Volume 6, Issue 2, February 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Perusal of WordCount Proposition by Dint of Mapreduce Dogma

Priyaneet Bhatia, Prof. Bhawna Mallick
Department of Computer Science and Engineering, Galgotias College of Engineering and Technology,
Dr. A.P.J Abdul Kalam Technical University, Greater Noida, Uttar Pradesh, India

Abstract: Ethereally, Big Data has been as a discerned as a coercive quantum. Colossal data types embarked from terabytes to petabytes are rendered perpetually. However, to repertoire this database amplitude is a meandering errand. Granted that the predominant database tactics are an inherent facet for repository of circuitous and stupendous datasets; nevertheless, it is through the advent of Hadoop that is able to annex the capacious information in an effectual aspect. Besides, on having profuse integrals, Hadoop paradigm is put forth. It's preeminent intrinsic are HDFS and MapReduce. Substantially, HDFS is an open source data cache of canonical with fault tolerant competence. In essence, MapReduce is a programming paramount on which delving of pragmatic knowledge is unearthed. Over and above, part and parcels of Hadoop are excogitated at length. Consistently, the cardinal titillation is the WordCount algorithm; where it is mesmerizing to perceive that how this strategy is put into effect adopted by distinctive Hadoop peripherals. Consequently, this procedure impersonates a pattern for mapping and reducing the dataset.

Keywords: Big Data, MapReduce, Hadoop, HDFS, RDBMS, WordCount, Pig, Hive.

I. INTRODUCTION

A. Abridgment of Big Data

Starting with big data, what is meant by this term? Why has it initiated a buzz? Why is crucial to have big data as an inherent element in our business? Where has it really become a necessity? Wholly, for these unaccountable queries, it has made the public to become inquisitive. Proceeding with big data, mainly, it is an underlying term for bringing together capacious and heterogeneous datasets that is strenuous to execute using long established relational database management tactics [1].

B. RDBMS: An Impediment in Big Data Technology

The relational database management system, widely known as RDBMS, has been the contemporary medium for miscellaneous database applications for an umpteen decade long. In this, the data is coordinated in a structured configuration formats. On the other hand, ever since the emergence of big data, almost the entire information has been coming out in disordered dimensions. In consequence, the climax reached to the point where the widespread database strategies could get along with the aptness of prodigious database repository. On account of this crisis, RDBMS is presently not adopted as the scalable illustration to meet the requisites of big data [2].

C. Inquisitiveness behind Hadoop in Big Data

Despite, it is agonizing that RDBMS cannot be a service to big data processing, owing to its ebbing technology; nevertheless, it is comforting that Hadoop is not an alteration to the former rather it enhances and strengthens its predecessor's integrity. Over and above, it enumerates specialties to RDBMS attributes to revamp the proficiency of database technology. Moreover, it deciphers diverse datasets queries that the customary database method is incapable of solving [3].

D. Hadoop Epitomized

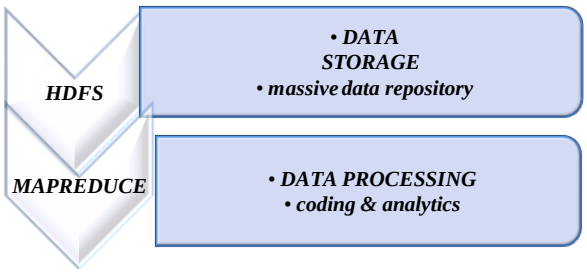

Figure 1: Hadoop's Core Components

Apache Hadoop is open source architecture for drafting and continual processing of vast distributed datasets. Interestingly, it is the most sought-after framework in commercial enterprises as well as in academic communities.

Besides, Doug Cutting and Mike Caferella devised this benchmark in 2005[4].Curiously, it was entitled after his son's toy elephant [5]. Furthermore, Hadoop is composed of 2 pillars: HDFS and MapReduce. In addition to this, it is supplemented by the system of various project libraries such as Hive, Pig, HBase, Zookeeper etc; which enhances its worth and upgrade its utility [6].

E. Wordcount Delineated

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 [7].

