Hadoop: An Effective Framework for Big Data Analytics

Dilbag Singh, PhD
(Associate Professor)
Dept. of Computer Science and Applications,
Chaudhary Devi Lal University, Sirsa

Chirag Goyal
(M.Tech Scholar)
Dept. of Computer Science and Applications,
Chaudhary Devi Lal University, Sirsa

ABSTRACT
In this modern era, analysis of enormous amount of data is becoming a big challenge to the decision makers. Big data is the datasets in size as well as high in variety, velocity and volume. So there is a need of the mean to handle and extract valuable insights from these datasets for better precision. It is very tedious rather impossible in some cases to handle enormous data using traditional databases and techniques their being the need for massive parallel processing and scalability which is not supported by the existing methods. Hadoop supports the scalability as it provides big storage and distribute big data sets over large no of servers operating in parallel. Traditional relational database systems don't scale to process the big data. Scaling of traditional RDBMS to such big data increases cost in many folds which is not affordable. Making efforts to reduce cost, the organizations have had to down-sample data and classify the data on assumptions by deleting raw data that may be useful only for a short term. Hadoop is designed as a scale out architecture and can affordably store company's data for use in future. In the present paper the Big Data Analytics has been carried out using experimental research method. Structured Queries are executed by setting up Hadoop Cluster and RDBMS environment using secondary datasets. The response time of RDBMS with Hadoop framework will be compared.

Keywords
Big Data, Hadoop cluster, HDFS, Map Reduce.

1. INTRODUCTION
In today's global era, extensive data is being generated with the velocity of light. Data from various sources like social platforms, web activities, life sciences, stock exchanges etc. contributes to this explosion. In present scenario understanding the meaning and importance of data, and use it to aid them in decision making is an alarming issue. Hence, the analysis has had a significant impact on research and technologies so as to draw patterns from the available data.

Hidden sights are directly proportional to data size. So as to recover the hidden patterns an effective mean is needed to learn about the trends based on the past and current data. So as to deal with huge amount of data Object Relational Databases concepts was used. Backups, recovery and searching made its use very complicated. In addition the RDBMS solutions lack scalability and parallel processing, the need of hour [2]. Hence, the need for a new framework to deal with the problem of huge amount of data has arisen. So as to overcome these problems Hadoop Framework has been developed.

Earlier the File Systems has been replaced by Database so as to overcome the problems with the file system. But in the changing scenario it is being felt that File System needs to be exercised again so as to cope up with increasing size of data in many folds [3]. In the days to come Big Data technology is going to be used in each and every area based on data requirements.

With existing RDBMS solutions, scalability exploitation is not easy tasks even using the multiple partitioning and parallelizing means. Moreover, structured data is also required to draw inference. Hadoop has the capability of processing unstructured data as well. Fault Tolerant, Reliability, Scalability and Cost-effectiveness are desirous features of the analytics and are supported by Hadoop.

Hadoop Framework provides distributed processing of PB of data sets across multiple machine(s) using Map Reduce programming models. Scaling from one machine to thousands of machines even as commodity hardware has been achieved by offering computation and storage. Framework is designed to encounter and manage failures itself so as to deliver a highly-available service on top of a cluster of computers. Hadoop provides two means in terms of Storage and Computation. It can be compared to a coin where one side is storage and other side is processing [7]. Storage is supported by highly distributed file system named Hadoop Distributed File System. HDFS is an immutable file system that runs on large clusters of commodity machines, designed for storing very large data (of sized petabytes). The HDFS file system is managed by two daemons operating in a master-worker pattern. The name node owns all the meta information of each and every block. It maintains the file system and the metadata for all the files and directories. Data nodes are the workhorses of the file system. Data Nodes physically store the actual data in form of blocks, and they report back to the Name node periodically with lists of blocks [1].

Computation is performed by Map Reduce Programming Framework. Map Reduce is a distributed parallel processing engine of Hadoop, which processes the data in parallel steps with two phases Map and Reduce [4].

2. OBJECTIVES OF THE STUDY
Present paper has been designed by taking into account the following objects:

- To study the concept of Big Data.
- To analyze Big Data with Hadoop Framework.
- To carry out the comparison between the RDBMS and HADOOP Systems.

