ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 3, Issue 4 , April 2016

Proficiency of Hadoop Integrants in Virtualization Milieu by wielding Word Count Doctrine

Priyaneet Bhatia, Siddharth Gupta

M.Tech Student, Department Of Computer Science Engineering, Galgotia College of Engineering and Technology (GCET), A.P.J. Abdul Kalam Technical University, Greater Noida, UP, India,

Assistant Professor ,Department Of Computer Science, Maharani Girls Engineering College (MGEC), Rajasthan Technical University, Jaipur, Rajasthan, India,

ABSTRACT: Ubiquitously, Big Data is contemplated as a cardinal imperative quantum in the world. Prodigious datatypes commencing from Terabytes to Petabytes are consumed persistently. But, to reserve these databases amplitude is an enervating quest. Although, the predominant database tactics are the indispensable facet for the repository of convoluted and inestimable datasets, nonetheless, it is at the hand of Apache Hadoop that is able to amalgamate the onerous information in an adept style. Furthermore, the Hadoop benchmark is utilized which has copious constituents. Its preeminent segments are HDFS and MapReduce. Substantially, HDFS is open source data stock architecture with fault tolerant competence. In essence, MapReduce is the programming paradigm on which quarry of pragmatic knowledge is excavated. Besides, the auxiliary components of Hadoop are cogitated at length. Momentarily, the outstanding enchantment is the WordCount algorithm where it is really charismatic to be appraised of, that how its proposition is accomplished on distinctive Hadoop modules. Consequently, this postulate impersonates a canon for mapping and reducing the dataset beneath the sustenance of Cloudera distribution CDH 4.7 in UNIX operating system, Ubuntu 14.5 LTS.

KEYWORDS: Big Data, Hadoop, HDFS, MapReduce, RDBMS, Pig, Hive, Word Count, Cloudera CDH, UNIX OS Ubuntu.

I. INTRODUCTION

Preliminarily with Big Data, it is the core idiom for bringing simultaneously the prodigious and assorted datasets, which is normally backbreaking to execute using customary relational database management strategies [1]. The relational database management schemes, extensively branded as RDBMS, have been the fashionable habitat for diverse database applications for innumerable period. In this, the data is synchronized in a structured pattern. Yet, ever since the dawn of Big Data, mostly, the unabridged information is accessible in topsy-turvy proportions. In consequence, it has hit the peak where the pervasive database tactics could not get alongside with the pertinent of capacious database repertoires. In view of this predicament, RDBMS is contemporary not embraced as the scalable elucidation to meet the rudiments of Big Data. What's more, it is really heartbreaking to be acquainted with the reality that RDBMS cannot be a service to Big Data evolution, owing to its dwindling technology [2]. Nonetheless, it is comforting to discern that Hadoop does not metamorphose the predecessor; however, it boosts and fortifies its antecedent's probity. Over and above, it appends attributes to RDBMS specialties to give a face-lift to the adeptness of database technology. Moreover, it decodes miscellaneous datasets queries where the long-established database techniques were inept to decipher [3]. Originally devised at Google, Apache Hadoop was devised by Doug Cutting and Mike Caferella in 2005.Curiously; it was captioned after his son's toy elephant [5]. Furthermore, Hadoop is poised of 2 masts: HDFS and MapReduce. In addition to this, it is complemented by a variety of project libraries such as Hive, Pig, HBase, Oozie etc; which augments its merit and promotes its effectiveness [6].

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 3, Issue 4 , April 2016

II. RELATED WORKS

Analogous to [7], UNIX shell commands are used to unearth LinkCount application under the assist of WordCount algorithm on the Hadoop cluster. Big data is extracted from Wikipedia to analyse the WordCount MapReduce application via 5-6 nodes on Hadoop cluster. The LinkCount is used to calculate the number of links accessible in the file. However, in reality the Linux commands are quite complicated for the clients to peerform the WordCount application, since it is obligatory to ponder over UNIX in order to perform MapReduce application. Subsequently, the user friendly GUI platform is crucial for execution of the WordCount MapReduce application.