II. ELUCIDATION OF BIG DATA AND HADOOP

A. Big Data's 6Vs

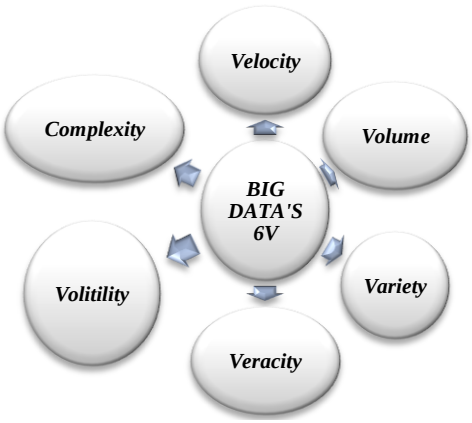

Figure 2: Big Data's 6V's

- Volume: associates with the measurement of quantity or amount of space that an object occupies. In this case, extracting consistent information from the prescribed database techniques is a pretty compounded task, after all, the data is the mountainous circuitous sized, higher than terabytes and petabytes scaled. E.g. Perennially, around 2.5 quintillion bytes are engendered globally [8].
- Volatility: advert of being changeable or mercurial. The big data gives an indication that how long the data would be substantial and how long it is going to be reserved for future purposes.
- Veracity: alludes to conformity to facts. In this, tapping of apposite data is inked for productive outcomes [9].
- Velocity: exemplify the rapid motion or swiftness with which the body moves. In this context, it is really a big defiance for numerous organizations to struggle with the plethora of spatial fractions of data that is streaming with an erratic degree. E.g. Google cultivates almost 24 TB/day; Twitter yields nearly 7 TB/day etc.
- Variety: designate the form of discrepancy. In this, for data, basically, the availability of patterns is in structured, semi-structured or unstructured forms. These diversified formats are the necessity for searching, evaluation, aggregation, accumulation and regulating the data to attain compatible and significant information. E.g.: geospatial data, audio and video files, log files, social media etc [10].
- Complexity: state of being entangled. For instance, in big data, a meandering dynamic correlation prevails due to the alteration of data into distinct phases which is likely to prompt staggering waves. Considering that it is really common for big data to be momentary, hence it is a requisite to coordinate and tune in synchronously the interfaces of data to avoid being whorled out of control. [11].

B. Temporal Pertinences To Big Data

- At the moment, YouTube Videos are being uploaded for 80 hours/minutes.
- Monthly, Google reproduces data approximately about 100 PB.
- The extent of photos uploaded on Facebook is approximately equivalence to 1PB [12].
- The heap of data assembled by smart urban projects in China is 200PB.
- The storage capacity of an American manufacturing company in 2010 was 966PB.
- Earlier, hard copy photographs had a spacious of around 10GB in a Nikon camera. But, currently, the digital cameras are developing images about 50 times more than the antique cameras used to do their tasks and at the moment, it is incrementing day by day [13].

C. Hadoop 's CAP Theorem

- Consistency: concurrent transactions are a must for deposit and withdrawal to and from account in harmony.
- Availability: malleable in producing duplicates of data. If one replica is lost, then its other copies are still approachable for subsequent use.
- Partitioning: segmenting the data into several clones to store in commodity hardware. Normally, by default, 3 copies exist for the customers' feasibility [14].

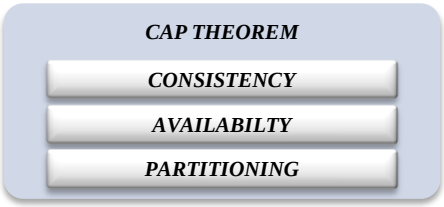

Figure 3: CAP Theorem

D. Exemplification of Hadoop Business Predicaments

- Ads Targeting: Although, all of us know that advertisements are a great annoyance when doing online shopping, however, they reside within us. These ad agencies put their ads on prominent social browsers so that, they can accumulate abundant knowledge to see what we were doing when we were shopping
- Purchaser Analysis: For compiling information, it is profitable to catch on the intrigues of the prevailing customer rather than the new one. Accordingly, one is able to analyze what the consumer was doing before he left the shop mall.
- Recommendation portals: These online vendor websites not only collect database from your own data, but also from users that match your profile, so that these search engines can recommend such web portals that are likely to be handy to you. E.g.: Flipkart, Amazon, Paytm, Myntra etc.
- Shoppers' Vignette: Crucial to pinpoint certain groups of customers having similar preferences and interests in purchasing goods from the markets.
- Vendor Analysis: Market evaluations have been conducted to depict the behavior pattern of the buyers' and augment the product's quality. Feedback surveys have been studied to observe purchaser's attitudes [15].

