IJCSE International Journal of Computer Sciences and Engineering **Open Access**

Review Paper **Volume-3, Issue-9** **E-ISSN: 2347-2693**

Query Execution Performance Analysis of Big Data Using Hive and Pig of Hadoop

Anshu Choudhary [1*] and C.S. Satsangi [2]

[1,2] Department of Computer Science,*
Medi-Caps Institute of Technology and Management, Indore, India

www.ijcseonline.org

Received: Aug/22/2015 Revised: Sep/03/2015 Accepted: Sep/23/2015 Published: Sep/30/2015

***Abstract*—**The cloud platform requires an efficient computational infrastructure. On this platform a huge amount of data gets generated in a fraction of a second, therefore, traditional computing techniques are not enough. The Big Data provides an answer for such huge computing and also provides support to scale the storage according to the application's need. Big Data is a new generation storage infrastructure (hardware and software). In this paper the Big Data environment is investigated and the comparative study is performed among most frequently used data retrieval techniques. In order to perform the comparative study, Pig and Hive of Hadoop technology are selected. These techniques provide efficient data processing ability. In order to perform comparative study Hadoop storage is prepared first and then with the help of MapReduce framework the Pig and Hive are configured. Additionally, for evaluating the efficiency of query execution in terms of processing time, a list of similar queries is prepared and for each query the experiment was performed. The result evaluation is done for both the techniques. It is observed that query processing time of the Hive is less as compared to the Pig for the selected new_songs dataset, but both the data models are working to achieve the different goals thus both the technologies are adaptable for different kinds of computer configuration.

***Keywords*—** Big Data; Hive; Pig; Performance Analysis; Data Processing; Query Execution Time;

I. INTRODUCTION

The rapid development of technology and inventions in the IT arena makes the internet money and efficient management of resources becomes bonus on the existing wealth. Therefore, a number of organizations utilize the services of data warehouses for analytics. In these organizations, management makes the decisions for the organization's growth by analyzing related data. Thus, for making decisions accurately, we need to process data accurately. But this data is found in huge quantity that is counted in terms of peta bytes. Such amount of data cannot be handled by any single centralized server.

On the other hand internet has provided help to improve business and their growth, even smaller scale and internet tycoons like Google, both are managing their data using Big Data. For example, Facebook holds about 10 billion photos or 2-3TB image data per day [1]. The huge data size and distributed computing infrastructures create a new set of challenges for management and computation like data mining, machine learning and others. A large amount of time and cost is invested in managing and extracting the targeted data from this huge amount of data. Therefore, efficiency is key a requirement of the analytics.

The Rapid growth of technology increases the need of end users as well as increases the processing cost of the data in an organization. Resource utilization in organizations such as computing power and the network transfer abilities is also increasing. So, traditional computation technology becomes out-dated and new technologies and tools are required. In order to successfully resolve the issue of processing huge data, Big Data and Big Data analytics provide a trustworthy solution. Big Data [4] is a huge amount of data to be processed additionally adds up the technique and infrastructures (software and hardware) to find the required data according to the end user need. Hadoop [5] is an open source software platform, designed to store and process Big Data.

In this work, the Big Data and its environment have been evaluated and investigated. Two programming tools of Hadoop are utilized namely Pig and Hive [6]. Both the techniques are used for efficient processing of data and delivering the high quality of results. Thus, first of all it is required to find a way, how the data is taken as input to these programming tools and, how an end user can find the required data from the system. Both programming tools are utilized to find the best technique for evaluation of data according to the need of end clients. Firstly, the initial steps of installation are performed and then the data is stored over

HDFS file system. Further for comparative performance analysis a process model is provided to execute the user query and performance evaluation.

Section 2 explains the system architecture of the proposed work along with Pig and Hive architectures. Section 3 explains the performance analysis of the experiment. Section 4 contains the conclusion and then references are given.

II. SYSTEM ARCHITECTURE

The proposed system architecture for performing the comparative study between Pig and Hive is demonstrated in Fig. 1.