- The experimentation with MapReduce applications was conducted via Hadoop cluster.
- The Hadoop infrastructure composed of one cluster having 6 nodes geographically disseminated in single lab. For the string of experiments, the authors used Intel Core2@93GHZ 4 CPUs and 4GB RAM for each node via TCP sockets of 100 MB/s, Ubuntu 11.10 and Sun Java JDK 1.6.0.
- Firstly, they used WordCount illustration to construe text files and enumerate how often the words emerge. Both the input and output were text files where each line contained a word and count, separated by a tab.
- Secondly, LinkCount model was also presided to comprehend the text files and tally how recurrently the links transpired. Identical procedure was operated for LinkCount testing as in WordCount.
- To perform the paradigm, the authors used the command syntax such as bin/hadoop jar hadoop-*-examples.jar wordcount [-m<#maps>] [-r <#reducers>] <in-dir> <out-dir>, bin/hadoop dfs copyFromlocal<local-dir><HDFS-dir>.
- According to the WordCount assessment for data size 140 MB, if the magnitude of nodes was more and data capacity was fewer, subsequently it would construct the overhead predicament. Besides, if the nodes aptitude were fit to data enormity i.e. if MapReduce job necessitates 4 nodes which were passable to achieve the inclusive program, then there was no requirement to oscillate the cluster range.

Corresponding to [8], the research objective was to mull over Hadoop and associated technologies correlated with MapReduce. Hence, the university research dataset was used as a scrutiny to recognize the purposeful research vicinity in Zoology and Botany sectors.

- For extensive research, to get the desired results, the experimental setup of MapReduce WordCount algorithm was based on the perception of university research dataset.
- Consequently, the research study required pertinent technique of data analysis under MapReduce WordCount algorithm. Statistics compilation was equipped primarily from website of Dr. Babasaheb Ambedkar University.
- The research dataset enclosed records from 1980 to 2010 stored on the local file system. A sample mass of Zoology dataset was 577, while Botany dataset included 335 samples. Text cleaning was prepared by eliminating corrupted, erroneous, misleading and empty fields. It was then copied from local file system to HDFS.
- The authors used the WordCount module in the Java programming language and then JAR file was uploaded to the single node storage. The data was tokenized using the MapReduce algorithm in order to ascertain the interested area of research. Afterwards, only tokens/data that were most frequent and imperative to the task were selected. Top 14 keywords were selected from each dataset which were having paramount occurrence. After processing the data, the yield was accumulated on HDFS and copied back to the local file system. The upshot was the catalogue of words with the computation of manifestation of each word.
- In their analysis, the authors analyzed that in Zoology province, mainstream of persistent research vicinity was parasites and fishes. Further, in Botany speciality, essentially research meditation was on airospora and fungi.

III. PROPOSED WORKS

A. Personification of WordCount Algorithm

The Hadoop shell commands are relatively strenuous to accomplish the WordCount algorithm using the Linux terminal command lines. Hence, virtualization software known as Cloudera CDH (Cloudera Distributed including Hadoop) is used as a hosted virtual machine to implement various libraries of Hadoop on a single node setup [9]. The WordCount is the Hello World of MapReduce program. Originate at Google; it attempts to illustrate the dataset queries by estimating the number of occurrences of each word in a large collection of documents. Mainly, it is the standard established technique for starting with Hadoop programming. It acquires an input as a huge quantity of datasets (megabytes MB or gigabytes GB) and delivers a list of words and start

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 4 , April 2016

counting on them [10]. Using WordCount MapReduce application, the diverse libraries of Hadoop such as Hive, Pig, Oozie etc are conducted against the big data for evaluation of the execution time in seconds. The input is the IBM datasets and output is the catalogue of words occurring in the massive dataset.

B. Algorithm Proposition [11]

- Map step: In this stride, it congregates key/value duals of input data and bestows outcome in the shape of intermediary list of key/ value combo.

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

- Reduce step: In this step, after shuffling/sorting, the upshot transitional key/value doublets is passed in the course of the reduce task where these values are amalgamated to form a smaller sets of values.