3. RESEARCH DESIGN AND METHODOLOGY
In the present paper experimental research method has been used in which trails are executed to draw conclusions. For the study, secondary data are gathered from Career Education Council and hardware from Amazon Web Services is bought. Amazon Elastic Compute Cloud is a web service that provides resizable compute capacity in the cloud. On the rented commodity hardware Hadoop Cluster and RDBMS environment set up is done. Datasets are taken and queries are

executed over the huge data and performance of RDBMS and Hadoop is evaluated. In traditional methods data is stored at one location and to process the data it is moved to computational location it needs to archive a big portion of raw data, and the processing is carried out on small chunk of which lacks the efficient data analytics. This problem may be overcome using the HADOOP framework as the HADOOP Framework facilitates the processing on the whole data. In Hadoop, storage and computation is performed at the same location which will give insights into the whole data and hence no need of data archiving [6].

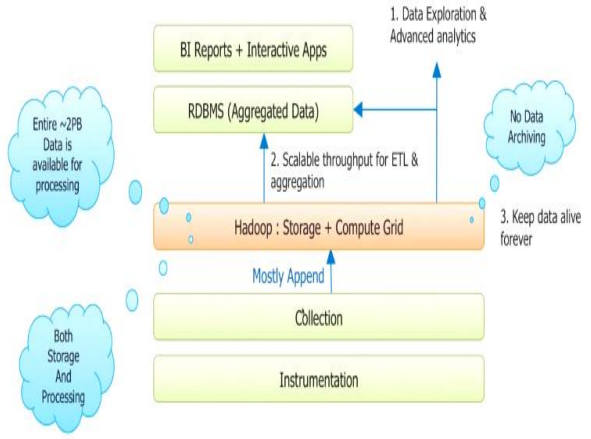

Figure 1: Hadoop: A Combined Storage Computational Layer

4. PERFORMANCE EVALUATION

In this section, performance has been evaluated in terms of execution time with RDBMS and Hadoop. It enables comparison between the traditional RDBMS and Hadoop distributed parallel processing architecture. In the experimentation, execution time evaluated based on growing input data size in respect for both RDBMS and Hadoop Framework [5].

Test Environment: It comprises the configuration of the systems on which the performance of Hadoop and RDBMS is evaluated.

Hadoop Environment: Hadoop cluster is set up on Amazon Web Services using Elastic Cloud Compute [8] Table shows hardware details of Master and Slave nodes.

Table 1: Hadoop Cluster Setup Details

Cores	4
Cache Size	4 GB
OS	Linux
Memory	32 GB
CPU	2.4 Intel Xeon E52676
Hadoop version	2.7
Hive Version	0.11.0.1

RDBMS Environment: Configuration of RDBMS is tabulated below.

Table 2: RDBMS Machine Details

OS	Linux 64 bit
CPU	Intel Xeon E5-2670 v2
RAM	64 GB
MySQL version	5.7

Dataset: Dataset used in this paper is taken from the Career Education Council (http://careereducationcouncil.ca/).

Field	Type	Null	Key	Default
state	varchar(4)	YES		NULL
email	varchar(100)	YES		NULL
leadscore	varchar(4)	YES		NULL
previouseducation	varchar(30)	YES		NULL
campus	varchar(12)	YES		NULL
leadage	varchar(10)	YES		NULL
applicationage	varchar(10)	YES		NULL
channel	varchar(30)	YES		NULL
militarystatus	varchar(4)	YES		NULL
degreeofinterest	varchar(40)	YES		NULL
leadconcentration	varchar(100)	YES		NULL
hrflag	int(1)	YES		NULL
hrhistory	int(1)	YES		NULL
totalsubmissions	varchar(4)	YES		NULL
currentstatus	varchar(10)	YES		NULL
custid	varchar(10)	NO		
date	date	YES		NULL

Figure 2: Lead Data Description

4.1 Hadoop Performance Metrics and Stats

This experiment analyzed query execution times with different volume of data.

Query 1: select count(*) from crm_data (table-name): This query gives us the number of records in the table mentioned with time taken by the query on Hadoop Cluster and MySQL.

