An Implementation of Map Reduce on the Hadoop for Analyzing Big Data

Gul Shaira Banu Jahangeer, T. Diliphan Rajkumar

Abstract: The Speedy development of Internet has led to huge quantities of digital data available online and vast capacity of digital data is increasing and successfully stored. In demand to the process, analyzed, and linked huge volume of stored data to achieve correct Information, some computation is required. Even efficient processing and implementation is needed for scientific data performance analysis. We will compare with already existing MapReduce Technique with Hadoop to afford high performance and efficiency for large volume of dataset. Hadoop distributed architecture with MapReduce programming is analysis here.

Keywords : MapReduce, Hadoop, Distributed Computing.

I. INTRODUCTION

We are existing in an era where large volume of information has been generated every single second. Surplus data are generated and stored from electronic Gadgets, social media, scientific data, websites and enterprise. We have heavy blast in internet world by funding huge data [1].

A. Why Big Data Important?

Recent trend in the hottest IT sector today is big data where we have bulky amount of data. In today's world trendy and developing Business, multiple sectors, scientific sectors got developed and contributed in economic gains in every division. [2].

The pride of big data is not about how much bulk of data you have in your record. It is all about what you will do with it. Analysis and find the answer from the big data collected from any source that permit to reduce the cost , reduce the timing ,for developing new product , good offers and decision makings .

High ranked analytics with big data can achieve Enterprises responsibilities such as:

Defining the cause grounds of issues, failures, and faults in actual time.

- ✓ Defining the cause grounds of issues, failures, and faults in actual time.
- ✓ At the opinion of sales based on customers buying habit, the seller can make vouchers.
- ✓ The entire risk portfolios can be recalculated in minutes.

Revised Manuscript Received on December 05, 2019.

Gul Shaira Banu Jahangeer, Computer Science and Engineering, Kalasalingam Academy of Research and Education, India.shairaamjath@gmail.com

Dr. T. Diliphan Rajkumar, Faculty of Computer Science and Engineering, Kalasalingam Academy of Research and Education, Tamilnadu, India. t.dhiliphan@klu.ac.in

Finding out fake behavior earlier before it shakes the entire business concern. Processing large amount of data in the enterprises is real challenge. None of the present centralized architectures find to be more efficient in handling large amount of data. When handling the lump sum data, best to attach and utilize in the distributed architecture. [3].

B. Apache Hadoop

In Distributed system use open source software module called apache Hadoop .The main components used in the module apache Hadoop is MapReduce, which is used to process the data and system is known as Hadoop Distributed File System (HDFS).MapReduce paradigm was very old and most popular tool still now for batch-processing large datasets .Using simple model, complex distributed program can be build [4].

Big data is nothing but large Chunks of data. It can be store and analyze using MapReduce programming models, which is in apache Hadoop [3].Popular social media like yahoo, Facebook, twitter use MapReduce programming model[4]. Some Common technique is used to enlarge the cluster by attaching new storage devices, computers and other needful thing, when Hadoop cluster is undersize. Resort for additional resources on another computer cluster is alternative Method. Real computer r virtual computer may be used as another computer for enlargement of cluster.

In 2004, Google introduce programming model called MapReduce paradigm, now it is supposed to say as Apache Hadoop. [5].Hence the Apache Hadoop covers the job like managing processing part by Hadoop MapReduce and managing distributed part by Hadoop Distributed File System
(HDFS).