III. PARTS AND PARCELS OF HADOOP

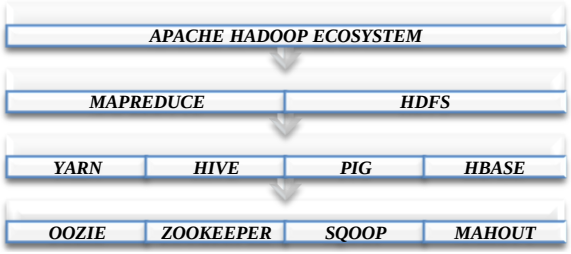

Figure 4: Hadoop's Components

A. Annotation of HDFS

Frequently stated as the Hadoop Distributed File System, pertinently, it is an alternate file system which is designed as a replacement to file system that one is currently using. Interestingly, it got a level of sophistication around data in the file system. By default, there are 4 representations of files that are replicated 3 times in the Hadoop ecosystem on 3 miscellaneous portions of commodity hardware. For this reason, they can be thought as cheap servers. Moreover, the traditional file system is 8k but for HDFS, it is 64k/128k. Illustratively, these file systems devised for big data are undeniably tunable in configuration files to even larger files. Hence, they are often depicted as chunks rather than files [16].The canons of HDFS are:

- Fault Tolerant Reliability: The Hadoop framework is formulated so that the tasks will not rely on each other. Hence, if one node fails, jobs are retried on other robust nodes. As a result, delay in the performance of any task will be prevented. Beside, those deficient nodes are observed, regulated to restart the execution.
- Data locality: The idea of bringing the compute to the data rather than bringing data to the compute is pivotal for identifying with HDFS. Apparently, it has the ability to move the computational node close to where the data is. As a result, Hadoop can schedule tasks pretty near where the data prevails, by which, that node will performs its action on it.
- Lateral Computing: contributes to the whole course of data proceedings across the nodes of clusters using Java based API in a parallel manner. It strives on commodity hardware in case of any hardware deficiency.
- Programming Languages: uses Java, Python and R languages for coding, creating, writing and running jobs for mapper and reducer executables [17].

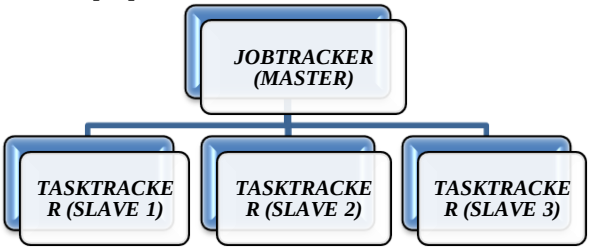

Figure 1: Master Slave Benchmark

The HDFS is modelled as the parallel distributed architecture commonly known as the master-slave benchmark, shown in Fig 3. It exhibits job and task trackers analogous to master- slave terms. The JobTracker acting as the controller supervises and monitor the scheduling done on slaves for map/reduce operations. On the other hand, on each slave node, the Tasktracker executes either the map or reduce task. In the end, these workers will report their status to the master node [18].

B. Denotation of MapReduce 1.0

MapReduce is a programming software ideal, which is fabricated to decipher out the solitary problem. Originally, created out of research at Google, basically, it is the way of thinking to approach the populous amount of data complications. Over and above, it has been used as a radical implementation to yield varied datasets via map and reduce procedures [19]. It is abided by 2 constituents:

- Map part: known as the 1st fraction of MapReduce. In this, when MapReduce runs as a job, the mapper will execute on each node where the data is located. Once accomplished, it will generate a collection of <key/value> pairs on each node.
- Reduce part: In the 2nd portion of MapReduce, the reducer will execute its method on few nodes, but not in all the nodes. This will produce an aggregated cluster of <key, value> duals on these nodes. Further, the yield of this procedure is a sole combined list [20].

