ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

Performance Evaluation of Word Frequency Count in Hadoop Environment

K.Madasamy [1], M.Ramaswami [2]

Research Scholar *(Part-Time)*, Department of Computer Applications, Madurai Kamaraj University, Madurai, India.[1]

Associate Professor, Department of Computer Applications, Madurai Kamaraj University, Madurai, India.[2]

ABSTRACT:Big data has become one of the most important paradigm in scientific computing and business analytics. Hadoop MapReduce is the modern tool to analyze the volume of data by Giga bytes, Tera Bytes, Peta Bytes, Zeta Bytes and so on. The research objective of this study is to measure the execution time on different sizes of text files by performing a simple MapReduce simulation on the Word count program which is very popular in the Big data arena. Also an improved version of the Word count program has been designed and simulated using the same set of text files. A comparative analytical study of both the methods has been carried out and critically reviewed along with the sequential version of the Word count program.

KEYWORDs:Big data analytics, Hadoop, Hadoop Distributed File System (HDFS),MapReduce, Parallel Processing.

I. INTRODUCTION

Data processing using traditional enterprise systems normally have a centralized server to store and process normalized data. Currently, this technology is not ideal to process huge volumes of scalable data and that cannot be accommodated in standard database servers. Moreover, the centralized system creates too much of a bottleneck like heavy data traffic while processing multiple files simultaneously. Many data science experts consider that, humongous volumes of data not only brings the biggest challenges, but also a most thrilling opening in the recent years. Analysis of such data with minimum delay is also a challenging task. The high volume of unprecedented data collected from heterogeneous sources is called “Big data”. Many sources of big data are actually semi-structured or multi-structured and unstructured. The possible sources of big data include sensor networks, nuclear plants, scanning devices, airplane engines and consumer-driven data from social media. Big data producers that exist within organizations includes legal, sales, marketing, procurement, finance, and human resources departments. Big data has also increased velocity, complexity, and variety compared to data sources of the past.

Google’s MapReduce framework [1] have been deployed on commodity machines and exploit massive parallelism to efficiently analyse enormous datasets. Later, several implementations of the MapReduce programming model have been implemented and among them Apache Hadoop framework [15] being the most widely adopted open-source tool. MapReduce model was inspired by functional programming (FP), an alternative programming paradigm that involves writing programs purely by composition of functions rather than executing sequential tasks [16]. The MapReduce programming models are mainly deployed of data-intensive and compute-intensive applications in data mining and machine learning domains. Examples include, such as Text tokenization, indexing, searching, sentiment analysis, distributed grep, distributed sort, Web link graph reversal and document clustering.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

This paper is organized as follows: Section II describes the related work in Big Data Analytics. The architecture of Hadoop, MapReduce is presented in Section III. Section IV explains the Word count and the Improved version of Word count problems. Section V describes the experimental setup with the brief execution time analysis of single node, multi node along with sequential execution of the Word count problem. Finally, Section VI presents conclusion.

II. RELATED WORK

This survey is about the core themes of research articles published in various journals in the domain of big data analytics. At present the big data analytics arein the limelight. Hadoop is an ideal and an easy framework for dealing with big data. Hadoop has two main components, namely Hadoop Distributed File System (HDFS) and MapReduce. In this literature survey, an attempt is made to focus the research articles covering the various issues related with big data analytics and it has been presented here.

