Comparative Analysis of Big Data Technologies

Rachit Singhal¹, Mehak Jain² and Shilpa Gupta³

¹Computer Science Department, Bharati Vidyapeeth’s College of Engineering, New Delhi, India.
²Computer Science Department, Bharati Vidyapeeth’s College of Engineering, New Delhi, India.
³Computer Science Department, Bharati Vidyapeeth’s College of Engineering, New Delhi, India.

Abstract

Recent technological advances and reduction in storage prices has led to accumulation of huge amount of data known as Big Data. This data, belonging to different applications and timelines, is difficult for organisations to process. In order to solve this difficulty, Doug Cutting and Mike Cafarella came up with a framework called Hadoop. Becoming open source in 2012, Hadoop went on to include Pig, Hive and many more products. Following this, Spark was developed by Matei Zaharia in 2009 which was open sourced in 2010. Meanwhile, many organisations came up with their own platforms to deal with Big Data. Hence, sprouting from Google’s MapReduce paper, these tools have grown into a wide array of technologies. This project focusses on comparing three main big data technologies which are used widely these days namely Pig, Hive and R. Similar problem statements are executed on all three platforms and performance is judged based upon the query execution time.

Keywords: Hadoop, HDFS, Big Data, Pig, Hive, R.

INTRODUCTION

Big Data

Big Data defines a large amount of data, usually greater than 1 TB of data. Also, Big Data refers to a collection of data that grows at an exponential rate. Hence, within a small span of time, this data becomes so big and complex that traditional data management tools cannot store or process this data any longer. Some examples of Big Data are –

1. Twitter produces somewhere around 90 million tweets every day.
2. Walmart handles close to 1 million customer transactions every single day. As a result, Walmart imports close to 2.5 petabytes of data into their databases every single day.
3. eBay uses two data warehouses, containing 7.4 petabytes and 40 petabytes of data and a Hadoop cluster to store its entire data. This data contains search, customer recommendations and merchandising.
4. NYSE (New York Stock Exchange) generates close to 1 terabyte of new trade data every day.
5. Facebook generates more than 500 terabytes of data every day and puts that data into its databases. This data contains photos, video uploads, messages that are exchanged, comments and many other things.
6. A jet engine is capable of generating more than 10 terabytes of data in a 30 minute of a flight. Since a thousand flights take off every day, the total data generated in a single day goes into petabytes.

Big Data encompasses three types of data. These are –

1. Structured Data – Data that has a well-defined structure falls under Structured Data. This data has records divided into rows and columns. As a result, it is easy to read and manage this data. Usually, data in databases falls under Structured Data. Well defined software are present to read structured data. Furthermore, it is easy to set constrains for this type of data.

   However, Structured Data accounts for only 20% of the total data available. Usually, this data has two types of sources, machines and humans. Hence, specialized programs can be written to read this data and write into databases. On the other hand, users can manually enter data into databases.

2. Semi-structured data – The main difference between Structured and Semi-structured is that Structured data cannot be stored in traditional databases. However, Semi-Structured still retains some organizational properties which makes it easier to process than Unstructured data.

3. Some common examples of Semi-structured data is data stores in CSV files and NoSQL documents.

4. Unstructured data – In structured data, data has a particular pre-defined format in which is gets stores. However, in Unstructured data, no such format is present. Hence, unstructured data can be used to store any kind of data to it. The downside of using unstructured data is that the data does not have any constraints to it. Hence storing this data and managing and manipulating this data proves to be a
difficult task. Unstructured data provides a way to store data like images and videos which cannot be processed using traditional databases.

**5 V’s of Big Data**

Big Data is always described as having at least three distinct dimensions: volume, velocity, and variety. With the passage of time, two more Vs were added to the list, namely variability and value. These are defined as the five V’s of Big Data. These are –

1. **Volume:** Big data first and foremost has to be “big,” and size in this case is measured as volume. From clinical data associated with lab tests and physician visits, to the administrative data surrounding transactions, this well of information is already expanding. When that data is coupled with greater use of precision medicine and stock exchange, there will be a big data explosion in any industry, be it medicine or business.

