International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Web Site: www.ijaiem.org Email: editor@ijaiem.org
Volume 3, Issue 3, March 2014 ISSN 2319 - 4847

UNDERSTANDING THE BIG DATA PROBLEMS AND THEIR SOLUTIONS USING HADOOP AND MAP-REDUCE

Mr. Swapnil A. Kale[1], Prof. Sangram S.Dandge[2]

[1]ME (CSE), First Year, Department of CSE, Prof. Ram Meghe Institute Of Technology and Research, Badnera,Amravati.
Sant Gadgebaba Amravati University, Amarvati, Maharashtra, India - 444701

[2]Assitantant Professor, Department of CSE, Prof. Ram Meghe Institute Of Technology and Research, Badnera,Amravati.
Sant Gadgebaba Amravati University, Amarvati, Maharashtra, India - 444701

Abstract

Big data connotes performing database operations and computations for substantial amounts of data, remotely from the data owner's enterprise. Information is being created and gathered at a rate that is rapidly nearing the Exabyte/year range. But, its creation and collection are speeding up and will come close to the Zettabyte/year range in a few years. The challenge is not only to store and manage this enormous volume, but also to analyze and draw significant value from it. The 'Big' question arise here is "How to meet the future pace and complexity that data demands so as to deliver right information at right time?" This paper focuses on analysis of problems and challenges in conforming and going for Big data technology and its optimal solution using Hadoop Distributed File System (HDFS) and Map Reduce framework.

Keywords: Big Data Problem, Big Data Characteristics, Hadoop Distributed File System, Map Reduce Framework

1. INTRODUCTION

'BIG DATA' has been getting much importance in IT industry over the last year or two, on a scale that has caused experienced industry-watchers to whiff the air for the familiar aroma of industry exaggeration. Big Data is a term applied to data sets of very large size such that the available tools are unable to undertake their acquisition, access, analytics and/or application in a reasonable amount of time. Typical difficulties include capture, storage, search, sharing, analysis, and visualising [2], [3]. For Storing large volumes of data there is a need to deviate from traditional relational database management approach. If ACID rules are implemented for searching huge volumes of Big data, it would be time consuming. Consider for example if we are to search certain information using Google, it returns number of pages of which only one is visible and rest are in soft state. This is the rule followed by Big databases which are generally NoSQL databases – 'basically available soft state and eventually consistent'. Big data bases follow only three properties – consistency, availability and partition (CAP) [1].

The world's information is more than doubling every two years. According to Cisco 1 zettabyte is equal to 250 billion DVDs that would take a long time to watch for an individual. By 2015 the majority of internet traffic around 61% will be in some form of video [5]. The following table shows the data ranges.

Table 1: Data Values and their Conversions

VALUE	NAME	CONVERSION
1000^1	Kilobytes (KB)	1KB=1024Bytes
1000^2	Megabytes(MB)	1MB = 1024 KB
1000^3	Gigabytes(GB)	1GB = 1024 MB
1000^4	Terabytes(TB)	1TB= 1024 GB
1000^5	Petabytes(PB)	1PB = 1024TB
1000^6	Exabytes(EB)	1EB= 1024PB
1000^7	Zettabytes(ZB)	1ZB = 1024EB
1000^8	Yottabytes(YB)	1YB = 1024ZB

In 2004, Google published a paper on Google File Systems (GFS) and a subsequent paper on MapReduce, which is a framework for distributed computing and large datasets on a scale-out shared-nothing architecture to address processing large unstructured data sets. Hadoop is an open source Apache project, which was implemented on these concepts by Google [2].

Big data systems these days have already formed the central part of technologies powering enterprises like IBM, Yahoo! and Facebook [1]. To deal with the costly system deployment and management problem faced by small enterprises, cloud-based big data analysis has been widely suggested. It enables flexible services framework to the users in a pay-as-

you-go manner, and largely reduces their deployment and management costs. There are various sources from which enormous data volume is gathered daily.

