RESEARCHERID
THOMSON REUTERS

[Srikanth* 4(5): May, 2017] ISSN 2349-4506
Impact Factor: 2.785

Global Journal of Engineering Science and Research Management

A FUNDAMENTAL CONCEPT OF MAPREDUCE WITH MASSIVE FILES DATASET IN BIG DATA USING HADOOP PSEUDO-DISTRIBUTION MODE

K. Srikanth*, P. Venkateswarlu, Ashok Suragala
* Department of Information Technology, JNTUK-UCEV Vizianagaram, INDIA
Department of Information Technology, JNTUK-UCEV Vizianagaram, INDIA
Department of CSE, JNTUK-UCEV Vizianagaram, INDIA

DOI: 10.5281/zenodo.801301

KEYWORDS: HDFS, Hadoop, MapReduce, Name Node, Data Node.

ABSTRACT

Hadoop Distributed File System (HDFS) and MapReduce programming model is used for storage and retrieval of the big data. Big data can be any structured collection which results incapability of conventional data management methods. The Tera Bytes size file can be easily stored on the HDFS and can be analyzed with MapReduce. This paper provides introduction to Hadoop HDFS and MapReduce for storing large number of files and retrieve information from these files. In this paper we present our experimental work done on Hadoop by applying a number of files as input to the system and then analyzing the performance of the Hadoop system. We have studied the amount of bytes written and read by the system and by the MapReduce. We have analyzed the behavior of the map method and the reduce method with increasing number of files and the amount of bytes written and read by these tasks.

INTRODUCTION

Distributed storage system can be used for storing this huge amount of data. There is lots of a distributed storage system currently available. There can be three states of the data. Data can be in structured form, semi structured form and unstructured form. The amount of structured data is very less as compared to the semi structured and the unstructured data. Internet Data Center (IDC) has given statistics of the amount of structured and unstructured data present scenario and the predicted growth of the structured and the semi structured data. There are 3V first purposed for big data, Velocity, Veracity, and Volume. A distributed system should have access transparency, location transparency, concurrency transparency, failure transparency, heterogeneity, scalability, replication transparency, migration transparency. It should support fine- grained distribution of data tolerance for network partitioning it should provide a Distributed file system service for handling files. The main aim of a distributed file system (DFS) is to allow users of physically distributed Computers to share data and storage resources by using a common file system. DFS consists of multiple software components that are on multiple computers, but run as a single system. The power of distributed computing can clearly be seen in some of the most ubiquitous of modern applications.

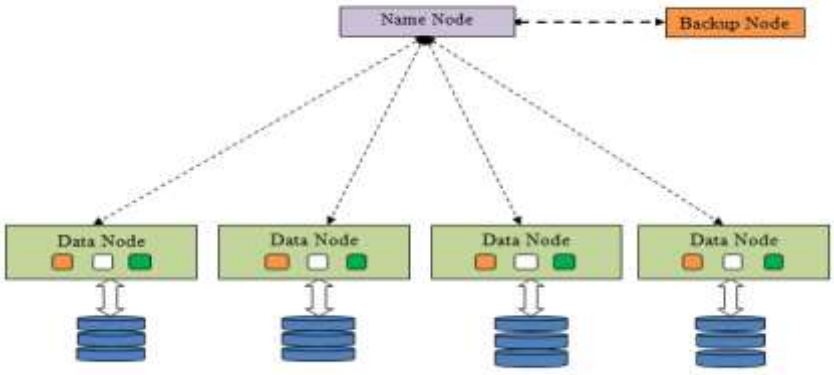


Figure 1. HDFS Architecture

RELATED WORK

Big data can be handled by using the Hadoop and MapReduce. The multiple node clusters can be set up using the Hadoop and MapReduce. Different size files can be stored in this cluster [5]. A shared disk analysis of the Hadoop is done [6]. Big data can also be provided as a service [7] in cloud computing (Data-as- service). With this big data analysis becomes important in a cloud computing environment. For management of data there is a traditional database management system. It is tested that Hadoop provides the best service in the context of cost, scalability and unstructured data [8].

