SPARQL Optimization Using Heuristi Approach

Dhanushri Varshney[1], Rupal Gupta[2]
[1]*Student of MCA, CCSIT, Teerthanker Mahaveer University, Moradabad*
[2]*Assistant Professor, CCSIT, Teerthanker Mahaveer University, Moradabad*
[1]dhanushree.varshney@gmail.com
[2]rupal.gupta07@gmail.com

Abstract— **The concept of Semantic web was introduced by Tim Berner's Lee, inventor of the WWW, URIs, HTTP and HTML. The semantic web is a development and addition of the existing web that allow computer to manage information and data. The idea of the semantic web is still undergoing research and development. There is a great demand in a web that has the impending proficiency to 'discern' and 'comprehend'. In Semantic web the data is stored in RDF (Resource Description Framework). RDF is a mark-up language used for describing information and resources on the web. A language is SPARQL (simple protocol and RDF query language). It is basically used for retrieving an RDF data. In this paper we have illustrated SPARQL semantic along with the approaches used for optimizing SPARQL query also shown comparative study of SPARQL and SQL twinkle tools for better understanding. Further we have shown optimization comparison of SPARQL and SQL using heuristics rules along with cost based optimization technique.**

***Keywords*— SPARQL, RDF, Query Optimization, Heuristic rules.**

I. INTRODUCTION

A. *Semantic Web*

Semantic web was an unique idea twisted by Tim Berner's Lee designer of the WWW, URIs, HTTP and HTML. The semantic web is an additional version of the current web that offer an easier way to find , share, reuse and combine information[3]. The aim of the Semantic Web is to provide a universal medium and it also provide a common formats for the interchange data. Semantic Web is about making the Web more understandable by machines. Semantic web provide the tools and mechanism with advanced searching types on web.

B. *SPARQL*

SPARQL(simple protocol and RDF query language), a W3C commendation, is a pattern-matching query language. SPARQL is designed for managing data over the semantic web. SPARQL is use to retrieve and data stored in Resource Description Framework (RDF). It is also called RDF query language or Semantic query language. It provides a method to represent constraints and facts and the entities returned to the user those are matching with the constraints. SPARQL query language is much similar to SQL query language because it uses several keywords such as SELECT, WHERE etc. [4]. It also has new keywords such as OPTIONAL, FILTER and much more. SPARQL 1.0 is the first version of SPARQL and SPARQL 1.1 is the additional feature of SPARQL.A SPARQL query is executed to retrieve data from RDF.[5]

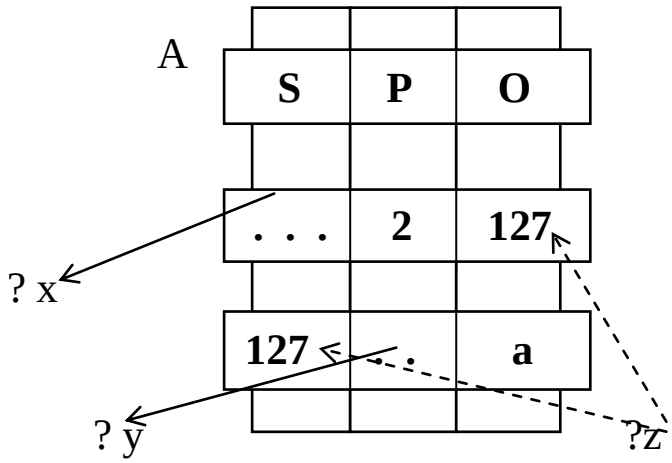


Figure 1: Example of SPARQL

SPARQL Query:

```
SELECT ?x ?y
from A
WHERE { ?x  2  ?z.  ?z  ?y  a. }
```

In SPARQL query the first keyword is SELECT ?x ?y ?z are the triple component variable. Afterwards, we catch the WHERE keyword which is shadowed by the triple pattern. This triple is the most interesting part of the query. The triple in the queries must contain of s-o-p. In SPARQL s-o-p means subject-object-predicate. Where subject & object are the variable but predicate is the constant value. This triple in the queries is checked beside

all the RDF triple in which data is stored in the RDF file and then filtered the data as per the constraint defined.

