iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** XI **Month of publication:** November 2020

DOI: https://doi.org/10.22214/ijraset.2020.32082

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue XI Nov 2020- Available at www.ijraset.com

A Survey Paper on Effective Query Processing for Semantic Web Data using Hadoop Components

C. Lakshmi[1], Dr. K. Usha Rani[2]
[1]*Research Scholar,* [2]*Professor, Department of Computer Science, Sri Padmavathi Mahila Visvavidyalayam, Tirupati, A.P, India.*

Abstract: ***The combination of the two quick creating logical exploration regions Semantic Web and Web Mining is called Semantic Web Mining. The immense increment in the measure of Semantic Web information turned into an ideal objective for some specialists to apply Data Mining methods on it. Semantic Web Data is an extra for the World Wide Web, and the primary goal of this is to make the web information machine-comprehensible. Resource Description Framework (RDF) is one of the advancements used to encode and speak to the semantics information as metadata. It's most likely to host the semantic web data on cloud due to its vast requirement, and also it can be managed well in terms of storage and evaluation. Map-reduce is a programming model that is well known for its scalability, flexibility, parallel processing, and cost-effective solution. Hadoop and Spark are the popular open-source tools for handling (Map-Reduce) and storing (HDFS) a huge amount of data. Semantic web data can be processed using the SPARQL, which is a primary query language for processing the RDF. In terms of performance, SPARQL has a significant drawback comparing to Map-reduce. For Querying the RDF data, we use Spark and Hadoop components (PIG and HIVE). Where Considering Directed Acyclic Graph (DAG) scheduler as a specific feature for In-memory processing in spark. In this paper, evaluate and analyse performance results using RDF data, which contains 5000 triples by executing the benchmark queries in PIG, HIVE, and SPARK. A scalable and faster framework can be obtained based on practical evaluation and analysis.***

Keywords: RDF, HDFS, SPARK, DAG, PIG, HIVE, Semantic Web Data

I. INTRODUCTION

The combination of the two quick creating logical exploration regions Semantic Web and Web Mining is called Semantic Web Mining. The immense increment in the measure of Semantic Web information turned into an ideal objective for some specialists to apply Data Mining methods on it. Semantic Web Data is an extra for the World Wide Web, and the primary goal of this is to make the web information machine-comprehensible. RDF is one of the advancements used to encode and speak to the semantics information as metadata. Here each structure of this record is considered as a triple. The subject represents as resources (Tree), and the Property acts as the relationship between the subject and object (Branches, leaves, trunk, root), Subjet can be considered as URI (Uniform Resource Identifier) or blank nodes. Objects are literals, whether it can approach URI or a value. Under the semantic web data, the Automation of the Information Retrieval, Internet of things, and Personal Assistants can be made possible with the help of this data. The gigantic developing in the amount of semantic information and information in various fields, as the condition in biomedical and clinical situations, might make an ideal and significant objective in the mining cycle. The Semantic Web Mining originated from consolidating two fascinating fields: the Semantic Web and the information mining. A potential design of this sort of mining proposed by is depicted in Figure 1.

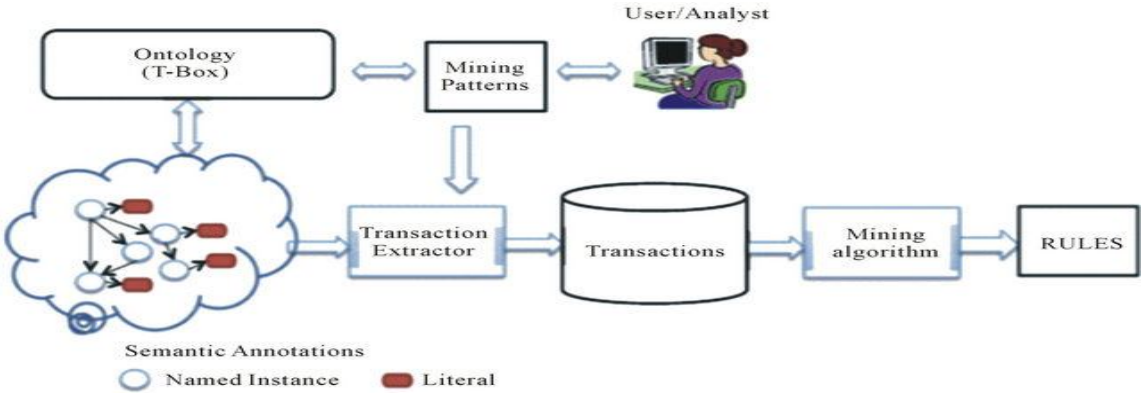


Fig 1 Semantic Web data example [21]