HADOOP HDFS AND MAPREDUCE

Hadoop comes with its default distributed file system which is Hadoop distributed file system [9]. It stores file in blocks of 64 MB. It can store files of varying size from 100MB to GB, TB. Hadoop architecture contains the Name node, data nodes, secondary name node, Task tracker and job tracker. Name node maintained the Metadata information about the block stored in the Hadoop distributed file system. Files are stored in blocks in a distributed manner. The Secondary name node does the work of maintaining the validity of the Name Node and updating the Name Node Information time to time. Data node actually stores the data. The Job Tracker actually receives the job from the user and split it into parts. Job Tracker then assigns these split jobs to the Task Tracker. Task Tracker has fixed number of the slots for running the tasks. The Job tracker selects the Task Tracker which has the free available slots. It is useful to choose the Task Tracker on the same rack where the data is stored this is known as rack awareness. With this inter rack bandwidth can be saved. Figure 2 shows the arrangement of the different component of Hadoop on a single node. In this arrangement all the component Name Node, Secondary Name Node, Data Node, and Task Tracker are on the same system. The User submits its job in the form of MapReduce task. The data Node and the Task Tracker are on the same system so that the best speed for the read and write can be achieved.

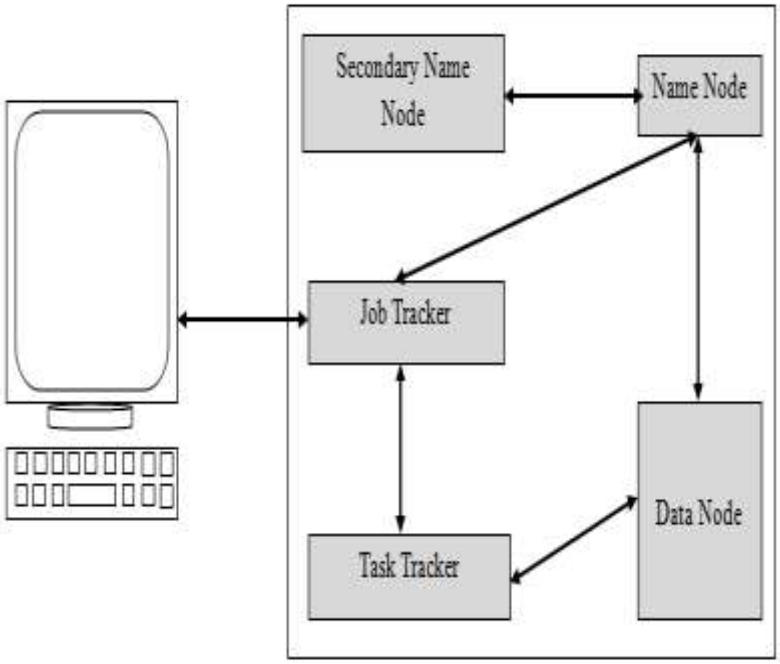
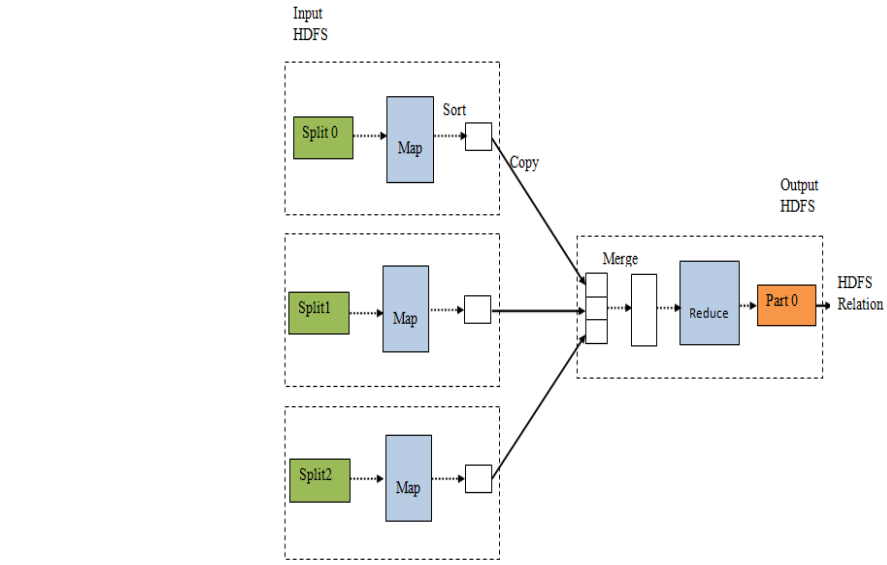


Figure 2 Single Node Architecture of Hadoop

Map when used in the HDFS first maps all the requested file blocks in the HDFS then reduces them according to the required result.