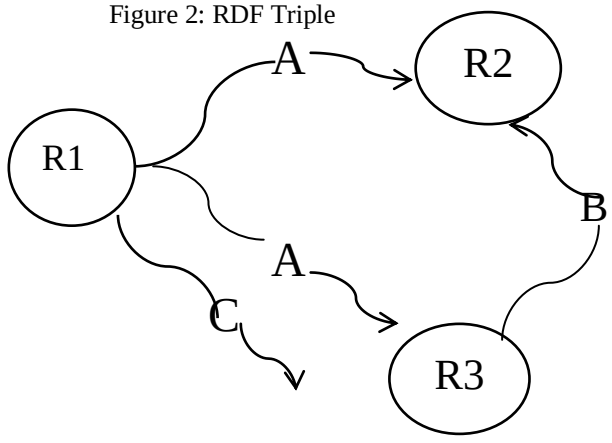
c. RDF

RDF (Resource Description Framework) is the data model of the Semantic web. It means data in a Semantic Web tools it is denoted by a RDF. RDF defines the resources which are presented in web. In RDF data model can be stowed in a universal format. RDF is originally designed as a metadata model. The RDF web resource in the procedure of s-o-p terms[6]. s-o-p means subject-object-predicate. These terms also called triples in the RDF terminology. In RDF the subject represents resource and the predicate represents the aspects of the resource. It is also show the relationship between subject and object[6]. For example "The sky has the colour blue" in the RDF a subject is representing "the sky", predicate representing "has" and object representing the "the colour blue".

Triple Representation:

Subject	Predicate	Object
R1	A	R2
R1	A	R3
R3	B	R2
R1	C	D

Figure 2: RDF Triple



D

Figure 3: RDF Graph Triple

II. Comparison of SQL and SPARQL

S.No	SQL	SPARQL
1.	SQL based on Tuple Relation Calculus.	SPARQL based on Triple Relation Calculus.
2.	SQL is designed to query Relational data.	SPARQL is designed to query RDF data.
3.	In SQL char, varchar, number, long etc. data type use.	In SPARQL subject, predicate, object, uri, literal etc. data type use.
4.	In SQL data access from Table.	In SPARQL data access from RDF data files.
5.	Relational data model store data in the Structured form.	RDF data is stored in the Unstructured form.
6.	Syntax: select< column_list > from< table_list > where< condition >	Syntax: Select < variable_list> where{< graph_pattern>}
7.	Example: select emp_id, salary from emp;	Example: select ?id ?sal where { ?id HR: salary ?sal }

III. SPARQL Optimization

This section focus on SPARQL Optimization. We describe the general optimization rule considered in our optimization framework which are used to rewrite SPARQL queries in order to get an optimized. Rules include a prepare and a transform stages. Component generates an execution plan for the query. In SPARQL the query optimizer is a key feature of our system it optimize the SPARQL queries. In SPARQL many type of query optimization approaches use. Query processing is optimized by evaluating the query patterns in an effective manner. Triple pattern are set in an order such that the equivalent result of a pattern server as input for the next pattern plan. Since the result of each pattern is checked for validity at every processing step, the number of intermediate result is substantially reduced.

ISBN-978-93-5288-834-3

IV. APPROACHES OF SPARQL OPTIMIZATION

A. Heuristics Rules

We will enlist the process of SPARQL query optimization based on Heuristics rules. The area of query optimization is very largest and most complex. Two key component of the SPARQL system are query optimizer and query execution[1]. It is basically tries to minimize the number of triples. Heuristic optimization is less expensive than that of cost based optimization[7][8]. Heuristics rules can be useful to multiple type of SPARQL query. With the help of Heuristic rule we can easily optimize a query and save a lots of time.

- H1: Minimise optional graph pattern: In SPARQL query we can easily decrease many triple pattern via finding triple pattern for a specified queries that will be sure with dataset. The SPARQL graph expression assumed the combination of all the triple pattern and assembled in the situations will be calculated in O(|P| . |D|) times, where the several triple pattern in SPARQL queries is |P| and the several RDF tuple in dataset |D|. In this example we can minimise the graph pattern.

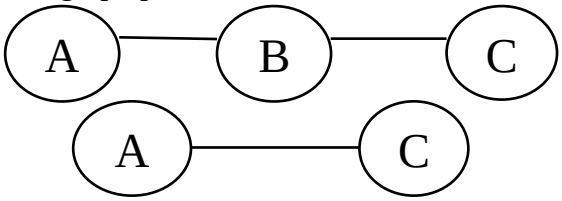


Figure 4: Example of H1 rule

- H2: Used named graph to contain SPARQL sub-graph pattern: The SPARQL we can used named graph to identify the subgroup of triple in dataset that shares a queries should be calculated against. The runtime presentation of every SPARQL queries has a progressive relationship to the several RDF triple calculated against for example

```
select *
from m1
from m2
where { ?A s ?B . ?A t ?C }
```

If predicate s occur in m1 and m2 and predicate t occur in m2. In the query we can modified as:

```
select *
from m1
from m2
from named m2
where { ?A s ?B . { m2 GRAPH { ?A t ?C }
} }
```

GRAPH operator use in the above query. GRAPH operator decrease the rate of the calculating the triple pattern { ?A t ?C }.

