ACSIJ Advances in Computer Science: an International Journal, Vol. 4, Issue 4, No.16 , July 2015
ISSN : 2322-5157
www.ACSIJ.org

ACSIJ
WWW.ACSIJ.ORG

Correlated Appraisal of Big Data, Hadoop and MapReduce

Priyaneet Bhatia[1], Siddarth Gupta[2]

[1] Department of Computer Science and Engineering, Galgotias College of Engineering and Technology
Uttar Pradesh Technical University
Greater Noida, Uttar Pradesh 201306, India
priyaneet2800@gmail.com

[2]Department of Computer Science and Engineering, Galgotias University
Greater Noida, Uttar Pradesh 203208, India
siddharthgupta1602@gmail.com

Abstract

Big data has been an imperative quantum globally. Gargantuan data types starting from terabytes to petabytes are used incessantly. But, to cache these database competencies is an arduous task. Although, conventional database mechanisms were integral elements for reservoir of intricate and immeasurable datasets, however, it is through the approach of NoSQL that is able to accumulate the prodigious information in a proficient style. Furthermore, the Hadoop framework is used which has numerous components. One of its foremost constituent is the MapReduce. The MapReduce is the programming quintessential on which mining of purposive knowledge is extracted. In this paper, the postulates of big data are discussed. Moreover, the Hadoop architecture is shown as a master- slave procedure to distribute the jobs evenly in a parallel style. The MapReduce has been epitomized with the help of an algorithm. It represents WordCount as the criterion for mapping and reducing the datasets.

Keywords: *Big Data, Hadoop, MapReduce, RDBMS, NoSQL, Wordcount*

1. Introduction

As immense amount of data is being generated day by day, efficiency of storage capacity for this huge information becomes a painful task [1]. Therefore exabytes or petabytes of database known as big data need to be scaled down to smaller datasets through an architecture called Hadoop.

Apache Hadoop is an open source framework on which the big data is processed with the help of MapReduce [2]. A programming model, MapReduce uses basic divide and conquer technique to its own map and reduce functions. On copious datasets it processes key/value pairs to generate intermediate key/value pairs and then, with the help of these pairs, merges the intermediate values to form a smaller sets of key/values sets [3][4]. The reduced data bytes of massive information are produced. The rest of the paper is formulated as follows. Section 2 covers the concepts of big data, its 3 V's and its applications in real world scenarios. Section 3 describes the comparison between RDBMS and NoSQL and why NoSQL rather than RDBMS is used in today's world. Section 4 explores the Apache Hadoop in detail and its use in big data. Section 5 analyzes the MapReduce paradigm, its use in Hadoop paradigm, and its significance in enormous data reduction. Section 6 explicates the table comparisons of big data and Hadoop of various survey papers. Finally, the paper is concluded in the end.

2. Big Data Concepts

2.1 Outline of Big Data

Let's start with big data. What is big data? Why has it created a buzz? Why is big data so essential in our daily chores of life? Where is it used? All these unanswerable questions have made everyone curious. Moving on, big data is actually a collection of large and complex datasets that has become very difficult to handle using traditional relational database management tools [5].

2.2 Four Vs' of Big Data

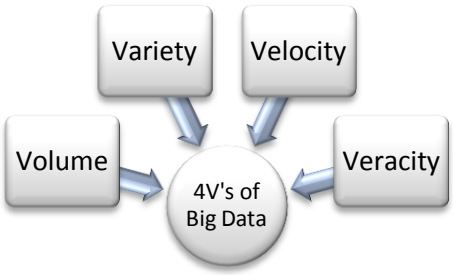

Figure 1: 4Vs' of Big Data

Big data has its own 4 characteristics shown above in Fig 1 as:

i. Volume: refers to the amount of space or quantity of something. Since data is huge scaled complex sized, even larger than 1024 terabytes, it becomes a challenge to extract relevance information from traditional database techniques. E.g. the world produces 2.5 quintillion bytes in a year.

