TEST Engineering & Management

May – June 2020
ISSN: 0193-4120 Page No. 5764-5769

Effective Query Processing for Web-scale RDF Data using Hadoop Components

C. Lakshmi[1], K. UshaRani[2]
[1]Research Scholar, [2]Professor
Department of Computer Science, Sri Padmavati Mahila Visvavidyalayam, Tirupati, A.P, India.
[1]lakshmibhavya@gmail.com, [2]usharanikuruba@yahoo.co.in

Article Info
Volume 83
Page Number: 5764 - 5769
Publication Issue:
May - June 2020

Article History
Article Received: *19 November 2019*
Revised: *27 January 2020*
Accepted: *24 February 2020*
Publication: *17 May 2020*

Abstract

Semantic Web Data is an add-on for the World Wide Web, and the main objective of this is to make the internet data machine-readable. Resource Description Framework (RDF) is one of the technologies used to encode and represent the semantics data in the form of metadata. Generation of the semantic data is growing day by day into large number and it's becoming complicated to Process and Store using the Traditional database systems, Hadoop and Spark are the popular open-source tools for Processing (Map-Reduce) and Storing (HDFS) a large amount of data. Using these bigdata tools can analyze the terabytes of the data in a distributed parallel process. In this paper, by executing the benchmark queries in Hive and Spark by using RDF data, Spark has an in-memory computation that can give faster results using Resilient Distributed Datasets (RDD). A scalable and faster framework can be obtained based on practical evaluation and analysis. Hence, by experimenting with the proposed system Spark has been given better performance results in processing the semantic web data when compared with the Hive.

Keywords: *RDF, HDFS, Spark, DAG, Hive, and Query Processing.*

I. INTRODUCTION

Fast increasing of the semantic web data is creating a complex challenge regarding the processing of the query. It is represented as a Resource Description Framework (RDF) [2, 3], which is a core method in representing the semantic data in machine Readable Formats. RDF plays an essential role in semantic data and data integration [22]. RDF involves in making semantic web data in a structured format of subject-property-object expression [5]. Following Figure 1 gives the example for the sematic Web data.

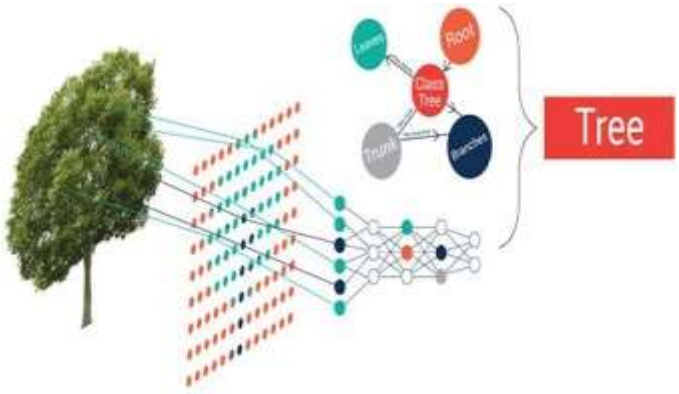

Fig 1. Semantic Web data example [20]

Here each structure of this record is considered as a triple. The subject in Figure 1 represents as resources (Tree), and the property in Figure 1 acts as the relationship between the subject and object i e . , Branches, leaves, trunk, root, Subject can be considered as URI (Uniform Resource Identifier)

May – June 2020
ISSN: 0193-4120 Page No. 5764-5769

or blank nodes [18]. Objects are literals, whether it can approach URI or a value.

Semantic Web data can be of two types

1) Linked data
2) Open data

Linked data is considered as a significant part of the semantic web data. Open data can be freely available and can be considered without any objections and it's not equal to linked data and no more links related to other data [18, 5].

Open data can be freely available and can be considered without any objections.nd it's not equal to linked data and no more links related to other data [18].