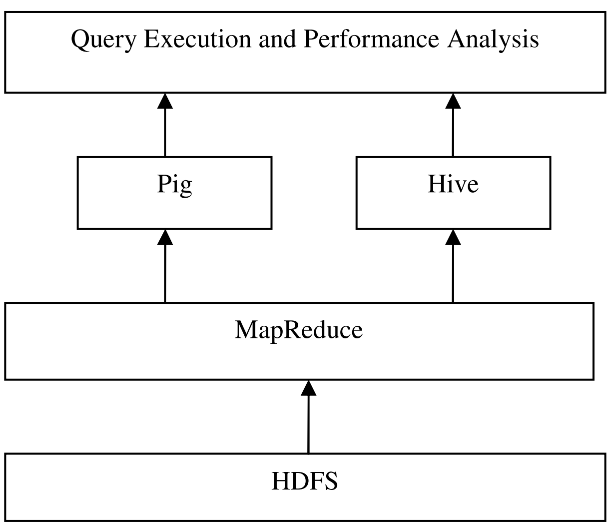

Fig. 1: Proposed System

In order to perform the comparative study of both the targeted technologies, a Big Data environment is required to develop first. The proposed comparative performance study platform is developed using the Hadoop and MapReduce technology. Hadoop is basically a storage technology that scales self for storing huge amount of data as required by the application. Additionally the MapReduce framework provides support to reduce and map the data for the data analytics.

Therefore, the input data is first of all hosted over the Hadoop repository and then using the MapReduce framework the data is processed in Pig and Hive infrastructures. The command line interface is used to make queries on the data over Pig and Hive with the similar dataset and the similar query one by one. After processing of data and execution of user queries over both the environments, the amount of time is estimated as performance analysis of the system.

The layered architecture of Pig is given in Fig. 2. In this diagram the initial HDFS file system is used to store the data and MapReduce is utilized for further processing. In order to scale the performance of MapReduce the Pig is attached as the supporting tool to the MapReduce.

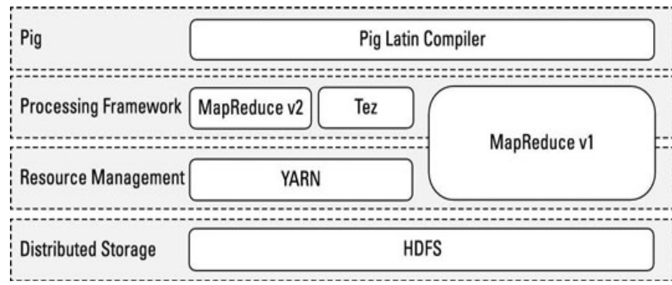

Fig. 2: Pig Architecture

Pig is an application that works on top of the MapReduce, Pig is written in Java and compiles Pig Latin scripts into MapReduce jobs. Think of Pig as a compiler that takes Pig Latin scripts and transforms them in Java.

- Pig is an application that runs on top of the MapReduce and abstracts Java MapReduce jobs away from developers.
- Pig Latin uses a lot fewer lines of code than the Java MapReduce script.
- The Pig Latin script is easier to read by someone without a Java background.
- MapReduce jobs can be written in Pig Latin.
- Java is a great and powerful language, but it has a higher learning curve than anything like Pig Latin. Therefore, using a higher-level language, like Pig Latin, enables many more developers/analysts to write MapReduce jobs.

Pig is an open-source programming tool, projects are developed under Apache Software Foundation. Pig is described as a data flow engine that is used to process large data sets. Companies like Yahoo use Pig to deal with their data. The language used by Pig is Pig Latin which handles one or more physical data flow jobs and then also carries out execution of these jobs. Pig currently uses the Hadoop open-source Map-Reduce implementation as its physical dataflow engine .Pig allows three modes of user interaction

1. **Interactive mode:** In this mode an interactive shell, called Grunt, accepts Pig commands and is triggered only when the user asks for output through the STORE command.

2. **Batch mode:** In this mode a series of Pig commands, typically ending with STORE are submitted by users as a prewritten script. The semantics are identical to interactive mode.

3. **Embedded mode:** Pig Latin commands can be written using Java program via method invocations which in turn permits dynamic construction of Pig Latin programs, as well as dynamic control flow.

The component diagram of Hive with their different functional units is defined as:

1. **User Interface:** Hive is data warehouse infrastructure software. The user interface is prepared to create interaction between user and HDFS. The user interfaces that Hive supports are a Web User Interface, Hive command line, and Hive HD Insight.

2. **Meta Store:** Hive has a Meta Store database server to store the schema or Metadata of tables, databases, columns in a table, their data types, and HDFS mapping.