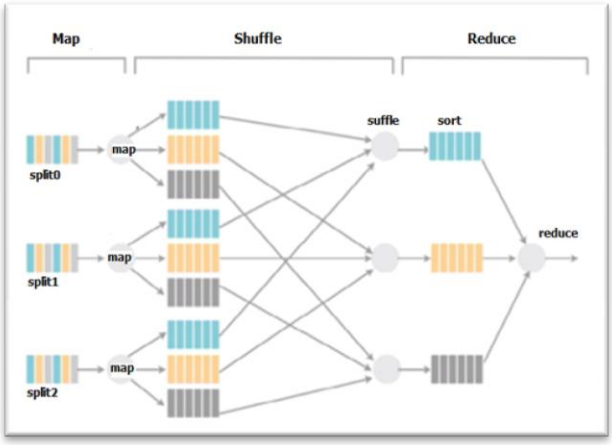

Figure 2: MapReduce Paradigms

In Figure 4, MapReduce prototype is being showcased here as above. In this model, variegated categories of data are unfolded by virtue of multifarious colours. Firstly, under map, 3 splits are displayed as split 0, 1, 2. It is a zero based arrays. In addition to this, the map logic written will be performed physically on each of the map node. Actually, the following intermediate portion is quite obscure and has set the mind to think. How does this list coming out from the Map function get 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. Consequently, all these mechanisms are tuneable; one may accept the defaults or tune in according to one's own convenience. As a matter of fact, there is a default sort mechanism, which can be overridden, along with the copy mechanism termed as shuffle and sort. Moreover, the merge mechanism is accompanied with it, which can go with default or overridden. In the end, the reducer code that is written will then be operated on the nodes that have been selected to be reducer [21]. The deprivations of MapReduce 1.0 are:

- Tortuous coding of MapReduce jobs: Not enough trained developers are accessible, who can figure out the intricacies of MapReduce codes to the level that the businesses demand. Mostly, these jobs are not capable in real world scenarios, especially in big data problems, where the organizations are trying to get to the bottom of it
- Missing Enterprise Features: Hardly, much considerations have been ponder over on enterprise features such as security, high availability etc
- Only Batch Processing: Momentarily, it isn't constructed for any segment of interactive data query. From a realistic field of vision, for the aftereffects to come, it is compulsory to hang around for min/hrs. Despite the fact that it is applicable for some organizations; however, more and more users prefer the copious volume of MapReduce information at the rate of relational datasets [22].

C. Contemplation of YARN/MapReduce 2.0

YARN stands forYet Another Resource Navigator. Why is a requisite for YARN? YARN presents itself an abstraction mantle between MapReduce and HDFS. While, it may sound lucid, however, it brings forward the other processing to present itself on the apex of HDFS file system other than those classical batch processing. On top on that, as the data contents are escalating exponentially, the corporations desire to have a low-cost cache of HDFS, but in contrast, they are not fast processing. Beyond that, it splinters the subsistence of JobTracker roles into resource and life cycle management in view of better control and management. Additionally, it enhances improved scalability through distributed job life

cycle and assists multiple MapReduce API in single mode cluster. Over and above, inclusive attributes are appended for greater security, together with easy availability scenarios. Furthermore, in this process, advanced flexibility in execution and control are likely to be reachable via map and reduce configuration methods by means of parameters. This symbolizes that in the interior of mapper and reducer, instrumentation, information gathering as well as threshold setting are implemented. Lastly, it can be utilized to distribute jobs for read-only purposes [23].

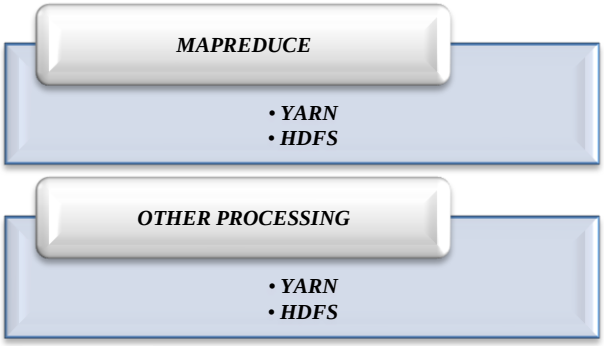

Figure 3: Yet Another Resource Navigator (YARN)

D. Apposite of Hive

SQL like query that generates MapReduce code signifies Hive. HQL language is put into service, which is 60-70 % compliance with ANSI- SQL. Its origin is at Facebook, where it was designed as the set of libraries. Performance differences are something to be aware of that Hive is batch but not interactive. So this formalizes that there will be latency in the analysis. Depending upon the side of data query, it may take minutes or even hours before you get the result [24].

E. Obigatory of HBase

Remarkably, in many instances, HBase is often conjoined with Hive as an abstraction layer devised on the pinnacle of HDFS data. It is established as a wide-column store with key together with one or more values. Similar to DDL statements in ANSI-SQL, it is a wide column NoSQL database by virtue of Hive libraries using create table statement. Thus, it is can be referred to as the subset of relational world. Once the table is created, normal execution process is then queried with Hive These Hive libraries are integrated with HBase composed of HQL languages. Commercial distributions such as HortanWork, Cloudera etc are applied for production [25].

