[FRTSSDS- June 2018]
DOI: 10.5281/zenodo.1288445

ISSN 2348 – 8034
Impact Factor- 5.070

GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

SURVEY PAPER ON BIG DATA PROCESSING AND TECHNOLOGIES

MakhanKumbhkar* & Yashwant Singh Chouhan
Christian Eminent College Indore, MP, India

ABSTRACT

Big data is data sets that are so voluminous and complex that traditional data-processing application software is inadequate to deal with them. Big data challenges include capturing data, data storage, data analysis, search, sharing, transfer, visualization, querying, updating, information privacy and data source. Hadoop is an open source distributed processing framework that manages data processing and storage for big data applications running in clustered systems. It is at the centre of a growing ecosystem of big data technologies that are primarily used to support advanced analytics initiatives, including predictive analytics, data mining and machine learning applications. Hadoop can handle various forms of structured and unstructured data, giving users more flexibility for collecting, processing and analyzing data than relational databases and data warehouses provide.
This paper introduced different technologies like Hadoop, Map Reduce, Hive, Hbase, Distributed Data, Relational Database, and NoSql.

Keywords- *Big data, Hadoop, Map Reduce, Hive, Hbase, Distributed Data, Relational Database, NoSql,Hadoop technologies.*

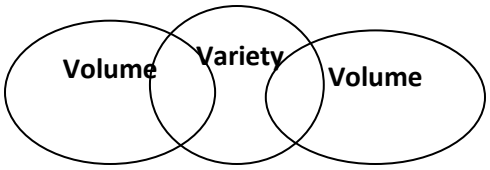
I. INTRODUCTION

Big Data
Big data is a blanket term for the non-traditional strategies and technologies needed to gather, organize, process, and gather insights from large datasets. An exact definition of "big data" is difficult to nail down because projects, vendors, practitioners, and business professionals use it quite differently. With that in mind, generally speaking, big data is:

Large datasets
The category of computing strategies and technologies that are used to handle large datasets
In this context, "large dataset" means a dataset too large to reasonably process or store with traditional tooling or on a single computer.

Vs of Big Data
3Vs (volume, variety and velocity) are three defining properties or dimensions of big data.

Volume: Volume refers to the amount of data.
Variety:variety refers to the number of types of data.
Velocity:velocity refers to the speed of data processing.



ISSN 2348 – 8034
Impact Factor- 5.070

II. DISCUSSION

Traditional Database Systems and Hadoop

What is a Relational Database?

Apache's Hadoop is somehow compared to relational databases. In most of those comparisons, Hadoop is presented as a non-relational database, as something that's totally different from classic database servers, such as IBM's DB2, Microsoft SQL Server, and Oracle11g. Comparing Hadoop this way makes no sense. Hadoop can be as relational as those classic database servers.

Whether a system is relational does not depend on how data is stored on disk, but fully depends on how the data is perceived by the applications. It depends on what language and/or API the applications use to insert, query, and manipulate the data.

In a nutshell, when the relational model was defined and introduced by TeddCodd, Chris Date and others, the rule was that if a system could present all the data as tables and columns, and if that data could be accessed through a language supporting relational operators such as join, select, and project, that system was a relational system.

NoSQL

NoSQL (commonly referred to as "Not Only SQL") represents a totally different framework of databases that give high-performance, agile processing of information at huge scale. In other words, it is a database infrastructure that has been very well-adapted to the heavy demands of big data. The efficiency of NoSQL can be achieved because unlike relational databases that are highly structured, NoSQL databases are unstructured in nature, trading off stringent consistency requirements for speed and agility. NoSQL centers on the concept of distributed databases, where unstructured data may be stored across multiple processing nodes, and often across multiple servers.

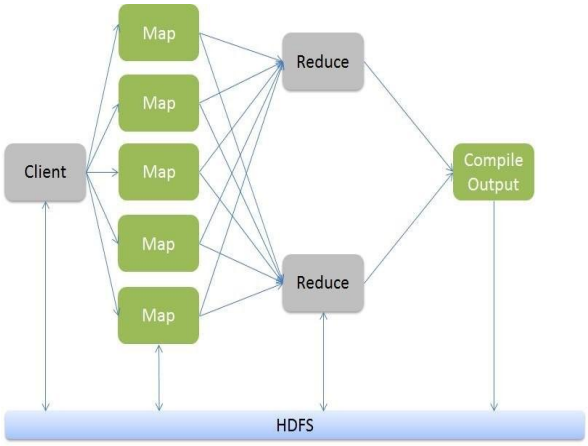
Hadoop

Hadoop is an open source distributed processing framework that manages data processing and storage for big data applications running in clustered systems. It is at the center of a growing ecosystem of big data technologies that are primarily used to support advanced analytics initiatives, including predictive analytics, data mining and machine learning applications. Hadoop can handle various forms of structured and unstructured data, giving users more flexibility for collecting, processing and analyzing data than relational databases and data warehouses provide.

MapReduce Framework

Hadoop MapReduce (Hadoop Map/Reduce) is a software framework for distributed processing of large data sets on computing clusters. It is a sub-project of the Apache Hadoop project. Apache Hadoop is an open-source framework that allows to store and process big data in a distributed environment across clusters of computers using simple programming models. MapReduce is the core component for data processing in Hadoop framework.in layman's term Mapreduce helps to split the input data set into a number of parts and run a program on all data parts parallel at once. The term MapReduce refers to two separate and distinct tasks. The first is the map operation, takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). The reduce operation combines those data tuples based on the key and accordingly modifies the value of the key.

ISSN 2348 – 8034
Impact Factor- 5.070



Fig(1). MapReduce Framework

Master-Slave Architecture
The Master (NameNode) manages the file system namespace operations like opening, closing, and renaming files and directories and determines the mapping of blocks to DataNodes along with regulating access to files by clients. Slaves (DataNodes) are responsible for serving read and write requests from the file system's clients along with perform block creation, deletion, and replication upon instruction from the Master (NameNode).

Programming Model of MapReduce
Map Reduce programming paradigm is based on the concept of key-value pairs. It also provides powerful paradigms for parallel data processing. For processing data in Map Reduce, we need to be able to map a given input, and expected output into the MapReduce paradigm, that is both Input and Output needs to be mapped into the format of multiple key-value pairs. A single key value pair is also referred to as a record.

For example, we have a text file 'input.txt' with 100 lines of text in it, and we want to find out the frequency of occurrence of each word in the file. Each line in the input.txt file is considered as a value and the offset of the line from the start of the file is considered as a key, here (offset, line) is an input key-value pair. For counting how many times a word occurred (frequency of word) in the input.txt, a single word is considered as an output key and a frequency of a word is considered as an output value. Our input key-value is (offset of a line, line) and output key-value is (word, frequency of word).

A Map-Reduce job is divided into four simple phases, 1. Map phase, 2. Combine phase, 3. Shuffle phase, and 4. Reduce phase. In our example of word count, Combine and Reduce phase perform same operation of aggregating word frequency. Now, let's look at how each phase is implemented using a sample code. Each phase takes a key-value as an input and emits one or more key-value pairs as an output. Generally in the Map phase we explode the input records; from one input key-value pair we create one or more output key-value pairs. In Reduce and Combine phases, we reduce input key-value pairs into less number of key-value pairs.

Consider the following pseudo code for map reduce to find the frequency of words in a collection of documents:

```
map(String key, String value)
// key: document name
// value: document contents
for each word w in value
EmitIntermediate(w, "1")

reduce(String key, Iterator values):
// key: word
// values: a list of counts
```

ISSN 2348 – 8034
Impact Factor- 5.070

```
for each v in values:
result += ParseInt(v);
Emit(AsString(result));
```

Hadoop Component

Hadoop Distributed File System:

It is the most important component of Hadoop Ecosystem. HDFS is the primary storage system of Hadoop. Hadoop distributed file system (HDFS) is a java based file system that provides scalable, fault tolerance, reliable and cost efficient data storage for Big data. HDFS is a distributed filesystem that runs on commodity hardware. HDFS is already configured with default configuration for many installations. Most of the time for large clusters configuration is needed. Hadoop interact directly with HDFS by shell-like commands.

HDFS Components

There are two major components of Hadoop HDFS- NameNode and DataNode. Let's now discuss these Hadoop HDFS Components-

NameNode

It is also known as Master node. NameNode does not store actual data or dataset. NameNode stores Metadata i.e. number of blocks, their location, on which Rack, which Datanode the data is stored and other details. It consists of files and directories.

Tasks of HDFS NameNode

Manage file system namespace.

Regulates client's access to files.

Executes file system execution such as naming, closing, opening files and directories.

DataNode

It is also known as Slave. HDFS Datanode is responsible for storing actual data in HDFS. Datanode performs read and write operation as per the request of the clients. Replica block of Datanode consists of 2 files on the file system. The first file is for data and second file is for recording the block's metadata. HDFS Metadata includes checksums for data. At startup, each Datanode connects to its corresponding Namenode and does handshaking. Verification of namespace ID and software version of DataNode take place by handshaking. At the time of mismatch found, DataNode goes down automatically.

MapReduce

Hadoop MapReduce is the core Hadoop ecosystem component which provides data processing. MapReduce is a software framework for easily writing applications that process the vast amount of structured and unstructured data stored in the Hadoop Distributed File system.MapReduce programs are parallel in nature, thus are very useful for performing large-scale data analysis using multiple machines in the cluster. Thus, it improves the speed and reliability of cluster this parallel processing.

YARN

Hadoop YARN (Yet Another Resource Negotiator) is a Hadoop ecosystem component that provides the resource management. Yarn is also one the most important component of Hadoop Ecosystem. YARN is called as the operating system of Hadoop as it is responsible for managing and monitoring workloads. It allows multiple data processing engines such as real-time streaming and batch processing to handle data stored on a single platform.

Hive

The Hadoop ecosystem component, Apache Hive, is an open source data warehouse system for querying and analyzing large datasets stored in Hadoop files. Hive do three main functions: data summarization, query, and analysis.

ISSN 2348 – 8034
Impact Factor- 5.070

Hive use language called HiveQL (HQL), which is similar to SQL. HiveQL automatically translates SQL-like queries into MapReduce jobs which will execute on Hadoop.

Pig

Apache Pig is a high-level language platform for analyzing and querying huge dataset that are stored in HDFS. Pig as a component of Hadoop Ecosystem uses **PigLatin** language. It is very similar to SQL. It loads the data, applies the required filters and dumps the data in the required format. For Programs execution, pig requires Java runtime environment.

HBase

Apache HBase is a Hadoop ecosystem component which is distributed database that was designed to store structured data in tables that could have billions of row and millions of columns. HBase is scalable, distributed, and Nosql database that is built on top of HDFS. HBase, provide real time access to read or write data in HDFS.

HCatalog

It is a table and storage management layer for Hadoop. HCatalog supports different components available in Hadoop ecosystem like MapReduce, Hive, and Pig to easily read and write data from the cluster. HCatalog is a key component of Hive that enables the user to store their data in any format and structure.
By default, HCatalog supports RCFile, CSV, JSON, sequenceFile and ORC file formats.

Avro

Acro is a part of Hadoop and is a most popular Data serialization system. **Avro** is an open source project that provides data serialization and data exchange services for Hadoop. These services can be used together or independently. Big data can exchange programs written in different languages using Avro.

Thrift

It is a software framework for scalable cross-language services development. Thrift is an interface definition language for RPC(Remote procedure call) communication. Hadoop does a lot of RPC calls so there is a possibility of using Hadoop Ecosystem componet Apache Thrift for performance or other reasons.

Apache Drill

The main purpose of the Hadoop Component is large-scale data processing including structured and semi-structured data. It is a low latency distributed query engine that is designed to scale to several thousands of nodes and query petabytes of data. The drill is the first distributed SQL query engine that has a schema-free model.

Apache Mahout

Mahout is open source framework for creating scalable **machine learning** algorithm and data mining library. Once data is stored in Hadoop HDFS, mahout provides the data science tools to automatically find meaningful patterns in those big data set.

Apache Sqoop

Sqoop imports data from external sources into related Hadoop components like HDFS, Hbase or Hive. It also exports data from Hadoop to other external sources. Sqoop works with relational databases such as teradata, Netezza, oracle, MySQL.

Apache Flume

Flume efficiently collects, aggregate and moves a large amount of data from its origin and sending it back to HDFS. It is fault tolerant and reliable mechanism. This Hadoop component allows the data flow from the source into Hadoop environment. It uses a simple extensible data model that allows for the online analytic application. Using Flume, we can get the data from multiple servers immediately into hadoop.

ISSN 2348 – 8034
Impact Factor- 5.070

Ambari

Ambari, another Hadoop component, is a management platform for provisioning, managing, monitoring and securing apache Hadoop cluster. Hadoop management gets simpler as Ambari provide consistent, secure platform for operational control.

Zookeeper

Apache Zookeeper is a centralized service and a Hadoop component for maintaining configuration information, naming, providing distributed synchronization, and providing group services. Zookeeper manages and coordinates a large cluster of machines.

Oozie

It is a workflow scheduler system for managing apache Hadoop jobs. Oozie combines multiple jobs sequentially into one logical unit of work. Oozie framework is fully integrated with apache Hadoop stack, YARN as an architecture center and supports Hadoop jobs for apache MapReduce, Pig, Hive, and Sqoop.

III. CONCLUSIONS AND FUTURE SCOPE

Hadoop is an open source framework, from the Apache foundation, capable of processing large amounts of heterogeneous data sets in a distributed fashion across clusters of commodity computers and hardware using a simplified programming model. Hadoop provides a reliable shared storage and analysis system.Hadoop is very cost effective as it can work with commodity hardware and does not require expensive high-end hardware .this paper provides many hadoop technologies like Hadoop Distributed File System, Map Reduce, YARN,Hive,Pig, HBase etc.

REFERENCES

[1] *Subramanian," A STUDY ON BIG DATA", International Journal of Applied Environmental Sciences (IJAES) ISSN 0973-6077 Vol. 10 No.1 (2015).*

[2] *Poonam S. Patil, "Survey Paper on Big Data Processing and Hadoop Components", International Journal of Science and Research (IJSR)*

[3] *Kyoo-sung Noh," Bigdata Platform Design and Implementation Model", Indian Journal of Science and Technology, Vol 8(18), DOI: 10.17485/ijst/2015/v8i18/75864, August 2015.*

[4] *RaghavToshniwal," Big Data Security Issues and Challenges", International Journal of Innovative Research in Advanced Engineering (IJIRAE) ISSN: 2349-2163Issue 2, Volume 2 (February 2015).*