ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

A Survey on Big Data and Hadoop

C.Willson Joseph, B.Pushpalatha

Assistant Professor, Department of Computer Science, Karpagam University, Coimbatore, India

Assistant Professor, Department of Computer Science, Karpagam University, Coimbatore, India

ABSTRACT: The term 'Big Data', refers to data sets whose size (volume), complexity (variability), and rate of growth (velocity) make them difficult to capture, manage, process or analyzed. To analyze this enormous amount of data Hadoop can be used. Hadoop is an open source software project that enables the distributed processing of large data sets across clusters of commodity servers. It is designed to scale up from a single server to thousands of machines, with a very high degree of fault tolerance. The technologies used by big data application to handle the massive data are Hadoop, Map Reduce, Apache Hive, No SQL and HPCC. These technologies handle massive amount of data in MB, PB, YB, ZB, KB and TB.

KEYWORDS: Big data, Hadoop, HDFS, Map Reduce

I. INTRODUCTION

What is Big Data?

There is no hard and fast rule about exactly what size a database needs to be in order for the data inside of it to be considered "big." Instead, what typically defines big data is the need for new techniques and tools in order to be able to process it [1].

Big Data in the beginning aimed the dimensions of data that could not be processed efficiently by traditional database methods and tools [11]. In order to use big data, you need programs which span multiple physical and/or virtual machines working together in concert in order to process all of the data in a reasonable span of time.

Getting programs on multiple machines to work together in an efficient way, so that each program knows which components of the data to process, and then being able to put the results from all of the machines together to make sense of a large pool of data takes special programming techniques[1]. Since it is typically much faster for programs to access data stored locally instead of over a network, the distribution of data across a cluster and how those machines are networked together are also important considerations which must be made when thinking about big data problems.

Characteristics:

The original three 'V' Dimension Characteristics of Big Data identified in 2001 are:

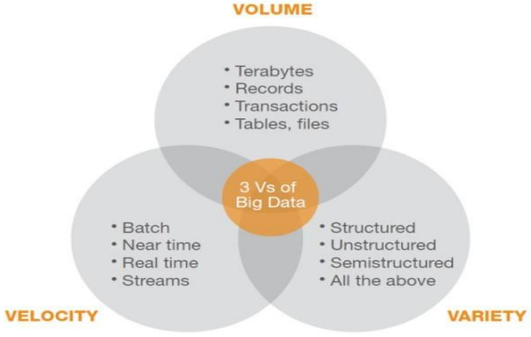

Fig 1: Dimension Characteristics of Big Data

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

Volume: Amount of data the size of the data set.

Volume Refers to the vast amounts of data generated every second. We are not talking Terabytes but Zettabytes or Brontobytes.but now we can process the with decreasing storage costs, better storage solutions like Hadoop and the algorithms to create meaning from all that data this is not a problem at all[2].

Velocity: Speed of data in and out or data in motion.

Velocity Refers to the speed at which new data is generated and the speed at which data moves around. Just think of social media messages going viral in seconds. Technology allows us now to analyze the data while it is being generated (sometimes referred to as in-memory analytics), without ever putting it into databases.

The Velocity is the speed at which the data is created, stored, analyzed and visualized. Every minute we upload 100 hours of video on You Tube. In addition, every minute over 200 million emails are sent, around 20 million photos are viewed and 30,000 uploaded on Flickr, almost 300,000 tweets are sent and almost 2.5 million queries on Google are performed[2].

Variety: Range of data types, domains and sources.

Variety Refers to the different types of data we can now use. In the past we only focused on structured data that neatly fitted into tables or relational databases, such as financial data. n fact, 80% of the world's data is unstructured (text, images, video, voice, etc.) [3]. With big data technology we can now analyze and bring together data of different types such as messages, social media conversations, photos, sensor data, video or voice recordings.

II. BIG DATA ANALYSIS TOOL

How is big data analyzed?

The best-known methods for turning raw data into useful information is by what is known as MapReduce. MapReduce is a method for taking a large data set and performing computations on it across multiple computers, in parallel. It serves as a model for how to program, and is often used to refer to the actual implementation of this model.

In essence, MapReduce consists of two parts. The Map function does sorting and filtering, taking data and placing it inside of categories so that it can be analyzed. The Reduce function provides a summary of this data by combining it all together[4]. While largely credited to research which took place at Google, MapReduce is now a generic term and refers to a general model used by many technologies.

Apache Hadoop:

The most influential and established tool for analyzing big data is known as **Apache Hadoop**. Apache Hadoop is an open-source software framework for storing and processing data in a large scale. Hadoop can run on commodity hardware, making it easy to use with an existing data center, or even to conduct analysis in the cloud.