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

C. **Elucidation** [12]

Let's mull over while guesstimate the manifestation of each word from a huge verses.
Quiz: there are 2 input documents and MapReduce functions are necessitated to carry out their errands

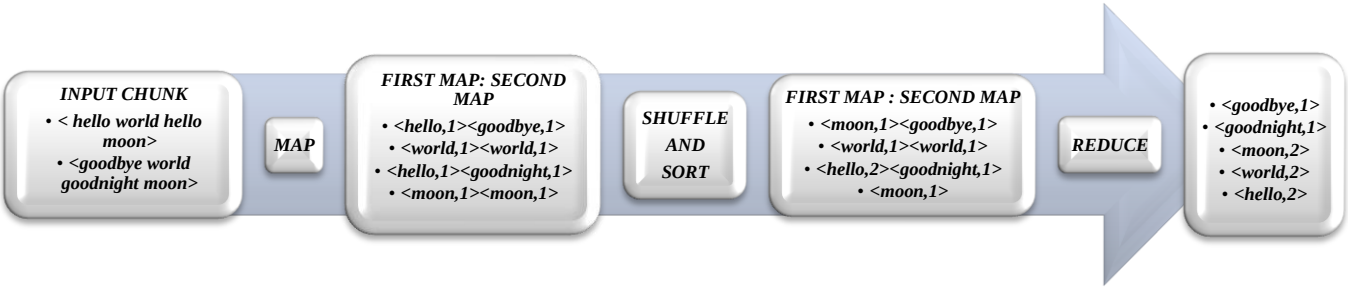


Figure 1: MapReduce Illustration

D. **Obligatory** [13]

- UNIX operating system: crucial to have of 2 operating systems: Ubuntu plus Windows. Frequently, Cloudera CDH performs on UNIX OS in the compatible situation exclusive of plummeting the alacrity of CPU processor, while, permitting additional programs to run at their own pace. Current version: Ubuntu 14.04/15.10 LTS.
- Cloudera CDH and Cloudera Manager: These 64-bit VMs entail 64-bit host OS and a virtualization product that can support 64-bit guest OS. Its contemporary version: CDH4 (4.7.0)/CDH 5 (5.4/5.5). The Cloudera comprises of CDH and Cloudera Manager. In its installed condition, it encloses Hue, Eclipse, and UNIX Terminal for Hadoop shell commands. Cloudera QuickStart VMs are accessible via Zip archives in VMware, KVM, and VirtualBox layouts.
- RAM requisites: at least 4GB or more is essential for RAM quantity in VM, but ought not to have fewer than 4GB. Thus, it is healthier to install extra 4GB RAM from the vendors so that this software can perform in compatible state.
- Oracle Virtualbox: Hadoop virtual platform on which Cloudera CDH executes its actions. Preferably, virtualbox should be downloaded for UNIX OS.
- Accounts: Once, VM commenced, automatically one is logged in as Cloudera user. Hue and Cloudera Manager use the same credentials i.e. username: cloudera, password: cloudera.
- HDFS in single setup node (by default)
- Methodical familiarity with Java codes and UNIX shell commands.

E. WordCount with Hive

SQL like query that engender MapReduce codes implies Hive. HQL language is put into service, which is 60-70% compliance with ANSI-SQL. Its origin is at Facebook, where it was premeditated as the position for libraries. Performance differences need to be conscious of, that Hive is batch excluding not interactive. So this sanctifies that there will be latency in the scrutiny. Depending upon the side of data query, it may take min/hrs to acquire the outcome [14].

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 4 , April 2016

F. WordCount with Pig

ETL library for Hadoop that propagates MapReduce jobs entitled Pig. Evolved at Yahoo, ETL symbolizes Extract Transform and Load. The pig libraries use pig Latin language. Now the real query is 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 [15].

G. WordCount with Oozie

It is phrased as the workflow scheduler system 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. One should use Oozie to set up workflows that execute MapReduce jobs and to set up a coordinator that manages workflows. The main content is the job properties which are needed to be added by using property names and values: mapred.mapper.class, mapred.reducer.class, mapred.output.dir and mapred.input.dir. The jar file is uploaded with the fully-qualified path to the JAR file with the classes that implement the mapper and reducer functions [16].