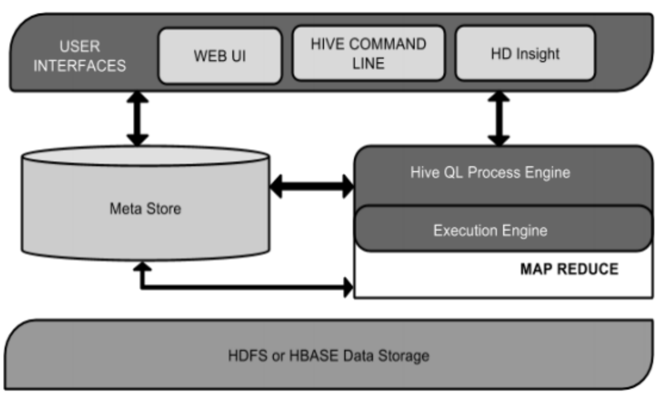

Fig. 3: Hive Architecture

3. **HiveQL Process Engine:** HiveQL is similar to SQL for querying of data. It is one of the replacements of traditional approach for MapReduce program. Instead of writing MapReduce program in Java, we can write a query for MapReduce job and process it.

4. **Execution Engine:** The conjunction part of HiveQL process Engine and MapReduce are Hive Execution Engine. Execution engine processes the query and generates results as same as MapReduce results.

5. **HDFS or HBASE:** Hadoop distributed file system or HBASE is the data storage techniques to store data into the file system.

The Fig. 4 shows the data flow of the Hive data processing system and its sub processes are described as:

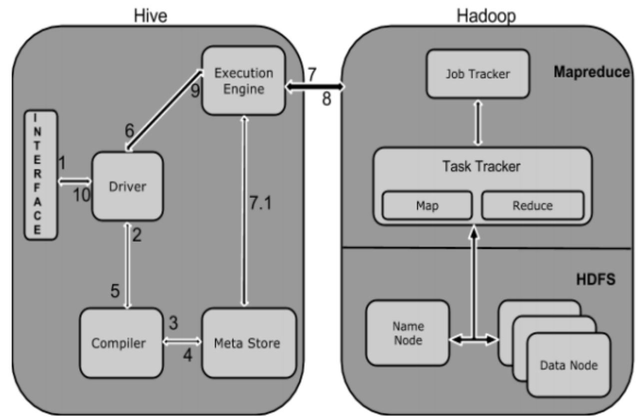

Fig. 4: Interaction between Hive and Hadoop

1. **Execute Query:** The Hive interface, such as Command Line or Web UI sends query to the Driver (any database driver such as JDBC, ODBC, etc.) to execute.

2. **Get Plan:** The driver takes the help of query compiler that parses the query to check the syntax and query plan or the requirement of query.

3. **Get Metadata:** The compiler sends metadata request to Metastore (any database).

4. **Send Metadata:** Metastore sends metadata as a response to the compiler.

5. **Send Plan:** The compiler checks the requirement and resends the plan to the driver. Up to here, the parsing and compiling of a query is complete.

6. **Execute Plan:** The driver sends the execute plan to the execution engine.

7. **Execute Job:** Internally, the process of execution job is a MapReduce job. The execution engine sends the job to JobTracker, which is in Name node and it assigns this job to TaskTracker, which

is in Data node. Here, the query executes MapReduce job.

8. **Metadata Operation:** Meanwhile, in execution, the execution engine can execute metadata operations with Metastore.

9. **Fetch Result:** The execution engine receives the results from Data nodes.

10. **Send Results:** The execution engine sends those resultant values to the driver.

11. **Send Results:** The driver sends the results to Hive Interfaces.

III. PERFORMANCE ANALYSIS

The authors developed the system and performed the experiment on two computers. Each machine having 4GB RAM and one machine has intel core i3 processor and another one has intel core i5 processor. In this multi node Hadoop setup, one machine acts like master and the other as slave. The dataset [19] used here is 1GB file which has 16115583 rows. The file is a TSV (Tab Separated Value) file.

After setting up the experimental environment, the queries that are listed in Table1 are fired on both PIG and Hive query interfaces and their performance in terms of query execution time is evaluated and reported in this section.

A. Experimentation with Hive

The amount of time consumed during input a user query for finding records from the Hive technique is termed here as the query execution time. In order to measure the query execution time, below listed queries are fired on the Hive interface and their performance is observed. After completing the observations first time for all the queries, the same queries are repeated for five times and their performance is visualized using Fig. 5 and 6.

