A Review of Large-Scale RDF Document Processing in Hadoop MapReduce Framework

Khushboo Tiwari
Central College of Engineering and Management
Dept. of Computer Science and Engineering
Raipur, Chhattisgarh, India

Prof. Abhishek Badholia
Central College of Engineering and Management
Dept. of Computer Science and Engineering
Raipur, Chhattisgarh, India

Abstract

Resource Description Framework (RDF) is Meta data model which can be used to store Meta data information of various large complex datasets which can further be used to extract or infer some meaningful out of it. To process these vast amount of data and infer some reasoning out of these is tedious task. Traditional centralized reasoning methods are not sufficient to process large ontologies. Distributed reasoning methods are thus required to improve the scalability and performance of inferences. MapReduce is widely used parallel computing model which able to process these huge amount of data in no time. Various methods are introduced to process RDF document. This paper reviews various methods and techniques used for processing RDF documents.

Keywords— *Information retrieval, RDF Document, Semantic Information, XML, MapReduce, Hadoop.*

I. INTRODUCTION

With the advancement of internet, the RDF data is increasing day by day and its amount reach at the level at which serious data loss can occur if not handled properly. So to process RDF data becomes a key and important problem which need to address quickly. MapReduce programming methodology provide better solution for processing large scale dataset, with an outstanding performance.

With a huge amount of RDF data and their rapid growth, extensive applications have evolved in a majority of domains such as life science and healthcare [1] , business process management [2], expert systems, cloud management, e-marketing etc. The Semantic Web [3] which is defined by WWW organization was estimated to contain 5 billion triples in 2010 and has now reached over 30 billion triples.

Growth of RFD data is still increasing. Resource description framework (RDF) is representation of ontologies which are used to describe knowledge in the form of Semantic Web. Triples are the fundamental unit of RDF document. These can be used for reasoning of various important queries. Triples shows important relationship between objects and subjects. Deriving inferences from these huge amount of data is very challenging task. The 3 of the challenges are:

- Distribution of data around the different server: The data is distributed around the data center. The task for aggregation of data and getting appropriate result is one of the challenging tasks.
- Growing amount of data: Day by day the amount of data generated by the internet is increasing at record rate. To construct a system which can handle those bulky data are needed. Not only bulky, but also complex in structure. Large system with higher computational capability also needed.
- To satisfy online query generated by the user, is one of the most important and crucial task in reasonable amount of time.

There are many performance lack in centralized approach. To overcome the performance issues with centralized method, Hadoop MapReduce programming is introduced for parallel processing.

However, existing distributed reasoning methods [5]–[9] focus on computing RDF closure for reasoning, which takes too much time (usually several hours or even days for large ontologies) and space (generally the ontology size is more than the original data size). Moreover, each time when new RDF arrives, full reasoning over the entire dataset is needed to compute the new RDF closure.

This process occurs at every update, which is too time-consuming in practice. WebPIE distinguishes newly-arrived RDF triples and old ones but fails to consider the relations between them, thus resulting in a huge number of duplicated triples during the reasoning thereby hampering its performance.

A. RDF

RDF document is constructed using extensible markup language (XML). RDF follow XML for syntax and universal resource identifier (URI) for naming conventions. RDF is assertion language used to infer meaning information and relation from objects and subjects. The fundamental unit of RDF is triple. Triple is used to describe relationship between objects.

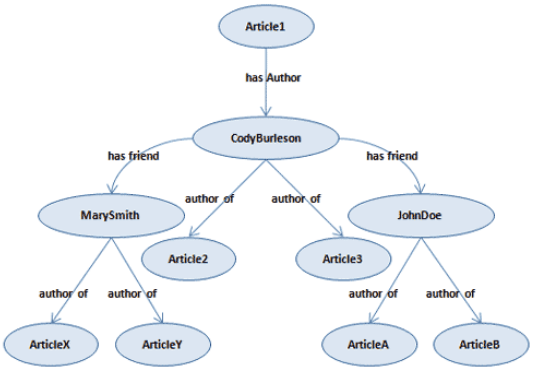

Fig. 1. Shows RDF Document Graph

The formal definition is:

