International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-10, August 2020

The File System Recommendations to Reduce the Space and Time Parameters in Hadoop File Storage and Map Reduce Processing of Big Data Applications

Rama Naga Kiran Kumar. K, Ramesh Babu. I

Abstract: ***The study of Hadoop Distributed File System (HDFS) and Map Reduce (MR) are the key aspects of the Hadoop framework. The big data scenarios like Face Book (FB) data processing or the twitter analytics such as storing the tweets and processing the tweets is other scenario of big data which can depends on Hadoop framework to perform the storage and processing through which further analytics can be done. The point here is the usage of space and time in the processing of the above-mentioned huge amounts of the data definitely leads to higher amounts of space and time consumption of the Hadoop framework. The problem here is usage of huge amounts of the space and at the same time the processing time is also high which need to be reduced so as to get the fastest response from the framework. The attempt is important as all the other eco system tools also depends on HDFS and MR so as to perform the data storage and processing of the data and alternative architecture so as to improve the usage of the space and effective utilization of the resources so as to reduce the time requirements of the framework. The outcome of the work is faster data processing and less space utilization of the framework in the processing of MR along with other eco system tools like Hive, Flume, Sqoop and Pig Latin. The work is proposing an alternative framework of the HDFS and MR and the name we are assigning is Unified Space Allocation and Data Processing with Metadata based Distributed File System (USAMDFS).***

Keywords: Analytics, Hadoop Framework, Meta Data based File system, Eco System, Unified Space Allocation.

I. INTRODUCTION

Hadoop distributed File System (HDFS) is the heart of MapReduce (MR) and other eco system tools such as Hive, Pig Latin, Sqoop and Flume. The MR is parallel and distributed processing programming model in Hadoop Distributed File System which completely depends on Distributed File System. All the other eco system tools also depend on HDFS and MR so as to store and perform the processing of the data in the Hadoop eco system. As all these tools again use the HDFS and MR obviously the transfer of the data and storage of the files requires high amount of the time as there is a data traversal from the tool storage to HDFS.

Revised Manuscript Received on August 30, 2020.
* Correspondence Author
Rama Naga Kiran Kumar. K, Research Scholar, Dept of Computer Science & Engineering, Acharya Nagarjuna University, Guntur, Andhra Pradesh, India. E-Mail: karasala_be@hotmail.com.
Prof. Ramesh Babu. I, Professor, Dept of Computer Science & Engineering, Acharya Nagarjuna University, Guntur, Andhra Pradesh, India. E-Mail: rinampudi@outlook.com.

In the current work the discussion is related to reducing the time requirements so as to store and process the data with effective usage of proposed unified approach of data storage which in turn gives the performance aspect of the framework in the efficient manner. We believe that the work is a revolutionary approach in the study of big data scenarios and analytics point of view. Till now the storage requirements and processing time of the huge amounts of the data is very high. The attempt we are making to provide an alternative way of storing and processing of the data in the context of HDFS and MR will sure open the avenues in the new age of the research.

The proposed architecture embeds the storage provisions other than the HDFS and provision of alternative log file usage and possible provision of local file system and other provisions so as to make best usage of the available space and time for at least the pilot or trial run projects in the industry. The organization of the work can be observed as below in Section II the issues related to existing architecture can be observed in Section III proposed architecture of Unified Framework and the process of data storage and programming can be observed along with the architecture model. In section IV the results and discussion can be seen and in Section V the conclusion can be observed.

II. ISSUES IN THE HDFS AND MR MEMORY USAGE

According to the Hadoop framework all the eco system tools along with MR need to depend on the HDFS storage so as to store and access the data. The problem with this methodology is every time the block wise data need to be accessed and make use of the data storage, once the processing is done then writing the data into HDFS is the second stage. The other stage of action is once the data stored into HDFS by MR or other eco system tools data processing it should be accessed via HDFS only. Due to this lot of time will be consumed and incurs additional time requirements for normal data processing also.

The context switching between the interface and HDFS consumes more time and incurs additional resource utilization for the completion of the one normal procedure of storing and performing some king of the processing on that data.