Sun.Z, et.al., [12] commented that, Big data analytics is an emerging science and technology involving the multidisciplinary state-of-art Information and Communication Technology (ICT), Mathematics, Operations Research (OR), Machine Learning (ML), and Decision Sciences.The core fundamentals of Big data analytics consists of Mathematics, Statistics, Engineering, Computer Science and Information Technology.MapReduce was proposed by Dean.J et.al.,[1] as a programming model for processing and generating large data sets. MapReduce is suggested as a possible means to perform elastic data processing in the cloud. The most successful implementation is the Google's MapReduce framework, which hides the complexity of data distribution, communication and task scheduling and offers a simple programming model for writing analytical applications, while also providingfault-tolerance.Vasiliki Kalavri et.al.,[17] have extensively studied the limitations, optimizations and many open issues related to MapReduce. Several implementations of the MapReduce model have been investigated. Among the many variations of MapReduce model, they suggested Apache Hadoop framework being the most widely adopted.Christopher Garcia [18] presented a study on MapReduce, which enables computational processes can be scaled up to very large sizes with the advent of Cloud computing. Also, the author have shed light on the types of problems for which MapReduce is well-suited, as well as those for which it is not. Further, the author suggested that, MapReduce is well-suited for processing tasks where relatively few processing steps depend implicitly or explicitly on one another.Elgendy et.al., [3] proposed that, the term "Big data" has recently been applied to datasets that grow so large that they become awkward to work with using traditional database management systems. Effectiveness and efficiency are the evident keys in Big data analytics. Cradling the gems of knowledge extracted out of Big data would only be effective if: (i) not a single important fact is left in the burden—which means completeness and (ii) these facts are faceted adequately for further inference—which means expressiveness and granularity. Efficiency may be interpreted as the ratio of spent effort to the utility of result. Shimin Chen and Steven W.Schlosser [13] have studied the extensive use of applications of Hadoop MapReduce Programming model. In particular, they have implemented many application tasks, including a large-scale machine learning computation, a physical simulation task and a digital media processing task. E. Lim et.al., [6] focused their research directions towards Business Intelligence and Analytics. They consider business intelligence and analytics (BIA) as a current form, replacing the traditional Business Intelligence. Gandomiandet.al., [5], discussed how Big data analytics has captured the attention of business and government leaders through decomposing Big data analytics into text analytics, audio analytics, video analytics, social media analytics, and predictive analytics. This implies that data has been classified into text data, audio data, video data, and social media data.

III. BIG DATA ANALYTICS

Big data analytics (BDA) is where advanced techniques operate on big data sets [10]. These include systems like Massively Parallel Processing (MPP) database systems and MapReduce that provide analytical capabilities for

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

retrospective and complex analysis that may touch most or all of the data. Big data is a collection of large datasets that cannot be processed using conventional computing techniques. It is not a single technique or a tool rather it involves many areas of business and technology. Many definitions are available for big data. Merv Adrian et.al., [9] said, "Big data exceeds the reach of commonly used hardware environments and software tools to capture, manage, and process it within a tolerable elapsed time for its user population". McKinsey [8] proposed that "Big data refers to data sets whose size is beyond the ability of typical database software tools to capture, store, manage and analyze". Big Data can be defined as volumes of data available in changeable amounts of density, produced at diverse velocities and changeable amounts of uncertainty, that cannot be processed using conventional technologies, processing techniques, algorithms, or any profitable off-the-shelf solutions. MapReduce provides a new method of analyzing data that are complementary to the capabilities provided by SQL. Also, MapReduce can be scaled up from single servers to thousands of high and low-end commodity hardware. Further, the Relational Database management system (RDBMS) uses SQL and emphasis on the ACID *(Atomicity, Consistency, Isolation, and Durability)* properties. As Big Data handles different types of data emphasize is given on BASE *(Basic Availability, Soft-state, Eventual consistent)* properties [4]. Big data also uses NoSQL database to store data, there is no need to have predefined schema. This section gives a brief outline about the Hadoop architecture and the MapReduce programming model.

A. Hadoop

Hadoop was created by *Doug Cutting* [2], the creator of *Apache Lucene*, the widely usedtext search library. Hadoop is an Apache open source, Java-based programming framework [11] that allows distributed processing of large datasets across clusters of computers using simple programming models. The two major components of Hadoop are Hadoop Distributed File System (HDFS) and MapReduce. Hadoop has many different vendors. Cloudera, Horton works, MapR, Amazon Elastic MapReduce, IBM Infosphere Big Insights are some of the most popular vendors. A Hadoop framework application works in an environment that provides distributed storage and computation across clusters of computers. Hadoop is designed to scale up from single server to thousands of machines, each offering local computation and storage. Hadoop runs applications using the MapReduce algorithm, where the data is processed in parallel on different commodity nodes. In short, Hadoop framework is competent to build up applications capable of running on clusters of computers and they could carry out complete statistical analysis for enormous amounts of data.Hadoop architecture is shown in figure 1.Hadoop architecture is categorized into three: Client Machines, Master Servers and Slave Servers. These are basically daemons or programs that run on different physical servers – NameNode, DataNode, Secondary NameNode, JobTracker and TaskTracker. These nodes are designated for different tasks and are briefed as under.