ii. Variety: represent the state of being varied or diversified. In this, all types of formats are available. Structured, unstructured, semi-structured data etc. These varieties of formats are needed to be searched, analyzed, stored and managed to get the useful information. E.g.: geospatial data, climate information, audio and video files, log files, mobile data, social media etc [6].

iii. Velocity: the rate of speed with which something happens. In this, to deal with bulky spatial dimensional data which is streaming at an eccentric rate is still an eminent challenge for many organizations. E.g. Google produces 24 TB/day; Twitter handles 7 TB/day etc.

iv. Veracity: refers to the accuracy of the information extracted. In this, data is mined for profitable purposes [7].

2.3 Big Data in Real World Scenarios

a) Facebook generates 10-20 billions photos which is approximately equivalence to 1 petabytes.
b) Earlier, hard copy photographs take space around 10 gigabytes in a canon camera. But, nowadays, digital camera is producing photographic data more than 35 times the old camera used to take and it is increasing day by day [8].
c) Videos on youtube are being uploaded in 72 hours/min.
d) Data produced by google is approximately 100 peta-bytes per month [9].

3. RDMS VS NOSQL

3.1 RDBMS

For several decades, relational database management system has been the contemporary benchmark in various database applications. It organizes the data in a well structured pattern. Unfortunately, ever since the dawn of big data era, the information comes mostly in unstructured dimensions. This culminated the traditional database system in not able to handle the competency of prodigious storage database. In consequence, it is not opted as a scalable resolution to meet the demands for big data [10].

3.2 NoSQL

NoSQL commonly refers to 'Not Only SQL', has become a necessity in replacement to RDBMS, since its main characteristics focuses on data duplication and unstructured schemes. It allows unstructured compositions to be reserved and replicated across multiple servers for future needs. Eventually, no slowdown in performance occurs unlike in RDBMS. Companies such as Facebook, Google and Twitter use NoSQL for their high performance, scalability, availability of data with regards to the expectations of the users [11].

4. Hadoop

4.1 Hadoop in Brief

Hadoop was created by Doug Cutting and Mike Caferella in 2005 [12]. It was named after his son's toy elephant [13]. It comprises of 2 components and other project libraries such as Hive, Pig, HBase, Zookeeper etc:

a. HDFS: open source data storage architecture with fault tolerant capacity.
b. MapReduce: programming model for distributed processing that works with all types of datasets. [14].

4.2 Motive behind Hadoop in Big Data

Despite, one might get worried that since RDBMS is a dwindling technology, it cannot be used in big data processing; however, Hadoop is not a replacement to RDBMS. Rather, it is a supplement to it. It adds characteristics to RDBMS features to improve the efficiency of database technology. Moreover, it is designed to solve the different sets of data problems that the traditional database system is unable to solve.

4.3 CAP Theorem for Hadoop

Cap theorem shown in Fig 2, can be defined as consistency, scalability and flexibility.

a) Consistency: simultaneous transactions are needed in continuity for withdrawing from the account and saving into the account.
b) Availability: flexibility in making multiples copies of data. If one copy goes down, then another is still accessible for future use.
c) Partitioning: to partition the data in multiple copies for storage in commodity hardware. By default, 3 copies are normally present. This is to make for easy feasibility for the customers [15].