Table 1: Query Statements

S. No.	Query Statements
1.	How many users we have?
2.	How many songs we have?
3.	Particular user listens, how many songs.
4.	How many times a particular song is played.
5.	Particular user listens, particular song how many times.
6.	Particular user listens which song the most?
7.	Particular user listens which song the least?
8.	Which is the most played song?
9.	Which is the least played song?
10.	Select songs information whose play_count >= 1000.

The noticed performance of Hive infrastructure is given in table 2. Additionally the performance is noticed in terms of seconds that is reported in Fig. 5. After that, the average performance of the query execution time is given in the Table 4 and visualized using Fig. 6. In both the diagrams namely Fig. 5 and 6, the performance of hive is visualized. The X axis contains the listed queries and the Y axis contains the amount of time required to produce results by the Hive.

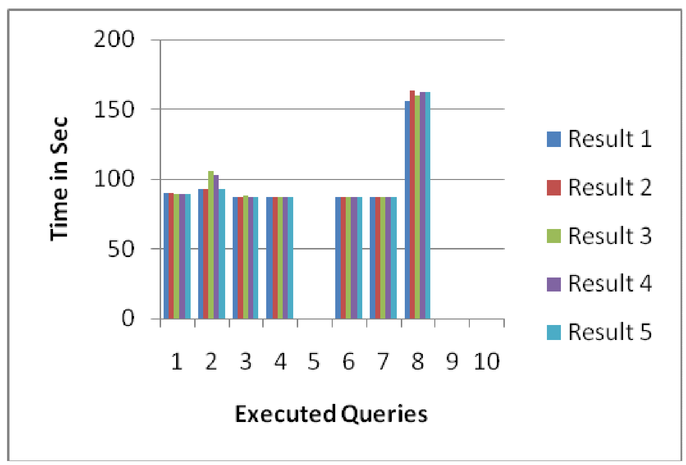

Fig. 5: Hive Query Execution Time

Table 2: Hive Query Execution Time

Query	Result 1(sec)	Result 2(sec)	Result 3(sec)	Result 4(sec)	Result 5(sec)
1.	89.686	89.535	89.375	89.375	89.435
2.	92.775	92.484	105.457	102.438	92.344
3.	87.405	87.612	87.866	87.538	87.423
4.	87.38	87.448	87.446	87.388	87.457
5.	0.08	0.086	0.078	0.062	0.072
6.	87.367	87.452	87.354	87.402	87.456
7.	87.368	87.437	87.43	87.427	87.351
8.	155.55	163.56	159.686	162.082	162.224
9.	0.112	0.194	0.095	0.064	0.065
10.	0.179	0.175	0.126	0.113	0.086

This section provided the information about the performance analysis of Hive infrastructure and in further section the Pig performance is reported.

B. Experimentation with Pig

The time required to execute the user request by the user input query is termed as query execution time of Pig.

Table 3: Pig Query Execution Time

Query	Result 1(sec)	Result 2(sec)	Result 3(sec)	Result 4(sec)	Result 5(sec)
1.	119	118	118	118	113
2.	133	134	133	135	133
3.	92	92	92	92	92
4.	92	92	92	91	92
5.	92	93	92	93	92
6.	230	225	226	225	221
7.	231	226	225	219	220
8.	225	249	244	244	250
9.	255	254	248	249	254
10.	92	92	98	98	98

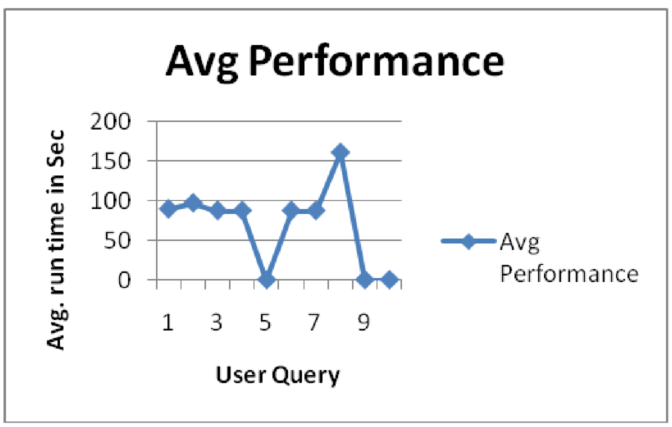

Fig. 6: Mean Performance of Hive Query Execution

Table 4: Mean Values of Hive and Pig

Query	Hive Mean	Pig Mean
1.	89.4812	117.2
2.	7.0996	133.6
3.	87.5688	92
4.	87.4238	91.8
5.	0.0756	92.4
6.	87.4062	225.4
7.	87.4026	224.5
8.	160.6214	242.4

