Optimal Query Processing in Semantic Web using Cloud Computing

S. Joan Jebamani[1], K. Padmaveni[2]

[1]Department of Computer Science and engineering, Hindustan University, Chennai, Tamil Nadu, India
joanjebamani1087@gmail.com

[2]Department of Computer Science and engineering, Hindustan University, Chennai, Tamil Nadu, India
kpadmaveni@hindustanuniv.ac.in

Abstract: ***Semantic web technologies can be useful for maintaining data in the Cloud. It adds new data and metadata to existing web documents. This extension of web documents to data will enable the web to be processed automatically by machines. To do this RDF is used to turn basic web data into structured data. In semantic technology large RDF graphs will be used. It poses significant challenges for the storage and retrieval of RDF graphs. So a framework can be built using Hadoop to store and retrieve large numbers of RDF triples. We proposed a sophisticated query model to answer the frequent queries. It will cache statistics for frequent queries and uses dynamic programming to exploit the statistics. It also proposes the history maintenance module to provide updated history information. We used Hadoop's Map reduce framework to answer the query and SPARQL query language to retrieve the data from RDF storage.***

Keywords: HDFS, RDF, SPARQL, Map reduce

1. Introduction

The proposed system uses the sophisticated query model to process the most frequent queries. The frequent query and its response will be stored in the cache memory. This sophisticated query first checks the cache memory for the given query. If the answer is there in the cache memory, it will directly fetch the answer from the cache memory instead of fetching from the Hadoop storage. It leads to reduce the processing time of frequent queries. It also proposes the history maintenance system. It is used to maintain the access time, data modification and other user information. It will frequently update the history information. It describes the framework that is build using Hadoop distributed file system (HDFS). It is used to store and retrieve large number RDF triples by exploiting the cloud computing. In addition It also has Map reduce programming model which is use to perform parallel processing of large amount of data. This paper uses semantic web technology to built efficient and scalable system for Cloud computing. It is the fast growing technology. It presents the data in standardized way; such data can be understood by both machine and human operator. It provides the user convenient search. So users can easily get the required data. For data storage it uses an algorithm to automatically convert the ordinary user data to RDF format. In fig1, first the data to be store in the cloud has to enter as a input data. This data will be automatically converted into RDF format. This input data will be pre-processed and then it will store in to the Hadoop cluster. In the data retrieval process we can retrieve the data by using SPARQL query. Here the user has to enter the query. And then the query will check the cache memory to fetch the answer. If the answer is not there, it will pass the query into Map reduce framework. This framework will generate the query plan and submit the job to the Hadoop cluster. Finally it will return the result to the user.

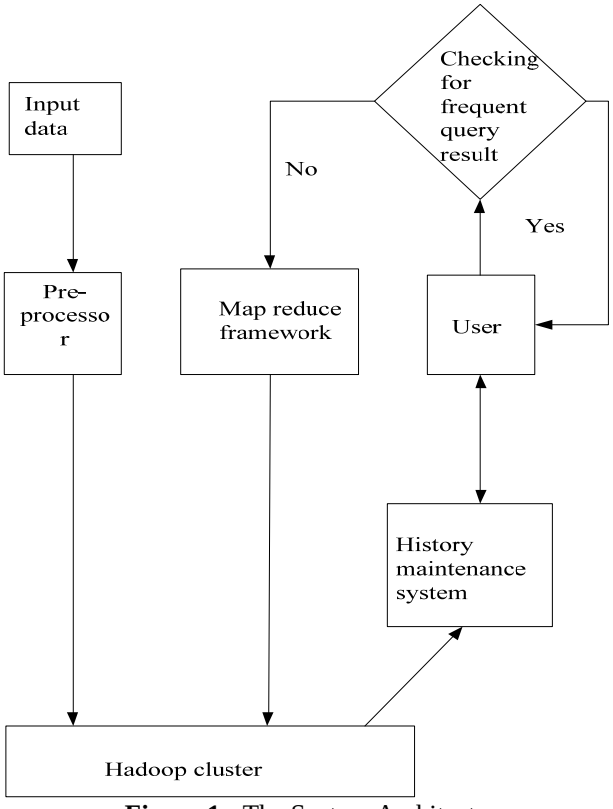

Figure 1: The System Architecture

The history information is automatically updated by the history maintenance system. User can view the updated history at any time.