2. **Velocity:** Velocity in the context of big data refers to two related concepts: the rapidly increasing speed at which new data is being created by technological advances, and the corresponding need for that data to be digested and analyzed in near real-time. For example, as more and more medical devices are designed to monitor patients and collect data, there is great demand to be able to analyze that data and then to transmit it back to clinicians and others.

3. **Variety:** With increasing volume and velocity comes increasing variety. This third “V” describes just what you’d think: the huge diversity of data types that organizations see every day. In case of medical applications, each medical device might collect a different kind of data, which in turn might be interpreted differently by different physicians—or made available to a specialist but not a primary care provider. Standardizing and distributing all of that information so that everyone involved is on the same page. With increasing adoption of population health and big data analytics, we are seeing greater variety of data by combining traditional clinical and administrative data with unstructured notes, socioeconomic data, and even social media data.

4. **Variability:** The way care is provided to any given patient depends on all kinds of factors—and the way the care is delivered and more importantly the way the data is captured may vary from time to time or place to place. For example, what a clinician reads in the medical literature, where they trained, or the professional opinion of a colleague down the hall, or how a patient expresses herself during her initial exam all may play a role in what happens next. Such variability means data can only be meaningfully interpreted when care setting and delivery process is taken into context. For example, a diagnosis of “CP” may mean chest pain when entered by a cardiologist or primary care physician but may mean “cerebral palsy” when entered by a neurologist or pediatrician. Because true interoperability is still somewhat elusive in health care data, variability remains a constant challenge.

5. **Value:** Last but not least, big data must have value. That is, if you’re going to invest in the infrastructure required to collect and interpret data on a systems-wide scale, it’s important to ensure that the insights that are generated are based on accurate data and lead to measurable improvements at the end of the day.

Three technologies i.e. Pig, Hive and R are described below.

1. **PIG**

Apache Pig [21] is a higher level of abstraction over MapReduce. Founded in 2006 at Yahoo, Pig was moved into the Apache Software Foundation in 2007. Pig has a few distinctive features that make it an ideal choice for big data processing.

- Uses a simple language called Pig Latin, for writing code to process data.
- Since it is a higher level of abstraction over MapReduce, Pig code gets converted into MapReduce code internally.
- Users can use UDF (User Defined Functions) that can be written in other programming languages like Java, etc.
- Compared to MapReduce, Pig requires a smaller amount of code to accomplish the same task.
- User can choose between two execution modes, Local mode and Map Reduce mode.
- Pig uses multiple-query approach, thus reducing the size of the code.
- There are many built-in operations and nested data types present in Pig.
- The tasks in Pig are automatically optimized by the Pig Engine.

Perhaps the biggest advantage that Pig offers is its ease to handle different types of data. Pig can handle all three types of data, structured, semi structured and unstructured with the same ease and using the same tools.

Pig consists of two parts, Latin Pig and Pig Engine. Latin Pig [20] is the high-level language in which all the Pig scripts are written. Pig Engine is the execution engine that takes in Pig script and outputs MapReduce code. This code then runs on the Hadoop platform on the data stored in HDFS and produces the information.

The various parts of Pig Engine are –

- Grunt shell
- Pig Server
- Parser
- Optimizer
- Compiler
- Execution Engine
However, since the code needs to be converted into MapReduce code, the time taken by the system to produce the results is high in Pig. Still, owing to its simplicity and ease to learn, Pig has been used by a large number of users as their primary Big Data analysis tool.

2. HIVE
Hive is a data warehouse infrastructure tool to process structured data in Hadoop. It resides on top of Hadoop to summarize Big Data, and makes querying and analyzing easy. Initially Hive was developed by Facebook, later the Apache Software Foundation took it up and developed it further as an open source under the name Apache Hive.

The features of Hive are -

- It stores schema in a database and processed data into HDFS.
- It is designed for OLAP.
- It provides SQL type language for querying called HiveQL [19] or HQL.
- It is familiar, fast, scalable, and extensible.