Connected Open Data gives information on the web in a machine coherent route with composed connections between related elements. Methods for getting to Linked Open Data incorporate slithering, looking, and questioning. Search in Linked Open Data takes into consideration something beyond catchphrase based, report situated information recovery. Just mind boggling inquiries across various information source can use the maximum capacity of Linked Open Data. In this sense Linked Open Data is more like dispersed/united information bases, yet with less participation between the information sources, which are kept up autonomously and may refresh their information without notice. Since Linked Open Data depends on principles like the RDF design and the SPARQL inquiry language, it is conceivable to actualize an organization framework without the requirement for explicit information coverings. Nonetheless, some structure issues of the current SPARQL standard cut off the proficiency and pertinence of inquiry execution systems. [21]

Semantic Web data can be of two types

1) *Linked Data:* is considered as a significant part of the semantic web data. Sematic is about creating the relation links between the datasets that can understand not only to humans but also to machines.
2) *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.

a) *Linked Open Data:* It's an efficient data which is collaborated with both linked data and open data. On text Graph database can handle the vast datasets coming from many sources and link them to open data. It provides richer queries and significant data-driven analytics.

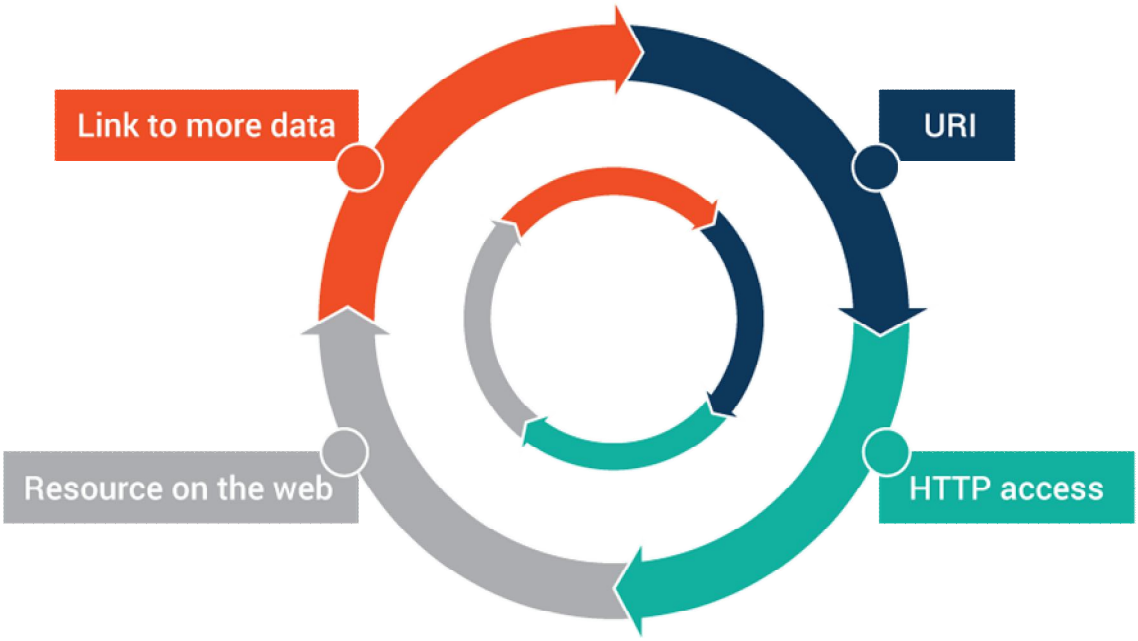


Fig 2 Linked Open data [22]

Linked open data givens a well-organized data integration, and browsing through complex data becomes more accessible and much more systematic.

