A preprocessing strategy of RDF data query based on relational database

Yuanyuan Chen[1, a]
[1]Nanchang Institute of Science and Technology, 330108
[a]262488739@qq.com

Keywords: Resource description framework; Relational database; Query optimization; preprocessing; LUMB data set

Abstract. To improve the efficiency of the query on the data of resource description framework (RDF), the method of storage and management of the data at home and abroad is studied. According to the characteristic of the data that the number of subjects and objects is large and the number of attributes is small, a preprocessing of the storage of the data based on the relational database is proposed. The query first judge the property, and then performed, thus the efficiency of the implementation of the query is improved. Finally, the data was generated uses the LUMB data set, and then, the designed query is executed on it. The experimental results verify the validity of the method.

Introduction

The semantic web was proposed by Tim Berners-Lee, the father of the world wide web, in 1998[1], its key point is to build a centric network of data,so the computer can understand the data which in the document and its semantic relations,in order to query the data on the Internet to get the formatted answer. The data of resource description framework (RDF) is a data storage format which designed to implement the Semantic Web. It is usually descrided by a three tuple (subject, property, object).For example, a film named "Terminator Genisys" directed by Alan in 2015,which actor Clarke take part in, using RDF form can be expressed as:

(id1,hasName, “Terminator Genisys”),(id1,producedInYear, “2015”),

(id1,directedBy,id2),(id1,hasCasting,id3),(id2,hasName, “Alan”),and so on.

It can be seem from this example, RDF form is very flexible and generic, which can decompose and storage all information.

RDF query language and data acquisition protocol (SPARQL) [3] is the official language of the RDF data query, which can support a number of operations such as merge and connection.

This paper mainly includes three aspects: first, research and analyze the advantages and disadvantages of the existing RDF data storage management methods. Second, the property table is designed and built in the relational. Third, the data set is used to verify the preprocessing strategy.

Current situation and analysis of domestic and international research

RDF data is widely used in multiple platforms, although the tuples data can semantic, which very convenient, however it also has some disadvantages. First of all, as the number of tuples becomes very large, as the data table increases very fast; second, for the complex queries, there will be a large number of connection operation in the execution process, it will greatly reduce the efficiency of the query.

In order to improve the efficiency of storage and query, researchers have proposed a lot of methods to store RDF data. In the early Jena[4] used the attribute table to store RDF data, the attribute table can solve the problems in the tuple storage, because of lots of the attributes organized together, reduced the query connection operations, eliminated the tuple storage in self join operations and improved the efficiency of the query. However the attribute table also have some shortcomings, first of all, it is not possible that put all attribute which the query needed into a table, therefore, there are still some connection operation, and existed the merge operation; secondly,

because the data of RDF is not structured, so there may be a lot of null value in the property table and a lot of storage space may be wasted.

In order to overcome the shortcomings in the attribute table, Professor Abadi proposed a vertical segmentation method which is very powerful [5]. This method will be classified according to the attribute tuples, independent tuples in a table for the N attribute, each table includes the two attributes of the subject and object, then the establishment of the table is stored in the column based on the C-Store database, and through the combination of pre-existing paths, the query operation connection between tuples are optimized to improve the efficiency of RDF data query. This method has many advantages:

1)It supports multiple values.

2)It supports heterogeneous data.

3)It does not need to design complex clustering algorithms.

There is no need for merge operations because of all the values of attributes are placed in a table.

In our experiments, when we stored the data in a relational database, the efficiency of vertical segmentation method and the attribute table in the same order of magnitude increased slightly; when we used the column Monet-db database management system, we has obtained a very good effect because the storage form of thought and column database vertical partition is very fit. Be relative to other queries, the attribute table be more than ten times faster. Although the vertical segmentation method greatly improves the query efficiency, but it is not suitable in some cases, such as a query need to cross multiple attributes, you need to connect to multiple tables.