- The initial point is data transmission from the interface of MR/other eco system tools need to traverse up to HDFS.

- The log file/temporary data need to be stored bit efficient as the big data analytics need huge amount of storage even for the log files also.
- Higher amount of the time incurred in the traversal, storage and again for the retrieval of the data to and fro from HDFS and MR logics.
- No unified process to be followed by MR/Hive/Flume/Sqoop/Pig to perform the access of the data and storage of the data.
- Root based access of the HDFS and MR logics so no need of configuring the same for the other eco system tools.
- The problem of HDFS dependency and replication of HDFS causes the additional memory burdens on the file system.
- The Configuration of HDFS and MR related to individual user configuration also leads to additional resource utilization and there is a need of separate monitoring tools for the observation of the cluster activities.

To address all the above-mentioned aspects the current work proposes the method of unified framework so as to handle the possible activities with the help of Local File System inclusion along with HDFS.

By doing so we can expect the usage of additional space in the LFS and reduce the time factor required to context switch between the LFS and HDFS.

To address all these aspects the current work proposes various solutions and a unified architecture of the data storage and processing model so as to considerably reduce the time and space requirements of the said requirements.

III. PROPOSED UNIFIED METHODOLOGY FOR THE EFFICIENT SPACE AND TIME USAGE OF THE FRAMEWORK.

To implement the current work, we have taken the following experimental setup. The operating system is Ubuntu 20.04 LTS and Hadoop 3.2.1 with HDFS and MR configuration. The corresponding files configured were hdfs-site.xml, core-site.xml, mapred-site.xml, hadoop-env.sh and yarn-site.xml.

The work here we considered are storing data with local file system (Lfs_work1) and we have estimated the time factor and importing the same file into HDFS and estimated the time factor.

Obviously, the time taken for the LFS storage is bit less when compared with HDFS import, the reason is for the LFS storage one can use the direct storage to point the source data into the file system.

In case of HDFS import the Hadoop configuration is playing a vital role and the same import (same file with same size) of LFS took higher time to store into HDFS, the reason is not the amount of the data consumed by the file only the thing is traversal and checking of configuration and recheck kind of the activities incurring the additional time and space requirements in to HDFS.

The same can be observed with the following figures.