9.ss	0.106	252
10.	0.1358	95.6

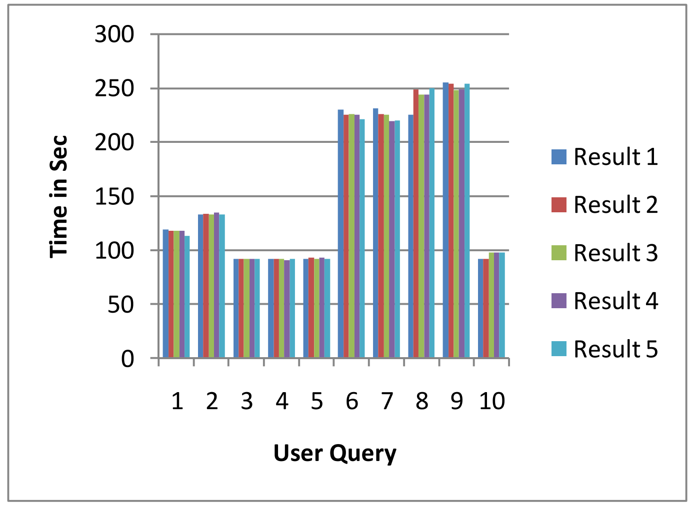

Fig. 7: Pig Query Execution Time

In order to evaluate the query execution time of Pig infrastructure the previously utilized query is resubmitted using the Pig interface and their performance is evaluated. In order to find the effective and accurate processing time the experiments are repeated with the same user queries five times and their observations are made.

The noticed performance of Pig infrastructure is given in Table 3. Additionally the reported performance in Table 3 is visualized using a bar chart as given in Fig. 7.

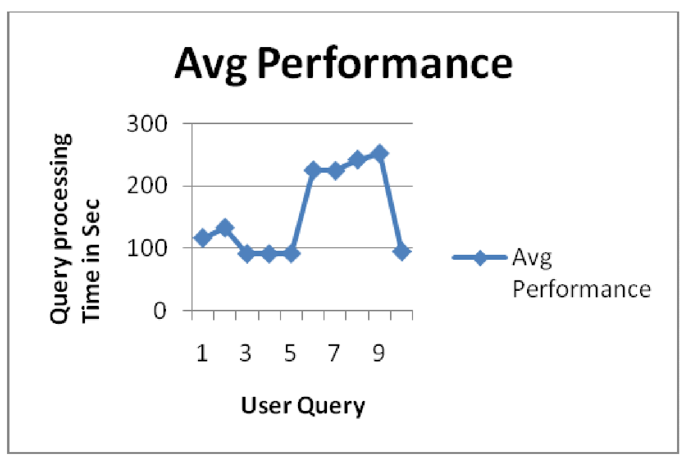

Fig. 8: Average Performance of Pig

After evaluation of performance using the different repetition of experiments a mean or average performance is also computed in Table 4 and its performance is shown using the Fig. 8. That is an average performance of the Pig infrastructure of the given query processing.

C. Comparative Performance

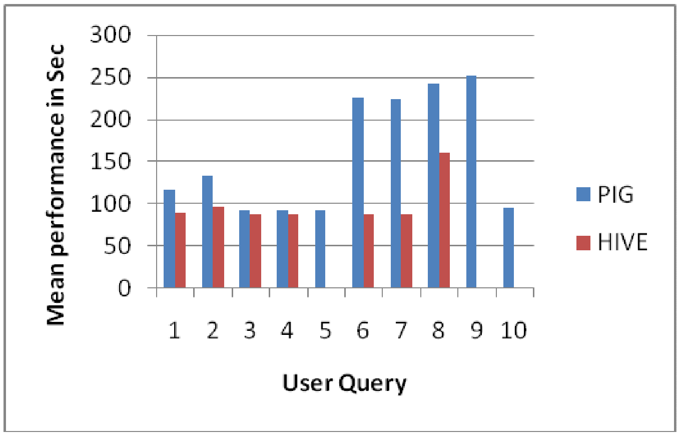

Fig. 9: Comparative Performance-1

The comparative performance in terms of query execution time for both the Big Data infrastructures is given using Fig. 9 and 10. In order to provide the performance of both the system, the X axis contains the user queries used for experimentation and the Y axis shows the amount of time consumed during similar query execution on different infrastructures. According to the obtained results the performance of the Hive is much more effective as compared to the Pig for the selected dataset.