Four standard rules for this Linked open data are:

- Uses URIs as names for things
- Uses HTTP URIs people understandable
- When someone looks up URIs, Providing information using standard (RDF)
- Include links to other URIs so that they can discover more things

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.

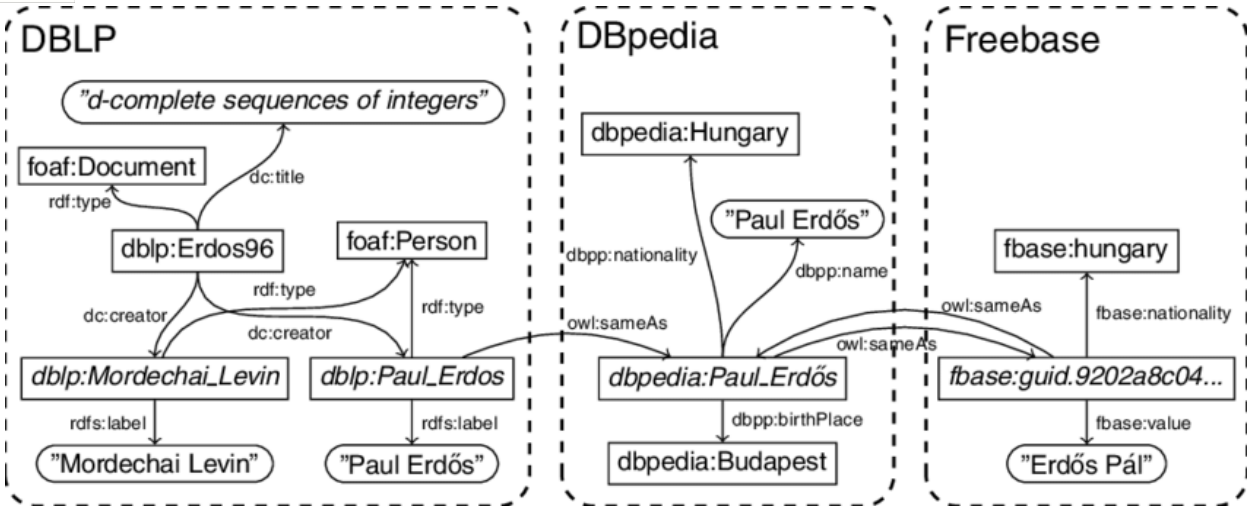


Fig 3 RDF data located at three different Linked Open Data sources, namely DBLP, DBpedia, and Freebase [23]

Fig. 3 delineates three Linked Open Data sources, for example DBLP, DBPedia, and Freebase, which contain RDF information about the mathematician Paul Erd˝os. The DBLP information depicts a distribution composed by Paul Erd˝os and a co-author. DBpedia and Freebase contain data about his ethnicity. The comparability of information substances is communicated through owl:sameAs relations.

b) *Semantic Web Technologies:* For the retrieval of the RDF data, the standardized query language came into existence, which is SPARQL. It is equivalent to the relational database like SQL used for the retrieval process for the RDF dataset.

c) *Cloud Computing Frameworks:*

- Amazon S3
- Amazon Ec2
- EMR(Elastic Map Reduce Service)

Utilizing these cloud administrations, we are going to assemble productive capacity for the semantic web information with an efficient inquiry system and high accessibility utilizing Amazon administrations. The most well-known question preparing utilizing the Map-lessen in Hadoop is PIG, Hive, and Spark SQL. With these three top handling instruments, we are going to actualize the semantic critical web information recovery measure. These instruments can conquer the perplexing issues looked during the time spent this standard question preparing (SPARQL). Till now, no framework has a very much arranged web system to scale countless triples because of the absence of dispersed structure and persevering stockpiling. Along these lines, Hadoop utilizes reasonable equipment in giving a conveyed structure extraordinary adaptation to non-critical failure and unwavering quality.

d) *Hadoop:* Hadoop is viewed as a mainstream answer for the preparing of a lot of information. It's is concocted the capacity as Hadoop disseminated File System (HDFS) and Processing as Map-Reduce (MR). It's a programming model that can cycle a lot of the datasets in an equal and conveyed calculation on a bunch, which can plan utilizing the sifting, arranging, and decrease strategy.