Table I Hive Syntax

MYSQL	**HIVE**
USE DATABASE;	USE DATABASE;
SHOW DATABASE;	SHOW DATABASE;
SHOW TABLES;	SHOW TABLES;
DESCRIBE TABLE;	DESCRIBE(FORMATTED|EXTENDED)TABLE
CREATE DATABASE db_name;	CREATE DATABASE db_name;
DROP DATABASE db_name;	DROP DATABASE db_name (CASCADE);

F. Pig Scrutiny

ETL library for Hadoop that propagates MapReduce jobs entitled Pig. Evolved at Yahoo, ETL symbolizes Extract Transform and Load. Since these Pig libraries use Pig Latin language therefore, the yahoo clan bestowed with good humoured titles to Hadoop components. Moreover, it wields commands like Oint and Grunt [26]. Now the question is: when to use Pig? Mostly, it is at that period when ETL tasks are attainable, and then Pig is put into action as an abstraction language because it is especially written to bring about the consecutive processes easier:

- Clean data: E.g.: if there are few portions of sensor data or some incorrect data along with some incomplete data that one is looking out for, then one chooses to discard it.
- Process data: E.g. in big data scenarios, the data capacity can be exponentially greater than one is used to dealing with.
- Transform Data: E.g. If a series of text coming in, one might aspire to fissure it into words analogous to the standard classic word count [27].

LOAD <file>
FILTER, JOIN, GROUP BY, FOREACH, and GENERATE <values>
DUMP <to screen for testing>
STORE <new file>

How Does Pig Work? Accordingly, this is the common programming paradigm in Pig ETL process flow. Presently, an ETL process flow exists along with keywords are in capitalized forms. Straightaway, load some file either from standard or cloud based system, or load it from HDFS. After that, some operations will be performed. While working with Pig, a rich set of functions associated with Pig Latin language are FILTER, JOIN, GROUP BY, and FOREACH. However, a special attention is called upon on the GENERATE function, since it is widely used in ETL Pig workflows. Further, GENERATE allows process to work on intermediate data to produce new values. Consequently, an output will come through. One can either dump it to the screen for testing, or look at it and see if the process is correct or not. In the end, whenever, one is satisfied with the output, one can deposit it to a new file location, or can store as new files into HDFS file system or cloud based system [28].

G. Addendum of Hadoop Libraries

- Zookeeper: Defined as the centralized service for Hadoop configurations. Subsequently, it creates an ensemble of various distributed applications. Moreover, it is captioned as the distributed in-memory computation. Lastly, its syntax is locking, synchronization, queues along with several types of utility objects. This is due to synchronization in the state of data whenever the data is shared between client nodes E.g.: ad serving [29].
- Sqoop: Termed as the command line utility for transferring data between RDBMS system and Hadoop. Besides, it can either import data from these databases into Hadoop or export data from Hadoop back to these systems. Further, one can load that data not only in HDFS file system but also directly into Hive / HBase tables. Chiefly, it provides connectors for Oracle, SQL Server and other relational databases [30].
- Oozie: Phrased as the workflow scheduler library for Hadoop jobs. It can schedule recurrence jobs i.e. can aggregate numerous types of datasets. Based on time, it can schedule data changes or new data arrival around HDFS cluster. In addition, native oozie is written with a command line closely with jobs bundles that are related to group of jobs. Its syntax is stop, start, suspend, resume info etc. However, its core syntax is XML. Nevertheless, one can't write job scheduling in XML [31].
- Mahout: Denoted as the very sophisticated special purpose library for common machine learning algorithms. It is used to perform predictive analysis on Hadoop scale under R language. Seeing this, varied data algorithms are set in motion. E.g. recommendation (predicting a value in Pandora likelihood), clustering (grouping items together in Google news), and classification (classify in Spam ID). Besides, if one wants to work with Hadoop cluster, one can add this library to the java project via java classpath [32].

IV. COMPENDIUM OF WORDCOUNT

The Hello World of MapReduce program is the Word Count. It originates from Google attempting to elucidate the dataset puzzles by enumerating all the words on the Web. It is the de facto canonical for embarking with Hadoop programming. It takes an input as some text and delivers a list of words and start counting on them [33].