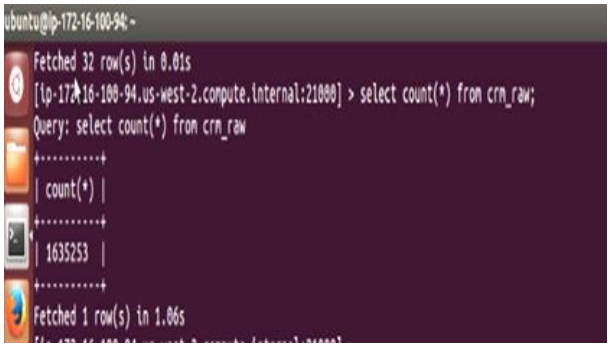

Figure 3: Snapshot for query to count the records on Hadoop

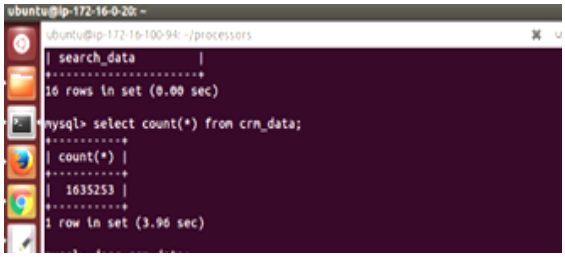

Figure 4: Snapshot for query to count the records on SQL

Query 2: select count(*) from alldomainsallsolutions (table-name): This query gives us the number of records in the table mentioned with time taken by the query on Hadoop Cluster.

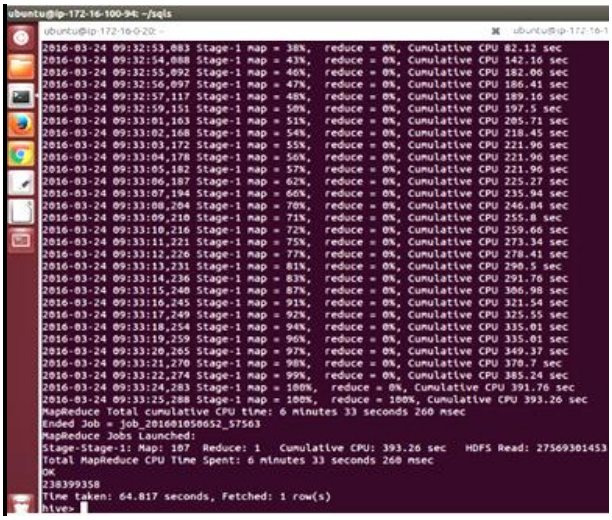

Figure 5: Snapshot of Map-Reduce functions Execution

Processing of data is done in two stages and Input-Output data at each stage is handled as Key-Value pairs. Map operates on one block of data, giving out transformed key/value pairs. Mapper output with the same key are sent to the same reducer. Input to reducer is always sorted by key. Number of mappers and reducers per node can be configured.

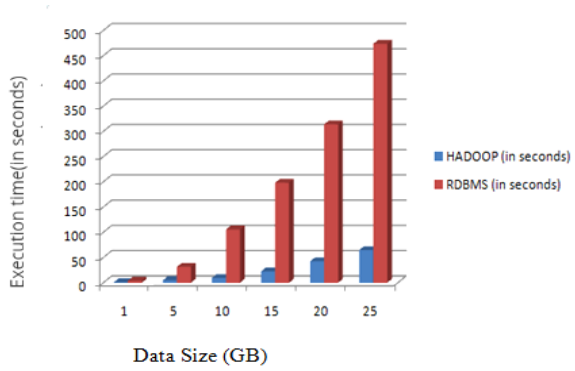

Figure 6: Representation of Execution time of Hadoop and RDBMS.

Table 3: Response Time for RDBMS viz-a-viz Hadoop Cluster

Data Size (in GB)	Hadoop	RDBMS
	select count(*) from table-name;	select count(*) from table-name;
1	1.06	3.96
5	5.1	31.8
10	9.6	105.7
15	22.5	198.3
20	42.6	314.2
25	64.8	473.5

4.2 RDBMS viz-a-viz Hadoop: Data Query Execution Performance Analysis

RDBMS: In this experiment, a query is run whose goal is to see the percentage of students of different backgrounds (on base of previous education) whose lead has been turned into the enrolment.