Hadoop is broken into four main parts:

- The Hadoop Distributed File System **(HDFS)**, which is a distributed file system designed for very high aggregate bandwidth;
- **YARN**, a platform for managing Hadoop's resources and scheduling programs which will run on the Hadoop infrastructure;
- **MapReduce**, as described above, a model for doing big data processing;
- And a common **set of libraries** for other modules to use.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

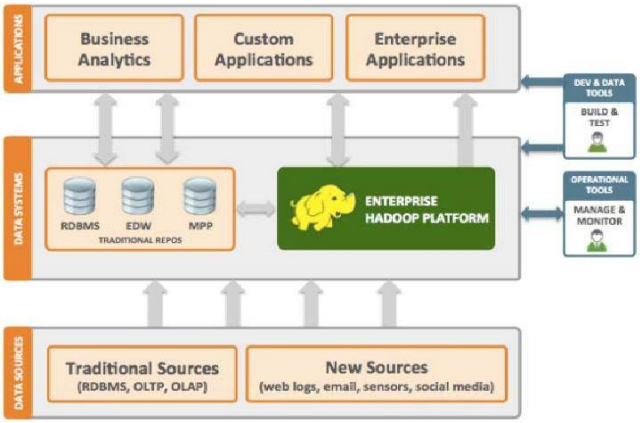

Fig 2: Hadoop system

HDFS

HDFS holds very large amount of data and provides easier access. To store such huge data, the files are stored across multiple machines [12]. These files are stored in redundant fashion to rescue the system from possible data losses in case of failure. HDFS also makes applications available to parallel processing.

Features of HDFS

- It is suitable for the distributed storage and processing.
- Hadoop provides a command interface to interact with HDFS.
- The built-in servers of namenode and datanode help users to easily check the status of cluster.
- Streaming access to file system data.
- HDFS provides file permissions and authentication.

HDFS Architecture

Given below is the architecture of a Hadoop File System.

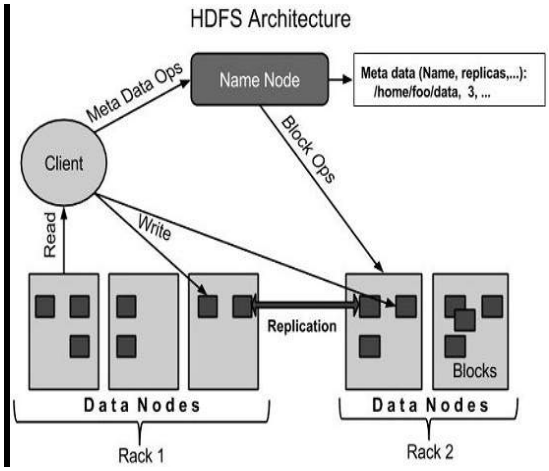

Fig 3:HDFS Architecture

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

HDFS follows the master-slave architecture and it has the following elements.

Namenode

The namenode is the commodity hardware that contains the GNU/Linux operating system and the namenode software[6]. It is a software that can be run on commodity hardware. The system having the namenode acts as the master server and it does the following tasks:

- Manages the file system namespace.
- Regulates client's access to files.
- It also executes file system operations such as renaming, closing, and opening files and directories.

Datanode

The datanode is a commodity hardware having the GNU/Linux operating system and datanode software. For every node (Commodity hardware/System) in a cluster, there will be a datanode. These nodes manage the data storage of their system.

Datanodes perform read-write operations on the file systems, as per client request.
They also perform operations such as block creation, deletion, and replication according to the instructions of the namenode[7].

Block

Generally the user data is stored in the files of HDFS. The file in a file system will be divided into one or more segments and/or stored in individual data nodes. These file segments are called as blocks. In other words, the minimum amount of data that HDFS can read or write is called a Block. The default block size is 64MB, but it can be increased as per the need to change in HDFS configuration.

Goals of HDFS

Fault detection and recovery : Since HDFS includes a large number of commodity hardware, failure of components is frequent. Therefore HDFS should have mechanisms for quick and automatic fault detection and recovery.

Huge datasets : HDFS should have hundreds of nodes per cluster to manage the applications having huge datasets.

Hardware at data : A requested task can be done efficiently, when the computation takes place near the data. Especially where huge datasets are involved, it reduces the network traffic and increases the throughput[8].

MapReduce

MapReduce is a processing technique and a program model for distributed computing based on java. The MapReduce algorithm contains two important tasks, namely Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job[9].

The major advantage of MapReduce is that it is easy to scale data processing over multiple computing nodes. Under the MapReduce model, the data processing primitives are called mappers and reducers. Decomposing a data processing application into mappers and reducers is sometimes nontrivial. But, once we write an application in the MapReduce form, scaling the application to run over hundreds, thousands, or even tens of thousands of machines in a cluster is merely a configuration change. This simple scalability is what has attracted many programmers to use the MapReduce model.

