Indian Journal of Science and Technology, Vol 9(39), DOI: 10.17485/ijst/2016/v9i39/102091, October 2016
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

# Performance Improvement in Ontology based Semantic Web using Multi-level Cache

**Dibya Raj Ghosh, E. Poovammal* and Ujjwal Sinha Mahapatra**

Department of Computer Science and Engineering, SRM University, Chennai - 603203, Tamil Nadu, India;
dibyaraj11@gmail.com, poovammal.e@ktr.srmuniv.ac.in, usmahapatra@gmail.com

## Abstract

**Objective**: The objective of the paper is to improve the performance of semantic web in a Hadoop environment. An analysis of the ontology and the improvement in the role of cache played a major part in the performance improvement. **Methods:** Ontology describes a formal specification of certain domain. Existing platform and servers are unable to process different types of data. Hence, it is better to have an effective large number of ontology in the web. Hadoop platform helps in implementing the incremental and distributed ontologies over and above existing ontologies in Semantic Web. One of the major challenges recorded while deploying the semantic technologies is the performances degradation of triple stores. **Findings:** A framework is proposed in order to improve the performance of Web applications which is relational database-backed. When implemented effectively, it provides a strong base to semantic computations. **Application:** We all know that inherent speed difference between disk and the processor can be reduced by File caching mechanism and hence Multilevel cache helped in improving the performance of semantic web.

**Keywords:** Big Data, Hadoop, Multilevel-Cache, Ontology Reasoning, RDF, Semantic Web

## 1. Introduction

Since it is not possible to understand normal web information by the machine, Semantic Web[1] is used which is developed on ontology[2] based approach. Semantic Web is where all the data elements are well defined so that both machine and human can understand easily. Hence semantic web can be viewed as the extension of current web. In general, web is used to display data, integration, automation etc. Semantic Web provides property of artificial intelligence to the machine and provides an easy and efficient way of searching. Resource Description Framework (RDF)[3] is the basic language used to describe the knowledge in Semantic Web. RDF uses graph data format. This format allows us to mark label and direction, which helps in representing information in the Web. RDF uses triple as its basic unit. Triple describes the relation between two things subject and object and it is represented as <subject, predicate, object>. Subject means resources and predicate means the properties by which subjects are related with objects. The technique of extracting web information is ontology based. Ontology helps in extracting concept hierarchy in an efficient way which in turn helps for ontology construction[4].

Ontology provides a common vocabulary in a particular domain. Ontology helps in many ways. The structure of the information can be understood from it. Secondly ontology helps in extracting the domain knowledge and also for reuse of domain knowledge. Using ontology we can differentiate domain knowledge from operational knowledge. Many other operations such as analyzing the domain knowledge become easier while using ontology. If some other domains like e-business sites, educational, healthcare sites use same ontology then OWL, SPARQL can extract knowledge from all the sites and aggregate the information. This aggregated information are used by semantic agents to answer user queries. Sometimes

*Author for correspondence

the aggregated information is used as input data to other applications.

Ontology design[5] inherited the concept of object-oriented design but it is not completely same as object-oriented design. Ontology designers take decision based on the structural properties of the class. Many institution use the same attribute but using different name like in customer database customer name attribute can be written as cust_name, c_name or customer_name. Machine can't understand that all these attributes contain same domain value which is customer name. So, some extra information has to be provided to the RDF. Ontology provides this kind of extra information.

WEBPIE[6] use RDF closure. RDF closure method allows multiple inputs for finding reasoning. But this method has a disadvantage that it allows duplicate values. Duplicate record unnecessarily takes more space to store the value and consumes more time for calculation.

SPARQL[7] is a one type of query language which is used to find the result from the RDF. The result of this graph can be represented in any format. Here, SPARQL just takes the description in the form of query, based on the requirement of the respective application. The retrieved descriptions are either in RDF graph format or a set of bindings. RDFS is recognizable as an ontology language. Ontology describes a formal specification of a certain domain[8].