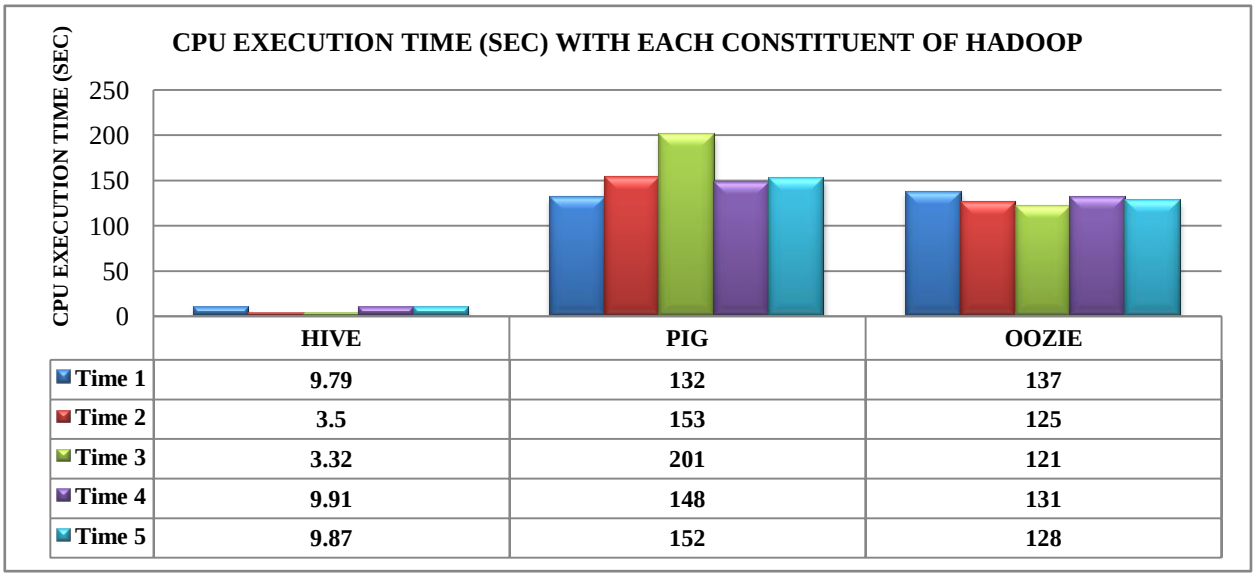
Table 1 : Wordcount Algorithm with various Hadoop Libraries in Cloudera CDH

HADOOP LIBRARIES	SYNTAX	WORDCOUNT WITH LIBRARIES	
HIVE	*CREATE TABLE table_1 (<column_name datatype>)* *LOAD DATA INPATH '<.csv/txt file>' OVERWRITE INTO TABLE table_1* *CREATE TABLE table_2 AS* *SELECT word, count(*) as count* *FROM (SELECT EXPLODE SPLIT (column_name, ')) AS word FROM table_1) table_2* *GROUP BY word* *ORDER BY word ASC , count DESC*	*CREATE TABLE american_time (col_1 string, col_2 int, age col_4 string, col_5 string, col_6 int, col_7 int, col_8 int, col_9 int, col_10 int, col_11 int, col_12 int, col_13 int, col_14 int, col_15 int, col_16 int, col_17 int)* *LOAD DATA INPATH '/user/cloudera/WA_American_Time_Use_Survey-lite.csv' OVERWRITE INTO TABLE american_time* *CREATE TABLE word_count AS* *SELECT word , count(*) as count* *FROM(SELECT EXPLODE SPLIT(col_4, ')) AS word FROM american_time) word_count* *GROUP BY word* *ORDER BY word ASC, count DESC*	
PIG	*lines= LOAD <file>AS (column_name: datatype);* *words=FOREACH LINES GENERATE FLATTEN (TOKENIZE(column_name)) AS word;* *grouped=GROUP words BY word;* *WordCount=FOREACH GROUPED GENERATE GROUP, count(words);* *DUMP <to screen for testing>* *STORE WordCount INTO <new file>*	*lines=LOAD '/user/cloudera/WA_American_Time_Use Survey-lite.csv' AS (line:CHARARRAY);* *words=FOREACH LINES GENERATE FLATTEN (TOKENIZE(line)) AS word;* *grouped=GROUP words BY word;* *WordCount=FOREACH GROUPED GENERATE GROUP, count(words);* *STORE WordCount INTO '/user/cloudera/training/ american_output;*	
OOZIE	*Name : Name of the Oozie worflow* *Description Description of the workflow* *Jar path: fully qualified path to the jar file with the classed that implement the mapper and reducer functions*	*Name: MapReduce_Job_Design* *Description: MapReduce job design action* *Jar path: /user/cloudera/oozie/MapReduceJob/oozie-examples-3.3.2- mapr.jar* *Job Properties:*	
		Property name	*Value*
		mapred.mapper.class	*org.apache.oozie.example. SampleMapper*