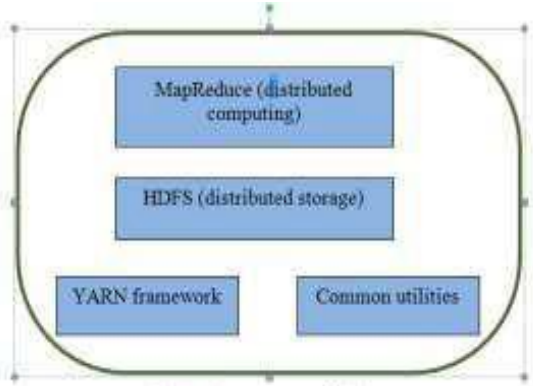

Fig. 1. Components of Hadoop

C. MapReduce Paradigm

MapReduce is a programming module which is modest and easy to understand which accompanying operation for handling and producing big data sets on cluster by using a parallel, distributed algorithm. It's well-known with clustered scale-out data handling solutions. Hadoop cluster supports huge scalability through hundreds or thousands of servers.

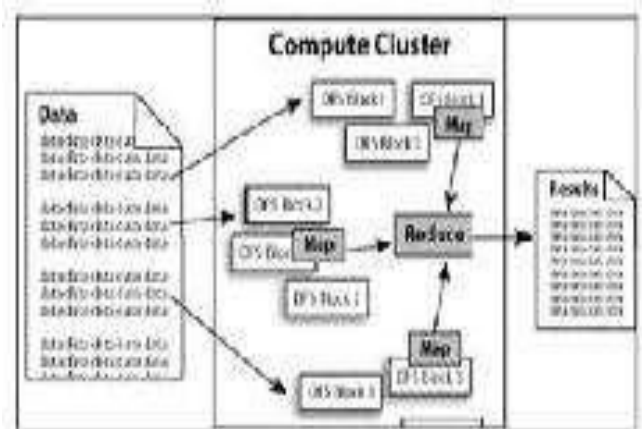

Fig. 2 Map Reduce in Hadoop

The MapReduce handler require two tasks called Map and Reduce, which work on data prepared in key/value sets. The MapReduce frameworks divide the first set of data. The divided data are processed by functions called Map function. The Map Function is generally applied on each line of every divide set of data. Each Map function of divide set of data gives a list of key/value pairs of output. If we take word count as example, the map function would create output of word by means of keys and map function would create number Instances of each word in every line by means of values. All middle new and smaller values in MapReduce framework groups together connected with the same intermediary key that permits them to decrease the task, it also groups or unifies
them.

Consider an example such word-count the Map Reduce function sums the total words and produce the outputs showing total number of instances for each word. The implementations of the Map and Reduce functions are mentioned to as Map and Reduce tasks, or shortly Mappers and Reducers. Usual responsibilities performed for one request are mentioned to as a MapReduce task. [4] .

D. Hadoop Distributed File System (HDFS)

Today much Service hardware is designed to use distributed file system which is said to be Hadoop Distributed File System (HDFS).Current distributed file systems has similarities with Hadoop Distributed File System (HDFS).Still Major differences from other distributed file systems. It is designed like low cost hardware and highly fault tolerant. Large number of data set use in HDFS, which provide high throughput entrée to application data, which is highly appropriate for many applications. Apache Hadoop is castoff to build subproject HDFS. Original infrastructure of Apache Notch web search engine project was built by HDFS. [6].

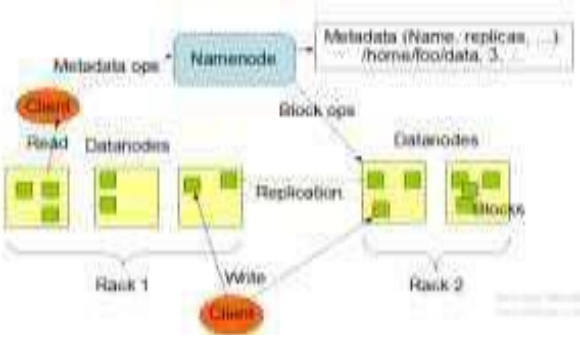

Fig .3 The prototypical MapReduce example

Hadoop distributed computing over traditional computing was analyzed in this paper and comparison is made with two different models with a simple program. This program uses multiple numbers of clusters with different size of data and the result is checked with execution time of work in each trial. The remaining paper is planned with proposed methodology and it is executing with Hadoop test performance analysis. Finally we come out with results find from our proposed methodology observations. Our work got concluded in last section.