Using File caching it is possible to reduce the inherent speed difference between processor and disk. In distributed system file system, access is based on a client-server computing model and server propagates through many instances.

In a distributed system data can be hold in four different places. First place is memory of the client. Second place is hard disk of the server. The third place is cache memory of the server and finally the fourth place is the client's disk. Since any interface connected to the server can check the centralized cache memory, data storage in server is not a challenge. But, storing data in client side is challenge because of cache consistency property.

## 1.1 Server Caching

Server caching process can be automated at the server, since the same buffer cache is used by all other files on the server. To avoid unexpected data loss, NFS-related writes are preferred. Even if the server dies, we don't want to lose data.

## 1.2 Client Caching

Normally, it is preferred to reduce remote operations and that is motto of client caching. Three pieces of information which are cached at the client side are, file attribute information, file data and bindings of pathname. It can be possible to cache the result of operations such as read, write, get attribute, lookup and redirect operations. The important aspect to be taken into consideration while dealing with cache is consistency to be maintained.

Semantic Web is used to develop intelligent web interface. OWL is used to model the Semantic Web[9] as well as it retrieves information from semantic web. OWL and RDF both are unable to store temporal information. A new language tOWL[10] (temporal OWL) provides the facility to store the temporary information in Semantic Web. Temporary information is very useful and necessary in some condition.

Query processor may use any searching technique like forward search, backward search, and valid node search. But if some process tries to search from RDF closure graph then it will take longer time. In this paper, we have created a sub graph of RDF named as Transfer Inference Forest TIF[11]. Any query language like OWL, SPARQL, when searches any element from RDF then it will search from TIF. If that element is not present in the TIF but is present in the RDF then one incremental edge is added to that TIF. So, for this incremental method next time a processor will take less time as it generates the result from TIF graph.

Semantics defines to meaning, which in contrast improves the time and the performance. All the data present in the database should be stored in compressed form. When a client asks any query, before processing that query, we have to perform two operations, spell checking and redundancy checking. These two checking methods help in creating valid TIF.

Most of the researchers have agreed or concluded that in most of the real world scenarios, the querying performance of triple stores becomes a decisive factor. Especially in the deployment of large-scale semantic technologies this decisive factor takes a lead role. To improve the triple stores' performance many alternatives were thought of. Some are special indexing, optimizing query and looking for better storage. On the other hand, querying a triple store is still usually slower when comparing the process of querying data stored in affixed relational

database schema. The speed of relational data base query is better by a factor of 2-20 (cf. e.g., BSBM results1).

This issue exists in triple store is due to the special feature of it. Triple store has special feature of editing or amending and reorganizing structures of schema easily and quickly. Before caching query results, we have to ensure that cached query results are selectively validated on knowledge base updates. The graph pattern of SPARQL is examined, in our approach. The query result is cached only when the updated triples do not match with any of the graph pattern query result. If it matches, the corresponding cache object is invalidated and the triple store starts re computing from the subsequent query execution.

Traditional Web applications will cache application objects. Also, they cache HTML page fragments, which are user interfaces. The whole parts of the generated user interface may be cached. Then the invalidation of these complex cache objects is taken care by application logic.

As a result, Semantic Web applications are significantly accelerated. The reason being the queries which are frequently asked are updated moderately. This improvement allows Semantic Web applications to reach the performance level of conventional web applications connected with relational databases.

## 2. Materials and Methods

### 2.1 XML

XML[12] (Extensible Markup Language), derived from SGML is simple, very flexible text format. It is designed for the purpose of describing raw data. For displaying data we can use HTML. XML is globally accepted and it is independent of hardware and software. System defined tag as well as User defined tags are also accepted. Namespace is defined by the Xmlns.