Linked Open Data

It is an efficient data which is collaborated with both linked data and open data. Ontext Graph database can handle the vast datasets coming from many sources and link them to open data [18]. It provides Richer queries and significant data-driven analytics. The Following Figure 2 flow represents the linked open data standard rules.

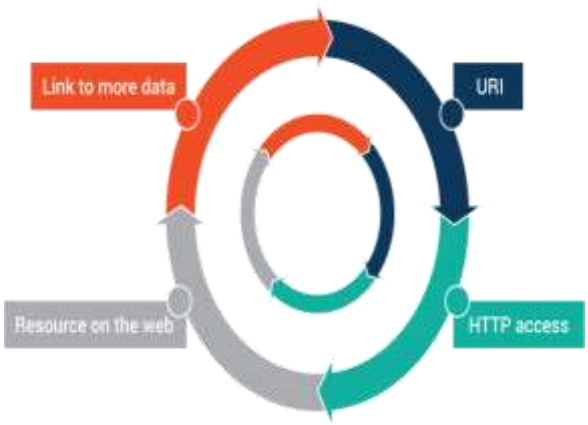

Fig 2. Linked Open data [21]

Linked open data givens a well-organized data integration, and browsing through complex data becomes more accessible and much more systematic [18]. It acts as the metadata for the retrieval of the better results from the web data and gives useful information to the people with enriching results.

Hadoop

Hadoop is considered a popular solution for the processing of a large amount of data [16]. It is come up with the storage as Hadoop Distributed File System (HDFS) and Processing as Map-Reduce (MR) [1].

It is an open source framework and distributed processing framework written in Java .It is used for storage for big data applications and data processing running in clustered systems [15]. It is at the centre of a developing ecosystem of big data technologies that are essentially used to reinforce advanced analytics initiatives, comprising of predictive analytics, machine learning applications and data mining. Hadoop can handle several forms of unstructured and structured forms of information, giving the users additional flexibility for processing, collecting and analyzing the information than relational databases and data warehouses provides [20]. It comprises of various components that permit the storage and processing of large volumes of data in a clustered environment [4].

It is a programming model that can process a large amount of the datasets in a parallel and distributed algorithm on a cluster, which can prepare using the filtering, sorting, and reduce method.

Hadoop Distributed File System

The increasing of a large amount of data leads to the solution as HDFS (Hadoop Distributed File System) which can stores and process the data [19,15].

1) Structured data

2) Unstructured data

Data volume

May – June 2020
ISSN: 0193-4120 Page No. 5764-5769

When we compare with the RDBMS a large amount of the data can be stored easily in the Hadoop and it can easily process and give the different encryption formats to transfer the data to other file systems [19].

Objective

The main objective of this to help the architects and the organization to do the bigdata analytics option to make available in the cloud to make it a dynamic process that will help the organization [15] and the industries to get the analysis on the up to date dynamically.

II. LITERATURE REVIEW

Bigdata tools played a crucial role in the processing of the RDF data with the Hadoop components, below are the some of the experimental studies carried out in querying the big sematic web data experiments. In order to process the data using the parallel distributed system processing time has been reduced in analyzing the number of increased bigdata.

T. Padiya, M. [14] explained the distribution of the RDF data processing using the Hadoop components and apache spark batch processing have applied the MapReduce model to RDF data to achieve parallel/distributed processing.

M. F. Husain, [15] experimented the query processing comparision for different bigdata frameworks using the cloud computing tools and Another differentiation between our approach and previous ones is our focus on parallelizing individual queries.

H. Zhang, [16] provides the in-memory big data processing using the Apach spark and explained the DAG(directed Acyclic Graph) speciality in processing the large data

Sara Landset et.al [17] proposed A survey of open source tools for machine learning with big data in the Hadoop ecosystem and improvesthe ability of HDFS to handle modern data by building data awareness modules that detect, distribute, and manage data over the scalable file system. Thus, the framework results in optimization and efficient resource usage of the Hadoop eco-system and other tools and services that use HDFS as a distributed storage.

