AN EFFICIENT MANAGEMENT OF UNCERTAIN MILLIONS NODE RDF WITH MAP-REDUCE BASED ALGORITHM BY USING COLUMN ORIENTED VERTICAL-PARTITIONING TECHNIQUES IN BIG DATA

[1]Foram M. Gohel,[2]Ashutosh A. Abhangi
[1]ME Scholar,[2]Assistant Professor
[1 2]Computer Engineering Department,
[1 2]Noble Group of Institution, Junagadh, Gujarat.

Abstract : Now-a-days management of large scale RDF graph is very challenging task. It is not easy to access and manage large-scale, million-node (big) RDF graphs. A possible solution to this problem is requiring Map-Reduced based algorithms and techniques, in using semantic web. So RDF data can management done more efficiently. From the relational database perspective, efficiency and scalability of RDF data model are derived from triplet model easily. so, in this research we describe the column oriented vertical-partitioning with triple nature of RDF. By using these different approaches we analyze that using vertical-partitioning in RDF triple nature, we can reduce storage.

Index Terms - RDF graphs, Map-Reduce, Semantic web, Vertical-Partitioning, COVP, etc….

I. INTRODUCTION

Big data is a covering all term for the non-traditional strategies and technologies needed to collect, organize and process from large datasets. While the problem of working with data that extends the computing power or storage of a single computer is not new, the ubiquity, scale, and value of this type of computing has greatly become larger in modernistic years. The Word Big Data Defines as Extremely huge data sets, which may be analyzed computationally to reveal patterns, trends, and associations, especially relation to human behavior and interactions.[1] The term big data applies to information that cannot be processed or analyzed using traditional processes or tools. Increasingly, organizations today we are facing more and more big data challenges. Big data challenges are indexing, shorting, search, manage, data creation, sharing, transfer updating and information privacy.[14]

A huge repository of terabytes of data is generated each day from modern information systems and digital technologies such as Internet of Things and cloud computing.[15] Analysis of these massive data requires a lot of efforts at multiple levels to extract knowledge for decision making. Therefore, big data analysis is a current area of research and development. Hence, for managing these terabytes of data the concept of RDF is arrived.[3]

The Resource Description Framework (RDF) is a general framework for how to describe any Internet resource such as a Web site and its content. An RDF description (such descriptions are often referred to as metadata, or (”data about data”) can include the authors of their source, date of creation or updating, the organization of the pages on a site (the sitemap),information that describes content in terms of audience or content rating, key words for search engine data collection, subject categories, and so forth. The Resource Description Framework will make it possible for everyone to share Website and other descriptions more easily and for software developers to build products that can use the metadata to provide better search engines and directories, to act as intelligent agents, and to give Web users more control of what they’re viewing.[2]

The Resource Description Framework (RDF) is a language for representing information about resources in the World Wide Web. This Primer is designed to provide the reader with the basic knowledge required to effectively use RDF. It introduces the basic concepts of RDF and describes its XML syntax. It describes how to define RDF vocabularies using the RDF Vocabulary Description Language, and gives an overview of some deployed RDF applications. It also describes the content and purpose of other RDF specification documents.[9]

The following example is taken from the W3C website describing a resource with statements "there is a Person identified by http://www.w3.org/People/EM/contact#me, whose name is Eric Miller, whose email address is e.miller123(at)example, and whose title is Dr.[8]

Fig 1.1: An RDF Graph[9]

The resource "http://www.w3.org/People/EM/contact#me" is the subject.

The objects are:

- "Eric Miller" (with a predicate "whose name is"),
- mailto:e.miller123(at)example (with a predicate "whose email address is"), and
- "Dr." (with a predicate "whose title is").

The subject is a URI.

The predicates also have URIs. For example, the URI for each predicate:

- "whose name is" is http://www.w3.org/2000/10/swap/pim/contact#fullName,
- "whose email address is" is http://www.w3.org/2000/10/swap/pim/contact#mailbox,
- "whose title is" is http://www.w3.org/2000/10/swap/pim/contact#personalTitle.