- ***Name Node***- Name node is the node which stores the filesystem metadata i.e. which file maps to what block locations and which blocks are stored on which data node. It is also known as master. The port id for name node is 50070.
- ***Data Node***-The data node is where the actual data resides. It is also called as slave. The port id for data node is 50075.
- ***Secondary Name Node***- Secondary Name node isa backup for the Name node.The port id is 50090 for secondary name node.
- ***Job Tracker***- The primary function of the Job Tracker is managing the task trackers, tracking resource availability, tracking its progress, fault tolerance etc. Job Tracker talks to the Name Node to determine the location of the data.The port id for job tracker is 50030.
- ***Task Tracker***- A Task Tracker is a node in the cluster that accepts tasks - Map, Reduce and Shuffle operations - from a Job Tracker. The port id for task tracker is 50060.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

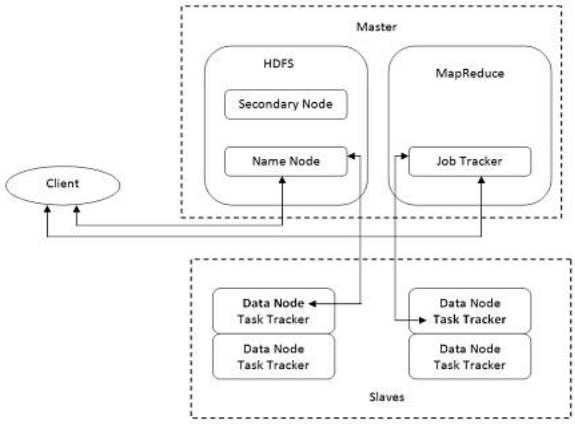

Fig.1. Hadoop Architecture

Hadoop's daemons uses many useful ports over TCP to access the various Web Interface for the common users. Some of these ports are used by Hadoop's daemons to communicate amongst themselves for scheduling jobs, replicating blocks etc.,. The other ports are listening directly to the users, either via an interposed Java client, which communicates via internal protocols, or via plain old HTTP. The default Hadoop ports are shown in table 1.

Table 1. Daemons and their Port addresses

Layer	Daemon	Default Port address
HDFS	Name node	50070
	Data node	50075
	Secondary Name node	50090
MapReduce	Job Tracker	50030
	Task Tracker	50060

B. MapReduce

MapReduce is a distributed programming model [1] proposed for dealing out enormous amounts of data in big clusters. MapReduce allows us to write distributed applications without worrying about the basic distributed computing infrastructure. The complexity that arises in the development of any application is handled by the Hadoop and the MapReduce framework itself. Thus, it helps program developer by handling all sorts of the problems. For example, it has the capacity to scale out the clusters by adding nodes. The automatic failover of data storage and data processing takes place with zero impact on applications. MapReduce is one of many application frameworks for data processing used to develop and run applications on Hadoop. The major advantage of MapReduce is that it is effortless to scale data processing over multiple computing nodes [14].

The MapReduce algorithm contains the tasks, namely Map and Reduce [7]. In the Map step, hundreds or even thousands of parallel processes, called "threads", perform a type of task called mapping, where a large mass of data is divided into pieces, and each performs a filtering process within a respective piece, creating a mass of values in the $<key_{in}, value_{in}>$ as input format [11].The output of the map function is the list of intermediate$<key_i, value_i>$ pair.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

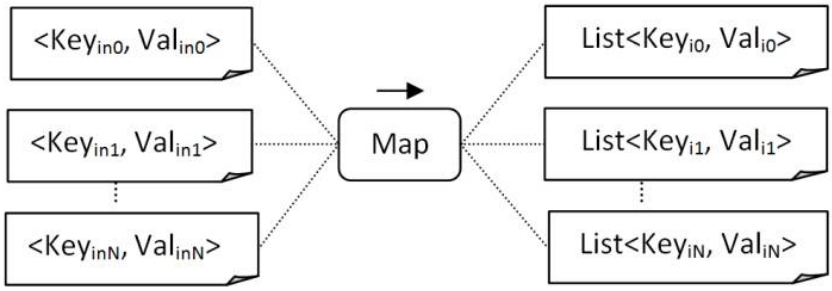

Fig.2. Map step in iteration

The mapping with input of the format <key$_{in}$, value$_{in}$> is executed simultaneously in multiple iterations and list (<key$_{in}$, value$_{in}$>) is obtained from the mapper process and is depicted in figure 2. To execute map and reduce functions in parallel, all the values associated with the same key is communicated from map to reduce in the shuffle stage.