Xindong Wu et.al [18] proposed several data mining techniques with Big Data and represents the performance evaluation.
K. Anusha et.al [19] provides an introduction about Big Data Characteristics and Hadoop Distributed File System.

III. EXPERIMENTAL SETUP

To process the 5000 tuples of the dataset, which has been taken from the DBpedia open source linked open data [17]. We propose the processing of semantic web data using cloud services for high availability and better performance results. Amazon web service provides the EMR (Elastic Map Reduce) functionality, which will effect the data retrieving process. Proposed Architecture works on the three open-source DBpedia data sets having the linked data related to them each. The Architecture is represented in the following Figure 3.

Data Loading

Loading of the semantic web data of RDF format file into the Hadoop storage is of different scripts in each tool. Hive uses the HIVE-QL and spark uses the SPARK-SQL syntax for loading and reading the data into corresponding servers and frameworks [12].

Processing Stage

After immediate loading of the data into respective frameworks processing stage starts, here the logic plays a crucial role in the retrieval process of the data to systematic business approaches

May – June 2020
ISSN: 0193-4120 Page No. 5764-5769

and analysis [15]. In this phase, Map-Reduce came into existence with the applying number of mappers and reducers to parallelize the process and get faster results. As the framework variation gives the different processing time for data retrieval [10].

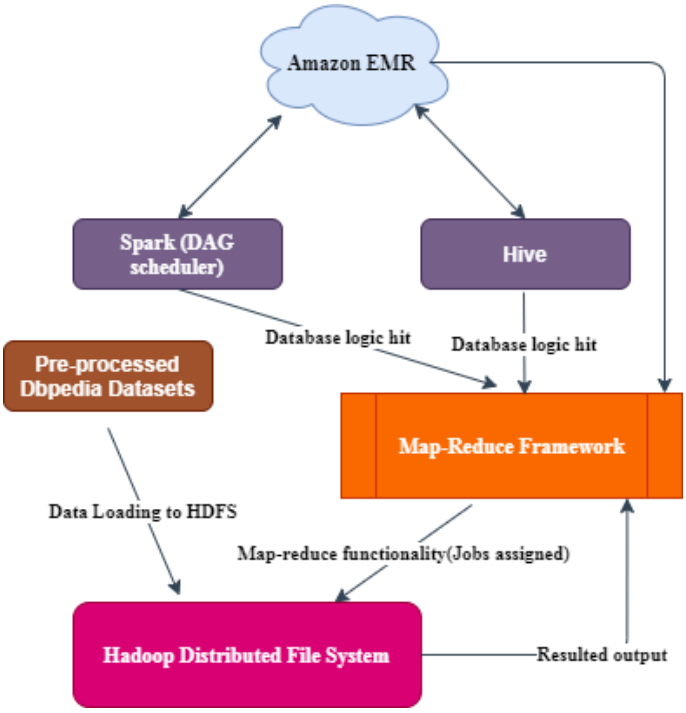

Fig 3. Proposed Architecture

Results Stage

With the resulted outputs from the different frameworks performed in the processing stage, we consider the CPU utilization, number of mappers, reducers, and jobs assigned. It can be compared with the two proposed frameworks and can conclude the best fit framework for the processing of the semantic web data [8].

Proposed Algorithm

Step 1: Collecting DbPedia Datasets.

Step 2: Load the data to HDFS Storage.

Step 3: Applying Map-Reduce Logic using Hive and Spark.

Step 4: Extracting the data from the Distributed File System.

Step 5: Evaluating the Query Processing time.

IV. IMPLEMENTATION

Using the Benchmark bottleneck queries from the DBpedia [22], we use the standard productive questions to retrieve and hit the databases to each framework. Following is the sample SPARQL query, this can be converted into each proposed frames and compare the best suitable results which can conclude our system [22] .