In addition, the subject has a type (with URI http://www.w3.org/1999/02/22-rdf-syntax-ns#type), which is person (with URI http://www.w3.org/2000/10/swap/pim/contact#Person).[3]

Therefore, the following "subject, predicate, object" RDF triples can be expressed:

- http://www.w3.org/People/EM/contact#me, http://www.w3.org/2000/10/swap/pim/contact#fullName, "Eric Miller"
- http://www.w3.org/People/EM/contact#me, http://www.w3.org/2000/10/swap/pim/contact#mailbox, mailto:e.miller123(at)example
- http://www.w3.org/People/EM/contact#me, http://www.w3.org/2000/10/swap/pim/contact#personalTitle, "Dr."
- http://www.w3.org/People/EM/contact#me, http://www.w3.org/1999/02/22-rdf-syntax-ns#type, http://www.w3.org/2000/10/swap/pim/contact#Person

Map-Reduce are a parallel programming model in displayed by Google. The thought is unique from utilitarian programming dialect in big data. Map-Reduce parts the issue them handling into two phases (outline and lessen arrange). The guide organize expends are (critical, esteem) of sets and gatherings of yield in (key, esteem) matches too. The organize forms the yield of guide arrange with keys and yields the last outcome. Map-Reduce structure are simply requires to the software engineer giving guide and reduce (join) strategy. However, just if the undertaking can be preoccupied as tasks over (key, esteem) Map-Reduce is reasonable.[5]

The Map-Reduce algorithm the important tasks, namely Map and Reduce.

1. The Map task takes a set of data and converts it into another set of data, where individual elements are broken down into tuples key-value pairs.

2. The Reduce task takes the output from the Map as an input and combines those data tuples key-value pairs into a smaller set of tuples.

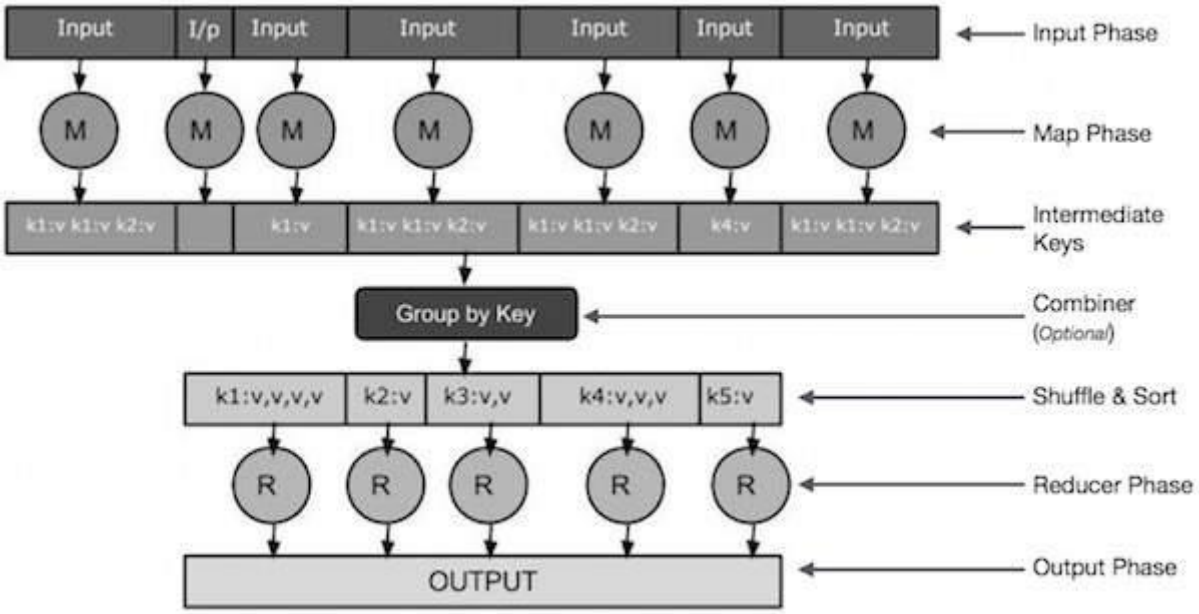


Fig 1.2: Map-Reduce Algorithm Phases[9]