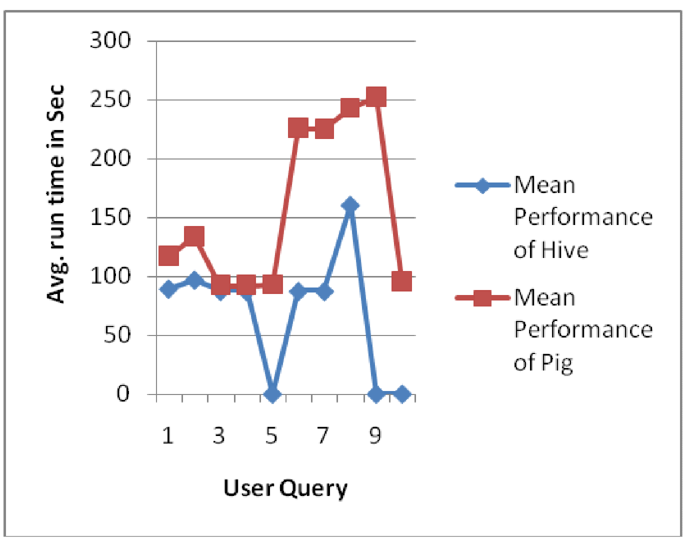

Fig. 10: Comparative Performance-2

IV. CONCLUSIONS

The key aim of the proposed study is to perform the comparative study among Hive and Pig data processing techniques. Therefore the query processing time is assumed as the key domain of study. In order to perform the experimentation a dataset is hosted on Hadoop and using the similar queries the performance is evaluated. According to the comparative study, the performance of the Hive is found more effective and consumes less time for data processing as compared to Pig for the selected dataset. On the basis of additional parameter comparison is given as:

Table 5: Additional Difference

Parameters	Pig	Hive
Language	Pig Latin	HiveQL(SQL-Like)
Language Support	Java	Java
Streaming	Yes	Yes
Server	No	Yes
Schema	Implicit	Explicit
Web Interface	No	Yes
JDBC/ODBC	No	Yes
DFS Direct Access	Explicit	Implicit
Partitions	No	Yes

Advantages of Pig

1. In Pig a multi query approach is being followed. Thus the number of times the data is scanned will be reduced.

2. For SQL, familiar user Pig Hadoop is very easy to learn, read and write.

3. Pig provides nested data types to the users such as Maps, Tuples and Bags. All these data types are not present in MapReduce.

4. The biggest advantage of Pig is that it uses very few lines code as compared to the Java MapReduce Program.
5. Pig is used by many companies like Yahoo, Twitter etc. For example, 90% of MapReduce is processed by Pig in Yahoo, 80% of MapReduce is also processed by Pig in Twitter. Like this various other companies such as LinkedIn, Sales force, and Nokia are also uses Pig.

Advantages of Hive

1. The Metastore or Metadata store is a big advantage of the Hive. This helps to make the lookup easier.

2. It has a lower learning curve as compared to Pig or MapReduce. The HiveQL query language is similar to SQL language, so for any SQL developer who wants to learn HiveQL will have a very low learning curve, almost negligible.

3. Hive can also be used with HBase for querying the data. Whereas Pig can be integrated with HBase. In case of Pig, we have to use a function named as HbaseStorage () for loading the data from HBase.

4. Hive supports external tables. That is a very great option. External table makes it possible to process data without actually storing in HDFS.

5. Apache Hive is used by many organizations. It has various user groups. For example, CNET, Facebook, Digg and so on uses Hive Programming tool.

In this work the comparative study among the Pig and Hive in Big Data environment is performed. During this study the different contributions and behavioral differences among Pig and Hive is observed. That concludes, during query processing both the data model supports the cloud infrastructures and both are having their own importance. In near future, it will be required to implement both the techniques with real world application, data processing and analytics.

REFERENCES

[1] Bharath Vissapragada, "Optimizing SQL Query Execution over Map-Reduce," M.S. thesis, Dept Comp. Sc., Center for Data Engineering International Institute of Information Technology, Hyderabad, India, September **2014**.