The working of Hive is as follows:
1. Execute Query: The Hive interface such as Command Line or Web UI sends query to 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.
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.
8. Metadata Ops: Meanwhile in execution, the execution engine can execute metadata operations with Metastore.
9. Fetch and send Result: The execution engine receives the results from Data nodes. The execution engine sends those resultant values to the driver the driver sends the results to Hive Interfaces.

3. R
R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R.

R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering and many more) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source route to participation in that activity.

One of R’s strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed. Great care has been taken over the defaults for the minor design choices in graphics, but the user retains full control.

R is a programming language and software environment for statistical analysis, graphics representation and reporting. The following are the important features of R –

- R is a well-developed, simple and effective programming language which includes conditionals, loops, user defined recursive functions and input and output facilities.
- R has an effective data handling and storage facility.
- R provides a suite of operators for calculations on arrays, lists, vectors and matrices.
- R provides a large, coherent and integrated collection of tools for data analysis.
- R provides graphical facilities for data analysis and display either directly at the computer or printing at the papers.

RELATED WORK
Big Data is present in almost every domain today. Lots of research has been done in this vast field. In [1] authors have proposed that sensing technologies, cloud computing, internet of things and big data analytics systems as emerging technologies which has made it possible to achieve impressive progress in the field of computational field and storage which play a major role in efficiency and effectiveness of the healthcare systems. Cloud computing [2] is nowhere left behind in big data as well. System consisting of monitoring agents, cloud infrastructure, and operation centre by using Hadoop MapReduce and Spark to enhance the processing by splitting and processing data streams concurrently. But with every positive thing comes the challenges and issues related as well. There is a model [3] proposed to tackle challenges like data complexity, computational complexity, and system complexity and also presented suggestions for carrying out Big Data projects. The emergence of big data, considered both the opportunities and the ethical challenges for the market research [7] as proposed in this research.

There are various applications that are developed in this domain as well. Big data is suitable for every trend these days. Therefore, applications like [4] an electricity generation forecasting system which can predict the required power generation close to 99% of the actual usage by using Big Data Analytics was proposed. Also, a Location-Aware Analytics
System [5] using effective spatio-textual indexes and incremental algorithms that has been deployed and widely accepted. Another application is where authors proposed a model [17] for monitoring and analyzing Internet Traffic which is in the form of Big Data. They provided analysis and forecasts, including traffic management and network upgrade to enhance the quality that can be used to promote investments as well.

Data privacy is another big issue these days. For this author have proposed [6] six data management research challenges for Big Data and Cloud such as Data Privacy, Approximate Results, Data exploration to enable Deep Analytics, Enterprise data enrichment with web and social media. Another application is where authors have reported [9] a bibliometric study of critical BI&A publications, researchers, and research topics based on more than a decade of related academic and industry publications.

One of the bigger research was proposed in one the publication where authors described the HACE theorem [12] that characterizes the features of the Big Data Revolution and also proposes a big data processing model, from the data mining perspective.

Harnessing the full potential of any technology is necessary as well. In a paper [13] authors proposed a comprehensive overview of the applications of data processing platform designed to harness the potential of big data in the field of road transport policies in Europe. Also, to implement big data algorithms efficiently, authors proposed [14] FB+-tree that builds fast indexing structure using multi-level Key ranges, which is explained on the basis of B+ trees. Point searches and range searches are helped by early termination of searches for non-existent data.

For any technology to perform up to its full potential it should be as robust it can be. In one of the paper, authors proposed a model [15] based on robust data analytics, high performance computing, efficient data network management and cloud computing techniques that are critical towards optimized performance of Smart Grid’s so as to reduce the cost for customer.

Big data is surely a very important and big terms these days. Organizations have understood the importance of big data for managing their data efficiently and to obtain day to day analysis of the data generated. In one the paper this functionality was proposed and a deep understanding of this whole concept was given to the users.

In another research [18] authors proposed an overview of big data, significance of big data, how hadoop works and systems, which is based on analysis of published implementation architectures of big data use cases, different flavours of Hadoop.

There are many researches that have been proposed in this domain and also so many are being going on to make the people aware of any new and unseen facts of this technology.