Table 2: Sources of Big Data

Big Data Datasets	Types of Data	Area of Use
Social Network **Facebook Dataset** **Twitter Dataset**	Text, Photo, Video Text, Photo	To keep in touch with friends, find new friends and information sharing
Email Network **Gmail Dataset**	Text, Photo, Video	To send email and information connection
GPS **Mobile Phones tracks Dataset**	Mobile positions	When mobile phone is on it sends the digital signal to locate current position of mobile
Online Shopping **EBay Dataset**	Items	Used by buyers and sellers to buy and sell items on Internet
Computer Network **Traffic Dataset**	Exchange, attack	Used anytime to get information of sent and received data
Satellite Data **Weather Dataset**	Temperature and Humidity	Digital signals are sent by the satellite giving weather details
Web Logs **Google contents Dataset**	Text, Image, Video	To get Information over the Internet
Video Streams **YouTube Dataset**	Video	To Upload videos online
Finance Transactions **Bank Transactions Dataset**	Account numbers, amount	For Online financial transactions
Smart Phones **Whatsapp Dataset** **WeChat Dataset**	Text, Image, Audio, Video Text, Image, Audio, Video	Based on the mobile number for keeping in touch with friends and making new friends

2. BIG DATA CHARACTERISTICS –

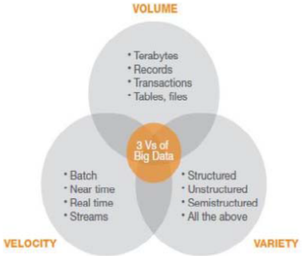


Figure 1: The three V's of Big Data

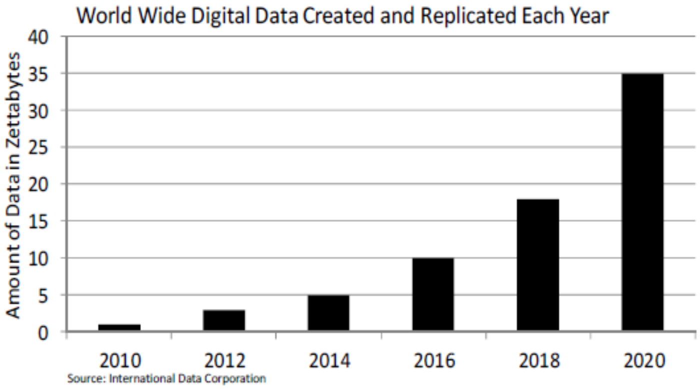
- ***Volume***: Data volume is the measurement of amount of data available to an organization, which does not necessarily have to possess all of it as long as it can access it [4]. There are different factors like age, quantity, type and richness etc. that decide the value of different data records which is going to decrease as volume of data increases. Take an example of social networking sites these days; they are themselves producing data in order of terabytes and considering traditional systems this amount of data is not easy to handle.

- ***Velocity***: Data velocity quantifies the speed of data creation, streaming, and aggregation. Big data is rapidly coming from various sources and this characteristic is not being limited to the speed of incoming data but also speed at which the data flows. eCommerce has swiftly increased the speed and richness of data used for different business transactions (for example, web-site clicks). Data velocity management is much more than a bandwidth concern; it is also a consumption issue (extract- transform-load) [4].

- ***Variety***: Data variety is a measure of the sumptuousness of the data representation from an analytic perspective, for example text, images video, audio, etc. [4]; it is probably the biggest barrier to effectively use large volumes of such data. Irreconcilable data formats, non-aligned data structures, and inconsistent data semantics represents momentous challenges that can lead to analytic difficulty.

3. BIG DATA PROBLEMS AND CHALLENGES-

The problem comes straight way when the data tsunami requires us to make specific decisions, about what data to keep and what to reject, and how to store what we keep reliably with the right metadata. Transforming unstructured content into structured format for later analysis is a major challenge [1]. Data analysis, organization, retrieval, and modelling are other foundational challenges. Since most data is directly generated in digital format today, the challenge is to influence the creation so as to facilitate later linkage and to automatically link previously created data.