[2] Ammar Fuad, Alva Erwin, and Heru PurnomoIpung, "Processing Performance on Apache Pig, Apache Hive and MySQL Cluster," International Conference on Information, Communication Technology and System, IEEE, **2014**.

[3] F. Provost, T. Fawcett, "Data Science and its relationship to Big Data and data-driven decision making," University of Massachusetts Amherst, DOI: 10.1089/big.2013.1508, March **2013**.

[4] Changqing Ji, Yu Li, Wenming Qiu, Uchechukwu Awada, and Keqiu Li, "Big Data Processing in Cloud Computing Environments," International Symposium on Pervasive Systems, Algorithms and Networks, IEEE, Dalian, China, **2012**.

[5] Apache Hadoop, Available: http://wiki.apache.org/hadoop.

[6] Munesh Kataria, Ms.Pooja Mittal, "Big Data and Hadoop with Components like Flume, Pig, Hive and Jaql," IJCSMC, Vol. 3, July **2014,** pp. **759 – 765**.

[7] Ashish Thusoo, Joydeep Sen Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Ning Zhang, Suresh Antony, Hao Liu and Raghotham Murthy, "Hive – A Petabyte Scale Data Warehouse Using Hadoop," ICDE Conference, IEEE, **2010**.

[8] Ashish Thusoo, Joydeep Sen Sarma, Namit Jain, Prasad Chakka, Suresh Anthony, Hao Liu, Pete Wyckoff and Raghotham Murthy, "Hive – A Warehousing Solution Over a Map-Reduce Framework," VLDB, ACM, Lyon, France, August **2009**, pp. **24-28**.

[9] Anja Gruenheid, Edward Omiecinski, and Leo Mark, "Query Optimization Using Column Statistics in Hive," IDEAS, ACM, Lisbon, Portugal, September **2011**, pp. **21-23**.

[10] Meng-Ju Hsieh, Chao-Rui Chang, Li-Yung Ho, Jan-Jan Wu, and Pangfeng Liu, "SQLMR: A Scalable Database Management System for Cloud Computing," DBLP, January **2011**.

[11] Avrilia Floratou, Umar Farooq Minhas, and Fatma Ozcan, "SQL-on-Hadoop: Full Circle Back to Shared-Nothing Database Architectures," Proceedings of the VLDB Endowment, Vol. 7, No. 12, **2014**.

[12] Rakesh Kumar, Neha Gupta, Shilpi Charu, Somya Bansal, and Kusum Yadav, "Comparison of SQL with HiveQL," International Journal for Research in Technological Studies, Vol. 1, Issue 9, August **2014**.

[13] Sai Prasad Potharaju, Shanmuk Srinivas, Ravi Kumar Tirandasu, "Case Study of Hive Using Hadoop," DBLP, Volume-1, Issue-3, **2014**.

[14] Madhuri Srinivas Palle, Konisa Jyothsna and B. Anusha, "Analyzing Failures of a Semi-Structured Supercomputer Log File Efficiently by Using Pig on Hadoop," International Journal of Computer Science and Engineering, Volume-2, Issue-1, **2014**.

[15] Tak Lon Wu, Abhilash Koppula, and Judy Qiu, "Integrating Pig with Harp to Support Iterative Applications with Fast Cache and Customized Communication", ACM, **2014**.

[16] Gang Zhao, "A Query Processing Framework based on Hadoop," International Journal of Database Theory and Application, Vol.7, No.4, **2014**, pp. **261-272**.

[17] James M. Harris, and Dr. Cynthia, and Z.F. Clark, "Strengthening Methodological Architecture with Multiple Frames and Data Sources," Proceedings 59th ISI World Statistics Congress, Hong Kong, August **2013**.

[18] J. Christy Jackson, V. Vijaya kumar, Md. Abdul Quadir, and C. Bharathi, "Survey on Programming Models and Environments for Cluster, Cloud, and Grid Computing that defends Big Data," 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15), ELSEVIER, **2015**.

[19] Dataset that is used in this project, Available: https://github.com/jasondbaker/seis734.

[20] Radhiya A. Arsekar, Ankita V. Chikhale, Vaibhav T. Kamble and Vinayak N. Malavade, "Comparative Study of MapReduce and Pig in Big Data", International Journal of Current Engineering and Technology, Vol.5, No.2, April **2015**.