II. PROPOSED METHODOLOGY

The Hadoop having splits which is fixed in size and also created distinct Map job for every split of Hadoop the Apache Hadoop Map Reducer which shown in (fig-1). The size of splitting the data is same as the size of HDFS block .The block of HDFS is 64 MB in size. The data locality optimization is performed by organizing the node input data exist in the HDFS with Map task. Creating partition for each reduces jobs and its output is written in partition disk. Various keys and related values for keys is contain for local disk partition. Reduce task is handled with related keys with recorded data and the reduced task can be achieved using practitioner function.

Hash function is employed on the key by the map reduce expert and create output from Map. Large number of reducer along with modulo function and reduce task index is applied with hash value for each key. Step by step map reduce dataflow is shown fig 1.function run systematically by reducing the task, different map task cannot delivered on data area. From the output of multiple map jobs, the single map reduce map job is formed, which is shown in (Fig.1). The output of the sorted map are combined before send to the reduce task, this is role play in reduce task node. The movement of information between Map and Reduce tasks is recognized as waddle. Here the unification is noted as a portion of the decrease task. Independently specify the number of reduce tasks for an allotted job, which is kept in the HDFS. Combiner the additional function is allowed by the Hadoop for receiving data send by the Map job as an input for the same node and customs the output that converts as the input to reduce function.

To achieve data reduction combiner function is used reduce the influence of the restricted communication bandwidth and data transfer over the network on the acts of a MapReduce job. [8]

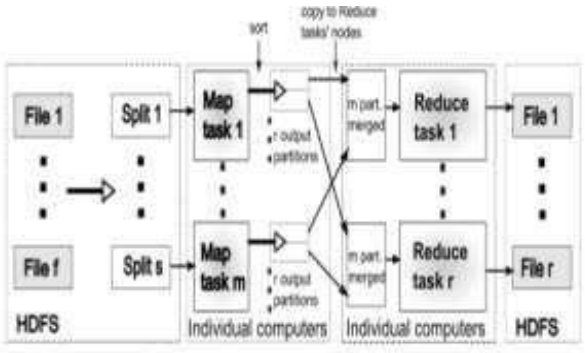

Fig. 4 Schematic representation of MapReduce data flow

A.. Algorithm

In a document the mapper produces a central key-value pair for each word in the file. The reducer will count and sum together all the reduced words. The data structure uses the field key and values esteem to the map and reduce function of Map reducer.one set of data was taken by map along with one set of data from domain. Finally return a list if set of pair to different domain.

$$\text{Map } (S1, A1) \rightarrow \text{ list } (S2, A2) \quad (1)$$

The Map function is applied in the input dataset and it is useful for matching to every pair (keyed by S1). This step is a list of pairs (keyed by S2) for individually demand. Later the MapReduce outline gathers all the pairs with the same key (S2) stating all the lists and crowd them together, making one cluster for individually key. Each group is applied with reduced function in parallel which yields a group of standards in the same domain:

$$\text{Reduce } (S2, \text{list } (A2)) \rightarrow \text{ list } (A3) \quad (2) \text{ Each}$$

Decrease call classically yields both one value A3 and an return which is empty, however single call is permitted to produce new than unique value. The desired result is collected depends upon the return of call. Therefore the MapReduce outline alters a list of (key, value) pairs into a list of values. This performance is dissimilar after the classic useful software design plot and the decrease combination, which takes a list of random values and returns one single value that associations all the values resumed by record. [9]