II. RELATED WORK

1) *SPARQL and RDF Framework:* SPARQL is considered as the standard inquiry handling for the RDF semantic web information, and already it's the main alternative for the recovery. And furthermore, the rehashed issues araised for the CRUD activity (Create, read, update, erase) workers. With regards to the security of the information and the getting to, it opens for the general population and doesn't contain any approval or verification.

2) *Massive Queries:* Creation workers set aside massive effort to get the necessary outcomes from the workers, which prompts the impact on business development, sway on the presentation of the creation framework. On the off chance that muti clients questioned to a solitary worker at once, the exhibition dropped out.

3) *Load:* A few times, it prompts the worker crash due to the monstrous dataset results and the limit, which will influence the CPU usage and expends high memory during the time spent getting the outcomes. Not ready to deal with the line cycle to execute the solicitations consistently.

4) *Scaling and Consistency:* Sparql neglected to build the group size to store the expanding measure of semantic web information. Same as the SQL social information base, it's effective in recovering the limited quantity of the information and gives better outcomes however not for Big information.
5) *Direct Access:* It gives direct admittance to the information bases, which prompts information instability. Indeed, even in SQL doesn't offer admittance to the information from the client. With the thought of the above restriction, pick the right fit for your dataset, for better outcomes and quality data recovery for examination.

K. Anusha et.al [18] provides an introduction about Big Data Characteristics and Hadoop Distributed File System.

The author [11] stated learning from the application studies, we explore the design space for supporting data-intensive and compute-intensive applications on large data-center-scale computer systems. Traditional data processing and storage approaches are facing many challenges in meeting the continuously increasing computing demands of Big Data.

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.

Sam Madden [5] explains that existing tools do not lend themselves to sophisticated data analysis at the scale many users would like.

Khalid Adam Ismail Hammad et.al [4] provides an introduction about Big Data frameworks, platforms, Databases for Big Data, data storage and Big Data Management and storage. They also presented a Big Data analysis and Management including Big Data with Data Mining, Big Data over Cloud Computing and Hadoop Distributed File System and Map Reduce

III.EXPERIMENTAL SETUP

To measure the 5000 tuples of the dataset, which has been taken from the DBpedia open source linked (open) information. We propose the preparing of semantic web information utilizing cloud administrations for high accessibility and better execution results. Amazon web administration gives the EMR (Elastic Map Reduce) usefulness, which can parallelize the information into a disseminated system and alloted the quantity of employments to finish the assignment in a negligible measure of the time. Utilizing these administrations, we can scale the no of hubs to build the presentation when the information increments and prompts the drop-down of the memory usage, which will impact the information recovering cycle. Proposed takes a shot at the three open-source DBpedia informational indexes having the connected information identified with them each.

