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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

A Survey on Storage and Analysis of Big Data

Jolly Khurana, Dr. A.K Sharma

M.Tech Scholar, Dept. of CSE, BSAITM, Faridabd, India

Professor, Dept. of CSE, BSAITM, Faridabd, India

ABSTRACT: Big Data is a term used for huge amount of data that is difficult to store and manage by traditional Data Management System. For managing Big Data, a Technology is used called HADOOP. Hadoop is a Open Source Framework for storing and processing Big Data. Hadoop introduced two terms i.e. HDFS and Map Reduce. HDFS is a special file system for storing Big Data and Map Reduce is a programming model used to process large dataset. In this paper I have provided a brief description about HDFS and Map Reduce.

KEYWORDS: Big Data; Hadoop; HDFS; Map Reduce

I. INTRODUCTION

The amount of data that exceeds the processing capacity of conventional database systems is what we call Big Data. This data is too big and moving too fast, does not fit the structures of existing database architectures. Sources for such large amount of data are Social Networking sites, Sensors, CCTV cameras, Airlines etc. In other words, Big Data refers to datasets whose size are beyond the ability of typical database software tools to capture, store, manage and analyze. New technology has to be in place to gain value from these data, there must be an alternative way to store and process it Hadoop is technology which is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models. Hadoop has two major technologies to that manage Big Data.

- HDFS
- Map Reduce

Hadoop includes a fault-tolerant storage system called the Hadoop Distributed File System, or HDFS. HDFS is able to store huge amounts of information, scale up incrementally and survive the failure of significant parts of the storage infrastructure without losing data. Hadoop creates clusters of machines and coordinates work among them. Clusters can be built with inexpensive computers. If one fails, Hadoop continues to operate the cluster without losing data or interrupting work, by shifting work to the remaining machines in the cluster. HDFS manages storage on the cluster by breaking incoming files into pieces, called “blocks,” and storing each of the blocks redundantly across the pool of servers. In the common case, HDFS stores three complete copies of each file by copying each piece to three different servers[2].

Hadoop has five major services, namely

- Name Node
- Secondary Name Node
- Data Node
- Job Tracker
- Task Tracker

Name Node and Data node are used by HDFS and JobTracker and Task Tracker are used by Map Reduce.

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

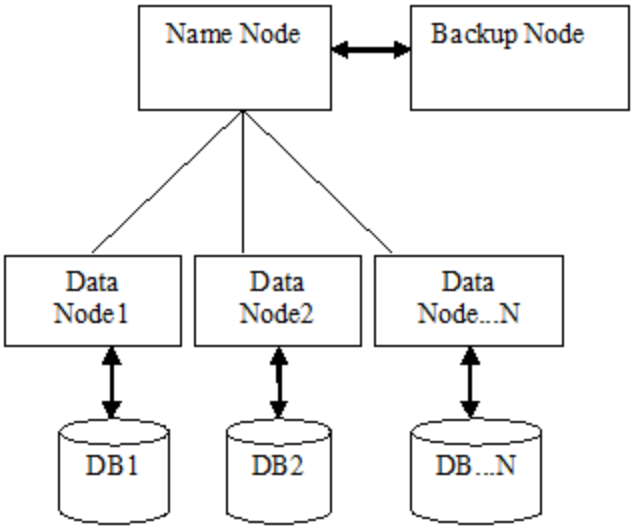

Fig: Architecture of HDFS

The design of HDFS is based on the design of GFS, the Google File System. Its design was described in a paper published by Google.
HDFS is a block-structured file system: individual files are broken into blocks of a fixed size. These blocks are stored across a cluster of one or more machines with data storage capacity. Individual machines in the cluster are referred to as Data Nodes. A file can be made of several blocks, and they are not necessarily stored on the same machine; the target machines which hold each block are chosen randomly on a block-by-block basis. Thus access to a file may require the cooperation of multiple machines, but supports file sizes far larger than a single-machine DFS; individual files can require more space than a single hard drive could hold.
Most block-structured file systems use a block size on the order of 4 or 8 KB. By contrast, the default block size in HDFS is 64MB -- orders of magnitude larger. This allows HDFS to decrease the amount of metadata storage required per file (the list of blocks per file will be smaller as the size of individual blocks increases).

The processing pillar in the Hadoop ecosystem is the MapReduce framework. The framework allows the specification of an operation to be applied to a huge data set, divide the problem and data, and run it in parallel. From an analyst's point of view, this can occur on multiple dimensions. For example, a very large dataset can be reduced into a smaller subset where analytics can be applied. In a traditional data warehousing scenario, this might entail applying an ETL operation on the data to produce something usable by the analyst. In Hadoop, these kinds of operations are written as MapReduce jobs in Java. There are a number of higher level languages like Hive and Pig that make writing these programs easier. The outputs of these jobs can be written back to either HDFS or placed in a traditional data warehouse. There are two functions in MapReduce as follows:

The purpose of the **map** phase is to organize the data in preparation for the processing done in the **reduce** phase. The input to the map function is in the form of key-value pairs, even though the input to a MapReduce program is a file or file(s). By default, the value is a data record and the key is generally the offset of the data record from the beginning of the data file.
map(inKey, inValue) → list(intermediateKey,intermediateValue)
Each **reduce** function processes the intermediate values for a particular key generated by the **map** function and generates the output. Essentially there exists a one-one mapping between keys and reducers. Several reducers can run in parallel, since they are independent of one another. The number of reducers is decided by the user. By default, the number of reducers is 1[4].
reduce(intermediateKey,list(intermediateValue)) → list (outKey,outValue)

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

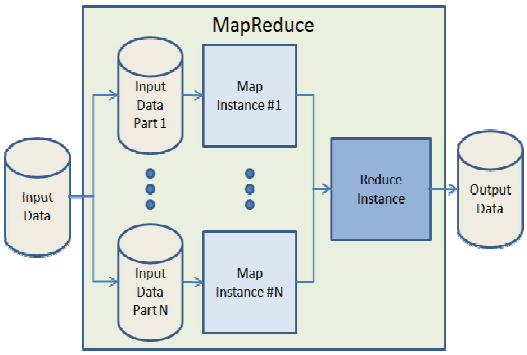

Fig: Architecture of Map Reduce

II. RELATED WORK

In [1] authors focused on the Big Data processing problems like lack of structure, error-handling, privacy ,timeline, provenance and visualization, at all the stages of the analysis pipeline from the data acquisition to the result interpretation. [2] Authors explained Data Storage is an important element of cloud computing. They discussed about working of HDFS and Map Reduce in Hadoop frame work. Combination of Data Mining and K-means clustering algorithm will make the data management is easier and quicker in cloud computing model. Cloud Computing will develop towards the security and reliable directions.
In [3] authors proposed a method to improve the efficiency of the map reduce scheduling algorithms. It works better than existing map reduce Scheduling algorithm by taking less amount of computations and gives high accuracy. They used the proposed k-means clustering algorithm together with the self-Adaptive Map Reduce(SAMR) algorithm. In [4] Explored through the Components of the Hadoop system, HDFS and Map Reduce. He also pointed out the architecture of HDFS, distribution of data across the cluster based on client application.In [5] published a research paper in which he explained about the need to process enormous quantities of data has never been greater. For engineers building information processing tools and applications, large and heterogeneous datasets which are generating continuous flow of data, lead to more effective algorithms for a wide range of tasks, from machine translation to spam detection.In [6] published a research paper in which he explored about the performance of Map reduce, and Hadoop in particular. Optimizing HDFS described in his paper will boost the overall efficiency of map reduce applications in Hadoop. The poor performance of HDFS can be attributed to challenges in maintaining portability, including disk scheduling under concurrent workloads, file system allocation, and file system cache overhead. HDFS performance under concurrent workloads can be significantly improved through the use of application level I/O scheduling while preserving portability.

III. ISSUES

There are many issues we have found during the research in Big Data. Such issues need to be taken into consideration for making the system work more efficiently. Some of the issues are:
HDFS architecture is not secure as there is no identity proof of the client as well as server means both client and server should prove their authenticity before requesting and granting service takes place.
If name node fails the whole system has to suffer because name node is the master node for data nodes. All information is sent to name node as name node maintains records for each activity takes place while storing the data.
Even if you can find and analyse data quickly and put it in the proper context for the audience that will be consuming the information, the value of data for decision-making purposes will be jeopardized if the data is not accurate or timely.
It takes a lot of understanding to get data in the right shape so that you can use visualization as part of data analysis. For example, if the data comes from social media content, you need to know who the user is in a general sense – such as a customer using a particular set of products – and understand what it is you're trying to visualize out of the data. Without some sort of context, visualization tools are likely to be of less value to the user.

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

IV. CONCLUSION AND FUTURE WORK

The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware. Hadoop is designed to run on cheap commodity hardware, It automatically handles data replication and node failure, It does the hard work – you can focus on processing data, Cost Saving and efficient and reliable data processing. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets. HDFS was originally built as infrastructure for the Apache Nutch web search engine project. HDFS is part of the Apache Hadoop Core project. In Future, will try to add security in HDFS cluster by making the client authentic and also try to reduce the tasks of client.

REFERENCES

1. Radhika M.Kharode and Anuradha R. Deshmukh,'Study of Hadoop Distributed File System in Cloud Computing', International Journal of Advanced Research in Computer Science and Software Engineering, Vol.5, Issue 1, pp. 990-993, 2015.
2. Harshawardhan S. Bhosle , Prof. Devendra P. Gadekar,'A Review Paper on Big Data and Hadoop', International Journal of Scientific and Research Publications, Vol.4, Issue 10,pp. 1-7, 2014.
3. S. Chandra Mouliswaran and Shyam Sathyan,'Study of Replica Management and High Availability in Hadoop Distributed File system', Journal of science, Vol.2, Issue 2, pp. 65-70, 2012.
4. Mrudula Varade and Vimla Jethani,' Distributed Metadata Management Scheme in HDFS', International Journal of Science and Research publications, Vol. 3, Issue 5, 2013.
5. Konstantin Shvachko, Chunlin LI, Z. Yang, Naji Hasan.A.H and X.Zhang ,' Improved the Energy of Ad hoc On- Demand Distance Vector Routing Protocol', International Conference on Future Computer Supported Education, Published by Elsevier, IERI, pp. 355-361, 2012.