There are four general types (most common categories) of NoSQL databases. Each of these categories has its own specific attributes and limitations. There is not a single solutions which is better than all the others, however there are some databases that are better to solve specific problems[19]. To clarify the NoSQL databases, let's discuss the most common categories:

- Key-value stores
- Column-oriented
- Graph
- Document oriented

II. RELATED WORK

An efficient RDF storage schema should offer both scalability in its data management performance and variety in its data storage, processing and representation.[5]

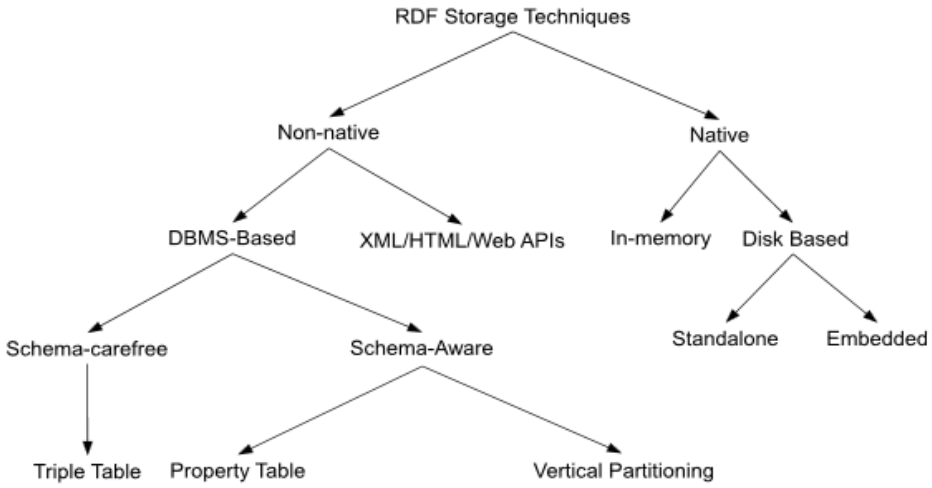


Fig 2.1: A Classification of RDF Storage Approaches.[5]

Vertical partitioning divides a table into multiple tables that contain fewer columns. The two types of vertical partitioning are normalization and row splitting:

- Normalization is the standard database process of removing redundant columns from a table and putting them in secondary tables that are linked to the primary table by primary key and foreign key relationships.
- Row splitting divides the original table vertically into tables with fewer columns. Each logical row in a split table matches the same logical row in the other tables as identified by a UNIQUE KEY column that is identical in all of the partitioned tables. For example, joining the row with ID 712 from each split table re-creates the original row.[6]

Like horizontal partitioning, vertical partitioning lets queries scan less data. This increases query performance. For example, a table that contains seven columns of which only the first four are generally referenced may benefit from splitting the last three columns into a separate table.[11]

Vertical partitioning should be considered carefully, because analyzing data from multiple partitions requires queries that join the tables. Vertical partitioning also could affect performance if partitions are very large.