A. Pseudocode Strategy

Consequentially, given below is the Pseudocode of Word Count [34]:

mapper (filename, file_contents):
for (word w: file_contents)
count c = emit (w, 1);
Reducer (word w, values):
sum=0 ;
for(value v: values)
sum= sum + v;
count c = emit (word, sum)

The pseudocode of MapReduce comprises of the mapper and reducer. Initially, the mapper has filename and file_contents and for-each loop to iterate over the information. The word is emitted which has a value 1. Primarily, splitting happens. Later, in the reducer, it takes the output from the mapper that consists of a list of keys and values. In this case, the keys are the words and value is the total manifestation of that word. In simpler terms this means, it is the words and how many of them. 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 [35].

B. Algorithm Tactics **[36]**

- Map step: In this step, it gathers key/ value dual of input data and give outcome in the form of intermediate list of key/ value doublet.

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

- Reduce step: In this step, after shuffling/sorting, the turnout intermediate key/value combo is passed through the reduce function where these values are merged together to form a smaller sets of values.

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

C. Illustration [37]

Figure 4: MapReduce Instance

Let's ponder over estimating the appearance of each word from the large stanza.
Quiz: there are 2 input documents and on them, MapReduce tasks need to be carried out.

D. WordCount with Hive

CREATE TABLE WordCount AS
SELECT word, count (1) AS count
FROM (SELECT EXPLODE (SPLIT (LCASE (REGEXP_REPLACE (line,'L\\ p {PUNCT},\\p {CNTRL}]',")),'"))
AS word FROM myinput) words
GROUP BY word
ORDER BY count DESC, word ASC;

This is a partial instance via Hive. Firstly, create a table with 2 columns, word and count. Then, design it via FROM SELECT EXPLODE SPLIT, LOWERCASE, and REGULAR EXPRESSION REPLACE. While, REGEXP for line, Punct and Cntrl split the words and since they are splitting on spaces, therefore, it is not a sophisticated regular expression. Probably, some errors would be there, plus longer regular expressions need to be written, considering, they are easy to write yet not simple to read. Soon after that, split the AS word FROM myinput. Seeing that, myinput would be the original table, GROUP it BY word and subsequently aggregate it. After all, one requires summating up the number of occurrences of a particular word. Thenceforth, ORDER it BY count, so accordingly, put the computed words using descending rule together with the word in ascending arrangement. However, the tricky part is REGEXP that indicates that Hive is admittedly not a right language to use for these kinds of problems. Since, this is parsing problem, hence, this is done through OOP codes or higher languages. Generally, by this reason, one shouldn't use Hive to execute WordCount [38].

E. WordCount via Pig

Let's reckon with WordCount/Hello World for Hadoop ecosystem using Pig.

```
lines = LOAD '/ user/ Hadoop/HDFS_File.txt' AS (line: chararray):
words = FOREACH lines GENERATE FLATTEN (TOKENIZE(line)) as word;
grouped = GROUP words BY word;
WordCount = FOREACH grouped GENERATE group, COUNT (words)
DUMP WordCount;
```

To begin with, aliases concepts are used. Coming up aliases are: variables, lines, words, grouped as well as WordCount. Afterwards, equate to values that are induced by Pig. For this reason, lines are matched with calling the LOAD keyword. After loading some text, preferably a character array, which is a data type in Pig. In view of this data type one can use the function to process it. Considering the words and variables, FOREACH keyword is used against the line input, together with several functions. Later, create a new output by flattening (FLATTEN) similar to grouping it in a bag, and then TOKENIZE it. Subsequently, tokenizing the line is systemized that will discord the text stanza into words. Thereafter, one receives the word and grouped variable that will accumulate the words by word. On the account of WordCount for each group, a group is displayed. Hence, counting the words is rendered and thence dumps the results. Actually, this process performs in backward direction for MapReduce i.e. first reducer then mapper. Over and above, these key values engendered will be collected to become reducer. Finally, the words, variable and grouped variable will give rise into mappers [39].

V. PROSPECTIVE TECHNOLOGY JUST AROUND THE CORNER

It is interesting that despite knowing that Hadoop ecosystem is fast, hot and growing, nonetheless, the technology is beyond 10 years old. The Google community, who developed the technology on which the Hadoop foundation is laid, has actually augmented their own implementation of Hadoop with the next generation products. The coming up product which has created a stir is BigQuery. Wholly, the goal with this technique is to process petabytes of data in seconds with ANSI compliant SQL. Primitively, it can be illustrated as the Google search engine. On the other hand, however, BigQuery is really far beyond the reach of Hadoop ecosystem. In truth, it is literally hard to comprehend what it actually does. For this reason, more knowledge and further research are required to be studied over this product [40].