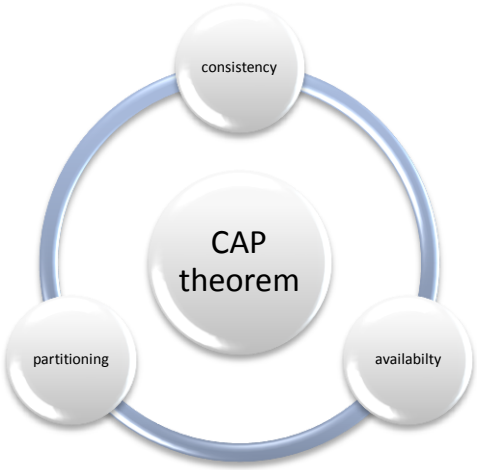

Figure 2: CAP Theorem

4.4 Hadoop Business Problems

i. Marketing analysis: market surveys are being used to understand the consumer behaviours and improve the quality of the product. Lot of companies used feedback survey to study shopper attitudes.
ii. Purchaser analysis: It is best to understand the interest of the current customer rather than the new one. Therefore, the best thing is collect as much information as one can to analyze what the buyer was doing before he left the shopping mall.
iii. Customer profiling: it is essential to identify specific group of consumers having similar interest and preferences in purchasing goods from the markets.
iv. Recommendation portals: These online shopping browsers not only collect database from your own data but also from those users who match the profile of yours, so that these search engines can make recommend webites that are likely to be useful to you. E.g.: Flipkart, Amazon, Paytm, Myntra etc.
v. Ads targeting: we all know ads are a great nuisance when we are doing online shopping, but they stay with us. These ad companies put their ads on popular social media sites so they can collect large amount of data to see what we are doing when we are actually shopping [16].

5. MapReduce

5.1 Understanding MapReduce

MapReduce is a programming paradigm, developed by Google, which is designed to solve a single problem. It is basically used as an implementation procedure to induce large datasets by using map and reduce operations [17].

5.2 Principles of MapReduce

a. Lateral computing: provides parallel data processing across the nodes of clusters using the Java based API. It works on commodity hardware in case of any hardware failure.
b. Programming languages: uses Java, Python and R languages for coding in creating and running jobs for mapper and reducer executables.
c. Data locality: ability to move the computational node close to where the data is. That means, the Hadoop will schedule MapReduce tasks close to where the data exist, on which that node will work on it. The idea of bringing the compute to the data rather than bringing data to the compute is the key of understanding MapReduce.
d. Fault tolerant with shared nothing: The Hadoop architecture is designed where the tasks have no dependency on each other. When node failure occurs, the MapReduce jobs are retried on other healthy nodes. This is to prevent any delays in the performance of any task. Moreover, these nodes failure are detected and handled automatically and programs are restarted as needed [18].

5.3 Parallel Distributed Architecture

The MapReduce is designed as the master slave framework shown in Fig 4, which works as job and task trackers. The master is the Jobtracker which performs execution across the mapper or reducer over the set of data. However, on each slave node, the Tasktracker executes either the map or reduce task. Each Tasktracker reports its status to its master.

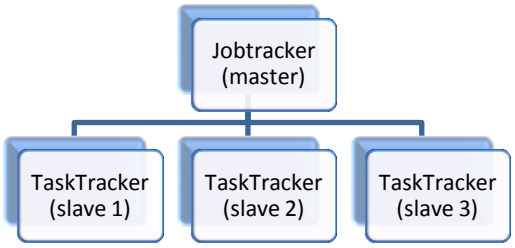

Figure 3: Master Slave Architecture

5.4 Programming Model

The MapReduce consists of 2 parts: map and reduce functions.

a) Map part: This is the 1st part of MapReduce. In this, when the MapReduce runs as a job, the mapper will run on each node where the data resides. Once it gets executed, it will

create a set of <key/value> pairs on each node.

b) Reduce part: In the 2nd part of MapReduce, the reducer will execute on some nodes, not all the nodes. It will create aggregated sets of <key, value> pairs on these nodes. The output of this function is a single combined list.