### 2.2 RDF

RDF[13] is an assertion language which is used to represent the data in the Semantic Web using some ontology. It describes the resources which are available in the Web and also find the relationship between them. Any data which have Universal Resource Identifier (URI) can be stored in RDF data model. RDF graphs provide the capability to define Ontology. RDF use to describe metadata. RDFS is a set of classes with certain properties in RDF.

```
<rdf:RDFxmlns:rdf="http://www.w3.org/rdf-syntax-ns#"xmlns:feature="http://www.flipkart.com/clothing- features#">
<rdf:Description rdf:about="http://www.flipkart.com/clothes#t-shirt">
<feature:size>22</feature:size>
</rdf:Description>
</rdf:RDF>
```

In line no 2 all the syntax about the RDF are written, feature is the object and size is the property of the subject. Actual application-specific classes and properties are not provided by RDF Schema. Instead RDF Schema provides the framework to describe application-specific classes and properties. RDF is designed to be read and understood by the computer. Using web identifiers (URI), RDF identifies things. It describes the identified things with property and property values.

### 2.3 SPARQL

It is a query language which is used in Semantic Web. RDF is used to describe Semantic Web. SPARQL is a query language is recommended by w3c which fetch data from RDF. It uses the triples, conjunction, conditions, attributes etc. for finding the results. The result can contain multiple values. The query in SPARQL first goes through the parser for syntax checking and after that it follows optimization stage.

Invalidation of cached query and respective updates on triple store are to be performed effectively. This effectiveness can be achieved by analysis of SPARQL queries. Based on a dependency tacking, we wanted to cache compound query by extending the results of plain query.

The checking of whether the query has already been cached or has to be re-executed is determined quickly by the SPARQL proxy. It computes the queries MD5 hash quickly as soon as receiving the query.

### 2.4 OWL

OWL is a language for processing web information. Owl is built on the top of RDF. Owl is written in XML. Ontology is about the exact description of web information and relationships between web information. OWL is used for formal representation and also helps to integrate information generated from different data sources. The machine interpretability of OWL is much greater than that of RDF. Also, it is a stronger language when compared to RDF since OWL has stronger syntax and rich vocabulary.

### 2.5 HBase

To have quick and random access of structured and high amount data Hbase was developed. It is designed based on Google's big table. Hbase uses the Hadoop file system and MapReduce[14] engine for its core data storage needs. It stores non-relational data. Hbase implementation is best suited for handling high-volume data. We prefer the same while gathering incremental data and real time data. When we know that there will be a lot of exchange of information or frequent saving of changed content, we prefer Hbase.

### 2.6 MapReduce

MapReduce[15] is a combination of map and reduce functions. The placement of the tasks in different nodes and load balancing is performed by the map component. The same map component manages at critical conditions by following recovery principles from failures. The second component, which is the reduce component aggregates the elements and provides the result.

## 3. Results and Discussion

All the operations have to work in the Semantic Web. Many interfaces are present to support the reasoning over the Semantic Web. In the existing system, for reasoning process RDF closure is used.

When SPARQL query is parsed and identifying that result of the query is not present in the cache, it goes to the original SPARQL endpoint. Following the principle and schema of cache, arrived result is stored along with the parsed query. Multi level Caches can cluster the service consumers into groups based on their searching similarity. Multi level cache formed in this fashion will improve the efficiency of service discovery and release the load of service registry. If the descriptions are not having matched keywords but having words which are semantically equal, then the special multi level cache service is ignored. For example General keyword based searching ignores this kind of service. Suppose, if the descriptions have all the keywords but are not semantically representing the functional requirements, then they can be named as proper candidates and to be treated accordingly. Following these principle researchers started developing approaches which are behavior-aware. These behavior aware approaches were aiming to improve the accuracy of service discovery by adding semantic information into the web services.

Since ontology and knowledge-representation languages are used in Semantic Web Service (SWS), it has a large potential to grow. A lot of challenges are to be faced by the researchers to take SWS to its greater heights with its complete benefits. SWS also provides infrastructure for approaches to describe, discover and invoke activities on the Web.