2. Basics of Cloud Computing

Cloud computing is a general term for anything that involves delivering the hosted service over the internet. The name cloud computing was inspired by the cloud symbol that's often used to represent the internet in flowcharts and diagrams. It is emerging paradigm in the IT

and data processing communities. Cloud computing is the fast growing technology. It allows the user to perform the remote access. We can store large amount of data in the cloud without the need of separate RAID storage. As the popularity of cloud computing grows, the service provider's face lot of challenges. They have to maintain huge quantities of heterogeneous data while providing efficient information retrieval. Thus the key emphasis for cloud computing solutions is scalability and query efficiency. It provides following four services that are, infrastructure as a service, platform as a service, software as a service and network as a service. End users can access the Cloud through web browser or mobile application or desktop while the user's data and business software are stored on servers at remote location.

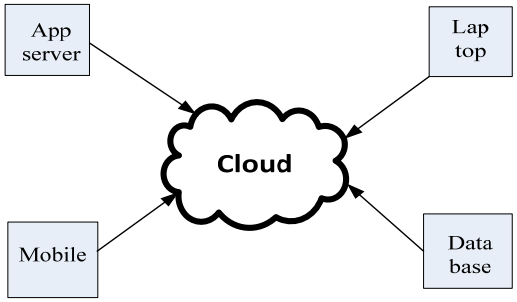

Figure 2: Cloud Computing Architecture

3. Basics of Semantic Web Technology

Semantic web will extend the web documents to data that will enable the web to be processed automatically by machines and also manually by humans. Computers will understand only syntax. So historically web pages are published in HTML files. Computers will understand the syntax based HTML data but it can't understand the semantics of the data. Computers can't understand the meaning behind the data. If it understand the meaning it can help us to find a exact require data while perform searching. Semantic web technology is used to enable the computers to understand the meaning behind the web data. Web of today is arguments where semantic web is about web of things. It means place, person, movies, sports, organizations and just about any concept you can think off. So it can easily perform searching, aggregating and combining of web information without a human operator. So it enables the user with efficiently store and retrieves the data for data intensive application.

4. Basics of RDF

RDF (resource description framework) is a family of w3c specification, originally designed as a Meta data model. It is similar to entity relationship or class diagram. RDF is used to represent the data in the web. It will make the statements about resources (web resources) in the form of subject – predicate – object expression. This form is called as triples. It is also called as N3 notation. Here subject refers to the thing that we are describing. It indicates the resources. Predicate refers to the attributes of the thing that we are describing. It indicates the relationship between the subjects and predicates. Object is the thing that we are referring to with the predicate. It indicates the aspect of the resources. Collection of RDF statements intrinsically represents a labeled directed multi graph. RDF is one of the promising technologies which can embed semantics in HTML documents. RDFa empower web developers to add semantics to web pages. So the computers can easily understand the semantics behind the web data. It will change the computers from passively helping us to actively helping us. Just like N3 is a syntax used to describe RDF to humans, RDFa is a syntax used to describe RDF to computers. RDF uses URI (Uniform Resource Identifier) to identify the subjects, predicates and objects. There can be many triples associated with the particular subject. The predicates points to vocabulary. No RDF has any inherent meaning until it pair them up with vocabulary. The vocabulary will define what the triples actually need. The vocabulary allows computers to understand when we are talking about specific concept. One such vocabulary that is quietly very popular on the semantic web is friend of a friend vocabulary. It is also known as foaf. Foaf contains several concepts that are used to identify people and identifying relationship between those peoples.