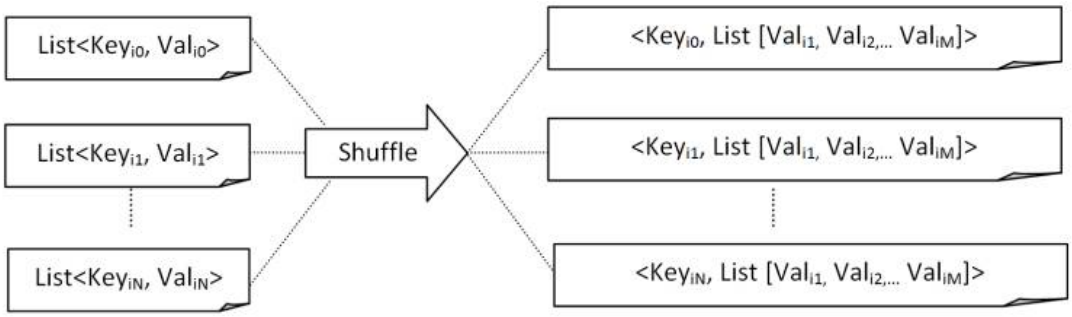

Fig.3. Shuffle step

After the Map phase and before the beginning of the Reduce phase is a handoff process, known as shuffle and sort. The shuffling phase will pass pairs with the same keytothesamereducerfrom differentmappers. The output of the shuffle process is of the form (<key$_i$, list[value$_{i1,}$value$_{i2...}$value$_{iM}$>) as illustrated in figure 3.

In the reduce step, the data generated by the map phase are again divided into pieces and passed with hundreds or even thousands of processes that perform processing on the received data bits and generate as a key-value output, which is the final output of the processing that is finally grouped into a mass of results. As the progression of the name MapReduce implies, the reduce task is always performed after the map job.

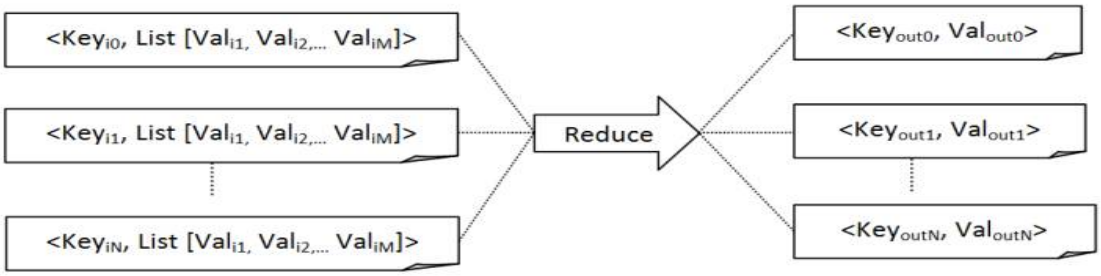

Fig.4. Reduce Step

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

The output of the reduce step is of the form <key_{out}, $value_{out}$> and is illustrated in figure 4.