The Algorithm

Generally MapReduce paradigm is based on sending the computer to where the data resides. MapReduce program executes in three stages, namely map stage, shuffle stage, and reduce stage.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

Map stage : The map or mapper's job is to process the input data. Generally the input data is in the form of file or directory and is stored in the Hadoop file system (HDFS). The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data.

Reduce stage : This stage is the combination of the **Shuffle** stage and the **Reduce** stage. The Reducer's job is to process the data that comes from the mapper. After processing, it produces a new set of output, which will be stored in the HDFS.
During a MapReduce job, Hadoop sends the Map and Reduce tasks to the appropriate servers in the cluster[9].

The framework manages all the details of data-passing such as issuing tasks, verifying task completion, and copying data around the cluster between the nodes.

Most of the computing takes place on nodes with data on local disks that reduces the network traffic.
After completion of the given tasks, the cluster collects and reduces the data to form an appropriate result, and sends it back to the Hadoop server.[12]

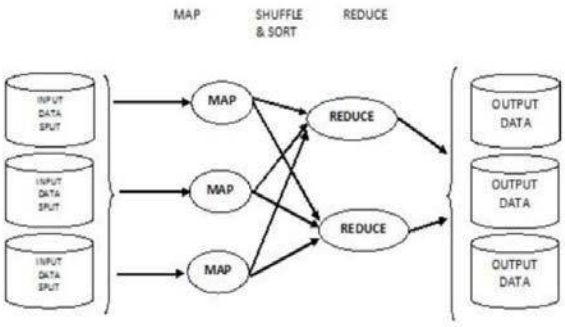

Fig 4:MapReduce operation

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[10].
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>)

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.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

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.

III. CONCLUSION

Hadoop MapReduce is a large scale, open source software framework dedicated to scalable, distributed, data-intensive computing. The framework breaks up large data into smaller parallelizable chunks and handles scheduling

- Maps each piece to an intermediate value
- Reduces intermediate values to a solution
- User-specified partition and combiner options
- Fault tolerant, reliable, and supports thousands of nodes and petabytes of data
- If you can rewrite algorithms into Maps and Reduces, and your problem can be broken up into small pieces solvable in parallel, then Hadoop's MapReduce is the way to go for a distributed problem solving approach to large datasets
- Tried and tested in production
- Many implementation options

We can present the design and evaluation of a data aware cache framework that requires minimum change to the original MapReduce programming model for provisioning incremental processing for Big data applications using the MapReduce model.

REFERENCES

[1]Dhole Poonam B, Gunjal Baisa L, "Survey Paper on Traditional Hadoop and Pipelined Map Reduce" International Journal of Computational Engineering Research||Vol, 03||Issue, 12||
[2]Nilam Kadale, U. A. Mande, "Survey of Task Scheduling Method for Mapreduce Framework in Hadoop" International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA 2nd National Conference on Innovative Paradigms in Engineering & Technology (NCIPET 2013) – www.ijais.org
[3]Suman Arora, Dr.Madhu Goel, "Survey Paper on Scheduling in Hadoop" International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 5, May 2014
[4]Wang, F. et al. Hadoop High Availability through Metadata Replication. ACM (2009).
[5]B.Thirumala Rao, Dr. L.S.S.Reddy, "Survey on Improved Scheduling in Hadoop MapReduce in Cloud Environments", International Journal of Computer Applications (0975 – 8887) Volume 34– No.9, November 2011
[6]Amogh Pramod Kulkarni, Mahesh Khandewal, "Survey on Hadoop and Introduction to YARN", International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 4, Issue 5, May 2014)
[7]Vishal S Patil, Pravin D. Soni, "HADOOP SKELETON & FAULT TOLERANCE IN HADOOP CLUSTERS", International Journal of Application or Innovation in Engineering & Management (IJAIEM)Volume 2, Issue 2, February 2013 ISSN 2319 - 4847
[8]Sanjay Rathe, "Big Data and Hadoop with components like Flume, Pig, Hive and Jaql" International Conference on Cloud, Big Data and Trust 2013, Nov 13-15, RGPV
[9]Yaxiong Zhao, Jie Wu and Cong Liu, "Dache: A Data Aware Caching for Big-Data Applications Using the MapReduce Framework",TSINGHUA SCIENCE AND TECHNOLOGYISSN 1007-0214 05/10 pp39-50 Volume 19, Number 1, February 2014
[10]Parmeshwari P. Sabnis, Chaitali A.Laulkar , "SURVEY OF MAPREDUCE OPTIMIZATION METHODS", ISSN (Print): 2319-2526, Volume - 3, Issue -1, 2014
[11]. M.Roopa, Dr.S.Manju Priya. "A Review of Big Data Analytics in Healthcare ",International Journal for Scientific Research & Development, Sp. Issue – Data Mining ,pp.6-9,2015.
[12]. Ms. Vibhavari Chavan, Prof. Rajesh. N. Phursule "Survey Paper On Big Data"
Vibhavari Chavan et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (6) , 2014, 7932-7939