PROPOSED MODEL

We have taken an Air Quality dataset from Kaggle.com containing approximately 7-8 million records. Here is a snippet of the dataset-

![Figure 1- Database Snippet 1](image1)

![Figure 2- Database Snippet 2](image2)

![Figure 3- Database Snippet 3](image3)

Based upon this dataset we have formulated two problem statements which will be executed on all three platforms.

Problem Statement 1- Sort all the parameter affecting the air quality according to mean values recorded over the year.

Problem Statement 2- Sort all the states which are affected due to parameters present in air according to standard deviation values calculated over the years.

For running these queries, we need to setup all components and then we have to execute them. All major steps are listed as below-

1. Setting up Hadoop Multi node Cluster

First and the foremost step in our project was to setup Hadoop multi node cluster. For this we used VirtualBox to host our
ubuntu 16.04 virtual machines. A master and a slave node setup was built up. Key steps involved in setting up the cluster are:

- Install VirtualBox and host Ubuntu 16.04 virtual machine
- Install Java on Ubuntu and setting up path variable in bash.rc file
- Create a user account
- In /etc/hosts file, slave IP address followed by name is mentioned
- Setting up SSH using RSA algorithm for secure communication without any prompt for password
- Install Hadoop in master node
- Configuring various Hadoop files like core-site.xml, hdfs-site.xml, mapred-site.xml and hadoop-env.sh
- Install Hadoop on slave node
- Configuring Master and slave node
- Format namenode on Master node and start the Hadoop services

Then we have to load data into Hadoop file system. Command used to put data from local directory into Hadoop FS is:

```
$ hadoop fs -put /path/in/linux /hdfs/path
```

### 2. Setting up and Using Pig

Now that we have multi node Hadoop cluster running, we can install pig and use Pig Latin to query our dataset. Steps involved in setup:

- Download and unzip Pig tar file from the official Apache Pig Website
- Edit the bash.rc file and setup the environment variables
- Compile the bash.rc file
- Check for pig -version

Now that the pig is successfully setup, we enter into Grunt shell where we can run pig commands to meet our requirements.

Following are the steps that we have taken to execute queries:

**For generating the results of first query we have to execute following commands in grunt shell of Pig.**

- `abcde = LOAD '/pigInput' using PigStorage(',') AS (pname:chararray,mean:chararray);`
• dump abcde;
• mean_order= order abcde by mean desc;
• dump mean_order;
• STORE mean_order into '/pigorderresult';

For generating the results of second query we have to execute following commands in grunt shell of Pig.

• abc = LOAD '/pigInput2' using PigStorage(',') AS (sdev:chararray,state:chararray);
• dump abc;
• sdev_order= order abc by sdev desc;
• dump sdev_order;
• STORE mean_order into '/pigsdevorderresult';

Output generated from both the queries was stored on HDFS in files pigorderresult and pigsdevorderresult.

Below is the real-time photo after the query was executed successfully.

Figure 7 - Query Executed on Pig

3. Setting up and using Hive

Now since we had multi node Hadoop cluster running and also executed Pig queries, our next aim was to install Hive and use HiveQL to query our dataset. Steps involved in setup-

• Download and unzip Hive tar file from the official Apache Hive Website
• Edit the bash.rc file and setup the environment variables
• Compile the bash.rc file
• Check for hive -version
• Create Hive directories within HDFS. The directory ‘warehouse’ is the location to store the table or data related to hive
• Configure Hive files

• Initialize Derby database
• Launch Hive

Figure 8 - Hive Installed

Now that the hive is successfully setup, we enter into hive shell where we can run commands to meet our requirements.

Following are the steps that we have taken to execute queries-

For generating the results of first query we have to execute following commands in hive shell of Hive.

• CREATE TABLE ABCDE
  (  	PNAME STRING,
  	MEAN FLOAT
  )
  
  ROW FORMAT DELIMITED
  FIELDS TERMINATED BY ',' ;

  LOAD DATA LOCAL INPATH /home/hduser/Downloads/Book1.csv OVERWRITE INTO TABLE ABCDE;

  INSERT OVERWRITE DIRECTORY '/tmp/hive' SELECT * FROM ABCDE SORT BY MEAN DESC;

For generating the results of second query we have to execute following commands in hive shell of Hive.