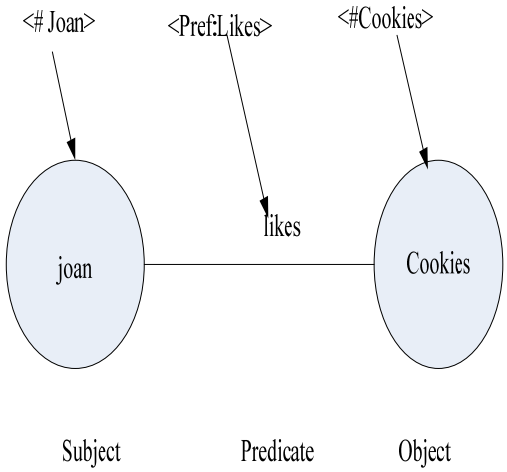

Figure 3: RDF Triples

5. Basics of HADOOP

Hadoop is the distributed file system. It is highly scalable and also it is portable from one platform to another. It is written in java. It has the features of highly fault tolerance and reliability. It achieves reliability by replicate the data across multiple hosts. It does not require RAID storage on hosts. Each node in a Hadoop instance typically has a single name node. Cluster of data nodes are form Hadoop clusters. Data nodes are serves as blocks of data over the network using a block protocol specific to HDFS. The name node is the repository for all Meta data. The system is designed in such a way that user data never flow through name node. The name node makes all decision about replication of blocks. It periodically receives the block report from each of the data nodes in the cluster. HDFS has the master/slave architecture. The name node act as a master. It has the responsibility to perform namespace operations like opening, closing and renaming the files and directories. The data node act as a slave. It has the responsibility to serve read and write request from the client. HDFS exposes the file system namespace and allows

user data to be stored in files. The file is split into one or more blocks internally. These blocks are stored in a set of data nodes. The file system uses TCP/IP for communication. Clients use RPC to communicate with each other. The advantage of HDFS is data awareness between the job tracker and task tracker. The job tracker schedules map/reduce jobs to task trackers with an awareness of the data location. It will reduce the data traffic over the network by preventing unnecessary messages.

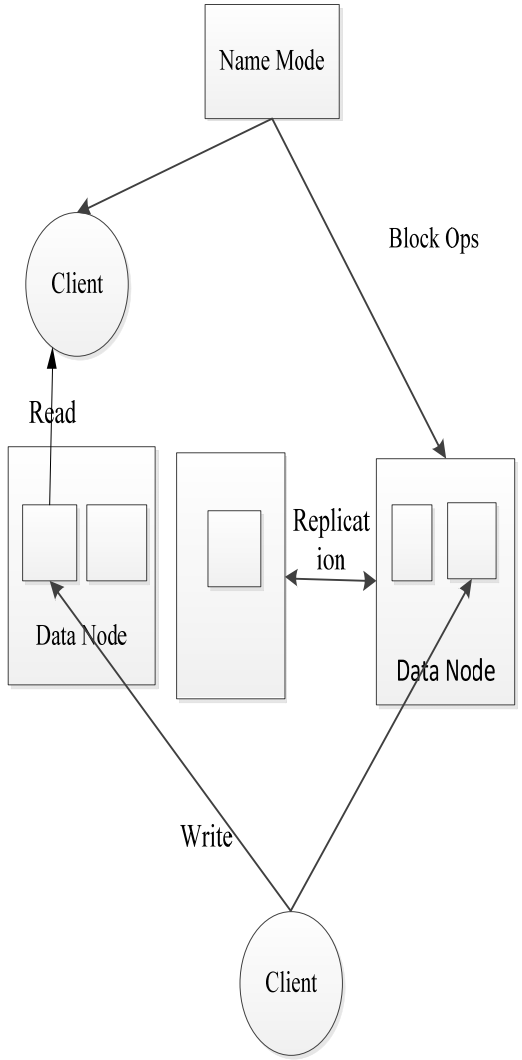

Figure 4: Hadoop Architecture

6. Basics of SPARQL

SPARQL is a recursive acronym for SPARQL protocol and RDF query language. As the name implies, SPARQL is a general term for both protocol and query language. Most uses of SPARQL acronym is refer to RDF query language. SPARQL is syntactically looking like a SQL language for querying RDF graphs via pattern matching. It has the features of conjunctive patterns, pattern disjunction, value filtering and optional patterns. It can able to retrieve and manipulate the data stored in RDF format. Some tools exist to translate SPARQL query into other query languages. For example: SPARQL to SQL. SPARQL query language for RDF is designed to meet the requirements of RDF data access. SPARQL protocol for RDF (SPROT) specification defines the remote protocol for issuing SPARQL queries and receiving the result. It is a method to perform remote invocation of SPARQL queries. It specifies a simple interface that can be supported via HTMA or SOAP that a client can used to send a SPARQL queries against some end point. Both the SPARQL query language and the protocol is the product of W3C's RDF data access working group.