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 4 , April 2016

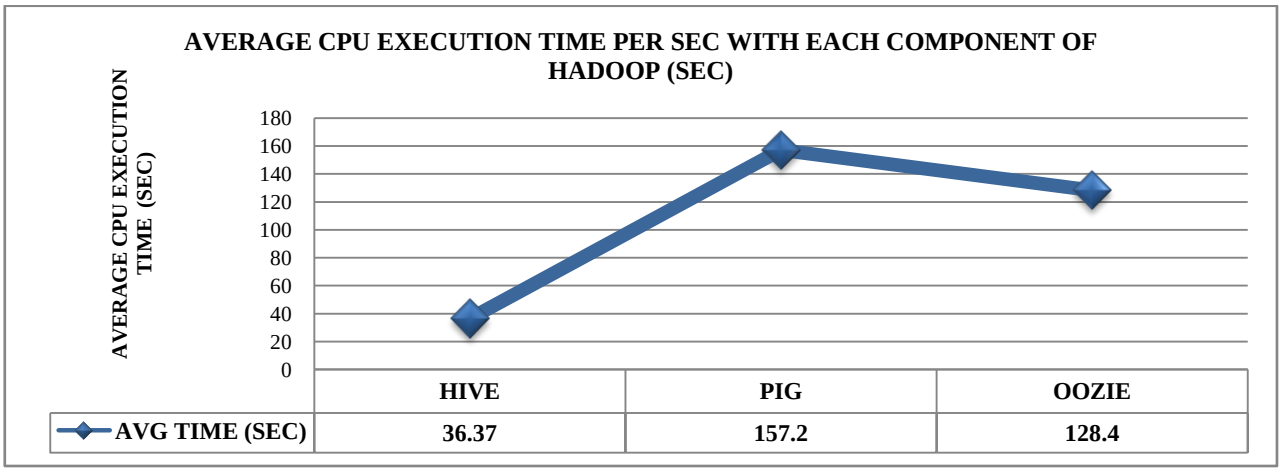
		mapred.reducer.class *mapred.output.dir* *mapred.input.dir*	*org.apache.oozie.example. SampleReducer* */user/cloudera/oozie/ MapRedueJob/ american_output* */user/cloudera/oozie/MapReduceJob/WA_American_Time_Use_Survey-lite.csv*

IV. EXPERIMENTAL RESULTS



	HIVE	PIG	OOZIE
Time 1	9.79	132	137
Time 2	3.5	153	125
Time 3	3.32	201	121
Time 4	9.91	148	131
Time 5	9.87	152	128