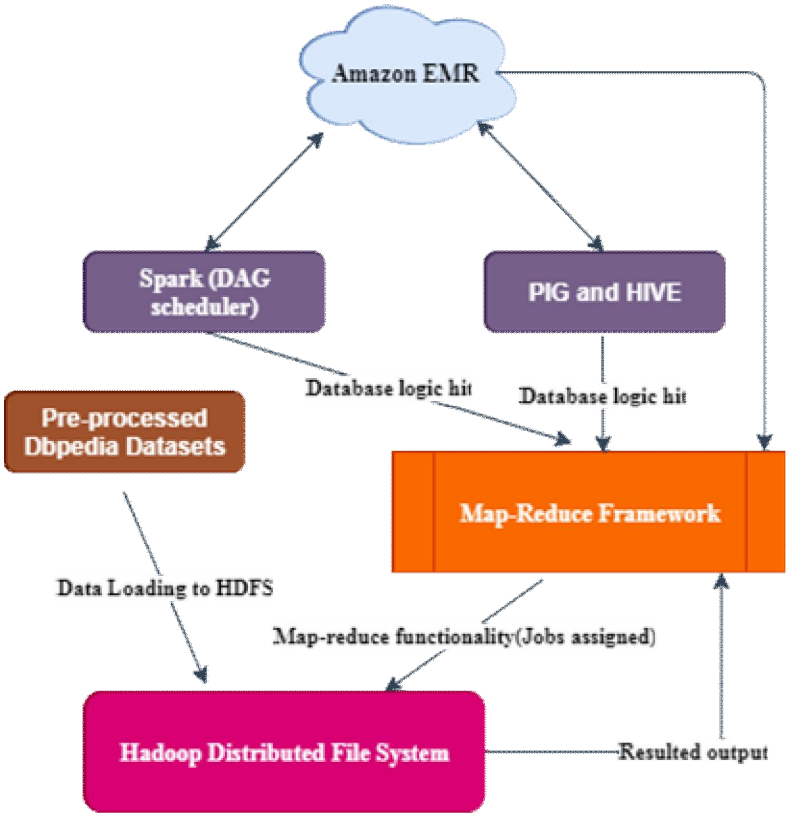


Fig 4 Proposed Architecture

1) *EMR setup:* Amazon services provide the complete Apache integrated tools in the name of service as EMR (Elastic Map-Reduce). Same as the standalone system here this service also provides the two high-level phases are storage (HDFS) and processing (Map-Reduce)

Below is the cluster setup CLI example after the creation of the cluster in amazon with EC2 service.

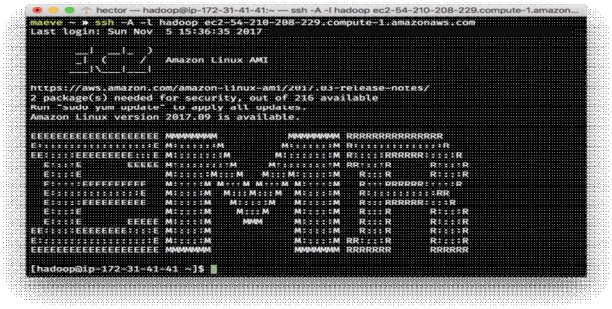


Fig 5 EMR Setup

So, the system has three stages proposed after the cluster setup on Amazon EC2.

a) Data Loading stage into HDFS
b) Processing Stage (Map-Reduce)
c) Result stage (Using Proposed Tools)

2) *Data Loading:* Loading of the semantic web data of RDF format file into the Hadoop storage is of different scripts in each tool. PIG uses the Pig Latin, and the Hive uses the HIVE-QL and spark uses the SPARK-SQL syntax for loading and reading the data into corresponding servers and frameworks.
3) *Hadoop data loading Commands*

a) Hadoop dfs – put geocoordinates-fixed.nt
b) Hadoop dfs – put homepages-fixed.nt
c) Hadoop dfs – put infoboxes-fixed.nt

4) *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 and analysis. In this phase, Map-reduce came into existence with the applying of no mappers and reducers to parallelize the process and get faster results. As the framework variation gives the different processing time for data retrieval.
5) *Results Stage:* With the resulted outputs from the different frameworks performed in the processing stage, we consider the CPU utilization, no of mappers, reducers, and jobs assigned. It can be compared with the three proposed frameworks and can conclude the best fit framework for the processing of the semantic web data.

IV. IMPLEMENTATION

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

1) *Sample SPARQL Query*

```
SELECT ?film1 ?actor1 ?film2 ?actor2
WHERE                    {
?film1 p:starring <http://dbpedia.org/resource/Kevin_Bacon> .
?film1 p:starring ?actor1 .
?film2 p:starring ?actor1 .
?film2 p:starring ?actor2 .
          }
```

Abvove query defines the identification of the subject as actor and the object represents film which is linked open data.

2) *Pig Case:* Pig has its query processing syntax as PIG-Latin as the above stages, the data loading, and the processing stage performs in this language. In contrast, the number of mappers and reducers is shown in the job-history server as a result.
3) *Hive Case:* It's 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. Same as the pig, the map-reduce method is applied in this, and the results are noted.
4) *Spark Case:* Here comes the unique feature in the spark with the DAG Scheduler, which is an in-memory processing unit; 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's integrated with the Spark for database hit.