It is essential however not enough to have implementations of the map and reduce concepts in demand to develop the MapReduce. Distributed employments of MapReduce need a income of linking the procedures to execute the Map and Reduce stages. [10] And it's similar to distributed file system. Other choices are promising, such as straight flowing from mappers to reducers or for the planning processors to attend their outcomes to reducers that request them. In Fig 5 the word count mapper algorithm is used as the reducer .

```
function map(String name, String document):
  // name: document name
  // document: document contents
  for each word w in document:
    emit (w, 1)

function reduce(String word, Iterator partialCounts):
  // word: a word
  // partialCounts: a list of aggregated partial counts
  sum = 0
  for each pc in partialCounts:
    sum += pc
  emit (word, sum)
```

B. Test Dataset

In our implementation, we complete a sequence of trials on a MapReduce program "word count" using map and reduce on Hadoop clusters differs starting 2-nodes to10-nodes. The information set used here differs from 500 MB to 1.5 GB with altered case for changed algorithm implementation using MapReduce. First we implement the package on a solo machine using old-style programming Language. We save this implementation time, and at the same time run the algorithm on different clusters and different datasets save the implementation time in each situation.

Word Count is a typical Hadoop test program. Here we compute the amount of happenings of each word in the quantified data set. It is single straightforward algorithms for handling text data. Throughout the testing procedure, word count program will practice the specified data and output the calculation result based on the number of amounts of the words.

Multiple text files available in our datasets, which contains the datasets with the file of 500MB, 750MB, 1GB and 1.5GB. The size of the work used in our Hadoop clusters contains 2-node, 4-node, 8-node and 10-node. The program is first showed with a old-style solo node execution of word count. Next data is uploaded and the process is completed in the HDFS .The operation of disk space of Data Nodes is stabilize dusing matching strategy in Hadoop.

Subsequent, we run the test program numerous times to progress and uploaded all data in the Hadoop cluster. The size of the data stored in each node is mostly similar, however owing to the conflicts of nodes' routine then the size of data treated by each node is not similar. In our results we have a distinct examination of the outcomes of numerous MapReduce programs.

III. RESULT AND ANALYSIS

The experiment is performed with different dataset. The different sizes of data led to cluster the data in the different size. Sequential programming method is use do the experiment .Map reducer is used to do the analysis and find to be more efficient. Initially we use 500 MB of dataset and implemented on a single system with sequence programming method. Later its experimented with multiple nodes of cluster (like 2, 4, 8, 10). Likewise, experiments are performed with different size of data. Experiment is conducted and executed at different size of data like 750 MB, 1 GB and 1.5 GB and it is recorded in seconds. Clear explanation about different execution time and dataset is provided in the below table. Our experiments are done using Hadoop cluster. The method is created on the commentaries of cluster scale-up by keeping the data size stable and increasing the size of cluster.

Table 1 – Comparison Result of Execution Time And No of Nodes.

Programming	500 MB	750MB	1 GB	1.5 GB
Sequence OOPS	590	1023	1146	2100
Hadoop 2 node	530	832	1024	1300
Hadoop 4 node	385	478	510	745
Hadoop 8 node	270	290	340	340
Hadoop 10 node	142	210	185	385

Method-1: The traditional object oriented Language is used to count the words with the data size of 500 Mb. The time taken to execute the program is 590 seconds and also calculated the word which is repeated in the document. Correspondingly we make sure to executed the MapReduce program with cluster of node like 2, 4,8,10. Using multiple of nodes in clusters and experiments the execution time is reduced. We have experimental result from which we concluded like the cluster size rises the time for execution will decrease.

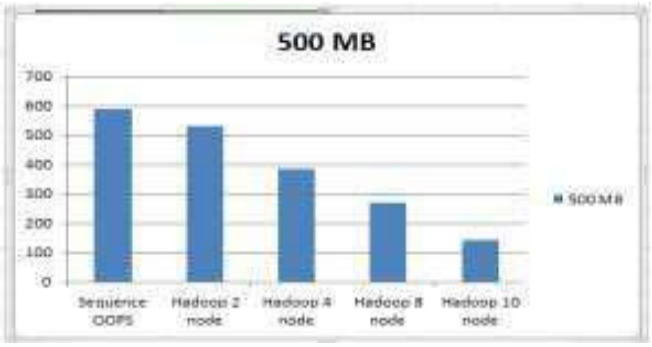

Fig: 5 Prediction result for clustered data for 500 MB data.