Figure 2: Results of MapReduce Application via WordCount Algorithm



	HIVE	PIG	OOZIE
AVG TIME (SEC)	36.37	157.2	128.4

Figure 3: Results of Average CPU time with Hadoop components

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 4 , April 2016

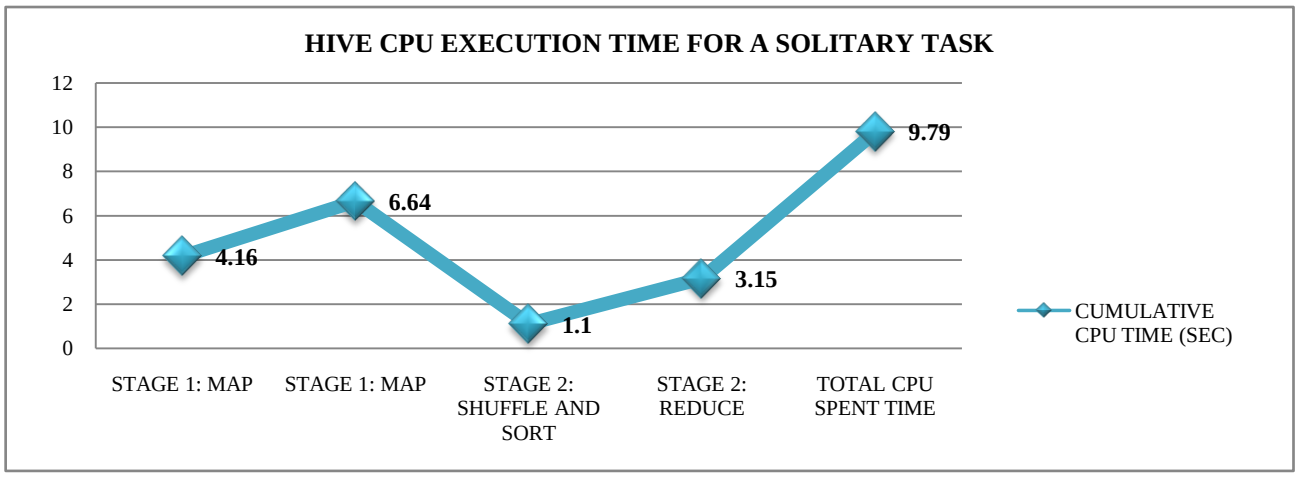


Figure 4: Hive CPU Execution Time for a single task

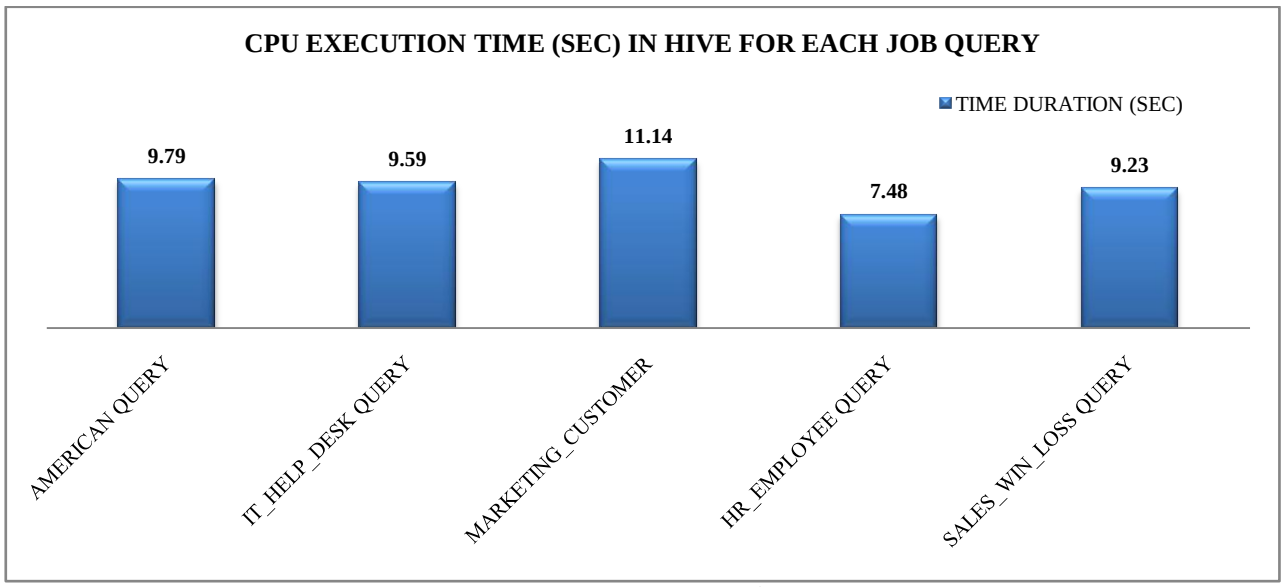


Figure 5: CPU Execution Time (sec) in Hive for each Job Query

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 4 , April 2016