Source: International Data Corporation

Figure 2: Worldwide Data Creation

1. Heterogeneity

Machine analysis algorithms expect homogeneous data, and cannot understand fine distinction. Computer systems work most expeditiously if they can store multiple items that are identical in size and structure. So, data must be carefully structured as a first step in (or prior to) data analysis [1].

2. Scale

The first thing that anyone thinks of in Big data is its size. Managing large and rapidly increasing volumes of data has been an exigent issue for many decades. In the past, this challenge was palliated by processors getting faster [1], [6]. But there is a fundamental shift happening now: 'data volume is scaling faster than computer resources'. Unluckily, parallel data processing techniques useful in the past for processing data across nodes don't directly apply for intra-node parallelism, since the architecture is very different.

3. Timeliness

The larger the data set to be processed, the longer it will take to analyze. The design of a system that effectively deals with size is likely also to result in a system that can process a given size of data set faster. However, it is not just this speed that is usually meant when one speaks of Velocity in the context of Big Data [2]. There are many situations in which the result of the analysis is required immediately. Given a large data set, it is often essential to find elements in it that meet a precise criterion. Scanning the entire data set to find suitable elements is obviously unfeasible [1].

4. Privacy and Security

A key value proposition of big data is access to data from multiple and diverse domains, security and privacy will play a very important role in big data research and technology. In domains like social media and health information, more data is gathered about individuals, so there is a fear that certain organizations will know too much about individuals.

Developing algorithms that randomize personal data among a large data set so as to ensure privacy is a key research problem [1].

5. System Architecture

Business data is examined and studied for many purposes that might include system log analytics and social media analytics for risk measurement, customer retention, and brand management etc. Typically, such diverse tasks have been handled by separate systems, even if each system includes common steps. The challenge here is not to build a system that is ideally suited for all processing tasks. Instead, the need is for the primary system architecture to be flexible enough that the components built on top of it for showing the various kinds of processing tasks can tune it to expeditiously run these different workloads [6].

4. HADOOP DISTRIBUTED FILE SYSTEM (HDFS) -

The principle of Hadoop was to process data in a distributed file system fashion. Therefore, a single file is split into blocks and the blocks are spread in the Hadoop cluster nodes. Hadoop applications require highly available distributed file systems with unconstrained capacity. The input data in HDFS is treated in write-once fashion and processed by MapReduce, and the results are written back in the HDFS [3], [5].

In HDFS information (terabytes or petabytes) is stored across many servers in larger file sizes. HDFS has default block size of 64MB, which results in fewer files to store and decreased metadata information stored for each file. It also provides streaming read performance, rather than random seek to arbitrary positions in files. The files are large in size and sequentially read so there is no local caching. HDFS reads a block start-to-finish for the Hadoop MapReduce application. The data in HDFS is protected by a replication mechanism among the nodes. This provides reliability and availability despite node failures. There are two types of HDFS nodes: DataNode, which stores the datablocks of the files in HDFS; and NameNode, which contains the metadata, with details of blocks of HDFS and a list of DataNodes in the cluster. There are two additional types of Map Reduce nodes: Job Tracker and Name Node [3].

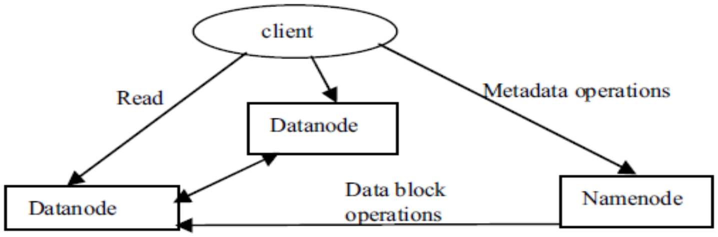


Figure 3: HDFS File Structure

5. MAPREDUCE FRAMEWORK -

A data can be of structured, semi structured or unstructured type. Data that resides in a fixed field within a record or file is called structured data. Structured data is organized in a highly mechanized and manageable way. Unstructured data refers to information that either does not fit well into relational tables or does not have a pre-defined data model [2].