7. Basics of Map Reduce

Map reduce is the programming model for large data sets. It is typically used to do distributed computing on clusters of computers. It is a framework to improve parallel processing of huge data sets by using a large number of computers. Computational processing will occur on either data stored on file system or database. Map reduce can take advantage of locality of data. It will process the data on or near storage assets to decrease transmission of data. It allows the distributed processing of map and reduces operations. It provides each mapping operation is independent of others and all maps can be performed in parallel. Similarly a set of reducers can perform the reduction phase. Map reduce process can be applied to large data sets than commodity server can handle. A large server will use map reduce to sort a peta byte data in a few hours. The parallelism also provides the possibility for recovery from the partial failure of servers or storage during the operation. If one mapper or reducer fails the work will be rescheduled. Map reduce achieves reliability by parceling out a number of operations on the set of data to each node in the network. Each node has to report back periodically with completed work and status updates. If a node falls silent for longer time than a particular interval, the master node will records that node as a dead node and sends out that node's assigned work to another nodes. The job execution will be start when the job is submitted to the job tracker. The job tracker will specifies the map and reduce function as well as the input and output path of the data. And then it will determine the number of split from the input path and select some task tracker based on the network proximity to the data sources, then it will send the request to task tracker. The task tracker will perform the map and reduce function.

7.1 Map Step

The master node takes the input and divides it into smaller sub problems, and distributes them to different worker nodes. The worker node may again split in turn. It leads to a multilevel tree structure. The worker nodes will process the smaller sub problems and passes the answer back to its master node.

7.2 Reduce Step

The master node then collects the answers from all the sub problems and combines them into some way to form the output. If the reduce phase task tracker crashes, the job tracker will rerun the reduce at different task tracker.

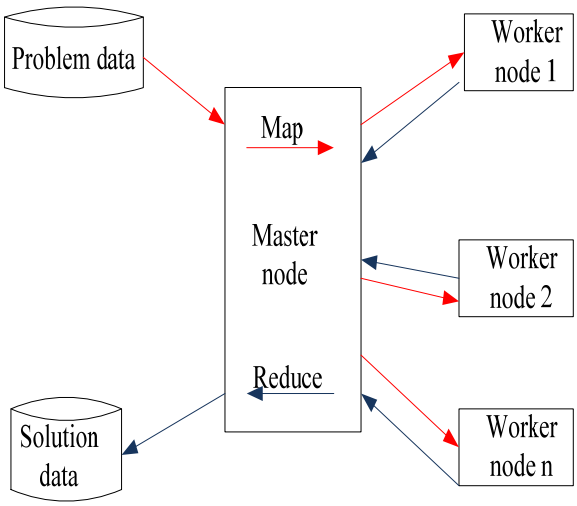

Figure 5: Map Reduce Framework

8. Pre Processor

Preprocessor will perform preprocessing on the input data to be stored. The data in RDF/XML format is not suitable to store in Hadoop. Because to retrieve a single triple we have to parse the entire file. So we have to convert the data to N – triples to store the data. In this format we have complete RDF triple (Subject, Predicate and Object) in one line of the file. It is very convenient to use with map reduce job. We can't store all the data in a single file. So we divides the data by using predicate split and predicate object split.

8.1 Predicate Split

In this split we divide the data according to the predicates. It will reduce the search space for SPARQL query which does not have a predicate variable. For such type of query we pick a file for each predicate and run the query on those files only.

8.2 Predicate Object Split

Here the split is done according to the object. There are two types of split will be done here. First the Split is done by using explicit type information object. Here the predicate rdf: type is used in RDF to denote that resource is an instance of a particular class. So the rdf-type file is first divided into many small files according to the distinct objects of RDF: type predicate. For example if the leaves of the class hierarchy are A1, A2…., An, then we will create files for each of these leaves.