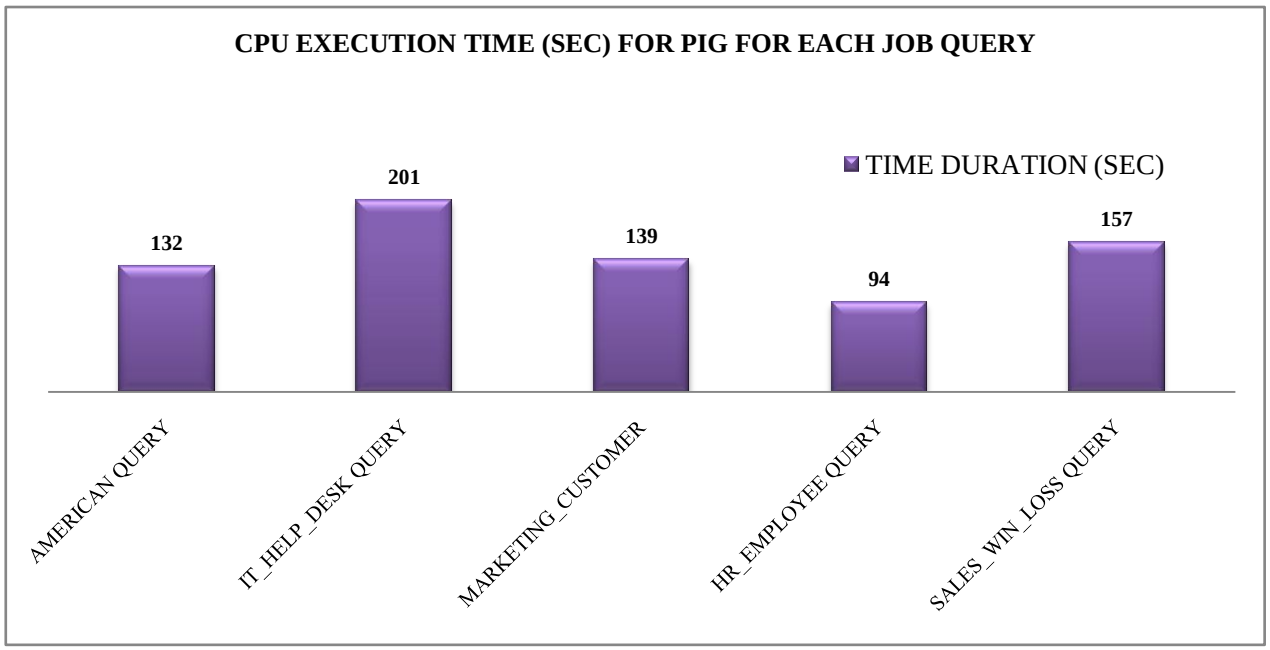


Figure 6: CPU Execution Time (sec) in Pig for each Job Query

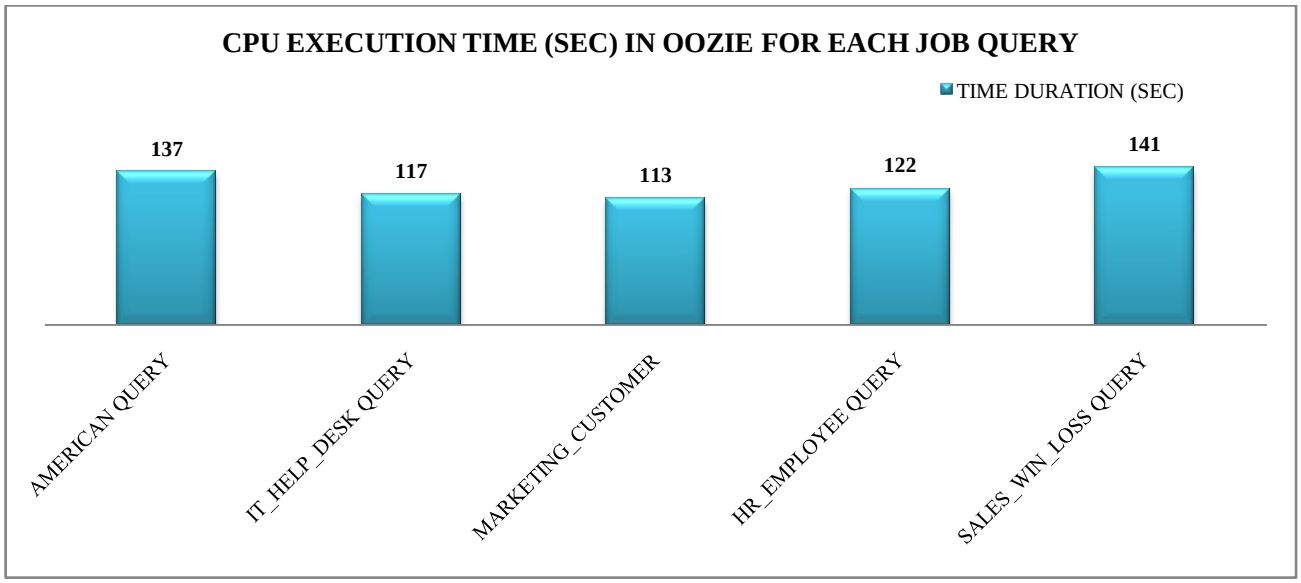


Figure 7 : CPU Execution Time (sec) in Oozie for each Job Query

In the above charts, CPU execution time is distinguished by using MapReduce purpose under the assist of WordCount algorithm on each Hadoop constituent. In this, the Hadoop libraries are Pig, Hive and Oozie. The conduct test is performed through a single text file named American_Time_Use_Survey-lite.csv. This dataset WA_American