2.1 Proposed Flow

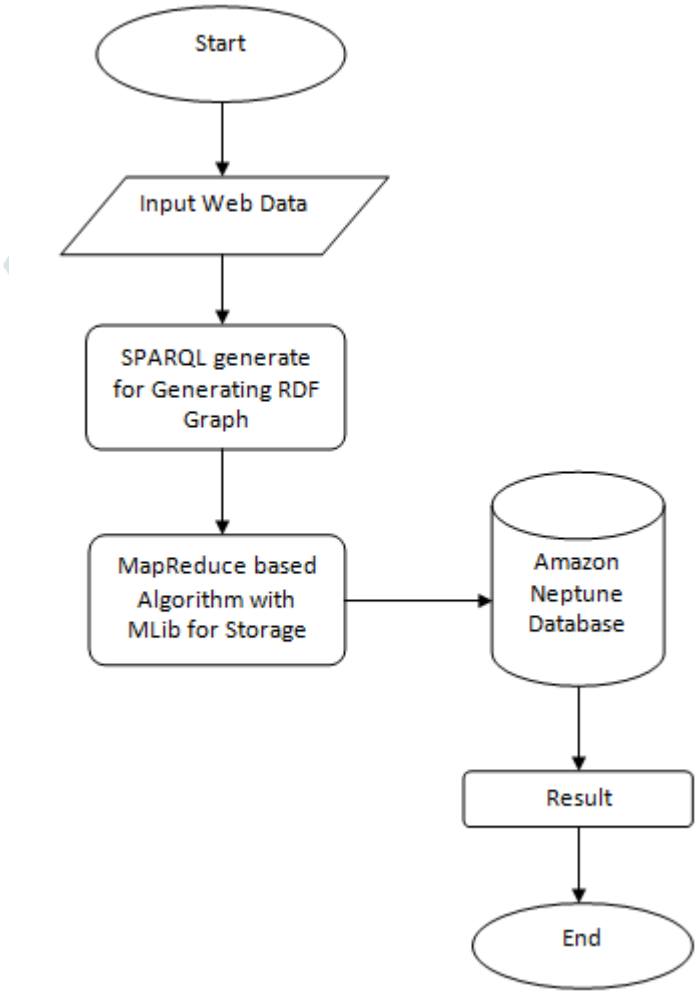


Fig 2.2: Proposed Flow

1. *Input Web Data*

   Take BSBM or LinkedMDB data set as an input.

2. *SPARQL generate for Generating RDF Graph*

   In this step We use SPARQL Generate Technique to convert large data set into RDF Graph.

3. *MapReduce based Algorithm with COVP1 Technique[7] for Storage*

   The MapReduce algorithm the important tasks, namely Map and Reduce.

   (a) The Map task takes a set of data and converts it into another set of data, where individual elements are broken down into tuples key value pairs.

(b) The Reduce task takes the output from the Map as an input and combines those data tuples key value pairs into a smaller set of tuples.

4. *Amazon Neptune Database*

In this step we take Amazon Neptune Database for storing the converted RDF graph.

5. *Result*

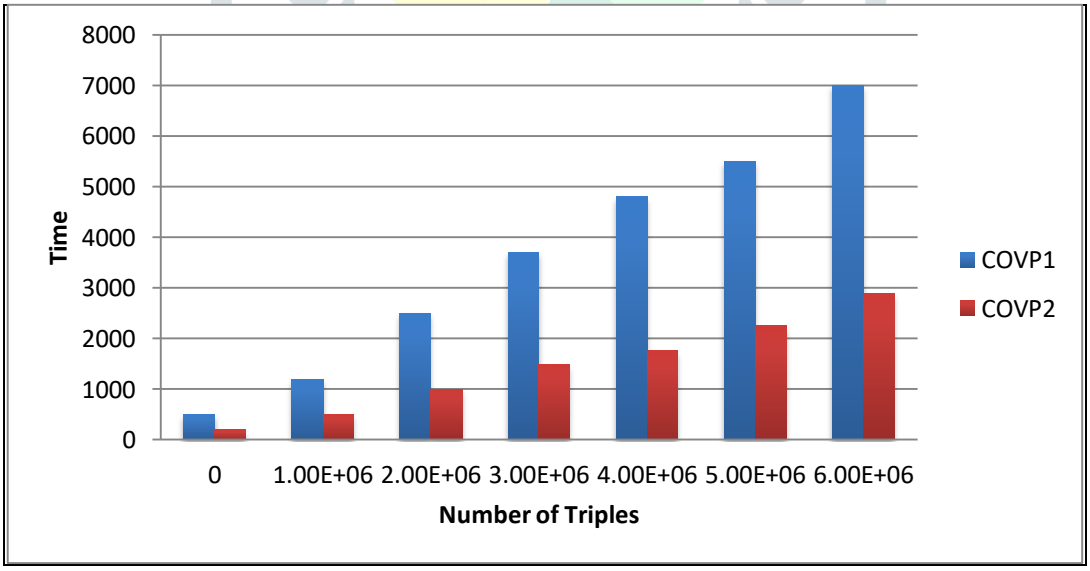
This is the last phase, in that we get the final accurate result.