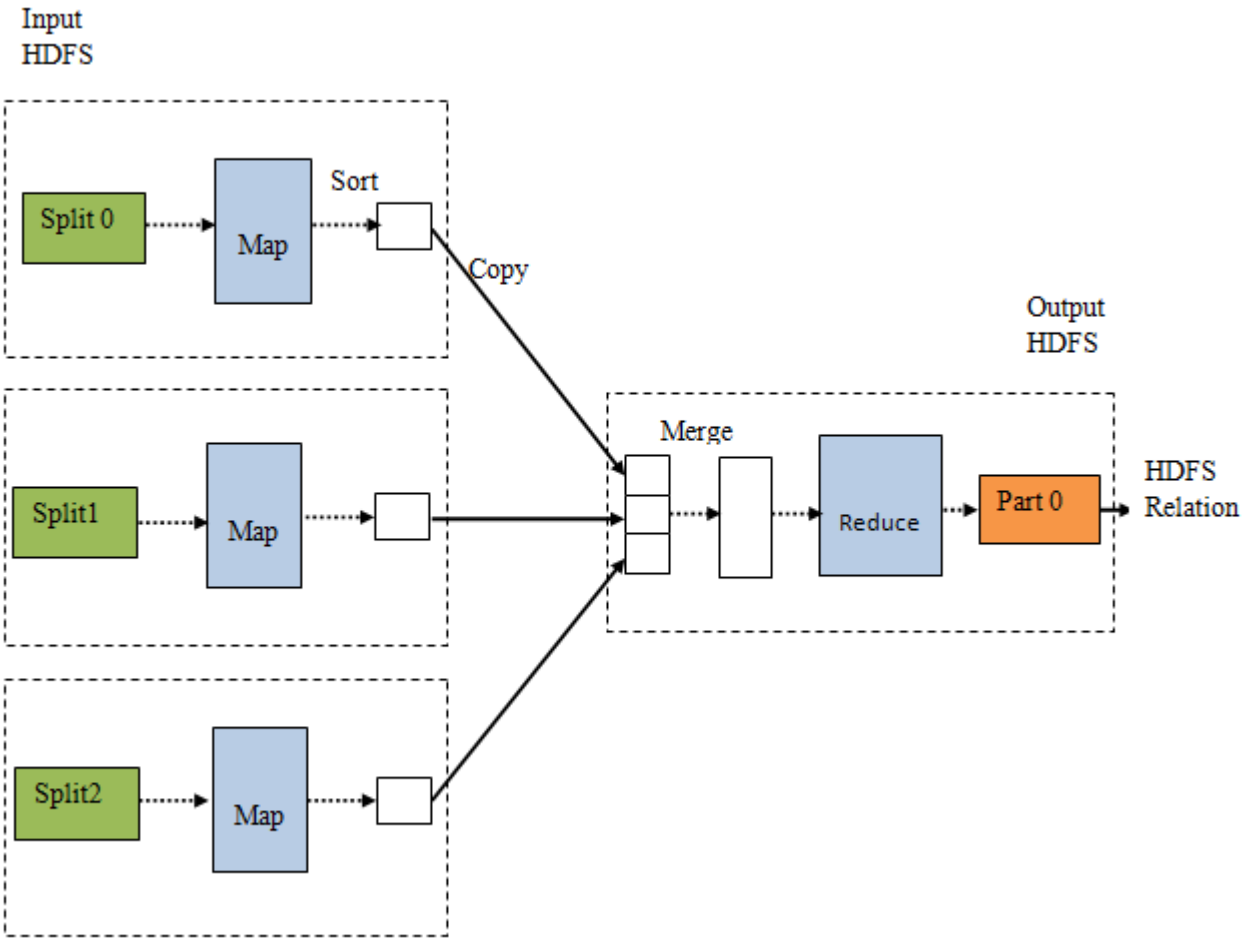


Figure 3. MapReduce data flow with a single reduce task

ISSN 2349-4506
Impact Factor: 2.785
Global Journal of Engineering Science and Research Management

WORD COUNT

WordCount is a simple MapReduce allocation. It counts the number of times a word is repeated into a text file of an input set. It imports Path, conf.*, io.*, maped.*, util.* of Hadoop. The Mapper is implemented by a map method; one line is processed at a time. The input can be strings or a text file. It splits the input into tokens, for separation it takes space as a tokenizer and uses the class String Tokenizer.

Output is in the form of <<word>, 1>. Different maps generate these key value pairs for example.
For input file having text like:
Hii
This is srikanth
Welcome to Hadoop Lab

Table1. Output of the Mapper

First map output	Second map output	Third map output
(<Hii>,1)	(<This>,1)	(<Welcome>,1)
	(<is>,1)	(<To>,1)
	(<Srikanth>,1)	(<Hadoop>,1)
		(<Lab>,1)

The Reducer for these maps is implemented by the Reduce method. It takes input from all the maps and like in merge sort it combines the output of all the map and produce output as (< word>, n) Where n is the number of times the occurrence of that word in the input.