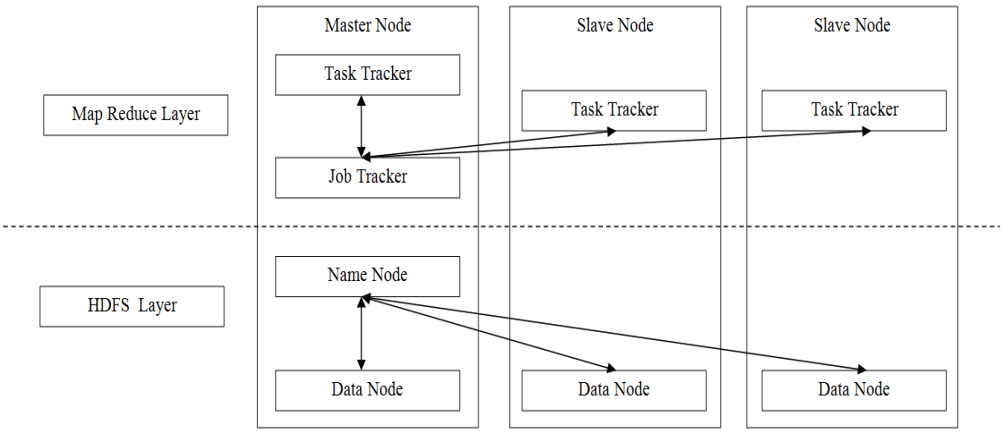

Fig. 5. Multinode MapReduce

MapReduce inputs for multinode typically come from input files loaded onto processing cluster in HDFS is illustrated in figure 5. These files are evenly distributed across all the nodes. Running a MapReduce program involves running mapping tasks on many or all of the nodes in the cluster. In each of these mapping tasks no mappers have particular "identities" associated with them. Therefore, any mapper can process any input file. Each mapper loads the set of files local to that machine and processes them.

When the mapping phase has completed, the intermediate (key, value) pairs must be exchanged between machines to send all values with the same key to a single reducer. The reduce tasks are spread across the same nodes in the cluster as the Mapper. This is the only communication step in MapReduce.

C. MapReduce Process

The MapReduce processes are explained as follows:

- Divide the input data into “n” number of chunks depending upon the amount of data and processing capacity of individual unit.
- Next, it is passed to the mapper functions. All the chunks are processed simultaneously at the same time.
- After mapping, shuffling happens which leads to aggregation of related patterns.
- Finally, reducers combine them all to get a consolidated output.
- This algorithm squeezes scalability as depending on the size of the input data, the number of the parallel processing units can also be increased.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

IV. WORD FREQUENCY COUNT PROBLEM

A. Basic Word count: Description

Word count is a simple Python application program that counts the number of occurrences of each word in a given input text file. The Word count program is executed in two stages: Map and Reduce. The map task maps the data in the file and counts each word in the data chunk provided to the map function. After the Map phase execution is completed successfully, shuffle phase is executed automatically wherein the key-value pairs generated in the Map phase are taken as input and then sorted in alphabetical order. In the Reduce phase, all the keys are grouped together, and the values for similar keys are added up to find the occurrences of a particular word. The Reducer phase takes the output of shuffle phase as input and then reduces the key-value pairs to unique keys with values added up.

B. Improved Word count: Description

The improved Word count program also works similar to the Word count program, but the difference lies in mapping the given input text. The improved Word count program maps the words present beside with the special character, numbers separately. The missing words with special characters in the Word count program are identified and counted using an improved Word Count program.

V. EXPERIMENTAL RESULTS

The Word count and the Improved Word count programs were tested both in single node as well as in multinode(1 Master with 2 slaves) Hadoop cluster environment. The host operating system has the following hardware and software specifications:

Intel Core i3-4150 CPU @ 3.50GHz and 8GB of RAM, System Type X64 based processor running under Windows 8 (Ultimate) version . Oracle VM Virtual Box Version 5.1.8 have been installed and in which Ubuntu 14.04 LTS was installed as the guest operating system. Hadoop- 2.7.2 and Sun Java JDK 1.7.0 have been configured in the guest operating system.

A. Implementation

The prime motivation of this study is to measure the execution time of different sizes of text files with parallel execution by performing a simple MapReduce simulation using the Word count program which is very popular in the Big data arena. The Word count program is written in Python script and it is designed to count the word occurrences in any given text files. Also an improved version of the Word count program has been designed and simulated using the same set of text files. An improved version of the Word count is designed to count the different punctuation symbols and numerical data in addition to counting the word occurrences. Both programs were tested in the single node as well as in multinode environment and their execution times were recorded. Finally a comparative analytical study of both methods have been carried out along with the sequential mode of execution and their results were critically reviewed.

B.Result analysis

Figure 6 and figure 7 depict the results of the execution of Basic Word count program in single node and multinode execution respectively. Text files with varying sizes like 2 MB,10 MB, 100 MB, 500 MB and 1 GB were used for both in Hadoop simulation and also for sequential mode of execution in all test cases. The graphs are plotted as "Size of data" versus "Execution time in seconds".