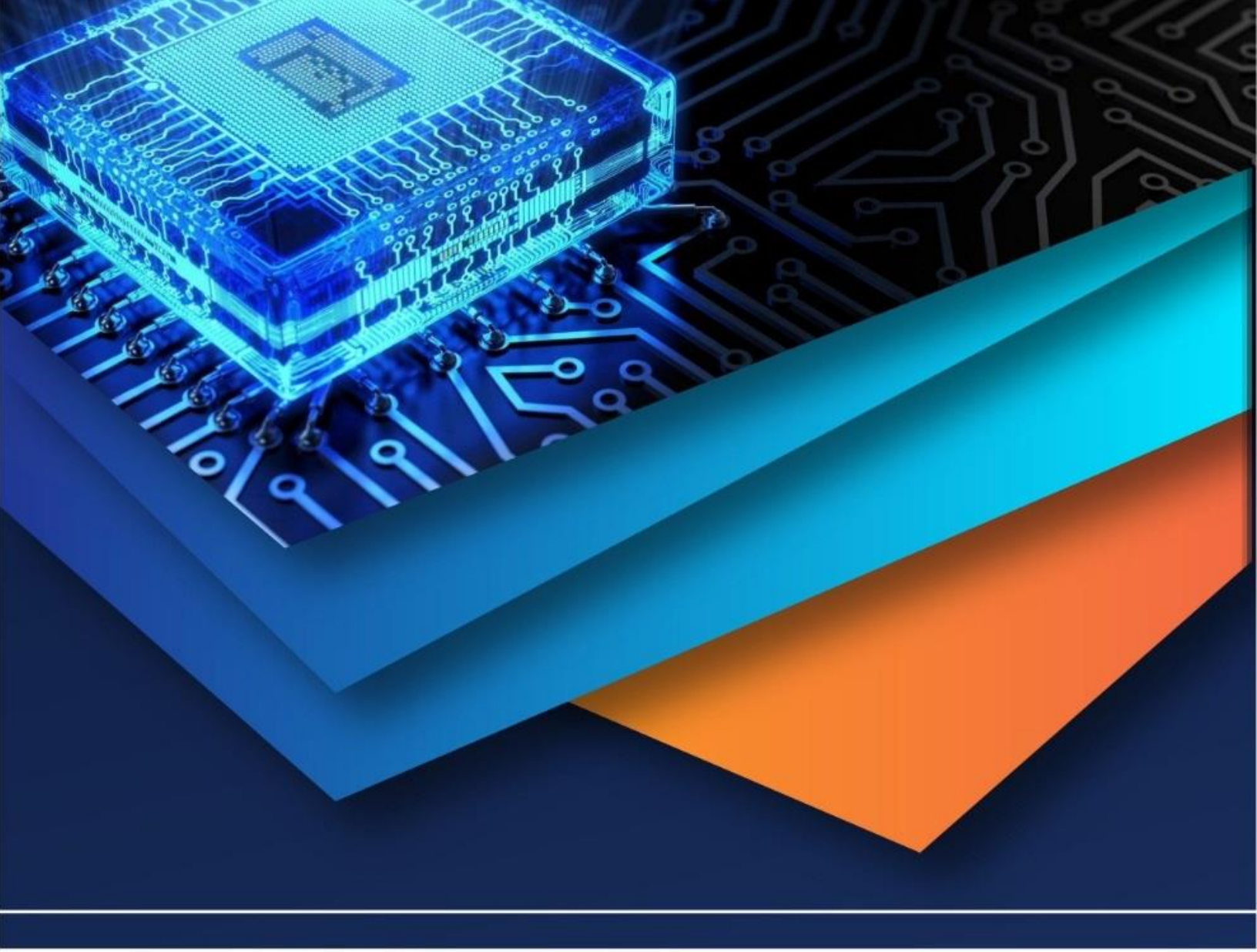
V. CONCLUSIONS AND FUTURE SCOPE

Efficient query processing is performed with the proposed big data techniques, in which standard dataset i.e., DBpedia is carried out for analysing 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, three big data cases i.e., Pig, Hive, and Spark are taken for implementing and analysis of performance of query processing. It is observed that Pig shown as outperformed the other in big data query processing. Pig shows better results, but it has a limitation in the increasing data that may result in the inactive job task. In the future work, it is planned to develop scalable and optimized distributed computing framework for reducing the require number of jobs and effective CPU utilization with the increased cluster size in the cloud. There are 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] O. Mustapaşa, A. Karahoca, D. Karahoca and H. Uzunboylu, "Hello World, Web Mining for E-Learning," Procedia Computer Science, Vol. 3, No. 2, 2011, pp. 1381- 1387. doi:10.1016/j.procs.2011.01.019 [Citation Time(s):6]
[3] 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).
[4] Khalid Adam Ismail Hammad, Mohammed Adam Ibrahim Fakharaldien, Jasni Mohamed Zain "Big Data Analysis and Storage", September 10, 2015.
[5] Sakr S., Wylot M., Mutharaju R., Le Phuoc D., Fundulaki I. Distributed RDF Query Processing. In: Linked Data. Springer, Cham (2018)
[6] Sam Madden "From Databases to Big Data" IEEE Computer Society (2012).
[7] 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).
[8] 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)
[9] Abadi, D.J., Marcus, A., Madden, S., Hollenbach, K.J.: Scalable Semantic Web Data Management Using Vertical Partitioning. In: VLDB, pp. 411–422 (2007)
[10] Bugiotti, F., Goasdoué, F., Kaoudi, Z., Manolescu, I.: RDF Data Management in the Amazon Cloud. In: DanaC Workshop (in Conjunction with EDBT) (2012)
[11] 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).
[12] Doulkeridis, C., Norvag, K.: A survey of large-scale analytical query processing in MapReduce. VLDB Journal (2013)
[13] Aditya B. Patel, Manashvi Birla and Ushma Nair, "Addressing Big Data Problem Using Hadoop and Map Reduce," in Proc. 2012 Nirma University International Conference On Engineering.