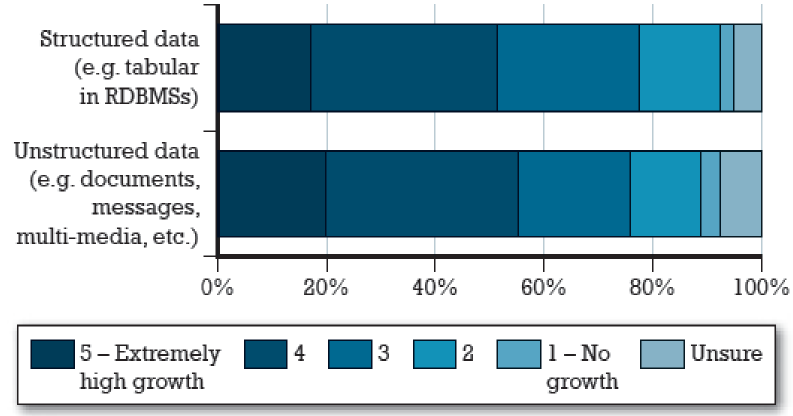


Figure 4: Organisations are seeing data volumes increase, with unstructured data it is also set to grow even faster than structured data in some cases Unstructured data is the fastest growing type of data, some example could be images, sensors, telemetry, video, documents, log files, and email data files. There are several techniques to address this problem of unstructured analytics. The techniques share common characteristics of scale-out, elasticity and high availability. MapReduce distributed data processing architecture has become the preferent choice for data-intensive analysis due to its excellent fault tolerance features, scalability and the ease of use.

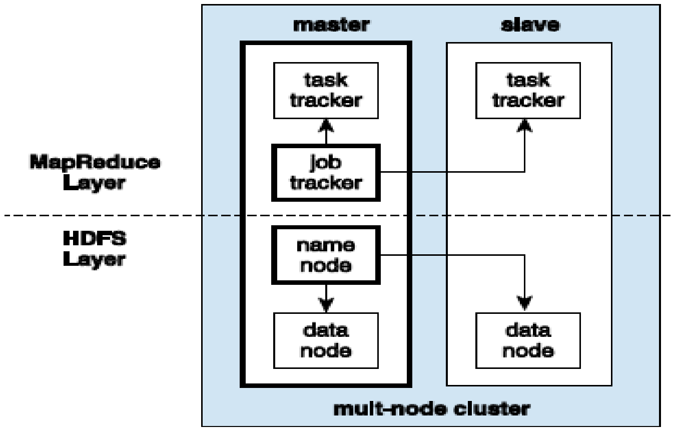


Figure 5: Map Reduce and HDFS System Architecture

Map Reduce is a programming model to process large volumes of data generally in tuple (pair) lists like [student | marks], [employee | salary], [liquid | density] this is accomplished by dividing the work into independent tasks that are spread across many nodes (servers) [3]. The data elements in Map Reduce are immutable i.e. they cannot be updated. For Example, if during a Map Reduce job, input data is changed (like modifying a student's marks) the change is not reflected in input files rather a new (key, value) pair is generated and passed by Hadoop to next phase of execution. If the amount of data to reduce is vast then all of the values with the same key are presented to a single reducing function together. This is performed independently of any reduce operations occurring on other lists of values, with different keys attached for example marks of 10th class students are reduced in one set and that of 11th class students in other [5].
The following section provides an introduction of how Map Reduce processes a job.