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

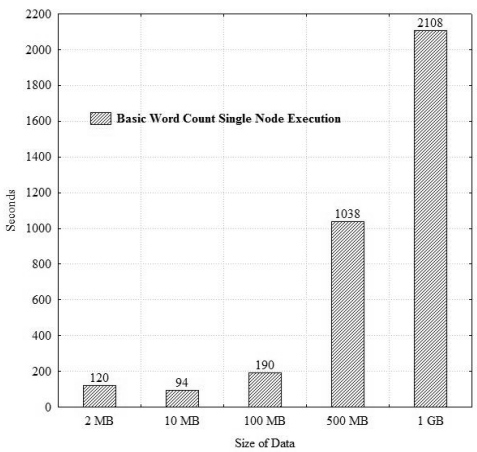

Fig.6. Basic Word count single node execution

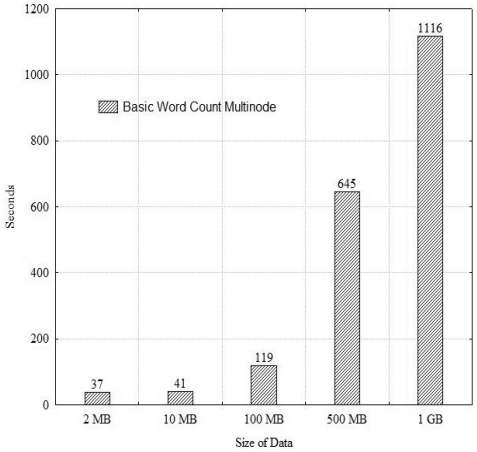

Fig.7. Basic Word count multinode execution

Further, the figure 6 shows that when the size of the data file is less, it consumes less time for execution and when the size of the data file is increasing, it consumes more time for execution. Figure 7 exhibits the same results as in the case of figure 6.

By comparing the results shown by figure 6 and figure 7, it is understood that the multinode execution consumes less time than the single node execution.

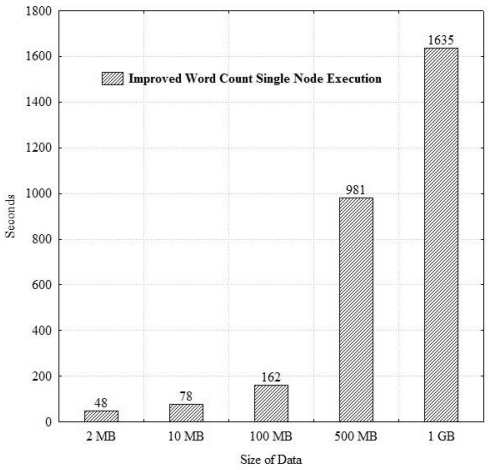

Fig.8. Improved Word count single node execution

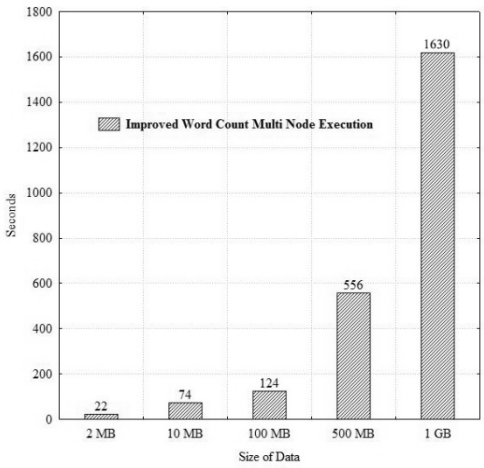

Fig. 9. Improved Word count multinode execution

Figure 8 and figure 9 depict the results of the execution of Improved Word count program in single node and multinode execution respectively. Figure 8 displays that when the size of the data file is less, it consumes less time for execution and when the size of the data file is increasing, it consumes more time for execution. Figure 9 reveals the same results as in the case of figure 8.