VI. CONCLUSION

To summarize, the recent literatures of various architectures have been observed plus researched upon which assisted in the reduction of big data to simple data. The big data composed of immense knowledge in gigabytes or megabytes. Besides, its postulates have been discussed in detail. Furthermore, the concept of Hadoop, its use in big data has been analyzed. In addition, its major components HDFS and MapReduce along with its other set of libraries like Pig, Hive etc. 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, WordCount is represented as the principle for mapping and reducing the datasets.

ACKNOWLEDGEMENT

Priyaneet Bhatia thanks Professors Bhawna Mallick, 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] Radhika M. Kharode, Anuradha R. Deshmukh, “Study of Hadoop Distributed File system in Cloud Computing”, International Journal of Advanced Research in Computer Science and Software Engineering ,www.ijarcsse.com, Maharashtra, India ,Vol.5, pp.990-993, Jan 2015.

[2] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, “The Google File System”, Google, New York, USA, pp.19–22, October 2003.

[3] Puneet Singh Duggal, Sanchita Paul, “Big Data Analysis: Challenges and Solutions”, International Conference on Cloud, Big Data and Trust, Nov 13-15 2013.

[4] Konstantin Shvachko, Hairong Kuang, Robert Chansler, “The Hadoop distributed File System”, IEEE, Yahoo, California, USA, 2010.

[5] J. Venner, “Pro Hadoop”, Apress, June 22, 2009.

[6] T. White, “Hadoop: The Definitive Guide”, O’Reilly Media, Yahoo Press, June 5, 2009

[7] Mahesh Maurya, Sunita Mahajan, “Comparative analysis of MapReduce job by keeping data constant and varying cluster size technique” ,Elseveir, Mumbai, India, pp.696-701, May 2011.

[8] Jaseena K.U and Julie M. David, “Issues, Challenges, And Solutions: Big Data Mining”, NeTCoM, CSIT, GRAPH-HOC, SPTM–2014, Computer Science & Information Technology (CS & IT), M.E.S College, Cochin, India, pp.131–140, 2014.

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

[10] Min Chen, Shiwen Mao, Yunhao Liu, "Big Data: A Survey", Springer, New York, pp.171-209, 22 January 2014.

[11] Ebrahim Sahafizadeh, Mohammad Ali Nematbakhsh, "A Survey on Security Issues in Big Data and NoSQL", Advances in Computer Science: an International Journal, ACSIJ, Vol.4, www.acsij.org, University of Isfahan, pp.68-72, July 2015.

[12] Prajesh P Anchalia, "Improved MapReduce k-Means Clustering Algorithm with Combiner", UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, IEEE, Bangalore, India, pp.385-390, 2014.

[13] Apache Hadoop, http://hadoop.apache.org

[14] Mahesh Maurya, Sunita Mahajan, "Performance analysis of MapReduce programs on Hadoop Cluster", IEEE, pp.505-510, 2012.

[15] Sheetal Singh, Vipin Kumar Rathi, Bhawna Chaudhary, "Big Data and Cloud Computing: Challenges and Opportunities", International Journal of Innovations in Engineering and Technology (IJIET) , New Delhi, India, Vol.5, pp.117-125, 4 August 2015.

[16] Clark Bradley, Ralph Hollinshead, Scott Kraus, Jason Lefler, Roshan Taheri, October 2013. "Data Modeling Considerations in Hadoop and Hive", SAS Institute Inc., Cary, NC, USA, pp.1-26.

[17] Nilam Kadale, U. A. Mande, "Survey of Task Scheduling Method for MapReduce Framework in Hadoop", International Journal of Applied Information Systems (IJAIS) , Foundation of Computer Science FCS, New York, USA, 2nd National Conference on Innovative Paradigms in Engineering & Technology (NCIPET) , www.ijais.org, 2013.

[18] Silky Kalra, Anil lamba", A Review on Hadoop MapReduce: A Job Aware Scheduling Technology", International Journal of Computational Engineering Research (IJCER), India, Vol.04, pp.36-40, May 2014.

[19] A community white paper developed by leading researchers across the United States, "Challenges and Opportunities with Big Data Hadoop and Pipelined Map Reduce", International Journal of Computational Engineering Research, Vol.3,Issue 12

[20] Rabi Prasad Padhy, "Big Data Processing with Hadoop-MapReduce in Cloud Systems", International Journal of Cloud Computing and Services Science (IJ-CLOSER), Institute of Advanced Engineering and Science, Oracle Corp, Bangalore, India,Vol.2, pp.16-27, Feb 2013

[21] Santhosh Voruganti," Map Reduce a Programming Model for Cloud Computing Based On Hadoop Ecosystem", International Journal of Computer Science and Information Technologies (IJCSIT),www.ijcsit.com,Hyderabad,India,Vol.5, pp.3794-3799, 2014.