- **Input**: The data is split into blocks and distributed to data nodes of the cluster and then loaded into HDFS. The blocks are replicated for availability in case of failures. The Name node keeps track of the data nodes and blocks.
- **Job**: Submits the Map Reduce job and its details to the JobTracker.
- **Job_Init**: The Job Tracker interacts with TaskTracker on each data node to schedule Map Reduce tasks.
- **Map**: Mapper processes the data blocks and generates a list of key value pairs.
- **Sort**: Mapper sorts the list of key value pairs.
- **Shuffle**: Transfers the mapped output to the reducers in a sorted fashion.
- **Reduce**: Reducers merge the list of key value pairs to generate the final result.
- Finally, the results are stored in HDFS and replicated as per the configuration. The results are finally read from the HDFS by the clients.

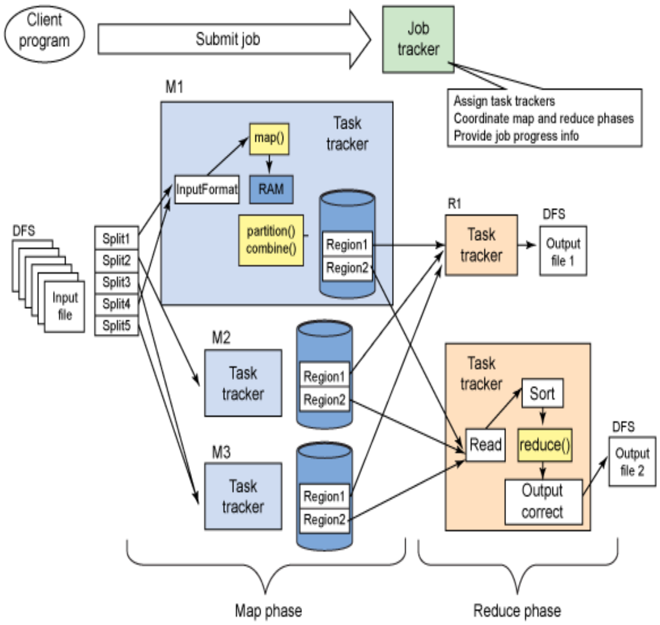


Figure 6: Map Reduce Data Flow Diagram

6. NEED OF HDFS FOR MAP REDUCE -

Input to Map Reduce come from input files which are evenly distributed across all servers. Mapping tasks are run on many or all of the nodes and are identical; therefore, any mapper can process any input file. Data is processed at the node where it exists and not copied to a central server [2]. So Individual map tasks do not communicate with one another, nor they are aware of one another's existence. After completing mapping phase, the intermediate (key, value) pairs must be exchanged between servers so all values with the same key are sent to a single reduce job. As illustrated in the example above, all the values from the class 10th must be sent to the reduce job responsible for class 10th; similarly for other classes. If servers in the Hadoop cluster fail, the map or reduce tasks must be able to be restarted. Individual task nodes i.e. TaskTrackers communicate with the head node of the system i.e. a JobTracker [3]. If a TaskTracker fails to communicate with the JobTracker in a default time (1 minute), the JobTracker will presume that the specific TaskTracker has failed. The JobTracker knows which map and reduce tasks were assigned to each TaskTracker; other TaskTrackers will re-execute a failed map or reduce job. This is a highly scalable and fault tolerant file system required for successful Map Reduce jobs.