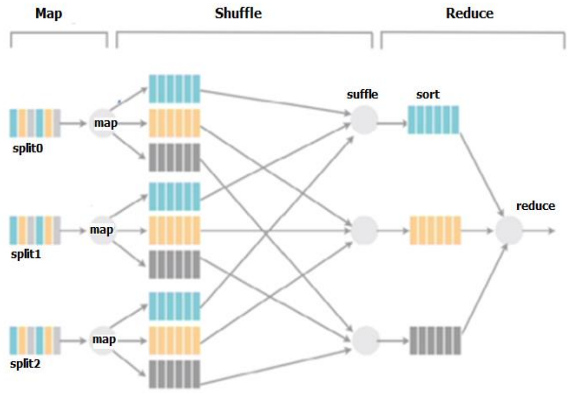

Figure 4: MapReduce Paradigm

Figure 4 displays the MapReduce prototype, comprised of three nodes under Map. Variegated categories of data have been represented through numerous colors in the Map. Accordingly, in essence, these nodes are running in three separate machines i.e. the commodity hardware. Thus, the chunks of information are implemented on discrete machines. Furthermore, the intermediate portion of this model resides the magical shuffle that is in fact quite complicated and hence is the key aspect of MapReduce. Literally, this framework has set the mind to think. How does this list come out from the Map function and is then aggregated to the Reduce function? Is this an automated process or some codes have to be written? Actually, in reality, this paradigm is the mixture of both. As a matter of fact, whenever the MapReduce jobs are written, the default implementations of shuffle and sort are studied. Consequently, all these mechanisms are tunable; one may accept the defaults or tune in or can change according to one's own convenience.

5.5 Word count

The Hello World of MapReduce program is the WordCount. It comes from Google trying to solve the data problem by counting all the words on the Web. It is de facto standard for starting with Hadoop programming. It takes an input as some text and produces a list of words and does counting on them.

Following below is the Pseudocode of WordCount [19][20]:

Mapper (filename, file contents):

For each word in file contents:

Emit (word, 1)

Reducer (word, values):

Sum=0

For each value in values

Sum+= value

Emit (word, sum)

The pseudocode of MapReduce contains the mapper and reducer. The mapper has the filename and file contents and a loop for each to iterate the information. The word is emitted which has a value 1. Basically, spliiting occurs. In the reducer, from the mapper, it takes the output and produces lists of keys and values. In this case, the keys are the words and value is the total manifestation of that word. After that, zero is started as an initializer and loop occurs again. For each value in values, the sum is taken and value is added to it. Then, the aggregated count is emitted.

5.6 Example [21]

Consider the question of counting the occurrence of each word in the accumulation of large documents. Let's take 2 input files and perform MapReduce operation on them.

File 1: bonjour sun hello moon goodbye world
File 2: bonjour hello goodbye goodluck world earth

Map:
First map: second map

<bonjour, 1><bonjour,1>
<sun,1><hello,1>
<hello, 1><goodbye,1>
<moon,1>< goodluck,1>
<goodbye,1><world,1>
<world,1><earth,1>

Reduce:

<bonjour, 2>
<sun,1>
<hello, 2>
<moon,1>
< goodluck,1>
<goodbye,2>
<world,2>
<earth,1>

5.7 Algorithm Steps [22]:

a) Map step: in this step, it takes key and value pairs of input data and transform into output intermediate list of key/ value pairs.

$$map(ky_{in}, value_{in}) \rightarrow list(key_{out}, value_{intermediate}) \quad (1)$$

b) Reduce step: in this step, after being shuffled and sorted, the output intermediate key/value pairs is passed through a reduce function where these values are merged together to form a smaller sets of values.

$$reduce\left(key_{out}, list(value_{intermediate}) \rightarrow list(value_{out})\right) \quad (2)$$