<Subject, Predicate, Object>

Subject used to denotes resource, predicate denotes properties of resource and expresses the relationship between object and resources.

```
<rdf:RDF
        xmlns:fs  = "http://www.ais.columbia.edu/sws/xmlns/cufs#"
        xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:dc  = "http://purl.org/dc/elements/1.0/"
        xmlns:cms = "http://www.ais.columbia.edu/sws/xmlns/cucms#"
    >
    <fs:File rdf:about = "/docs/manual/develop/rdf.xml">

        <cms:comment>added sample of RDF serialized in XML</cms:comment>

        <cms:editor>Alex Vigdor</cms:editor>

        <dc:title>RDF: Extensible Metadata</dc:title>

        <dc:format>text/xml</dc:format>

        <dc:date>2003-11-06</dc:date>

        <dc:type>http://purl.org/dc/dcmitype/Text</dc:type>

        <dc:identifier>/docs/manual/develop/rdf.xml</dc:identifier>

        <dc:creator>Alex Vigdor</dc:creator>

    </fs:File>
</rdf:RDF>
```

Fig. 2. Shows RDF Document Structure

B. Hadoop MapReduce

Hadoop MapReduce is a software framework for executing huge amount of data i.e. terabyte data sets in parallel environment on large clusters (in thousands of data nodes) which can be commodity hardware in a fault tolerant manner. MapReduce job splits the input data set into various chunks of files which then are processed by the map tasks in parallel fashion. The hadoop framework sorts the output of map phase
Which are then input to the reduce tasks. Both input and output files are stored on HDFS (Hadoop Distributed File System). The Hadoop framework has a responsibility of managing and scheduling tasks.

The MapReduce framework consists of a single master JobTracker and one slave TaskTracker per cluster-node. The master is responsible for scheduling the jobs' component tasks on the slaves, monitoring them and re-executing the failed tasks [4].

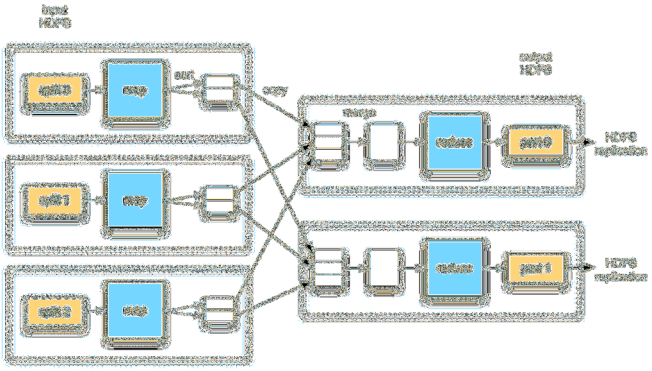

Fig. 3. Shows the flow of MapReduce job

II. LITERATURE SURVEY

A. Inference on heterogeneous e-marketplace activities

Massimo Melucci et. al. [5] focused on a class of RF algorithms inspired by quantum detection to re-weight the query terms and to re-rank the document retrieved by an IR system. These algorithms project the query vector on a subspace spanned by the eigenvector which maximizes the distance between the distribution of quantum probability of relevance and the distribution of quantum probability of non-relevance. The experiments showed that the RF algorithms inspired by quantum detection can outperform the state-of-the-art algorithms. In this paper author studied the problems on how to transform business documents between heterogeneous marketplaces and how to generate the possible result against a set of requirements and user preferences. Regarding to the first problem, we adopt XPM Schema to solve the semantic consistency problem.

Method Used: multiphase forward-chaining algorithm
Data Source: e-commerce RDF
Strength: Can able to process heterogeneous data.
Limitation: farward chaining process will weaken performance of the system. Although RDF data is sorted, there is still the before-mentioned performance loses.

B. TOWL: A temporal web ontology language

Kevyn Collins-Thompson et. al. [6] focused on models and algorithms to address the three key problems in improving relevance for search using reading difficulty: estimating user proficiency, estimating result difficulty, and re-ranking based on the difference between user and result reading level profiles. Author evaluate our methods on a large volume of Web query traffic and provide a large-scale log analysis that highlights the importance of finding results at an appropriate reading level for the user.

Method Used: OWL-based temporal formalism

Data Source: Financial Data

Strength: effectively recommend stocks using web based application.

Limitation: These ontology reasoning methods are conducted on a single machine or local cluster. The reasoning speed is directly dependent on the scale of the ontology, which is not suitable to a large ontology base.

C. Scalable Distributed Reasoning using MapReduce

Donna Harman et. al. [7] focused on experiments, using the Crartfield 1400 collection~ showed the importance of query expansion in addition to query reweighing, and showed that adding as few as 20 well-selected terms could result in performance improvements of over 100%. Additionally it was shown that performing multiple iterations of feedback is highly effective.

Method Used: closure of an RDF graph and naive bayes

Data Source: Falcon dataset, which contains 35 million triples

Strength: deployed it on a compute cluster of up to 64 commodity machines which gives better running time for algorithm.