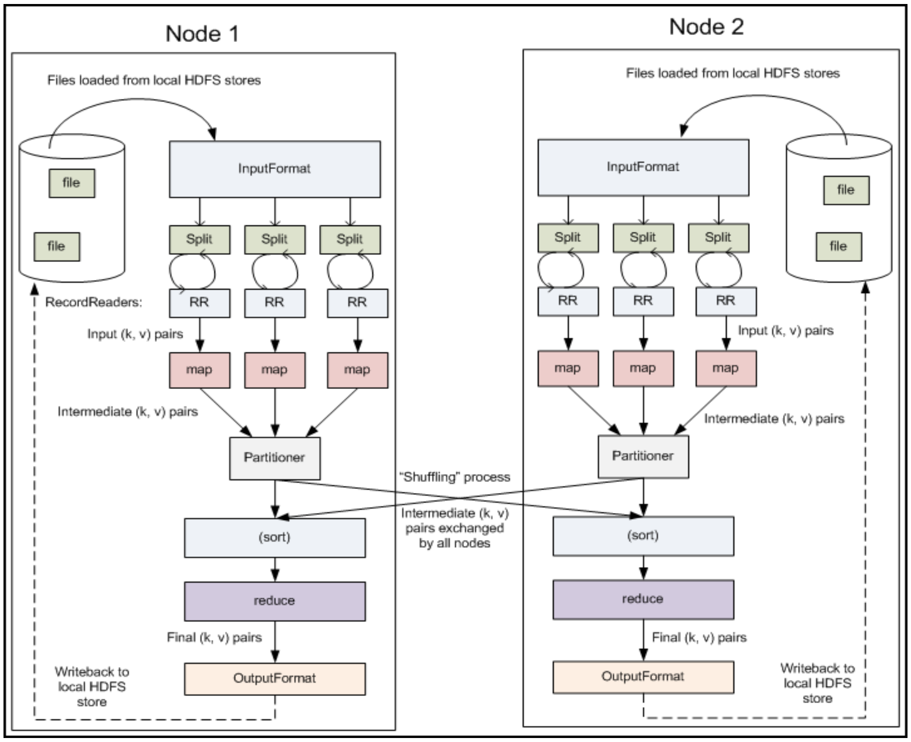


Figure 7: Two nodes with multiple mappers and reducers

7. CONCLUSIONS

This paper depicted the new concept of Big data, its significance and the future data requirements. To meet these new technology problems there is a need for better analysis of the large volumes of data that are becoming available. There are future prospects in making faster advances in many scientific areas using Big data technology. However, many technical challenges described in this paper must be looked for before these prospects can be realized fully. These problems and challenges will help the business organizations which are considering Big data technology for increasing the value of their business by dealing with them right in the beginning. Hadoop Distributed File System and Map Reduce framework for Big data are described in detail so that in future Big data can have technology and skills to work with.

REFERENCES

[1] Stephen Kaisler, Frank Armour, J. Alberto Espinosa, William Money, "Big Data: Issues and Challenges Moving Forward", *IEEE, 46th Hawaii International Conference on System Sciences*, 2013.

[2] Sam Madden, "From Databases to Big Data", *IEEE, Internet Computing*, May-June 2012.

[3] Kapil Bakshi, "Considerations for Big Data: Architecture and Approach", *IEEE, Aerospace Conference*, 2012.

[4] Sachchidanand Singh, Nirmala Singh, "Big Data Analytics", IEEE, International Conference on Communication, Information & Computing Technology (ICCICT), Oct. 19-20, 2012.

[5] Yuri Demchenko, Zhiming Zhao, Paola Grosso, Adianto Wibisono, Cees de Laat, “Addressing Big Data Challenges for Scientific Data Infrastructure”, *IEEE , 4th International Conference on Cloud Computing Technology and Science*, 2012.

[6] Martin Courtney, “The Larging-up of Big Data”, *IEEE, Engineering & Technology*, September 2012.

AUTHORS

Mr.Swapnil A. Kale, ME (CSE) ,First Year,Department of CSE,Prof. Ram Meghe Institute Of Technology and Research, Badnera, Amravati. Sant Gadgebaba Amravati University, Amravati, Maharashtra, India - 444701

Prof. Sangram S. Dandge, Assitantant Professor, Department of CSE, Prof. Ram Meghe Institute Of Technology and Research, Badnera,Amravati. Sant Gadgebaba Amravati University, Amarvati, Maharashtra, India - 444701