6. Tabular Comparisons on Big Data and Hadoop

Table 1: Approach on Big Data

S.No	Author's name	Year	Approach on Big Data	Results/ Conclusion
1.	Puneet Singh Duggal et al	2013	Big Data analysis tools, Hadoop, HDFS, MapReduce	Used for storing and managing Big Data. Help organizations to understand better customers & market
2.	Min Chen et al	2014	Cloud computing, Hadoop	Focus on 4 phases of value chain of Big Data i.e., data generation, data acquisition, data storage and data analysis.
3.	P.Sarada Devi et al	2014	Hadoop, extract transform load (ETL) tools like ELT, ELTL.	Introduces ETL process in taking business intelligence decisions in Hadoop
4.	Poonam S. Patil et al	2014	RDBMS, NoSQL, Hadoop, MapReduce	Study challenges to deal analysis of big data. Gives flexibility to use any language to write algorithms.
5.	K.Arun et al	2014	mining techniques like association rule learning, clustering classification	Study big data classifications to business needs. Helps in decision making in business environment by implementing data mining techniques,

Table 2: Approach on Hadoop

S.No	Author's Name	Year	Approach on Hadoop	Results/ Conclusion
1.	Mahesh Maurya et al	2011	MapReduce, Linkcount, WordCount	Experimental setup to count number of words & links (double square brackets) available in Wikipedia file. Results depend on data size & Hadoop cluster.
2.	Puneet Duggal et al	2013	HDFS, MapReduce, joins, indexing, clustering, classification	Studied Map Reduce techniques implemented for Big Data analysis using HDFS.
3.	Shreyas Kudale et al	2013	HDFS, MapReduce, ETL, Associative Rule Mining	Hadoop's not an ETL tool but platform supports ETL processes in parallel.
4.	Poonam S. Patil et al	2014	HDFS, MapReduce, HBase, Pig, Hive, Yarn,	Parallelize & distribute computations tolerant.
5.	Prajesh P. Anchalia et al	2014	k-means clustering algorithms	Experimental setup for MapReduce technique on k-means clustering algorithm which clustered over 10 million data points.
6.	Radhika M. Kharode et al	2015	HDFS, MapReduce , k-means algorithms, cloud computing	Combination of data mining & K-means clustering algorithm make data management easier and quicker in cloud computing model.

7. Conclusion

To summarize, the recent literature of various architectures have been surveyed that helped in the reduction of big data to simple data which mainly composed of immense knowledge in gigabytes or megabytes. The concept of Hadoop, its use in big data has been analyzed and its major component HDFS and MapReduce have been exemplified in detail. Overall, the MapReduce model is illustrated with its algorithm and an example for the readers to understand it clearly. To sum up, applications of big data in real world scenario has been elucidated.

Acknowledgements

Priyaneet Bhatia and Siddarth Gupta thanks **Mr. Deepak Kumar**, Assistant Professor, Department of Information Technology, and **Rajkumar Singh Rathore**, Assistant Professor, Department of Computer Science and Engineering, Galgotia

College of Engineering and Technology, Greater Noida, for their constant support and guidance throughout the course of whole survey.

References

[1] Shreyas Kudale, Advait Kulkarni and Leena A. Deshpande, "Predictive Analysis Using Hadoop: A Survey", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 1, Issue 8, 2013. pp 1868-1873

[2] P.Sarada Devi, V.Visweswara Rao and K.Raghavender, "Emerging Technology Big Data Hadoop Over Datawarehousing, ETL" in International Conference (IRF), 2014, pp 30-34.

[3] Jeffrey Dean and Sanjay Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters", Google, Inc in USENIX Association OSDI '04: 6th Symposium on Operating Systems Design and Implementation, 2009, pp 137-149.

[4] Jeffrey Dean And Sanjay Ghemawat, "MapReduce: Simplified data Processing on Large Cluster" in OSDI'04: Sixth Symposium on Operating System Design and Implementation, CS 739 Review Blog, 2004, http: //pages.cs.wisc.edu/~swift/classes/cs739-sp10/blog/2010/04/mapreduce_simplified_data_proc.html

[5] Munesh Kataria and Ms. Pooja Mittal, "Big Data and Hadoop with Components like Flume, Pig, Hive and Jaql", International Journal of Computer Science and Mobile Computing (IJCSMC), Vol. 3, Issue 7, 2014. pp.759 – 765,

[6] Jaseena K.U. and Julie M. David, "Issues, Challenges, and Solutions: Big Data Mining", NeTCoM, CSIT, GRAPH-HOC, SPTM – 2014, 2014, pp. 131–140

[7] K.Arun and Dr. L. Jabasheela, "Big Data: Review, Classification and Analysis Survey", International Journal of Innovative Research in Information Security (IJIRIS), 2014, Vol. 1, Issue 3, pp 17-23

[8] T. White, Hadoop: The Definitive Guide, O'Reilly Media, Yahoo! Press, 2009.