Though the figure 8 and figure 9 exhibit the same results, it is observed that the multinode execution consumes less time than the single node execution in the case of Improved Word count.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

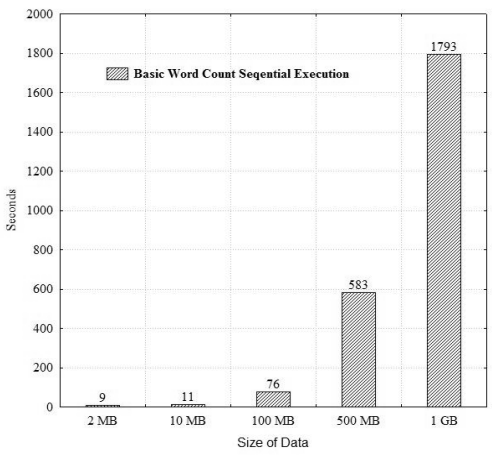

Fig.10. Basic Word count sequential execution

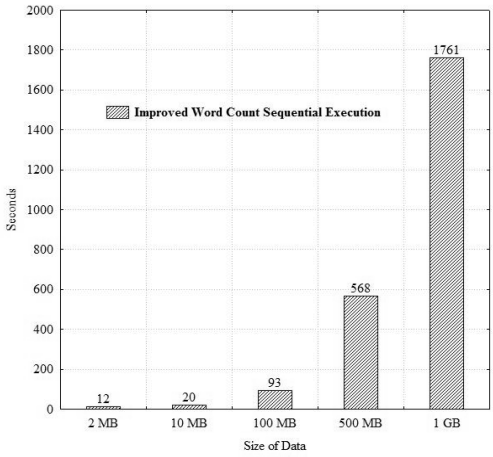

Fig.11. Improved Word count sequential Execution

Figure 10 and figure 11 depict the results of the execution of Basic Word count and Improved Word count program respectively in the normal sequential mode of execution. The same set of text files were used for the simulation. Figure 10 shows that when the size of the data file is less, it consumes very less time for execution and when the size of the data file is increasing, it consumes more time for execution. Figure 11 shows the same results as in the case of figure 10.

By comparing the results shown by figure 10 and figure 11, it is understood that when the size of the text file is less, the execution time is also less in both the cases, though it is not much significant; and at the same time, when the size of the text file is increasing, the execution time is significantly much lesser in Improved Word count than Basic Word count .

Table 2. Text files of varying sizes and their execution time

Size of Data in MB/GB	Basic Word count		Improved Word count		Sequential Mode Execution	
	Single node Execution	Multinode Execution	Single node Execution	Multinode Execution	Basic Word count	Improved Word count
2 MB	120	37	48	22	9	12
10 MB	94	41	78	74	11	20
100 MB	190	119	162	124	76	93
500 MB	1038	645	981	556	583	568
1 GB	2108	1116	1635	1630	1793	1761

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

Table 2 shows the consolidation of execution time on single node and multinode setup in both Basic Word count and Improved Word count under Hadoop environment. It also exhibits the execution time of both Basic Word count and Improved Word count in normal sequential mode of execution. Each input text file is executed consecutively for five times and the average time consumed for each run is recorded.

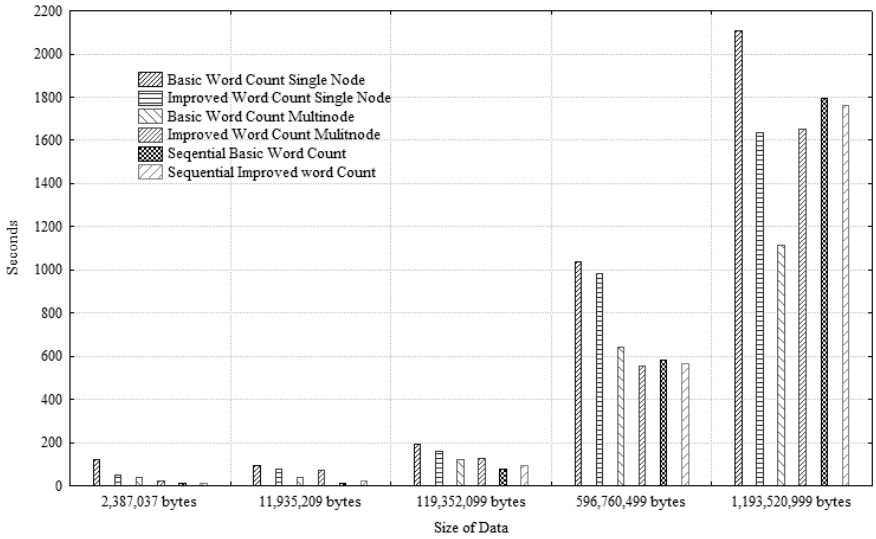

Fig. 12. Comparative execution time of Single Node vs. Multi node
(File Sizes are given in bytes)

Figure 12 shows the overall time taken by both the Basic Word count and Improved Word count program in single node and multinode setup.The time consumed by a sequential version of both the Basic Word count and Improved Word count programs are also plotted. It is evident that,multinode setup yields better results than single node setup when the file sizes arein increasing scale.