III. IMPLEMENTATION

When a theoretical design is converted into a working system, this phase is called implementation and therefore it is the most critical phase in entire research. Implementation stage requires proper planning and understanding and limitations of existing system. It is the phase in which we apply our own methodology to mitigate the limitations and to enhance the existing mechanism.

Here, We have some Experimental results of the techniques for vertical partitioning method which is COVP1 and COVP2. From these results We found the graph which is given bellow that tells that the COVP1 technique is better for store the RDF graph in Big data world.

Number of Triples	COVP1	COVP2
0	500	200
1.00E+06	1200	500
2.00E+06	2500	1000
3.00E+06	3700	1478
4.00E+06	4800	1753
5.00E+06	5500	2256
6.00E+06	7000	2900



CONCLUSION

In the real-life application and systems are characterized by imprecise and uncertain data. Big RDF graphs can be such a nature too, as originated by plethora of scientific applications which all naturally introduce impression and uncertainty in data. There is no any efficient method to manage Uncertain Big RDF data. So in our future work, we convert big web data into RDF graph and then manage them through COVP1 Technique and Amazon Neptune database. From this we could get better result as saw in implementation.

REFERENCES

PAPERS

[1] LaValle, S., Lesser, E., Shockley, R., Hopkins, M.S. and Kruschwitz, N., 2011. Big data, analytics and the path from insights to value. *MIT sloan management review*, *52*(2), p.21.

[2] Papailiou, N., Tsoumakos, D., Konstantinou, I., Karras, P. and Koziris, N., 2014, June. H 2 RDF+: an efficient data management system for big RDF graphs. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data* (pp. 909-912). ACM.

[3] Urbani, J., Maassen, J., Drost, N., Seinstra, F. and Bal, H., 2013. Scalable RDF data compression with MapReduce. *Concurrency and Computation: Practice and Experience*, *25*(1), pp.24-39.

[4] Katal, A., Wazid, M. and Goudar, R.H., 2013, August. Big data: issues, challenges, tools and good practices. In *2013 Sixth international conference on contemporary computing (IC3)* (pp. 404-409). IEEE.

[5] Faye, D.C., Cure, O. and Blin, G., 2012. A survey of RDF storage approaches. *Revue Africaine de la Recherche en Informatique et Mathématiques Appliquées*, *15*, pp.11-35.

[6] Shukla, V. and Tiwari, R., 2013. Column Oriented Database: Implementation and Performance Analysis. *vol*, *4*, pp.2013-2015.

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

WEBLINKS

[8] https://en.wikipedia.org/wiki/Resource_Description_Framework

[9] https://www.w3.org/TR/rdf-primer/

[10] https://www.tutorialspoint.com/map_reduce/map_reduce_quick_guide.htm

[11] https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/ms178148(v=sql.105)

BOOKS

[12] Zikopoulos, P. and Eaton, C., 2011. *Understanding big data: Analytics for enterprise class hadoop and streaming data*. McGraw-Hill Osborne Media.

[13] Mayer-Schönberger, V. and Cukier, K., 2013. *Big data: A revolution that will transform how we live, work, and think*. Houghton Mifflin Harcourt.

PATENTS

[14] ENHANCED HADOOP FRAMEWORK FOR BIG-DATA APPLICATIONS. US10268716B2, 2016.

[15] METHOD AND APPARATUS FOR DATA PARTITIONS IN THE KEY-VALUE DATABASE. CN102799628B,2012.