[9] Min Chen, Shiwen Mao and Yunhao Liu, "Big Data: A Survey", Springer, New York, 2014, pp-171-209

[10] Poonam S. Patil and Rajesh. N. Phursule, "Survey Paper on Big Data Processing and Hadoop Components", International Journal of Science and Research (IJSR), Vol.3, Issue 10, 2014 pp 585-590

[11] Leonardo Rocha, Fernando Vale, Elder Cirilo, Dárlinton Barbosa and Fernando Mourão, "A Framework for Migrating Relational Datasets to NoSQL", in International Conference on Computational Science, , Elsevier, Vol.51, 2015, pp 2593–2602

[12] Apache Hadoop, Wikipedia https://en.wikipedia.org/wiki/Apache_Hadoop

[13] Ronald C Taylor, "An overview of the Hadoop/ MapReduce/ HBase framework and its current applications in bioinformatics" in Bioinformatics Open Source Conference (BOSC), 2010, pp 1-6.

[14] Puneet Singh Duggal and Sanchita Paul, "Big Data Analysis: Challenges and Solutions" in International Conference on Cloud, Big Data and Trust, RGPV, 2013, pp 269-276

[15] lynn langit, Hadoop fundamentals, ACM, 2015, Lynda.com , http://www.lynda.com/Hadoop-tutorials/Hadoop-Fundamentals/191942-2.html

[16] ShaokunFana, Raymond Y.K.Laub, and J. LeonZhaob,. "Demystifying Big Data Analytics for Business Intelligence through the Lens of Marketing Mix", Elsevier, ScienceDirect, 2015, pp 28-32.

[17] Mahesh Maurya and Sunita Mahajan ,"Comparative analysis of MapReduce job by keeping data constant and varying cluster size technique", Elseveir, 2011, pp 696-701

[18] Dhole Poonam and Gunjal Baisa, "Survey Paper on Traditional Hadoop and Pipelined Map Reduce", International Journal of Computational Engineering Research (IJCER), Vol. 3, Issue 12, 2013, pp 32-36

[19] MapReduce, Apache Hadoop, Yahoo Developer Network, https://developer.yahoo.com/hadoop/tutorial/module4.html

[20] Mahesh Maurya and Sunita Mahajan, "Performance analysis of MapReduce programs on Hadoop Cluster" IEEE, World Congress on Information and Communication Technologies (WICT2012), 2012, pp 505-510.

[21] Ms. Vibhavari Chavan and Prof. Rajesh. N. Phursule, "Survey Paper on Big Data", International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 5, No.6, 2014, pp 7932-7939.

[22] 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 (IJARCSSE) , Vol.5, Issue 1, 2015, pp 990-993.

First Author Priyaneet Bhatia has done her B.Tech in IT from RTU, Jaipur, Rajasthan, India in 2012. Currently, she is pursuing M.Tech in CSE from Galgotia College of Engineering and Technology, UPTU, Greater Noida, Uttar Pradesh, India. She is working on the project "Big Data in Hadoop MapReduce".

Second Author Siddarth Gupta has done B.Tech in CSE from UPTU, Lucknow, Uttar Pradesh, India in 2012.He has completed M.tech in CSE from Galgotias University, Greater Noida, Uttar Pradesh, India in May 2015. He is currently working on "Big Data optimization in Hadoop"