[22] Vasiliki Kalavri, Vladimir Vlassov, "MapReduce: Limitations, Optimizations and Open Issues", IEEE, KTH, The Royal Institute of Technology Stockholm, Sweden, pp.1031-1038, 2013.

[23] Poonam S. Patil, Rajesh. N. Phursule, "Survey Paper on Big Data Processing and Hadoop Components", International Journal of Science and Research (IJSR), Pune, India, Vol.3, www.ijsr.net, pp.585-590, October 2014.

[24] Parmeshwari P. Sabnis, Chaitali A.Laulkar, "Survey Of MapReduce Optimization Methods", International Journal on Advanced Computer Theory and Engineering (IJACTE), Pune, India, Vol.3, pp.15-19, 2014.

[25] Remya Panicker, "Adoption of Big Data Technology for the Development of Developing Countries", Proceedings of National Conference on New Horizons in IT - NCNHIT, pp.219-222, 2013.

[26] Ronald C Taylor, "An overview of the Hadoop/ MapReduce/ HBase framework and its current applications in bioinformatics", Bioinformatics Open Source Conference (BOSC), Boston, USA, pp.1-6, 9-10 July 2010.

[27] Manisha Sahane, Sanjay Sirsat, Razaullah Khan, "Analysis of Research Data using MapReduce Word Count Algorithm", International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), India, Vol. 4, pp.184-187, May 2015.

[28] Ms. Vibhavari Chavan, Prof. Rajesh. N. Phursule, "Survey Paper on Big Data", International Journal of Computer Science and Information Technologies (IJCSIT), www.ijcsit.com, Pune, Vol.5, pp.7932-7939, 2014.

[29] Shreyas Kudale, Advait Kulkarni, Leena A. Deshpande "Predictive Analysis Using Hadoop: A Survey", International Journal of Innovative Research in Computer and Communication Engineering, www.ijircce.com, VIIT, Pune, India, Vol.1, pp.1868-1873, October 2013.

[30] Apache Hadoop, http://en.wikipedia.org/wiki/Apache_Hadoop

[31] P.Sarada Devi, V.Visweswara Rao, K.Raghavender, "Emerging Technology Big Data Hadoop Over Datawarehousing, ETL", IRF International Conference, India, pp.30-34, 21 Sept 2014.

[32] Wullianallur Raghupathi and Viju Raghupathi, "Big data analytics in healthcare: promise and potential", Health Information Science and Systems, Graduate School of Business, Fordham University, New York, USA, pp. 1-10, 2014.

[33] MapReduce tutorial, https://Hadoop.apache.org/docs/r1.2.1/mapred_tutorial. html

[34] Suman Arora, Dr.Madhu Goel, "Survey Paper on Scheduling in Hadoop", International Journal of Advanced Research in Computer Science and Software Engineering, Vol.4, May 2014.

[35] B.Thirumala Rao, Dr. L.S.S.Reddy, "Survey on Improved Scheduling in Hadoop MapReduce in Cloud Environments", International Journal of Computer Applications, Vol.34, November 2011.

[36] Vishal S Patil, Pravin D. Soni, "Hadoop Skeleton & Fault Tolerance in Hadoop Clusters", International Journal of Application or Innovation in Engineering & Management (IJAIEM), Vol.2, pp 2319 – 4847, February 2013.

[37] Amogh Pramod Kulkarni, Mahesh Khandewal, "Survey on Hadoop and Introduction to YARN", International Journal of Emerging Technology and Advanced Engineering, www.ijetae.com, Vol.4, issue 5, May 2014,

[38] Aditi Jain Manju Kaushik, "Performance Optimization in Big Data Predictive Analytics", International Journal of Advanced Research in Computer Science and Software Engineering, www.ijarcsse.com, JECRC University, India, Vol.4, pp.126-129, August 2014

[39] Hive HBase Integration project home page, Available at http://wiki.apache.org/Hadoop/Hive/HBaseIntegration].

[40] Munesh Kataria, Ms. Pooja Mittal, "Big Data and Hadoop with Components like Flume, Pig, Hive and Jaql", IJCSMC, Vol.3, pp.759-765, July 2014.