### 3.1 TIF Constraction

RDF is a simple connected graph. TIF[11] is a set of directed tree as constructed by the triples where predicates are, i>rdfs: subClassOf, ii>rdfs: subPropertyOf, iii>rdfs: domain, iv> rdfs: range

#### *3.1.1 Property TIF*

Property TIF means that, in the RDF graph two nodes will be connected if two graphs nodes belong to same class as shown in figure 1. But, if have different properties then they will be connected in Class TIF but no connections would be present in Property TIF.

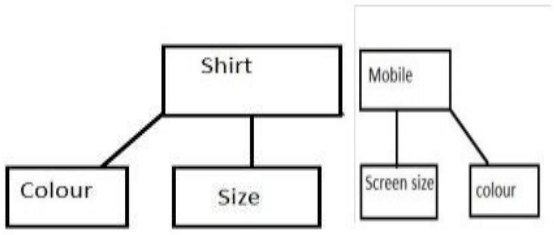


**Figure 1.** P-TIF.

In Figure 1, there are two classes, shirt and Mobile. Shirt has two properties colour and size while Mobile has two properties screen size and colour. But there is no property relation between these two entities shirt and mobile. So no edge connects shirt and mobile. In RDF these will be written like this:

Shirt rdfs: subPropertyOf: colour;
Shirt rdfs: subPropertyOf: size;
Mobile rdfs: subPropertyOf: colour;
Mobile rdfs: subPropertyOf: colour

#### *3.1.2 Class TIF*

In Class TIF two nodes are connected if they belong to the same class as shown in figure 2. But if two classes are connected then it does not assure that the properties of these two classes are same.

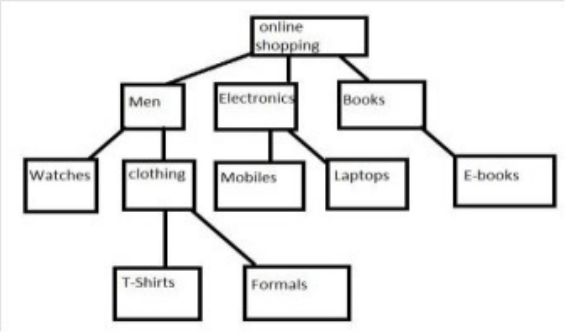


**Figure 2.** C-TIF.

In the Figure 2, if we consider online shopping as a super class then we can say that Men, Electronics and Books are subclasses of online shopping because these three categories are available which are different from each other. Electronics and books are two completely different entities so there are no connected edges in between. In RDF we can write:

OnilneShoppingrdfs: subClassOf: Men;
OnilneShoppingrdfs: subClassOf:Electronics;
OnilneShoppingrdfs: subClassOf: Books;
Men rdfs: subClassOf: Watches;
Men rdfs: subClassOf: clothing;
Electronics rdfs: subClassOf: Mobiles;
Electronics rdfs: subClassOf: Laptops;
Books rdfs: subClassOf: E-Books;

### *3.1.3 Domain and Range TIF*

Using domain and range knowledge we can draw some conclusions from properties of some individuals. Like, if we say that A is elder brother of B and B is the younger brother of A then we can say that both are male. So, both belong to the Male domain.

If we say about their age then it is sure that range cannot have negative values. It is the range of their age.

ObjectPropertyDomain(:hasElderBrother :Man)
DataPropertyRange(:hasAgexsd:nonNegativeInteger)

### *3.1.4 Domain Specific Search*

In recent years semantic methods proved themselves to perform intelligent operations and to provide some real time outcomes for varying input data.

Domain specific search results can be shown by using semantic methods and an organized semantic architecture will provide an efficient system to perform semantic operations. Here, architecture is proposed to efficiently implement and to provide a strong base to semantic computations.

Three searching techniques are present- Forward search, Reverse search, valid node search. In forward search searching follows top-down method. From the starting node it starts searching and stops until it finds the required value or if it reaches the last node. In Reverse search from the last node in a class it starts searching until it finds the required value or reaches the root node and in valid node search is used to check whether any information is present or not in any particular domain.