It is different form the vertical segmentation method , the attribute table be insisted on using three tuple, and aslo designed and implemented many methods to solve the problem of query efficiency, Professor Weiss of University Zurich for a class of queries: "when the required query attribute" is not specified, the vertical segmentation method is not appropriate, because it requires access to all of the N attribute table to get the answer, so he put forward the Hexastore[6] method, which is six times of the index method, this method build all indexed for the subject, object and attribute of all 6 arrangement forms, thus can improve the query efficiency, but its drawback is that the data storage space is expanded by about 5 times.

Thomas Neumann, Professor of German Marx Planck Institute of information also proposed a RDF-3X[7] storage architecture, first this architecture sorted the original three tuple, and the data in the same column are found to be very small. Therefore, this paper proposes a compression idea, that is, only the first data, the subsequent data is stored by the difference between the data and the first data. By this method, the amount of data reduced greatly, and then established the index, in RDF-3x the same as SPARQL analysis was studied, and put forward the related model, and several methods are used to optimize the query process, which improved the efficiency of the query and achieved great success. RDF-3X method also has some shortcomings, for the logic of complex queries, the efficiency is relatively low.

professor Zou Lei who is from China's Peking University pointed out that in recent years, the original system did not consider the use of wildcards in the query when it is designed, so he proposed a gStore[8] storage method based on graph to store RDF data. This method will store the RDF data in the form of map, then put the character encoding for each vertex, the original string used a hash function mapping for the 0 and 1 encoding, also encoded the wildcards, then established a S-tree in each encoding vertex as inputing, and generated the connection between nodes of each layer S-tree which between each vertex attributes, also built VS-tree, then optimized the querying process of the VS-tree then the establishment of VS*-tree, through this method, it can support Wildcard Queries, and improve the efficiency of query execution . The gStore method is very suitable for the star structure because it based on graph, so, the efficiency is very high, and support Wildcard Queries, but for complex data structure, the query efficiency will slow down.

In the recent research results, H2RDF+[9] provided the use of multi user interface to interact with different databases in the cloud computing environment. The literature [10] pointed out that different storage methods are suitable for a specific query type, and puted forward the system which

Workload sensitive adaptation. That is, according to the changes in the query, the dynamic adjustment of storage, in order to provide better service, the corresponding system is being developed in [10].

We compared the methods of storage and query in the above methods, get some inspiration that the vertical segmentation method can be used for data sets with less attribute values, and it can have a better query efficiency, however, when the value of the data set is very large, it is not available. Hexastore method have a high efficiency, but the storage space is greatly increases, so was Rarely used.Rdf-3x is still Current research hotspots ,because of its high compression rate and high efficiency, the author also put forward x-rdf-3x[11],which supported more online updates. The g-store method is higher the average query efficiency, but because it is based on a graph structure to build, so there are multiple penetration queries for a node, its efficiency compared with other methods, will be relatively much slower. In summary, now for the storage of RDF data is not a for the case of "master key", the research task is still a long way to go.

Pretreatment strategy and experimental configuration

Motivation

From the description of the relevant literature and the analysis of the experimental data set, we can draw a conclusion: in the RDF data, the number of attributes is far less than the number of tuples.

For example, in [5] literature, the number of tuples of data set in the Barton library is about 50 million 250 thousand, while the number of different attributes is only about 221. In our experiments, we find that there are about 8 million [12] in the data set DBLP, and the number of attributes is less than 100, the same conclusion in the LUBM data set [13].

And in the experiment, because of the existence of various factors, there is no final result in some query query result, this leads to a large amount of computing resources wasting, so we did some works to cope with this situation on the original query process pretreatment, the steps are as follows:

1)using related tools to import data into the database.

2)Defined all the attributes which get from the query, then stored in a new property table after the removal of the duplicate.

3)Judge in the property ta ble before executing the query, if the query does not exist, then the query directly to the end; if it exists, then proceed to the next step.

