution time and memory usage of the proposed methods are analyzed in this section.

A. Execution time

The existing query optimization techniques such as RDF-3X [13], PECA [13] and PEDA [13] are compared with statistics based optimization. The execution time of the statistics based optimization and other existing methods are shown in Table 1. This shows that the proposed method has a lower execution time compared to the other existing techniques. The statistics based optimization techniques uses the mathematical equation for the finding queries while other existing method involves in use of a complex algorithm. So, the statistics method has a lower execution time compared to the existing method.

Table I. Execution time of query optimization method

Execution time (milliseconds)				
Number of queries	**RDF-3X [13]**	**PECA [13]**	**PEDA [13]**	**Statistics based Optimization**
1	10,840,47	3,26,167	3,09,361	38
2	81,373	23,685	23,685	54
3	72, 257	10,239	10,368	35
4	7	753	753	27
5	6	125	125	39
6	355	3388	1914	72
7	1,46,325	1,43,779	46123	76

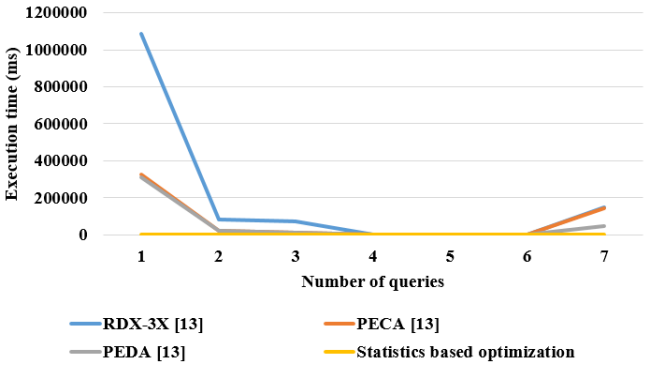

Fig. 2. Execution time of statistics and existing method

The execution time of the existing method and proposed statistics methods are compared in Fig. (2). The execution of the statistics method is low compared to other existing method in query optimization. The proposed statistics based optimization method has low execution time due to the optimization is involves in the statistics measure while existing method involves in complex methods. This shows proposed statistics method has low execution time compared to the other existing methods.

Table II. Comparison of different methods

Execution time (milliseconds)				
Number of query joins	**MARVEL [15]**	**MonetDB -triple [14]**	**MonetDB -emer [14]**	**Statistics based Optimization**
1	2400	480	50	38
2	1800	500	480	54
3	2100	470	48	35
4	3300	520	40	27
5	1200	50	48	39
6	2100	500	480	72
7	2400	420	300	76
8	1300	500	480	81

The statistics method in the query optimization is implemented in the LUBM dataset in several queries and execution time is calculated. The existing method such as MARVEL and MonteDB-emer method is compared with the proposed statistics method, as shown in table 2. This shows that the statistical method has a lower execution time compared to the other existing method. The proposed statistics method recommend the query based on the simple factors of query joints, triple pattern and node priority. So, the statistics method has lower execution time compared to the existing method.

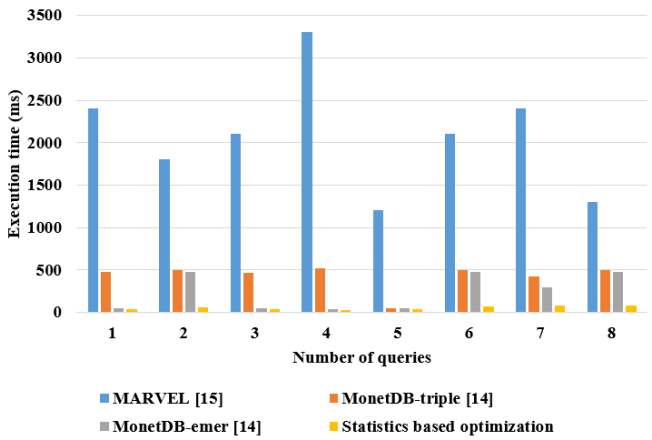

Fig. 3. Comparison of statistics and existing methods

The execution of the various method in the query optimization is implemented in the LUBM dataset and compared in Fig. (3). The existing techniques such as MARVEL and MonteDB-triples are compared with proposed statistics method. This shows that proposed statistical method has a low execution time compared to other existing methods. The proposed statistics based optimization is required only a few features in the query recommendation. So, the execution time of the statistics method is low compared to other existing method. The proposed method provides the query with low cost compared to the existing method.

B. Memory Size Analysis

The memory usage of the existing and statistics methods in query optimization are compared in the Table. 3. This shows that the proposed statistics method has lower memory usage compared to MARVEL method. The proposed method needs the storage space of 24.6 MB while existing method has 364 MB. The proposed statistics method need to store only a few features from the query and this in terms needs low space of storage.

Table III. Memory usage of MARVEL and statistics

Methodology	Memory usage
MARVEL [15]	365 mb
Statistics based Optimization	24.6 mb

Hence, the statistics method has lower execution time and require low storage space for the query optimization. The proposed statistics method has lower execution time than the existing method in query optimization in RDF distribution.