Hive case

It is equivalent to the SQL in syntax, but the processing is different when compared to the relational database and the hive-ql, which is a distributed processing system., The Map-Reduce method is applied in this, and the results are noted [14].

Spark Case

Here comes the unique feature in the spark with the DAG Scheduler [4], which is an in-memory processing unit [16], it differs from the stage to stage variation in consumption of the retrieval time. Spark is only meant for the processing, datasets are loaded into the Hadoop file system, and it is integrated with the Spark for database hit [17].

Result Analysis

By considering the proposed system, a comparison between the spark and hive for each query using the benchmark join queries [22] to get the efficient query processing results are tabulated in Table 1.

Table 1. Processing results

Queries	Time taken to Execution (sec)	
	Hadoop+Spark	**Hadoop+Hive**
1st query	41	52
2nd query	127	316
3rd query	48	76
4th query	74	104
5th query	88	171

May – June 2020
ISSN: 0193-4120 Page No. 5764-5769

The query processing using the proposed system can be analyzed Figure 3 using the benchmark queries which can analyze the best processing Map-Reduce tool. Hive and Spark have their own syntax for query logic transformed according to that. The proposed system says in comparing the best tool to process and analyze the data in Spark [12]. When we compare the difference between the Hive and Spark, it says that the Spark has given the best results in processing. The below graph represents the number of seconds are taken to process the data in different bigdata tools [6].

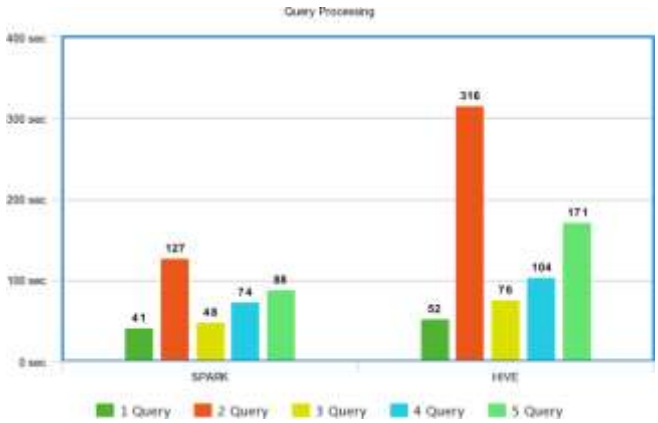

Fig 4. Query Processing between Hive vs Spark (Sec)

V. CONCLUSION AND FUTURE SCOPE

Efficient query processing is performed with the proposed big data techniques. In which standard datasets i.e., DBpedia is carried out for analyzing the efficiency of query processing and its empirical analysis. The datasets are not loaded based on the partitioning and bucketing for the Hive-ql, which may affect the faster data retrieval compared to other frameworks. In the experimental, two big data tools i.e., Hive, and Spark are taken for implementing and analysis of the performance of query processing. It is observed that Spark has shown as outperformed the other in big data query processing. Spark shows better results Figure 4. In the future work, it is planned to develop a scalable and optimized distributed computing framework for reducing the required number of jobs and effective CPU utilization with the increased cluster size in the cloud. Many frameworks came into existence to process the semantic web data in a distributed method can attempt these benchmark queries which may get better results.

REFERENCES

[1] Wang, X., Chai, L., Xu, Q. et al. Efficient Subgraph Matching on Large RDF Graphs Using MapReduce. Data Sci. Eng. 4, 24–43 (2019).

[2] Mouad Banane1, Abdessamad Belangour, "An Evaluation andComparative study of massive RDF Data management approachesbased onBig Data Technologies", International Journal of Emerging Trends in Engineering Research, vol 7, 48-53 (2019).