Method-2:

The traditional object oriented Language is used to count the words with the data size of 750 MB, 1GB, 1.5 GB. The time taken to execute the program is 530 seconds and also calculated the word which is repeated in the document. Correspondingly we make sure to executed the MapReduce program with cluster of node like 2, 4,8,10. Using multiple of nodes in clusters and experiments the execution time is reduced. We have experimental result from which we concluded like the cluster size rises the time for execution will decrease.

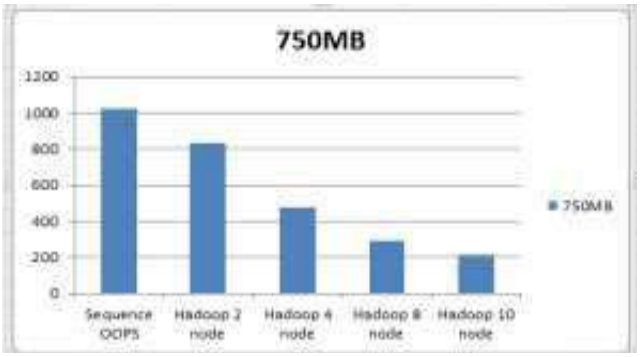

Fig: 6 Prediction result for clustered data for 750 MB data.

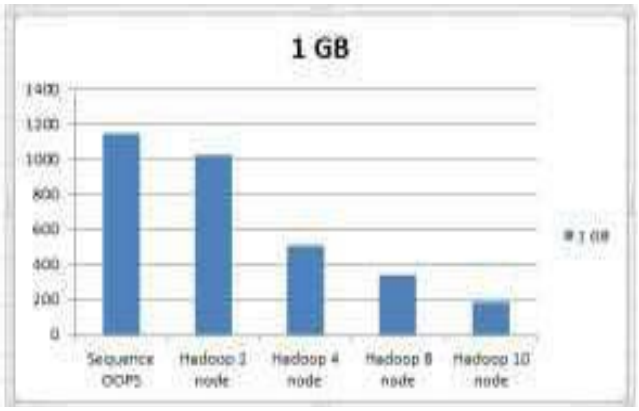

Fig: 7 Prediction result for clustered data for 1GB data

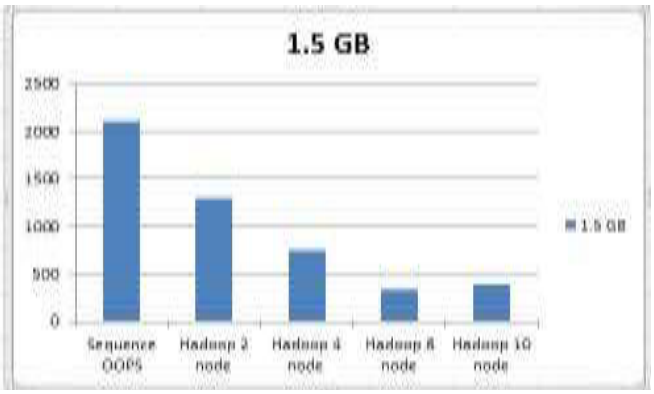

Fig: 8 Prediction result for clustered data for 1.5 GB data.

We have concluded the result obtained from all four experiments in order to analyses and conclude Hadoop performance.

IV. CONCLUSION

Chockfull Capacity of information can be produced for research and even data collected from internet can be clusters by using Hadoop MapReduce technique. Efficient code is developed for distributed computing using Hadoop .Four different size of hodoop cluster is used to perform the trials with four different sizes of dataset. Our trial in this paper has been witnessed if we increase number of nodes in the program the execution time will be decreases automatically.