V. Conclusion

Semantic web query optimization techniques are applied to provide an efficient query recommendation. Since optimization techniques used for the query recommendation that uses more number of feature and the query cost is high. In this research, the statistics based optimization technique is used to minimize the execution time of the query. The LUBM dataset is used to measure the performance of the proposed statistics method. The graph is plotted based on the SPARQL queries and the triple scores are measured from the graph. The proposed statistics method selects the features like triple score etc.., to process the query in the low cost. The proposed statistics method is compared with the existing method in query optimization and this shows that the proposed statistics method has low execution time. The proposed statistics method has the execution time of 27 ms for the query while the existing method has the execution time of 48 ms. The proposed statistics method has low execution time and required low storage space. The contribution of proposed statistics based optimization are given below

- The proposed statistics method has low execution time compared to the other existing method in query optimization because the proposed statistics method uses only fewer number features for optimization.
- The statistics based optimization method uses less number of features and this requires less storage space. The proposed method has the capacity to process more queries in low time and required less storage space.

The future work of the proposed method involves in implementing the real-time system and to evaluate its performance.

REFERENCES

1. C. Nikolaou, and M. Koubarakis, (2016). Querying incomplete information in RDF with SPARQL. *Artificial Intelligence, 237, pp. 138-171.*
2. Z. Gu, S. Zhang, and C. Cao, (2019). Reasoning and querying web-scale open data based on DL-LiteA in a divide-and-conquer way. *Journal of Web Semantics, 55,* pp. 122-144.
3. M. Acosta, E. Simperl, F. Flöck, and M. E. Vidal, (2017). Enhancing answer completeness of SPARQL queries via crowdsourcing. *Journal of Web Semantics, 45,* pp. 41-62.

4. J. Van Herwegen, R. Verborgh, E. Mannens, and R. Van de Walle, (2015). Query execution optimization for clients of triple pattern fragments. *In European Semantic Web Conference*, pp. 302-318. Springer, Cham.
5. G. Montoya, H. Skaf-Molli, and K. Hose. (2017). The Odyssey approach for optimizing federated SPARQL queries. *In International Semantic Web Conference* Springer, pp. 471-489.
6. Z. Chouiref, A. Belkhir, K. Benouaret, and A. Hadjali, (2016). A fuzzy framework for efficient user-centric Web service selection. *Applied Soft Computing, 41*, pp. 51-65.
7. D. Janke, S. Staab, and M. Thimm, (2018). Impact analysis of data placement strategies on query efforts in distributed rdf stores. *Journal of Web Semantics, 50*, pp. 21-48.
8. J. M. Almendros-Jimenez, A. Becerra-Terón, and G. Moreno, (2018). Fuzzy queries of social networks with FSA-SPARQL. *Expert Systems with Applications, 113*, pp.128-146.
9. E. G. Kalayci, T. E. Kalayci, and D. Birant, (2015). An ant colony optimisation approach for optimising SPARQL queries by reordering triple patterns. *Information Systems, 50*, pp.51-68.
10. J. Schoenfisch, and H. Stuckenschmidt, (2017). Analyzing real-world SPARQL queries and ontology-based data access in the context of probabilistic data. *International Journal of Approximate Reasoning, 90*, pp. 374-388.
11. F. Michel, C. Faron-Zucker, and J. Montagnat, (2019). Bridging the Semantic Web and NoSQL Worlds: Generic SPARQL Query Translation and Application to MongoDB. *In Transactions on Large-Scale Data-and Knowledge-Centered Systems XL*, Springer, pp. 125-165.
12. G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, and R. Rosati, (2018). Using ontologies for semantic data integration. *In A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years*, Springer, pp. 187-202.
13. P. Peng, L. Zou, M. T. Özsu, L. Chen, and D. Zhao. (2016). Processing SPARQL queries over distributed RDF graphs. *The VLDB Journal—The International Journal on Very Large Data Bases, 25(2)*, pp. 243-268.
14. M. D. Pham, and P. Boncz, (2016). Exploiting emergent schemas to make RDF systems more efficient. *In International Semantic Web Conference*, Springer, pp. 463-479.
15. D. Ibragimov, K. Hose, T. B. Pedersen, and E. Zimányi, (2016). Optimizing aggregate SPARQL queries using materialized RDF views. *In International Semantic Web Conference).* Springer, pp. 341-359.
16. Y. Guo, Z. Pan, and J. Heflin, (2005). LUBM: A benchmark for OWL knowledge base systems. W*eb Semantics: Science, Services and Agents on the World Wide Web, 3(2-3)*, pp. 158-182.
17. M. D. Pham, and P. Boncz, (2016). Exploiting emergent schemas to make RDF systems more efficient. *In International Semantic Web Conference*, Springer, pp. 463-479.

AUTHORS PROFILE

Rubin T Jose, MCA, M Tech. Scholar in the area of Semantic web and Ontology Engineering, already published 3 Journal Papers and 8 Conference publications in the area. Attended short course in the Protégé tool in Stanford University, USA. He has got a total of 15 years of teaching experience in the field of Computer Science and Engineering.

Dr P. Sojan Lal has more than 30 years of blended experiences, with major international petroleum companies in Middle East and premier educational institutions in India. He has authored 6 technical books, two of them published in Germany and other books published in India. He is an approved research supervisor of Mahatma Gandhi University, Kottayam; University of Petroleum and Energy Studies, Dehradun; and APJ Abdul Kalam Technological University, Kerala, India.

Dr. Sojan Lal has authored 65 National and International Journal and Conference papers and guided 4 PhD scholars. He has the world record for the highest number of publications within shortest period in 2014. He has been listed in "Marquis Who's Who in the World" since 2009, representing the world's most accomplished individuals.

Qualifications: BE (Mechanical Engineering, 1985), M.Tech(Computer Science, 1993), PhD (Faculty of Technology, 2002), MBA(2011, UK), DBA(2018, USA)