[14] 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)
[15] 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).
[16] Zhang, X., Chen, L., Tong, Y., Wang, M.: EAGRE: Towards Scalable I/O Efficient SPARQL Query Evaluation on the Cloud. In: ICDE (2013)
[17] T. Padiya, M. Bhise, "DWAHP: Workload Aware Hybrid Partitioning and Distribution of RDF Data", IDEAS-2017, pp. 235-241.
[18] 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).
[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] K. Anusha, Dr. K. Usha Rani, "Big Data Techniques for Efficient Storage and Processing of Weather Data" International Journal for Research in Applied Science & Engineering Technology (IJRASET), ISSN: 2321-9653; Volume 5 Issue VII, July 2017.
[21] K .Anusha, Dr. K .Usha Rani "Comparative Evaluation of Big Data Frameworks on Batch Processing" International Journal of Pure and Applied Mathematics (IJPAM) Scopus indexed journal ISSN: 1314-3395; Volume 119 No. 16, August 2018.
[22] K .Anusha, Dr. K. Usha Rani "Performance Evaluation of Spark SQL for Batch Processing" accepted for publication in Springer series "Advances in Intelligent Systems and Computing".
[23] Dr. K. Usha Rani, K. Anusha "Data Preprocessing on Cassandra Data through Spark SQL", accepted for publication in International Journal for Research in Engineering Application & Management (IJREAM) ISSN: 2454-9150; Volume - 05, Issue – 04, July 2019.

[24] Dr. K. Usha Rani, C Lakshmi "Effective Query Processing for Web-scale RDF Data using Hadoop Components", accepted for publication in TEST Engineering and Management, ISSN: 0193-4120; Volume - 83, Page No. 5764-5769, Publication Issue: May - June 2020.
[25] "Federated Data Management and Query Optimization for Linked Open Data" Olaf G¨orlitz and Steffen Staab Institute for Web Science and Technologies, University of Koblenz-Landau, Germany {goerlitz,staab}@uni-koblenz.de
[26] https://www.scirp.org/html/2-8701229_26994.htm
[27] https://www.ontotext.com/knowledgehub/fundam entals/linked-data-linked-open-data/
[28] https://www.researchgate.net/figure/Example-RDF-data-located-at-three-different-Linked-Open-Data-sources-namely-DBLP_fig1_225651676



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY