International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 24 Issue 3 – FEBRUARY 2017 (SPECIAL ISSUE).

A Review: Bigdata and Hadoop

U.Udhayakumar[#1], Dr. G.Murugaboopathi [*2]and G. Sarala[*3]

[#1]*Research Scholar, Department of Computer Science, Bharathiyar University, Coimbatore, Tamilnadu*

[*2]*Associate Professor, Department of Computer Science and Enginering, Kalasalingam University, Krishnankoil, Tamilnadu*

[*3]*Head & Assistant Professor, Department of Electronic Science, Shanmuga Industries Arts and Science College, Thiruvannamalai, Tamilnadu*

***Abstract*— The bigdata is the large amount data, Bigdata describes technologies to capture, store, distribute and analyze Petabyte. Bigdata can be Structured, Semi-Structured and Unstructured format. Data is generated from various sources and can arrive in the system at various rates. The biggest challenge is handling these bigdata. Hadoop is the platform for structure the data and solve the problem of making it useful for analytics purpose. Hadoop enables the distributed processing of large data sets across clusters of commodity servers. The core concepts of the Hadoop is HDFS and Mapreduce. HDFS is the store mechanism and Mapreduce is the programming model for processing large sets with distributed algorithm on cluster.**

***Index Terms*— Bigdata, Hadoop, HDFS, Mapreduce. Petabyte.**

I. INTRODUCTION

Bigdata is data whose scale, diversity and complexness need new design, technique, algorithms, and analytics to manage it and extract price and hidden data from it. Hadoop is that the open source software package for process massive datasets. Hadoop Distributed File System (HDFS) for storage and MapReduce for processing.

MapReduce is a programming for process massive datasets. MapReduce works with Map and Raduce functions. The Map, Written by the user, takes and input combine and produces a collection of itermediate key/value pairs. The Reduce function accepts an intermediate key sets the values for the key and it merges values to make a presumably smaller set of values.

II. BIGDATA

Bigdata may be a term applied to data sets whose size is on the far side the flexibility of usually used software system tools to capture, manage process the data. Bigdata needs ascendible technologies to expeditiously method massive quantities of data with in tolerable elapsed times. Bigdata ascendible technologies embrace databases, the Hadoop framework, the Internet and storage systems.

Volume: There is a lot of data to be analyzed and/or the analysis is extremely intense; either way, a lot of hardware is needed.

Variety: The data is not organized into simple, regular patterns as in a table; rather text, images and highly varied structures or structures unknown in advance are typical.

Velocity: The data comes into the data management system rapidly and often requires quick analysis or decision making.

Veracity: A lot of data generated area unit vociferous, e.g., data from sensors. Data area unit typically incorrect. for example, several websites you access might not have the proper data, it's tough to be completely sure concerning the truthfulness of huge data.

Value: Data by itself is of no price unless it's processed to get data victimization that one might initiate actions. the massive volume of data makes process tough. As luck would have it, computing power and storage capability have conjointly inflated hugely. An enormous variety of cheap processors operating in parallel has created it possible to extract helpful data to notice patterns from massive data.

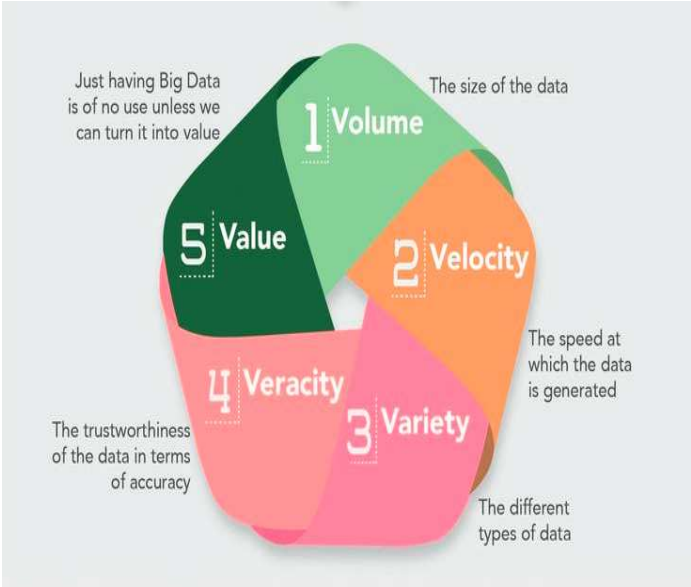

Fig 1. The 5V's of Bigdata

III. HADOOP

Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models. A Hadoop framework application works in an environment that provides distributed storage and computation across clusters

of computers. Hadoop is designed to scale up from single server to thousands of machines, each offering local computation and storage.

- Hadoop runs applications victimization the MapReduce algorithmic program, wherever the data is processed in parallel on totally different CPU nodes. In short, Hadoop framework is capable enough to develop applications capable of running on clusters of computers and that they may perform complete applied mathematics analysis for a large amounts of data**.**

A. *Hadoop Work Procedure*

1) *Stage 1*

A user/application will submit job to the Hadoop (a hadoop job client) for needed method by specifying the subsequent items:

1. The location of the input and output files within the distributed classification system.
2. The java classes within the kind of jar file containing the implementation of map and reduce functions.
3. The job configuration by setting completely different parameters specific to the task.

2) *Stage 2*

The Hadoop job shopper then submits the task (jar/executable etc) and configuration to the JobTracker that then assumes the responsibility of distributing the software/configuration to the slaves, scheduling tasks and observance them, providing standing and diagnostic info to the jobclient.

3) *Stage 3*

The TaskTrackers on completely different nodes execute the task as per MapReduce implementation and output of the reduce function is keep into the output files on the classification system.

B. *Advantages of Hadoop*

- Hadoop framework permits the user to quickly write and take a look at distributed systems. It is economical, and it automatic distributes the info and work across the machines and successively, utilizes the underlying correspondence of the CPU cores.
- Hadoop doesn't deem hardware to supply fault-tolerance and high handiness (FTHA), rather Hadoop library itself has been designed to discover and handle failures at the application layer.

Servers are additional or removed from the cluster dynamically and Hadoop continues to work while not interruption.

- Another huge advantage of Hadoop is that aside from being open source, it's compatible on all the platforms since it's Java based mostly.

IV. HADOOP ARCHITECTURE

Hadoop framework includes following four modules:

A. *Hadoop Common:*

These are Java libraries and utilities needed by alternative Hadoop modules. These libraries provides file system and OS level abstractions and contains the needed Java files and scripts required to begin Hadoop.

B. *Hadoop YARN:*

This is often a framework for job scheduling and cluster resource management.

C. *Hadoop Distributed File System:*

A distributed File system that gives high-throughput access to applicationdata.

D. *Hadoop MapReduce:*

This is often YARN based system for parallel processing of enormous data sets.

The term "Hadoop" typically refers not simply to the base modules mentioned on above however additionally to the gathering of further software packages that may be put in on top of or aboard Hadoop, like Apache Pig, Apache Hive, Apache HBase, Apache Spark etc.

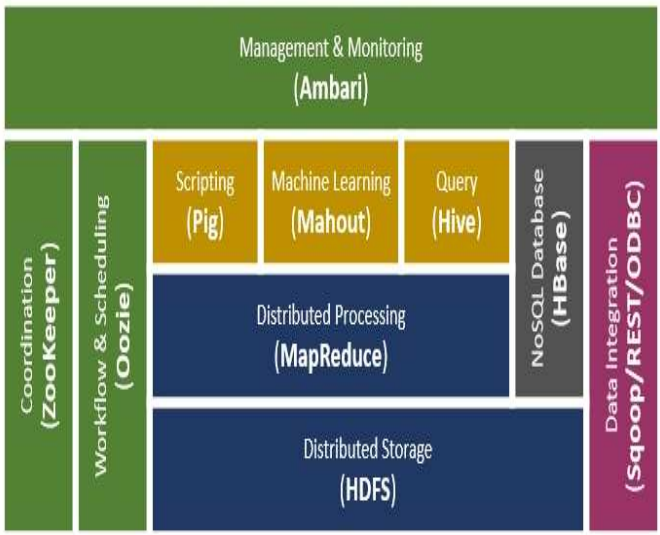

Fig. 2. Hadoop Ecosystem

V. HADOOP DISTRIBUTED FILE SYSTEM

The Hadoop Distributed file system (HDFS) is predicated on the Google file system (GFS) and provides a distributed file system that is designed to run on large clusters (thousands of computers) of tiny computer machines during a reliable, fault-tolerant manner.

HDFS uses a master/slave architecture wherever master consists of a SingleNameNode that manages the filing system metadata and one or a lot of SlaveDataNodes that store the particular data.

HDFS namespace is split into many blocks and people blocks square measure hold on during a set of DataNodes. The NameNode determines the mapping of blocks to the

DataNodes. The DataNodes takes care of read and write operation with the filing system. They additionally lookout of block creation, deletion and replication supported instruction given by NameNode.

HDFS provides a shell like several different filing system and an inventory of commands square measure accessible to act with the filing system. These shell commands are coated during a separate chapter along side applicable examples.

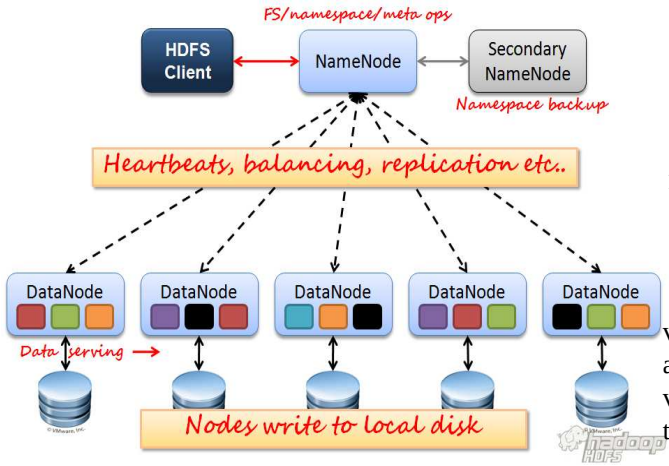

Fig 3. HDFS

VI. MAPREDUCE

Hadoop MapReduce could be a software framework for simply writing applications that method massive amounts of data in-parallel on massive clusters (thousands of nodes) of artifact hardware in an exceedingly reliable, fault-tolerant manner.