• CREATE TABLE ABCDEF
  (  	SDEV FLOAT,
  	STATE STRING
  )
  
  ROW FORMAT DELIMITED
  FIELDS TERMINATED BY ',' ;

  LOAD DATA LOCAL INPATH /home/hduser/Downloads/Book1.csv OVERWRITE INTO TABLE ABCDEF;
• LOAD DATA LOCAL INPATH '/home/hduser/Downloads/Book2.csv' OVERWRITE INTO TABLE ABCDEF;
• INSERT OVERWRITE DIRECTORY '/USER' SELECT * FROM ABCDEF SORT BY SDEV DESC;

Output generated from both the queries was stored on HDFS in directories /tmp/hive and /USER.

Below are some of the real-time photo after the query was executed successfully.

Figure 9- Query 1 executed

Figure 10- Query 2 executed

4. Setting up and using R
Rstudio can be downloaded from its official website www.rstudio.com

Following are the steps that we have taken to execute queries-
• reading in csv
  b1<-read.csv(file.choose())
  b2<-read.csv(file.choose())
• setting column name
  colnames(b2) <- c("SD","city")
  colnames(b1) <- c("para_name","mean_val")
• arranging columns "mean_val" and "SD" in descending order
  r1 <- order(b1$mean_val,decreasing= TRUE)
  r2 <- order(b2$SD,decreasing= TRUE)
• arranging entire dataFrame acc to columns "mean_val" and "SD"
  newb1<-b1[r1,]
  newb2<-b2[r2,]
• reading out the csv files
  write.csv(newb1, file = "MyData.csv")
  write.csv(newb2, file = "MyData22.csv")

Therefore, we have executed both the queries on same dataset in all three technologies. In the next section we will be analysing the results generated.

The following results were obtained after executing queries-

1. Pig Results-

Figure 11- First Problem Statement Output

Figure 12- Second Problem Statement Output

2. Hive Results-
So, both the queries are executed successfully and results of both the queries are realized using different type of graphs. It can be easily understood that due to presence of big amount of data, X-axis parameters are overlapping.

4. Now below are the graphs of execution times of both the queries in all three technologies-
   - Query 1 execution time comparison-

   ![Comparison of First Problem Statement Execution Times](image17)

   ![Comparison of Second Problem Statement Execution Times](image18)

   Therefore, it can be easily observed that execution time of both the queries in R is minimum as compared to other two technologies.

**CONCLUSION AND SCOPE**

In this article we have proposed the analysis of comparing three major technologies of handling big data i.e. Pig, Hive and R. Journey started from firstly setting up all the technologies in our virtual machine setup. Secondly, we had to formulate the queries and execute them accordingly in all the three technologies listed above. After successful query
execution, results are obtained and to make useful analysis from them we had to use python for plotting the graphs using python library.

From the results obtained, it can be clearly observed that query execution time in R is minimum as compared to Pig and Hive. Also, among Pig and Hive execution time is minimum in Hive. But this does not mean that in every case Pig will be slower than hive or hive can take place of Pig. It can be stated that depending upon the user requirement both the technologies can be used as both are based on map reduce concept. R is surely overtaking the big data business of Big Data by providing fancy algorithms of analysing these large datasets and also efficiently.

This research can take many paths from here like more technologies can be used for comparison of efficiency of big data technologies. Thereby, giving more accurate and more enhanced results. Also, from here we can go on to analyse streaming datasets using Apache Kafka, Spark etc. as the world is moving towards streaming data and real-time analysis is the need of the hour.

REFERENCES


[7] Daniel Nunan, Maria Di Domenico, Market research & the ethics of big data


[12] Xindong Wu, Xingquan Zhu, Gong-Qing Wu and Wei Ding, Data Mining with Big Data (IEEE), Vol. 26, NO. 1, JANUARY 2014