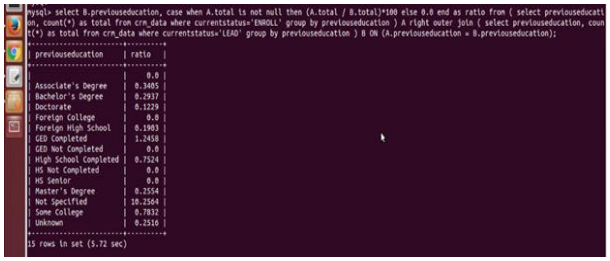

Figure 7: Snapshot of complex query (using joins, group by) on SQL Server.

Figure 8: Snapshot of complex query (using joins, group by) on Hadoop Cluster.

The above query has been run on Hadoop Framework and RDBMS. The comparison has been carried in terms of execution time. As shown in the Figure 8, it will take only (1.3 sec) to run the query which involves all major operations like join, group by, order by etc. on a large dataset, while RDBMS solution (Figure 7) takes 5.72 seconds for this problem.

Table 4: Response Time for RDBMS viz-a-viz Hadoop Cluster.

Data Size(in GB)	RDBMS (in minutes)	Hadoop (in minutes)
	Execution time	Execution time
1	0.095	0.0231
5	4.6	0.95
10	14.6	1.8
15	22.9	3.1
20	31.3	4.5
25	40.6	5.78

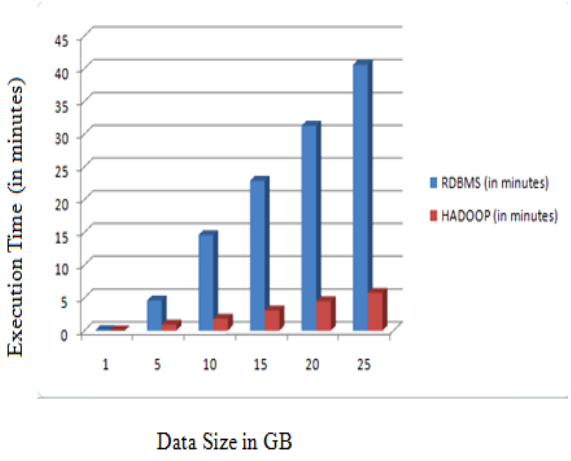

Figure 9: Representation of execution time by varying data size

5. CONCLUSION

Experiment has been conducted through the rented hardware from Amazon Web Services for both Hadoop as well as RDBMS. Some complex queries over large data sets are executed using both architectures and their performance is evaluated and compared. Response time of a query over 1 GB Data using Hadoop Architecture comes out to be 1.39 and for RDBMS is 5.72 seconds. For 10 GB data, Hadoop response time is 1.8 minutes and that of RDBMS is 14.6 minutes which explains its benefits over the large data. It is also found that for 25 GB data, Hadoop response time is 5.78 minutes and that of RDBMS is 40.6 minutes. From these results it is easy to perceive that the Hadoop framework is advantageous than that of RDBMS for analytics purpose. It can also be envisaged that with growth in size of the data response time of Hadoop framework is faster than RDBMS. Hadoop provides highly scalable storage platform which must be used for log processing, data warehousing and such other types of big data analytics. Availability of data and data processing tools make the faster processing of data.

Hence, it is recommended that the policy makers or the Data Scientists must use the Hadoop Framework for handling Big Data instead of RDBMS solutions as it provides better scalability and massive parallel processing which is must for effective inferences.

6. REFERENCES

[1] Tom White "Hadoop Definitive Guide" , Second Edition, O'Reilly Media, pp 1-9, October-2010.

[2] Shiqi Wu , Big Data Processing with Hadoop, pp. 13-16, June 2015.

[3] A Review Paper on Big Data and Hadoop, International Journal of Scientific and Research Publications, Volume 4, Issue 10, October 2014.

[4] Mark Kerzner and Sujee Maniyam, "Hadoop Illuminated", GitHub, pp. 28-30, 2014.

[5] Apache Hadoop, MapReduce Tutorial, 2013. https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html, accessed April 2014.

[6] Ketaki Subhash Raste, "Big Data Analytics-Hadoop Performance Analysis", pp. 18-22, 2014.

[7] Rui Xue, "SQL Engines for Big Data Analytics: SQL on Hadoop", pp 31-41, Nov 20,2015.

[8] Jefferey Shafer ,"A Storage Architecture for Data-Intesive Computing" , pp. 87-100, May 2010.