Experimental configuration

Experiment on a PC machine, the specific configuration is as follows:

CPU:Intel (R) CORE (TM) i7-4790 3.60GHz

Memory: 4GB

Operating system: Ubuntu 14.0464 bit Database: POSTGRES 9.3

Programming language: Java

Programming tools: Eclipse

In order to consist with other storage methods of environment and reduce the differences, experiments were carried out under Ubuntu operating system, Ubuntu Kylin is an open source project supported by China CCN joint laboratory, it is focusing on the usability of the system, through providing customized Chinese user experience by customizing the localized desktop user environment and developing applications that meet user specific needs, is one of the most China characteristics of the operating system, since its launch, is widely used in domestic research field.

The experiment uses the database for POSTGRES 9.3, which is developed by the Department of computer science, University of California at Berkeley. It can support most of the SQL standard and offers other functions of complex queries, such as foreign key, view, transaction integrity etc. And it can be extended, such as the addition of new data types, functions, operators, etc. It is also very rich on supporting for the interface, almost all types of database client interface, which is widely used in database research.

The experiment used Java language and Eclipse development tool, because the Jena and other

tools provides a wealth of features for the operation of RDF data.

Experimental process and result analysis

Experimental process

This thesis used LUBM as the experimental data set, LUBM is developed by the Lehigh (The Lehigh University), in order to promote the standardization and evaluation system of Semantic Web data.some semantic data which describes the school, including automatic data generator, standard query problem, and measure etc.

Specific process of experiment:

1)use LUBM automatic data generator to generate data (specific operation reference [13] provided instructions).

2)download and install JENA2[14], programming will generate the data into the database LUBM_DATA.

3)In order to do comparison with experimental results, Using the JENA2[14] parser generates data parsing to generate LUBM_data.txt files, then the resulting LUBM_data.txt format data is converted to the syntax format required by the gStore, and also into the format required by the RDF-3X language, import RDF-3X.

4)We used the LUBM which provided by the 14 Standard Query in the first 9 as a test query (the average value of the implementation of the, unit MS), the results shown in table 1.

5)Modify the properties of the query Q6, Q7, Q8,Q9, ensure that attributes do not exist in the dataset and other queries unchanged, again step 4 for the query, the last run in a relational database, as shown in table 2.

Experimental result analysis

Table 1 results of original query execution

	gStore	RDF-3X
Q1	25.3	37.3
Q2	39.3	135.0
Q3	24.0	17.0
Q4	224.3	19.7
Q5	213.0	10.7
Q6	230.3	1168.7
Q7	41.0	180.0
Q8	229.3	175.0
Q9	194.0	220.7

Table 2 results of the modified query execution

	gStore	RDF-3X	Pre process relational database
Q1	25.3	37.3	43.2
Q2	39.3	135.0	99.7
Q3	24.0	17.0	31.0
Q4	224.3	19.7	232.4
Q5	213.0	10.7	188.7
Q6	230.3	1168.7	5.3
Q7	41.0	180.0	4.8
Q8	229.3	175.0	5.0
Q9	194.0	220.7	4.9

From the experimental results we can see that for the Q1, Q2, Q3, Q4, Q5 query which unmodified, time in the pretreatment of relational database query and RDF-3X slightly longer than gStore. And the query Q6, Q7, Q8, Q9 which modified and its Characteristic that the property does not exist in the data. Therefore, only the property table in the relational database is searched, and exit the query directly after the property is not found. However, the length of the attribute table is

relatively small, and it is invariable, so greatly improve the efficiency of the inquiry, the example of the effect as shown in figure 1.