And then next the split is done by using implicit type information of object. Here we divide the remaining files according to the type of object. All the objects are not URI some of them are literals. The literals are remaining in the file named by predicate; no further processing is required for them. The URI objects are move to the respective file named as predicate-type. For example if a triple has the predicate p and type of the URI object is Ai, then the subject and object appears in one line in the file P-Ai.

.

9. History Maintenance System

The history maintenance system is used to store or maintain the history information of the Hadoop distributed file system. It will maintain the data base of history information about the details of the user who accesses the system through query and data access time. It is also used to store the information about the data updation. The history information is automatically updated often. So we can efficiently retrieve the updated history information from this system at any time.

10. Sophisticated Query Model

This paper proposes the sophisticated query model for most frequent queries. Some queries need to be accessed frequently. For that frequent queries, again perform searching and display the result to the user is the wastage of time. So we will propose the most efficient method of query access for frequent queries. Here we will cache statistics for the most frequent queries and use dynamic programming to exploit the statistics. So for every query from the user, first it will check the cache memory for frequent queries. If the result is there, it will directly fetch the result from the cache memory instead of getting from the storage engine. If the result is not there, it will submit the job to map reduce framework.

11. Conclusion

The proposed sophisticated query model is efficiently bringing the result for frequent queries. It reduces the query processing time for frequent queries.We have presented a framework capable of handling enormous amount of RDF data. Our framework is based on Hadoop, which is a distributed and highly fault tolerant system. And also it uses map reduce framework which leads to improve the parallel processing of large data sets. Our result shows that our framework is scalable and efficient and can handle large amounts of RDF data and it efficiently maintain the history information, unlike traditional approaches. And also it provides fast query processing for frequent queries.

References

[1] Daniel J. Abadi, Data Management in the Cloud: Limitations and Opportunities, IEEE Data Engg. Bull., Vol 32, No. 1, 2009.
[2] P. Mika and G. Tummarello, Web Semantics in the Clouds, IEEE Intelligent Systems, Volume 23, 2008.
[3] Henog Sikkim, padmashree Ravindra, Kemaforanyanwv "From SPARQL to Map reduce: The journey using a nested triple group algebra", North caroline state university, Raleigh, Nc, 2009.
[4] Prasad Kulkarni, "Distributed SPARQL query engine using Map Reduce," Master of Science, Computer Science, School of Informatics, University of Edinburgh, 2010.
[5] James p. Mcglathlin, Lalifur khan "A scalable data model for efficient querying of RDF data set" The university of Texas at dallas Richardson, Tx, 2009.

[6] James P. McGlothlin, "Framework and Schema for Semantic Web Knowledge Bases", the University of Texas at Dallas Richardson, TX, USA, 2009.
[7] Jay Liu, Department of Computer Science, "Distributed storage and query of large RDF graphs," The University of Texas at Austin, Austin, TX, USA, 2008.
[8] Mohammad Farhan Husain, Latifur khan,Murat kantarcioglu and Bhavani Thuraisingam, "Data intensive query processing for large RDF graphs using cloud computing Tools" IEEE cloud 2010, pp. 1-10 Miami, Florida, July 2010.
[9] J.J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson, "Jena: Implementing the Semantic Web Recommendations," Proc. 13th Int'l World Wide Web Conf. Alternate Track Papers and Posters, pp. 74-83, 2004.
[10] Jesse Weaver and James A. Hendler, Parallel Materialization of the Finite RDFS Closure for Hundreds of Millions of Triples, Proceedings of the 8th International Semantic Web Conference, 2009.
[11] Michael Schmidt, Thomas Hornung, Georg Lausen and Christoph pinkel, Sp2 Bench: A SPARQL Performance Benchmark, 25th International Conference on Data Engineering (ICDE'09).
[12] Jacopo Urbani, Spyros Kotoulas, Eyal Oren and Frank van Harmelen, Scalable Distributed Reasoning Using Map Reduce, International Semantic Web conference, 2009.

Author Profile

S. Joan Jebamani received a bachelor's degree (Vel Tech Engineering College) in Information Technology and Engineering from Anna University, Chennai, India in 2011 and doing Master's degree in Computer Science and Engineering from Hindustan University, Chennai, India.