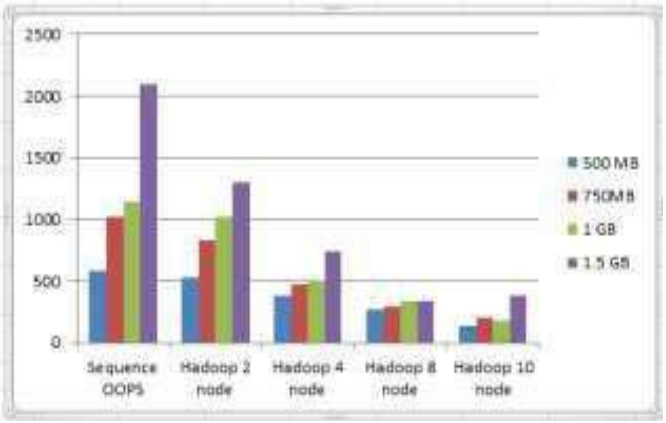

Fig:9 Prediction result for clustered data for 500MB,750MB,1GB,1.5GB data.

REFERENCES

[1] R.A.Fadnavis, Samrudhi Tabhane, "Big Data Processing Using Hadoop", International Journal of Computer Science and Information Technologies, Vol. 6 (1), 443-445, 2015.
[2] Hemlata S. Urade, "Analyzing Big Data Problem Using Hadoop and Cloud Technology", International Journal of Science and Research (IJSR), Volume 4 Issue 11, November 2015.
[3] R.A.Fadnavis, Samrudhi Tabhane, "Big Data Processing Using Hadoop", International Journal of Computer Science and Information Technologies, Vol. 6 (1), 443-445, 2015.
[4] I. Tomašić, A. Rashkovska and M. Depolli, "Using Hadoop MapReduce in a Multicluster Environment"," IEEE , MIPRO 2013, Opatija, Croatia, May 20-24, 2013.
[5] Shaymaa Taha Ahmed, Ramdas Naik, "Hadoop Data for Big Data Processing", International Journal of Innovative Technologies, vol. 3, Issue 2, June 2015.
[6] https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html.
[7] Praveen Kumar B1 , Nirmala Bariker, "Implementation of Hadoop Based Framework for Parallel Processing of Biological Data", International Journal of Science and Research (IJSR) , volume 4 issue 4, april 2015.
[8] J. Lin, and C. Dyer, "Limitations of MapReduce," Data-Intensive Text Processing with MapReduce, Synthesis Lectures on Human Language Technologies, pp. 143-145: Morgan & Claypool Publishers, 2010.
[9] https://en.wikipedia.org/wiki/MapReduce.
[10] AshaLatha, Naga Lakshmi, ""Implementation of Hadoop Based Framework for Parallel Processing of Biological Data", IOSR Journal of Computer Engineering (IOSR-JCE) , Volume 18, Issue 4, Ver. VI, PP 75-77, Jul.-Aug. 2016.

AUTHORS PROFILE

Gul Shaira Amjath is a Lecturer. She has a Master of computer Science and Engg from Mepco Schlenk Engineering college , Anna University, and Bachelor of Computer science and Engg from Mohamad sadak Engineering college , Madurai Kamraj University.Her Research area include image Processing , Machine Learning , Data Mining .

Dr. T. Diliphan Rajkumar completed his B.E (2009) Computer Science and Engineering in Arulmigu Kalasalingam College of Engineering, Anna University. He received M.E (2011) Computer Science and Engineering in Muthayammal Engineering College, Anna University. He pursed his Ph.D (2017) degree in Computer Science and Engineering, ManonmaniamSundaranar University, India. He has a strong passion in Web Mining, Pattern recognition and Social networking. He is currently working as Assistant Professor in Kalasalingam Academy of Research and Education.