Limitation: Above method considered no influence of increasing data volume, and did not answer how to process users' queries.

D. WebPIE: A web-scale parallel inference engine using mapreduce

Ingo Frommholz et. al. [8] focused on an approach inspired by quantum mechanics to represent queries and their reformulations as density operators. Differently constructed densities can potentially be applied for different types of query reformulation. To do so, we propose and discuss indicators that can hint us to the type of query reformulation we are dealing with.

Method Used: WebPIE to Calculate RDF closure

Data Source: protein sequence

Strength: achieved highly optimized execution of joins required to apply the RDFS and OWL-horst rules.

Limitation: the performance of incremental updates was highly dependent on input data. Furthermore, the relationship between newly-arrived data and existing data is not considered and the detailed implementation method is not given.

E. History Matters: Incremental Ontology Reasoning using modules

Yuanhua Lv et.al. [9] Focused on an approach to adaptively predict the optimal balance coefficient for each query and each collection. We propose three heuristics to characterize the balance between query and feedback information. Taking these three heuristics as a road map, we explore a number of features and combine them using a regression approach to predict the balance coefficient. Our experiments show that the proposed adaptive relevance feedback is more robust and effective than the regular fixed-coefficient feedback.

Method Used: OWL classification

Data Source: Truth Maintenance Ontologies

Strength: has significant improvement over regular classification time on a set of real-world ontologies.

Limitation: proposed method is used for OWL but not suitable to the increasing RDF data. Besides, since no distributed reasoning method is adopted, the reasoning speed is a huge problem when dealing with a large ontology base.

III. CONCLUSION

Symantec web search stores information in the RDF data format. Hence RDF document needs to be toughly analyzed to infer reasoning from it. MapReduce programing model helps in doing so. It process the job parallel and can scale over thousands of nodes if RDF document size increases.

Many researchers of RDF data processing focus on forward chaining RDF reasoning and query to get some meaningful out of those huge amount of data. But still there are some limitation in existing system to process RDF data which need further improvement in future.

IV. DISSCUSSION AND FUTURE SCOPE

Various authors have proposed various methods to identify and infer reasoning from the RDF documents. There are huge number of data are present for analysis. The standalone system cannot perform well with large number of document. Hence need some parallel method to overcome performance issues.

The existing algorithm performs incomplete RDFS reasoning. Authors ignore RDF axiomatic triples which is widely accepted practice and in line with most of the existing reasoners. Author also omit the rules with one antecedent since parallelizing their application is trivial and they are commonly ignored by reasoners as being uninteresting. If standard compliance is sought, these rules can be implemented with a single map over the final data, which very easy to parallelize and should not take more than some minutes.

Further study is needed to overcome performance of RDF document processing algorithm. The Existing system can solve the problem but increases overhead.

REFERENCES

[1] M. S. Marshall et al., "Emerging practices for mapping and linking life sciences data using RDF—

A case series," J. Web Semantics, vol. 14, pp. 2–13, Jul. 2012

[2] M. J. Ibánez, J. Fabra, P. Álvarez, and J. Ezpeleta, "Model checking analysis of semantically annotated business processes," IEEE Trans. Syst., Man, Cybern. A, Syst., Humans, vol. 42, no. 4, pp. 854–867, Jul. 2012.

[3] http://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/DataSets/Statistics

[4] Sandeep Kumar Dewangan, Shikha Pandey, Toran Verma. 2016. "A Distributed Framework for Event Log Analysis using MapReduce", IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), Tamil Nadu, ISBN: 978-1-4673-9544-1 pp. 587-590

[5] H. Paulheim and C. Bizer, "Type inference on noisy RDF data," in Proc. ISWC, Sydney, NSW, Australia, 2013, pp. 510–525.

[6] V. Milea, F. Frasincar, and U. Kaymak, "tOWL: A temporal web ontology language," IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 42, no. 1, pp. 268–281, Feb. 2012.

[7] J. Urbani, S. Kotoulas, E. Oren, and F. Harmelen, "Scalable distributed reasoning using mapreduce," in Proc. 8th Int. Semantic Web Conf., Chantilly, VA, USA, Oct. 2009, pp. 634–649.

[8] J. Urbani, S. Kotoulas, J. Maassen, F. V. Harmelen, and H. Bal, "WebPIE: A web-scale parallel inference engine using mapreduce," J. Web Semantics, vol. 10, pp. 59–75, Jan. 2012.

[9] B. C. Grau, C. Halaschek-Wiener, and Y. Kazakov, "History matters: Incremental ontology reasoning using modules," in Proc. ISWC/ASWC, Busan, Korea, 2007, pp. 183–196.