To compare with other schemes single-level sub graph architecture is considered, and the proposed method contains multiple levels of sub graphs using caching technique of computing architecture. The theoretical description and evaluation of the system is done in this paper which shows the variation of access time for different states of caches and the system is compared with single level sub graph system. Ultimate goal of this paper is to determine advancement of the system over existing system and evaluation results fulfils achievement of the goal.

## 3.2 System Architecture

In the existing system a prototype is implemented on the Hadoop platform. Map Reduce system discriminates some keywords entered by users and matches them with objects stored in triplets stored in triple Cache, this module implements Map Reduce technique where indexes in the Triplet cache are treated as keys and object field is treated as value to form key-value pair and ultimately stores them into object cache. Figure 3 presents the architecture of existing system.

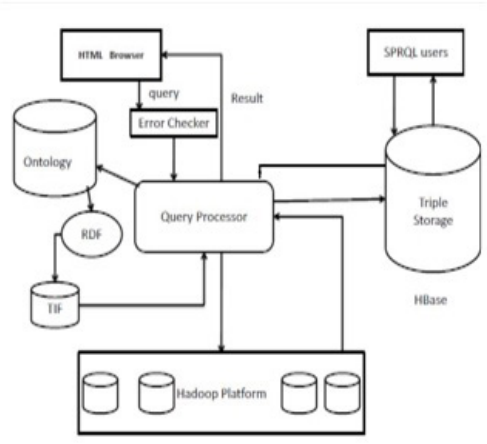


**Figure 3.** System architecture.

Map Reduce function is used to integrate different ontologies and RDF. Hbase is used for storing large tables, triples and data. When client ask query that pass through Error checker. Error checker check spelling. It does not allow searching meaningless word. In the existing system performance is very slow. To improve the system we proposed a new architecture using multilevel cache as shown in Figure 4.

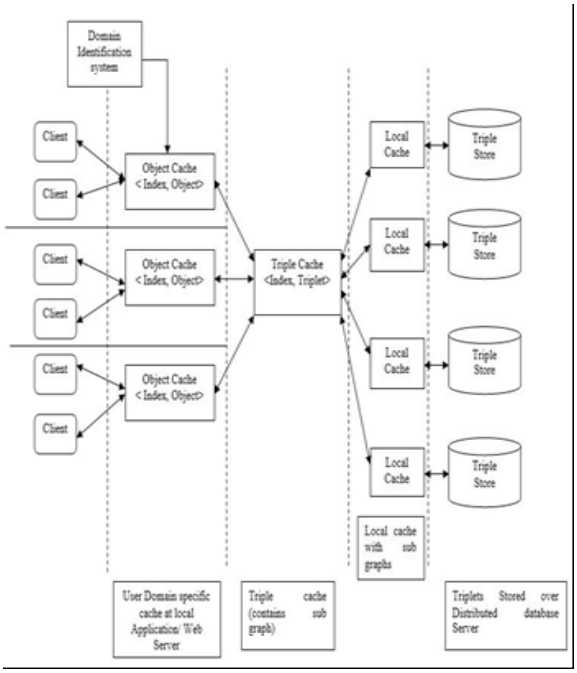


**Figure 4.** Proposed architecture.

### *3.2.1 Proposed Architecture*

The system consists of 5 modules:

- Distributed Database Server: The servers in the module mainly consist of servers, contains data in triple format.
- Local cache with sub graph: For the construction of local cache we are applying Map Reduce technique on distributed database server. To implement Map Reduce we consider object as key and remaining as value for the triples stored in the distributed database server and stores the intermediate set derived from mapping of triples into Local cache, and ultimate result after reduce operation is stored into Triple cache.
- Triple cache: It contains selected set of triples from the database store along with their indexes. The refined result set derived from the cache is stored into User Domain specific cache where the operation is performed by Domain identification system.
- User Domain specific cache at local Application/ web server: Domain identification system selects particular rows from the triple cache and stores them into the object cache. As a result, more user specific results will be derived on the basis of selecting particular set of objects which belong to specific domain in which user is interested.
- Domain Identification System: To understand this properly we can consider a condition that if a user Alice searches details about some particular word "apple" then we can say that Object cache will contain a set names of fruits and Triple cache will have names and information related to set of fruits and local cache attached to the triple server will contain names and information related to fruits which is stored in their respected triple server.