- H3: Reduce intermediate result: In SPARQL we use sequence path to exchange the linked of the triple pattern means the object of any triple pattern is use as the subject of any other triple pattern. It allow to require path of arbitrary size between two graph nodes. If a € I, then a is the property path. If the p1 & p2 is the property path formerly [p1]*, [p1/p2], [p1]? , [p1|p2] and [p1]+ all are also the property paths. [p1/p2] is also called sequence path. Here we use a method to increase the performance of any queries created on reduction of the intermediate result by using property path expressions. For example

```
select ?E ?F ?H
where
```

{ ?E s ?F . OPTIONAL { ?F t ?G . ?G u ?H } }

In the above queries we can understand the variable ?G is not essential outer the sub graph patterns { ?F t ?G . ?G u ?H }. If we change the sub graph patterns by system path (?F, [t/u], ?H) equally a approach to clean variable ?G. Its agree to decrease the amounts of intermediates result and decrease the cost of subsequent operation and also develop the calculation of the queries. Property path is used in project intermediate variables.

- H4: Reduce the properties of Cartesian products: The SPARQL we can used aggregate function which is used to decrease the size of the resolution sequences. In the heuristic, we suggested method to decrease redundant value arising in a solution sequence. The several solutions in an order is specified via the product of the several mapping found for every variable in the queries for example

  μ1: { μ1 (?s) = i, μ1 (?p) = j, μ1 (?o) = k }
  μ2: { μ2 (?s) = i, μ2 (?p) = j, μ2 (?o) = l }

  In the above example we can see that the variable ?s and the variable ?p value is same in both mapping but the variable ?o are different.
  In SPARQL1.1 present the set of seven aggregate function that pool the set of mappings for the related variables MAX, SUM, COUNT, AVG, MIN, GROUP_CONCAT and SAMPLE .

- H5: Specifying alternative URIs: In SPARQL use many ways of requiring the alternate URIs past UNION. The SPARQL design[2] a commends it use the UNION keywords that means match one or more alternatives graph pattern. For example

  ```
  select ?text
  where {
    { < uri1 > text ?text }
   union
    { < uri2 >text ?text }
  }
  ```

  In above query mapping will holds texts subordinate with either < uri1 > and < uri2 >.

B. Cost Based Optimization

In SPARQL we can use cost based query optimization. In cost based query optimization creates all the query execution plans. The cost of all the plan is predictable. In cost based query optimization select best estimation, plan with lowermost expected cost [7]. In cost based optimization quality depend on the complication and correctness of cost function use. It contain many different technique such as dynamic programming select for greatest plan. Its main disadvantage is that it is very costly. In cost based optimization two main points.

- Creates all possible query execution plans and then calculate cost.
- Quality depend on complexity and correctness of cost function.

C. SPARQL Query Rewriting

In SPARQL query it is likely to define instructions for converting query from one schema to another schema and also required from one data set to another data set. We extant a SPARQL query modifying the method which is used to accomplish

interoperability in semantic info retrieval and the knowledge discovery procedures over inter related the RDF data source. It presents SPARQL rewriting, a framework which provides transparent query access over RDF dataset. The approaches to data combination are comparable to the one agreed in the earl data management system[10] where query can be modified in the several phases, dependent on where the queries will be implemented. In the method we can classify the queries, source ontology used to express the query and data set as an input for executing on the data combination. The modified queries that fit mark ontology or the data set is the output of the way.

V. EXECUTION TOOLS OF SPARQL QUERY

Many types of tool presented for executing SPARQL query. Twinkle, Apache Jena Fuse Ki, Redland etc. are some of the tools which is used to execute the SPARQL query. In this paper we have analysed "Twinkle" tools used for executing the SPARQL query. To use these tools require jdk1.5 or advanced versions of java.

A. *Twinkle Tool*

Twinkle is a SPARQL query tool. It is used to execute the SPARQL queries. Twinkle is a GUI interface that stoles the ARQ SPARQL query engines. It is used to load, insert and save the SPARQL query.

Steps

- Download the Twinkle2.0 version.
- Then open command prompt & extent controlled folder of the twinkle.
  Example: E:\twinkle>
- Then execute the command
  E:\twinkle>java –jar twinkle.java
- "Twinkle SPARQL tool" window will open and then execute the SPARQL queries[11].