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 3, Issue 4 , April 2016

encompasses of demographic, select personal and other attributes of a subset of Americans. For example, use Explore to find interesting trends about how Americans spend their time.
According to the graph, in Fig.2, Hive obtains the slightest quantity of time to perform the experimentation setup, while Pig gets the extreme concluding time for the closing stages to be revealed. In Fig.3, the average CPU implementation time with each Hadoop component is prominent using the preceding chart of Fig.2. In Fig.4, the Hive query for a solitary task of american query i.e. American_Time_Use_Survey-lite.csv is put into practice by taking the observations of 2 stages of Hive in cumulative CPU time, which are calculated in seconds.

In Fig.5, 6, 7, the graphs are made known as the execution time (in seconds) for each Hadoop module like Oozie, Hive and Pig for a set of 5 queries. The job queries names are american_time, it_help_desk, sales_win_loss, hr_employee and marketing_customer_analysis which are all associated to IBM datasets in .csv form, available freely in IBM dataset website. Time durations (in sec) of each job queries are observed and premeditated upon and diverse charts are prepared according to the evaluation of each Hadoop components i.e. Hive, Oozie and Pig.

V. CONCLUSION

In summing up, the recent literatures of diverse architectures have been observed and researched upon to support in the reduction of big data to simple data. The big data is composed of immense knowledge in gigabytes, petabytes or zetabytes. Besides, its postulates have been exemplified in details. 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 studied in detail. Overall, the MapReduce model is scrutinized with the help of its algorithm and an illustration for the reader to understand it clearly. To sum up, WordCount is characterized as the principle for mapping and reducing the datasets using Cloudera virtual machine in UNIX operating system.

REFERENCES

[1] Puneet Singh Duggal, Sanchita Paul, "Big Data Analysis: Challenges and Solutions", International Conference on Cloud, Big Data and Trust, Nov 13-15 2013.
[2] T. White, "Hadoop: The Definitive Guide", O'Reilly Media, Yahoo Press, June 5, 2009
[3] 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.
[4] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, "The Google File System", Google, New York, USA, pp.19–22, October 2003.
[5] Konstantin Shvachko, Hairong Kuang, Robert Chansler, "The Hadoop distributed File System", IEEE, Yahoo, California, USA, 2010.
[6] 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.
[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] 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.
[9] Vasiliki Kalavri, Vladimir Vlassov, 2013. MapReduce: Limitations, Optimizations and Open Issues, *IEEE*, KTH, the Royal Institute of Technology Stockholm, Sweden, pp.1031-1038.
[10] 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.
[11] 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
[12] 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.
[13] Mirel Cosulschi, Mihai Gabroveanu, and Adriana Sbrcea, "Running Hadoop applications in virtualization environment", Annals of the University of Craiova, Mathematics and Computer Science Series, 12th International Conference on Artificial Intelligence and Digital Communications - AIDC 2012, Volume 39, pp.322-333, November 30, 2012.
[14] 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.
[15] 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.
[16] 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

AUTHOR'S BIOGRAPHY

Priyaneet Bhatia is presently pursuing M.Tech in CSE from Galgotia College of Engineering and Technology (GCET), UPTU, Greater Noida, Uttar Pradesh, India. She has completed B.Tech from RTU, Jaipur Rajasthan, India in 2012.

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 3, Issue 4 , April 2016

Momentarily, she has published four quality papers in renowned international journals and currently working on the project “Big Data in Hadoop MapReduce under WordCount algorithm”. Her field of interests are Databases, Big Data etc.

Siddarth Gupta is currently working as Assistant Professor at Maharani Girls Engineering College, RTU affiliated college in Jaipur. He has completed M.Tech in CSE from Galgotia University, Greater Noida in 2015 and B.Tech from UPTU in 2012. Presently, he has published two quality papers in reputed international journals and shortly working on Big Data Optimization using Hadoop. His field of interests are Cloud Computing, Digital Watermarking, and Web Services.