[3] Sakr S., Wylot M., Mutharaju R., Le Phuoc D., Fundulaki I. Distributed RDF Query Processing. In: Linked Data. Springer, Cham (2018)

[4] P Vyshnav, et.al "Parallel Approach of Visualized Clustering Approach (VCA) for Effective Big Data Partitioning", Jour of Adv Research in Dynamical & Control Systems, Vol. 10, 04-Special Issue (2018).

[5] Kaoudi Z., Kementsietsidis A., Query Processing for RDF Databases. In: Koubarakis M. et al. (eds) Reasoning Web. Reasoning on the Web in the Big Data Era. Reasoning Web 2014. Lecture Notes in Computer Science, vol 8714. Springer, Cham (2014)

[6] Abadi, D.J., Marcus, A., Madden, S., Hollenbach, K.J.: Scalable Semantic Web Data Management Using Vertical Partitioning. In: VLDB, pp. 411–422 (2007)

[7] Bugiotti, F., Goasdoué, F., Kaoudi, Z., Manolescu, I.: RDF Data Management in the Amazon Cloud. In: DanaC Workshop (in Conjunction with EDBT) (2012)

[8] Vyshnav et.al, "Intelligent System for Visualized Data Analytics a Review", International Journal of Pure and Applied Mathematics, Volume 116 No. 21, 217-224(2017).

[9] Doulkeridis, C., Norvag, K.: A survey of large-scale analytical query processing in MapReduce. VLDB Journal (2013).

May – June 2020
ISSN: 0193-4120 Page No. 5764-5769

[10] Li, F., Le, W., Duan, S., Kementsietsidis, A.: Scalable Keyword Search on Large RDF Data. IEEE Transactions on Knowledge and Data Engineering 99(PrePrints) (2014)

[11] P Anjaiah, et. al., "An Efficient Approach for Secure Storage, Search using AES in Cloud Storage", International Journal of Engineering & Technology, 7 (3.12) 661-665, (2018).

[12] Zhang, X., Chen, L., Tong, Y., Wang, M.: EAGRE: Towards Scalable I/O Efficient SPARQL Query Evaluation on the Cloud. In: ICDE (2013)

[13] Zhang, X., Chen, L., Wang, M.: Towards Efficient Join Processing over Large RDF Graph Using MapReduce. In: Ailamaki, A., Bowers, S. (eds.) SSDBM 2012. LNCS, vol. 7338, pp. 250–259. Springer, Heidelberg (2012)

[14] T. Padiya, M. Bhise, "DWAHP: Workload Aware Hybrid Partitioning and Distribution of RDF Data", *IDEAS-2017*, pp. 235-241.

[15] M. F. Husain, L. Khan, M. Kantarcioglu, B. Thuaisingham, "Data intensive query processing for large RDF graphs using cloud computing tools", *2010 IEEE 3rd International Conference on Cloud Computing. IEEE Press*, 2010.

[16] H. Zhang, G. Chen, B. C. Ooi, K. L. Tan, M. Zhang, "In-Memory Big Data Management and Processing: A Survey", *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 7, pp. 1920-1948, 2015.

[17] Sara Landset, Taghi M. Khoshgoftaar, Aaron N. Richter and Tawfiq Hasanin “A survey of open source tools for machine learning with big data in the Hadoop ecosystem” Journal of Big Data (2015).

[18] Xindong Wu, Xingquan Zhu, Gong-Qing Wu, Wei Ding, “Data Mining with Big Data”, (In Press) IEEE Transactions on Knowledge and Data Engineering, 2013.

[19] K.Anusha, K.Usha Rani, C. Lakshmi “A Survey on Big Data Techniques” Special Issue on Computational Science, Mathematics and Biology IJCSME- SCSMB-16-March-2016, ISSN-2349-8439.

[20] https://www.ontotext.com/knowledgehub/fundamentals/what-is-the-semantic-web/

https://www.ontotext.com/knowledgehub/fundamentals/linked-data-linked-open-data/

[22] http://docs.openlinksw.com/virtuoso/rdfperfgeneraldbpedia/