The term MapReduce really refers to the following two completely different tasks that Hadoop programs perform:

The Map Task: This is the first task, which takes input data and converts it into a set of data, where individual elements are broken down into tuples (key/value pairs).

The Reduce Task: This task takes the output from a map task as input and combines those data tuples into a smaller set of tuples. The reduce task is always performed after the map task.

Typically each the input and also the output area unit keep in a file system. The framework takes care of programming tasks, observation them and re executes the failing tasks.

The MapReduce framework consists of one master JobTracker and one slave TaskTracker per cluster node. The master is accountable for resource management, pursuit resource consumption and programming the roles part tasks on the slaves, observation them and re-executing the failing tasks. The slaves TaskTracker execute the tasks as directed by the master and supply task-status info to the master sporadically.

The JobTracker may be a single purpose of failure for the Hadoop MapReduce service which implies if JobTracker goes down, all running jobs area unit halted.

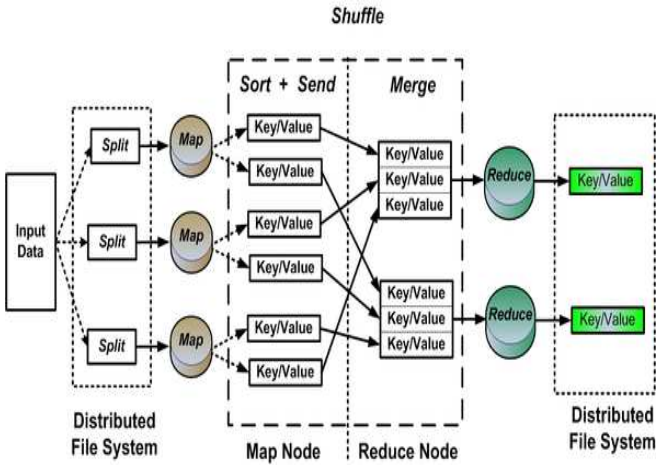

Fig 4. MapReduce

A. *Inputs and Outputs (Java Perspective)*

The MapReduce framework operates on <key, value> pairs, that is, the framework views the input to the job as a set of <key, value> pairs and produces a set of <key, value> pairs as the output of the job, conceivably of different types.

The key and the value classes should be in serialized manner by the framework and hence, need to implement the Writable interface. Additionally, the key classes have to implement the Writable-Comparable interface to facilitate sorting by the framework. Input and Output types of a MapReduce job: (Input) <k1, v1> → map → <k2, v2>→reduce → <k3, v3> (Output).

	Input	Output
Map	<k1, v1>	list (<k2, v2>)
Reduce	<k2, list(v2)>	list (<k3, v3>)

B. *Terminology*

- PayLoad - Applications implement the Map and the Reduce functions, and form the core of the job.
- Mapper - Mapper maps the input key/value pairs to a set of intermediate key/value pair.
- NamedNode - Node that manages the Hadoop Distributed File System (HDFS).
- DataNode - Node where data is presented in advance before any processing takes place.
- MasterNode - Node where JobTracker runs and which accepts job requests from clients.
- SlaveNode - Node where Map and Reduce program runs.
- JobTracker - Schedules jobs and tracks the assign jobs to Task tracker.
- Task Tracker - Tracks the task and reports status to JobTracker.
- Job - A program is an execution of a Mapper and Reducer across a dataset.
- Task - An execution of a Mapper or a Reducer on a

slice of data.

- Task Attempt - A particular instance of an attempt to execute a task on a SlaveNode.

VII. CONCLUSION

In Bigdata analytics, researchers divided generated data into various Bigdata application such as structured data analytics, text analytics, web analytics, multimedia analytics and mobile analytics. It also specifies the Hadoop environment, its architecture. This paper helps the researchers to understand the basic concepts of Bigdata, Hadoop, HDFS, MapReduce to work further.

REFERENCES

[1] Gali Halevi, Dr. Henk Moed: "The Evolution of Bigdata as a Research and Scientific Topic-Overview of the Literature".
[2] Jilia Lane "Bigdata Science Metrices and the black box of science policy" Research Trends Issue 30 September 2012.
[3] J. Dean and S. Ghemawat(2004), MapReduce: Simplified Data Processing on Large Clusters. p.10
[4] T. White. "MapReduce and the Hadoop distributed file system", Hadoop: The definition Guide, 1st edition. O'Reillly Media. Inc., Yahoo press, 2012.
[5] Shvachko.K, and Kuang, H., 2010, The Hadoop Distributed File System, Mass Storage Systems and Technologies(MSST), IEEE 26th Symposium.
[6] C.Lam, "Introducing Hadoop", in Hadoop in Action, MANNING, 2011.
[7] J.Venner and S.Cyrus, "The Mapreduce", in Pro Hadoop, vol 1, New York, APRESS, 2009.