VI. CONCLUSION

The revolution of Big Data Analytics has made a remarkable impact on all spheres of our modern living and also in the IT sector. The Hadoop MapReduce framework has found many important applications in Big data analytics. In this present study, a simple and an improved version of Word count programs have been simulated in the Hadoop MapReduce environment using different file sizes. It is found that the improved version of the Word count program counts all the tokens of the given text files including the special characters and provides better results than the simple Word count program in terms of execution time. Owing to the personal desktop hardware limitations, this program is now running with limited file size. With high-end desktop hardware support and with higher numbers of slave nodes, these programs could be run for higher volume file sizes in upward scalability to suit with any Big data related applications. As an extension work, the same working setup can be configured in any cloud environment by selecting the best hardware configuration with the desired number of slave nodes.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

REFERENCES

[1] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," Commun ACM, 51(1), pp. 107-113, 2008.
[2] Tom White, "Hadoop : The Definitive Guide", Second edition,2010, O'Reilly Media Inc.,
[3] Nada Elgendy and Ahmed Elragal,"Big Data Analytics: A Literature Review Paper",Department of Business Informatics & Operations,German University in Cairo (GUC), Cairo, Egypt
[4] R. Gupta, H. Gupta and M. Mohania, "Cloud Computing and Data Analytics: What Is New from Databases Perspective?" in Big data Analytics , Anonymous : Springer, 2012,pp.42-61.
[5] Gandomiand . M. Haider, "Beyond the hype: Big data concepts, methods, and analytics," *International Journal of Information Management 35 (2015)*, vol. 35, p. 137–144, 2015.
[6] E. Lim, H. Chen and G. Chen, "Business Intelligence and Analytics: Research Directions," *ACM Transactions on Management Information Systems*, vol. 3, no. 4, Article 17, pp. 1-10, 2013.
[7] Mani Bushan, Balaraj J, Oinam Martina Devi, "Comparison of Join Algorithms in Map Reduce Framework", International Journal of Innovative Research in Computer and Communication Engineering, Vol.2, Special Issue 5, October 2014
[8] McKinsey, Global Institute, Big Data: The Next Frontier for Innovation, Competition, and Productivity, May 2011.
[9] Merv Adrian, *"Big Data,"* Teradata Magazine,1:11, www.teradatamagazine.com /v11n01 /Features/Big-Data/
[10] M. Minelli, M. Chambers and A. Dhiraj, Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses, Wiley & Sons (Chinese Edition 2014), 2013.
[11] Ruchi Mittal, Ruhi Bagga, "Performance Analysis of Multi-Node Hadoop Clusters using Amazon EC2 Instances", International Journal of Science and Research (IJSR) ISSN (Online): 2319-7064
[12] Z. Sun, K. Strang and J. Yearwood, "Analytics service oriented architecture for enterprise information systems," in *Proceedings of iiWAS2014, CONFENIS 2014, 4 - 6 Dec 14*, Hanoi, 2014.
[13] Shimin Chen and Steven W.Schlosser, " Map-Reduce Meets Wider Varieties of Applications", Intel Research, Pittsburgh, USA
[14] Dr. Urmila R. Pol, "Big Data Analysis Using Hadoop Mapreduce", American Journal of Engineering Research (AJER), e-ISSN: 2320-0847 p-ISSN : 2320-0936, Volume-5, Issue-6, pp-146-151
[15] "Hadoop : Open-Source implementation of MapReduce", http:// Hadoop.apache.org
[16] Hughes.J, "Why Functional Programming Matters", The Computer Journal, 32(1989),98-107.
[17] Vasiliki Kalavri and Vladimir Vlassov, "Mapreduce : Limitations, Optimizations and Open Issues", In: Proceedings - 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2013, IEEE , 2013, pages.1031-1038.
[18] Christopher Garcia, "Demystifying MapReduce", Procedia Computer Science, 20(2013),484-489.