EXAMPLE:- The query retrieve the student_name, course, subject, marks. Student marks is greater than and equal to 70.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-
rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?student_name ?course ?subject ?marks
WHERE {
  ?x rdf:type foaf:Person .
  ?x foaf:studentname ?student_name .
?x foaf:course ?course .
?x foaf:course "MCA(lat)" .
?x foaf:subject ?subject .
?x foaf:marks ?marks .
FILTER(?marks >= "70")
  }
```

Output :-

student_name	course	subject	marks
"MANSI"	"MCA(lat)"	"JAVA"	"72"
"DHANUSHRI"	"MCA(lat)"	"JAVA"	"85"
"NAMAMI"	"MCA(lat)"	"DBMS"	"88"

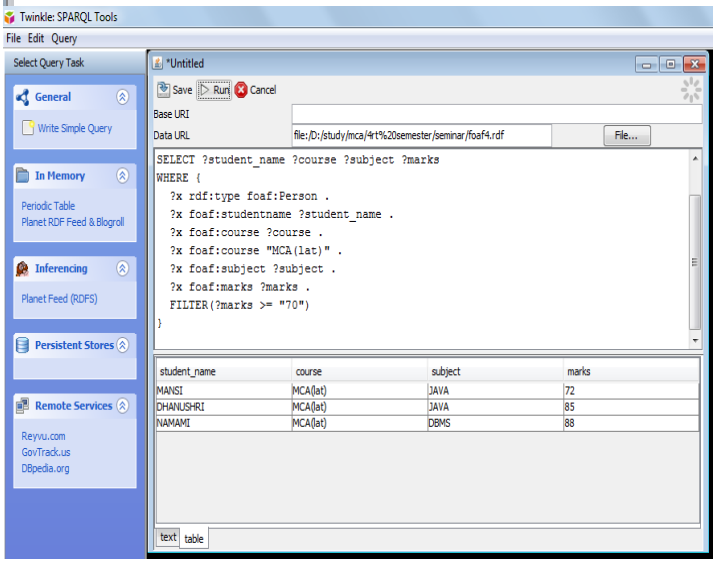


Figure 5: Execution Of Query Using Twinkle Tool

VI. CONCLUSION

SPARQL is a pattern matching query language it is used to RDF data store. In this

research paper we have discuss SPARQL semantic along with its comparison with SQL full batter understanding. Further we have shown the optimization need in various domains. In this paper we analysis different approaches of SPARQL optimization. Such as heuristic approach, cost based optimization approach and SPARQL query rewriting approach. In this paper we have also shown the various SPARQL tools. This tools help to execute the SPARQL queries.

REFERENCES

[1] "Query Optimizer plan Diagram: Production, Reduction and Application, Data Engineering (ICDE)", 2011 IEEE 27th International conference.

[2] E. Prud'hommeaux, A. Seaborne, SPARQL query language for RDF. W3C Recommendation, 2008. http://www.w3.org/TR/rdf-sparql-query/.

[3] Grigoris Antoniou , Frank Van Hermelen, "A Semantic Web Primer ." MIT press , Campbridge, England.

[4] Herman, I., Semantic Web Activity, W3C, 2007 http://www.w3.org/2001/sw

[5] Artem Chebotko, Shiyong Lu, Hasan M. Jamil, Farshad Fotouhi.. "Semantics Preserving SPARQL-to-SQL Query Translation for Optional Graph Patterns", Technical Report TR-DB-052006-CLJF, 2006.

[6] Olaf Hartig and Ralf Heese, "The SPARQL Query Graph Model for Query Optimization", Humboldt-University zu Berlin.

[7] David E. Goldschmidt and Mukkai Krishnamoorthy. "Architecting a Search Engine for the Semantic Web ," Renesselaer Polytechnic Institute, Troy, New York,USA.

[8] Yannis E.Ioannidis paper on "Query Optimization" Computer Sciences Department University of Wisconsin Madison, WI 53706 in 2011

[9] Leo Giakoumakis, Cesar Galindo-Legaria paper on "Testing SQL Server's Query Optimizer: Challenges, Techniques and Experiences" . IEEE Data Eng. Bull. 31(1): 36-43 (2010)

[10] A. Y. Halevy, Z. G. Ives, D. Suciu, and I. Tatarinov. Schema mediation in peer data management systems. In Proceedings of the 19th International Conference on Data Engineering, pages 505{516, 2003.

[11] R. Gupta, Sanjay Malik, "SPARQL Semantics and Execution Analysis in Semantic web using various Tools", CSNT, IEEE computer society, 2011.