#### *3.2.1.1 Algorithm*

- Input is provided by client.
- The domain computation will decide the object by determining the domain of input given by the client.
- Local web server will check the object entered by end user in object cache, it will lead to a miss because initially the object cache is empty, so it has to be updated for the first time and for that it will send an update request to the triple cache along with the input of client (which is considered to be the object).
- Then triple cache is accessed for data and it will also lead to a miss so, it will send an update request to the set of local caches which are tightly coupled to distributed database servers.
- The Map Reduce function will search the object field in the triple database server for the object taken as an input by the client and each of the tightly coupled local cache got updated for the first time.
- Similarly, triple cache will also get updated with respective triples along with indices for fast access and retrieval.

- Ultimately each of the local servers is updated with objects and indices in accordance to the domain provided by the user.
- Firstly, a client has to create an account Initial iterations/Training the caches: (caches are empty):
- The above steps were repeated and the caches will updated by sufficient number of values and the system gets trained by above set of iterations.

## 4. Conclusion

At present, providing efficient reasoning is a very difficult task reason is data size. Big Data can handle these different kinds of data and huge amounts of data. But due to these huge data we need to increase the size of the main memory otherwise cache size will get reduced and performance will decrease. SPARQL can execute fixed queries easily but for ad-hoc queries it faces difficulties. We can use some shortest path algorithm like Prime's, kruskal's algorithm to find the shortest path from RDF and then use SPARQL for query execution. It will decrease the execution time. It is not possible for one cache to improve the performance. So, multilevel cache can increase the performance but it has cache consistency problem.

## 5. References


1. Available from: https://en.wikipedia.org/wiki/SemanticWeb
2. Available from: https://en.wikipedia.org/wiki/Ontology
3. Available from: http://www.w3schools.com/webservices
4. Karthikeyan K, Karthikeyani V. Ontology based concept hierarchy extraction of web data. Indian Journal of Science and Technology. 2015 Mar; 8(6).
5. Viniba V. A hybrid layered approach for ontology matching. Indian Journal of Science and Technology. 2015 Aug; 8(17).
6. Urbani J, Kotoulas S, Maassen J, Harmelen FV, Bal H. WebPIE: A web-scale parallel inference engine using map reduce. Journal of Web Semantics. 2012 Jan; 10:59–75.
7. Available from: http://www.w3.org/TR/rdf-sparql-query
8. Vigneshwari S, Aramudhan M. Social information retrieval based on semantic annotation and hashing upon the multiple ontologies. Indian Journal of Science and Technology. 2015 Jan; 8(2).
9. Antoniou G, Bikakis A. Prolog: A system for defeasible reasoning with rules and ontologies on the Semantic Web. IEEE Trans Knowl Data Eng. 2007 Feb; 19(2):233–45.
10. Milea V, Frasincar F, Kaymak U. tOWL: A temporal web ontology language. IEEE Trans Syst, Man, Cybern B, Cybern. 2012 Feb; 42(1):268–81.
11. Liu B, Huang K, Li J, and Zhou MC. An incremental and distributed inference method for large-scale ontologies based on map reduce paradigm. IEEE Transactions on Cybernetics. 2015, Jan; 45.
12. Available from: w3schools.com/xml/xml_whatis.asp
13. Available from: http://www.w3schools.com/webservices
14. Urbani J, Kotoulas S, Oren E, Harmelen F. Scalable distributed reasoning using map reduce. Proc 8th Int Semantic Web Conf; Chantilly, VA, SA. 2009 Oct. p. 634.
15. Schlicht A, Stuckenschmidt H. Map resolve. Proc 5th Int Conf RR; Galway, Ireland. 2011 Aug. p. 294–9.