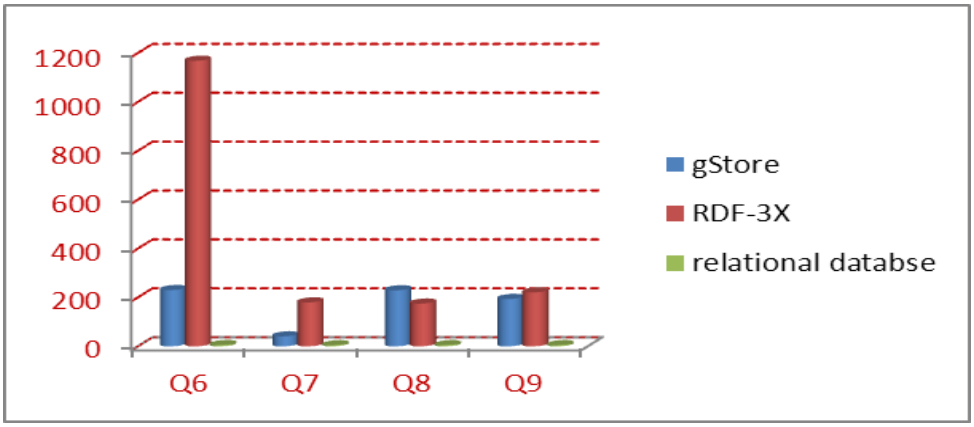

Figure 1 comparison of pretreatment query example

Conclusion

This paper makes a summary and analysis of the RDF data storage methods, according to the characteristics of data, the query process is pre processed in the relational database, so as to improve the efficiency of queries that do not contain data set attributes in the query. The next research project is divided into two parts, first, using more data sets continue to experiment to verify the effectiveness of the pretreatment such as DBLP and Yago2 data sets. Second, explore the discovery of a new form of data storage management, and support for more efficient and diverse query services.

References

[1] Berners-Lee T, Hendler J, Lassila O. The semantic web[J]. Scientific american, 2001, 284(5): 28-37.

[2] Pan J Z. Resource description framework[M]//Handbook on Ontologies. Springer Berlin Heidelberg, 2009: 71-90.

[3] Prud'Hommeaux E, Seaborne A. SPARQL query language for RDF[J]. W3C recommendation, 2008, 15:1-27.

[4] Wilkinson K, Wilkinson K. Jena property table implementation[J]. 2006:1-13.

[5] Abadi D J, Marcus A, Madden S R, et al. SW-Store: a vertically partitioned DBMS for Semantic Web data management[J]. The VLDB Journal—The International Journal on Very Large Data Bases, 2009, 18(2): 385-406.

[6] Abadi D J. Column Stores for Wide and Sparse Data[C]//CIDR. 2007: 292-297.

[7] Weiss C, Karras P, Bernstein A. Hexastore: sextuple indexing for semantic web data management[J]. Proceedings of the VLDB Endowment, 2008, 1(1): 1008-1019.

[8] Neumann T, Weikum G. The RDF-3X engine for scalable management of RDF data[J]. The VLDB Journal, 2010, 19(1): 91-113.

[9] Zou L, Mo J, Chen L, et al. gStore: answering SPARQL queries via subgraph matching[J]. Proceedings of the VLDB Endowment, 2011, 4(8): 482-493.

[10] Papailiou N, Tsoumakos D, Konstantinou I, et al. H2RDF+: an efficient data management system for big RDF graphs[C]//Proceedings of the 2014 ACM SIGMOD international conference on Management of data. ACM, 2014: 909-912.

[11] Aluç G, Özsu M T, Daudjee K. Workload matters: Why RDF databases need a new design[J]. Proceedings of the VLDB Endowment, 2014, 7(10): 837-840.

[12] Neumann T, Weikum G. x-RDF-3X: Fast querying, high update rates, and consistency for RDF databases[J]. Proceedings of the VLDB Endowment, 2010, 3(1-2): 256-263.

[13] Ley M. DBLP: some lessons learned[J]. Proceedings of the VLDB Endowment, 2009, 2(2): 1493-1500.

[14] Guo Y, Pan Z, Heflin J. LUBM: A benchmark for OWL knowledge base systems[J]. Web Semantics: Science, Services and Agents on the World Wide Web, 2005, 3(2): 158-182.

[15] Wilkinson K, Sayers C, Kuno H A, et al. Efficient RDF Storage and Retrieval in Jena2[C]//SWDB. 2003, 3: 131-150.