The output of the reducer will be like

(<Hii>, 1)
(<This>, 1)
(<Is>, 1)
(<Srikanth>, 1)
(<Welcome>, 1)
(<To>, 1)
(<Hadoop>, 1)
(<Lab>, 1)

The running component of the Hadoop can be checked using the jps command the output of the command is shown in Figure 4. We can access the name node at.

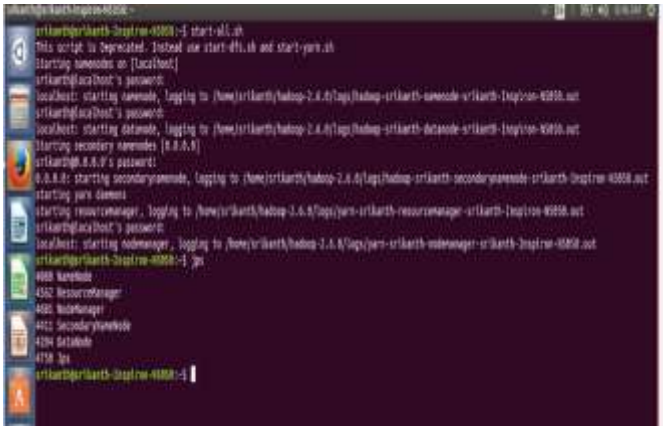


Figure 4. JPS Output

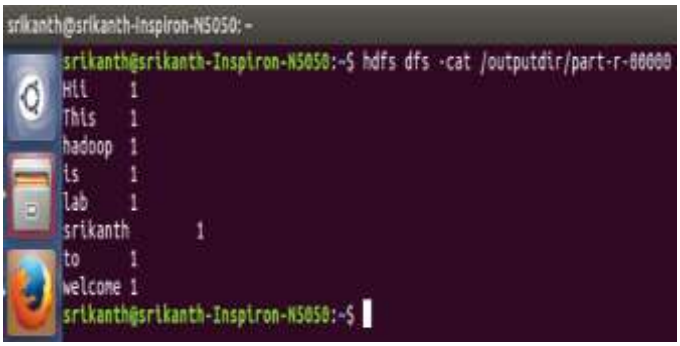


Figure 5. Output for wordcount

CONCLUSION AND FUTURE WORK

We have analyzed the performance of the map reduce task with the increase number of files. We have used the word count application of the Mapreduce for this analysis. The output shows that the Bytes written do not increase in the same proportion as compared to the amount of files increase. For our next work of study we will establish a cluster of nodes and then analyze the behavior of the Map reduce task with a number of files. This can be used in analyzing the files generated as logs. It can also be used for analyzing the sensor's output which is the number of files generated by the different sensing devices.

REFERENCES

1. http://home.web.cern.ch/about/updates/2013/02/cern-data-centre- passes-100-petabytes
2. http://gigaom.com/2012/08/22/facebook-is-collecting-your-data-500- terabytes-a-day/
3. Jeffrey Dean and Sanjay Ghemawat, "Map Reduce: Simplified Data Processing on Large Clusters", Google, Inc.
4. http://raconteur.net/technology/big-ataor- big- statistics.
5. Aditya B. Patel, Manashvi Birla, Ushma Nair, "Addressing Big Data Problem Using Hadoop and Map Reduce", NUiCONE-2012, 06-08DECEMBER, 2012
6. Anirban Mukherjee, Joydip Datta, Raghavendra Jorapur, Ravi Singhvi, Saurav Haloi, Wasim Akram, "Shared Disk Big Data Analytics with Apache Hadoop" 978-1-4673-2371-0/12/ ©2012 IEEE Bernice Purcell "The emergence of "big data" technology and analytics" Journal of Technology Research 2013.
7. Definting Big Data Architecture Framework: Outcome of the Brainstorming Session at the University of Amsterdam, 17 July 2013.Presented at NBD-WG, 24 July 2013 [Online].
8. Ahmed Eldawy, Mohamed F. Mokbel "A Demonstration of Spatial Hadoop: An Efficient Map Reduce Framework for Spatial Data" Proceedings of the VLDB Endowment, Vol. 6, No. 12Copyright 2013 VLDB Endowment 21508097/13/10.
9. A Workload Model for Map Reduce by Thomas A. deRuiter, Master's Thesis in Computer Science, Parallel and Distributed Systems Group Faculty of Electrical Engineering, Mathematics, and Computer Science. Delft University of Technology, 2nd June 2012.
10. http://hadoop.apache.org/docs/r1.2.1/hdfs_design.html