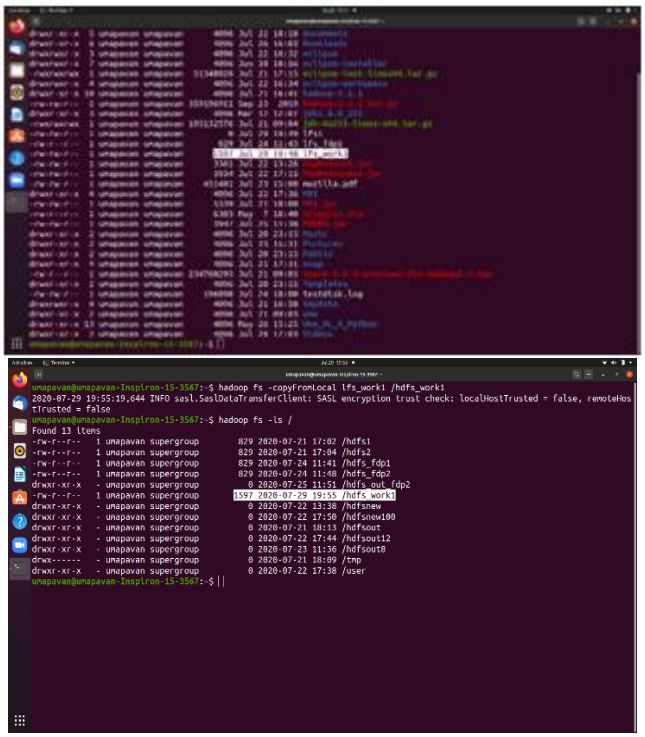
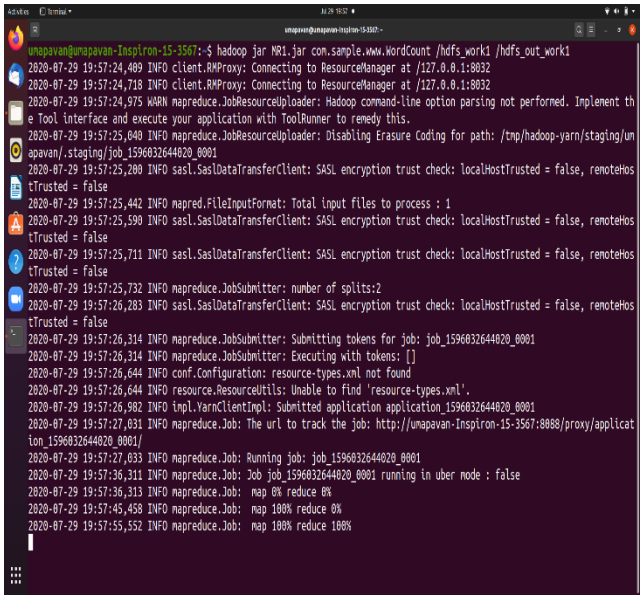
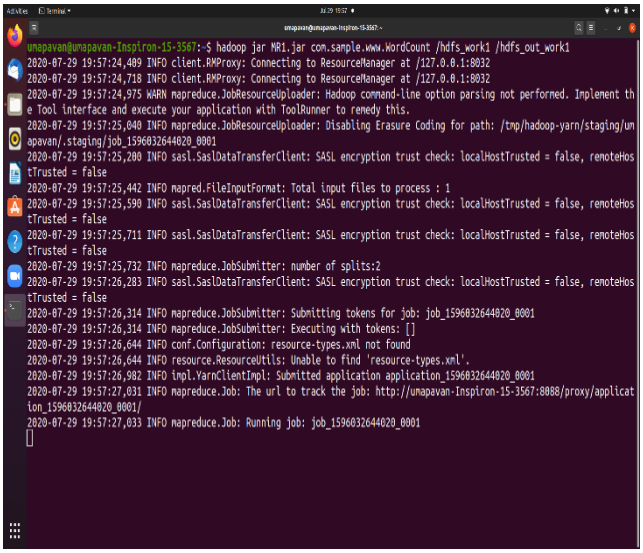


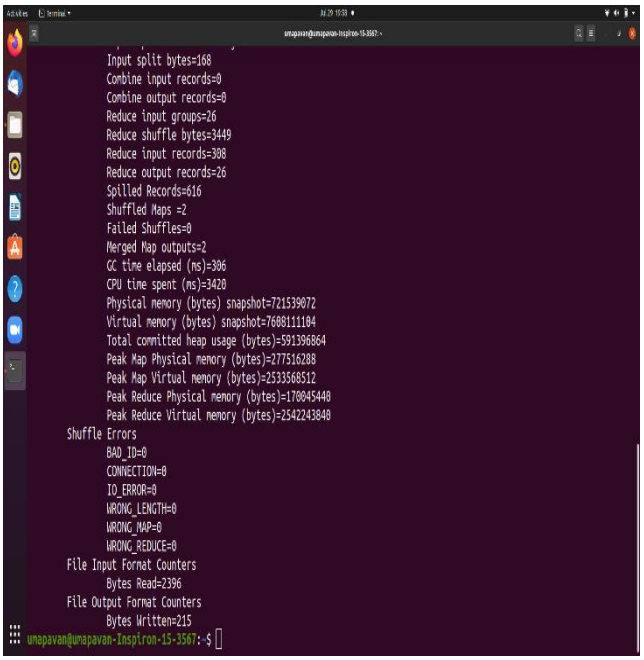
Figure 1: Time taken to store a file in LFS and HDFS.

From the above figure1, we can clearly observe the amount of time required while storing a file in local file system is required is less than the time taken to import the same file into HDFS. Here we are trying to make use of local file system also to perform certain kind of temporary file launchings and log file processing rather than make use of HDFS which obviously reduce the time required to perform import of the file and execution of the programs.

The other observation we are presenting now is the various parameters required to perform the processing of Map Reduce (MR) logic from the point of launching and we have recorded the time factor also the same can be observed from the following figure.







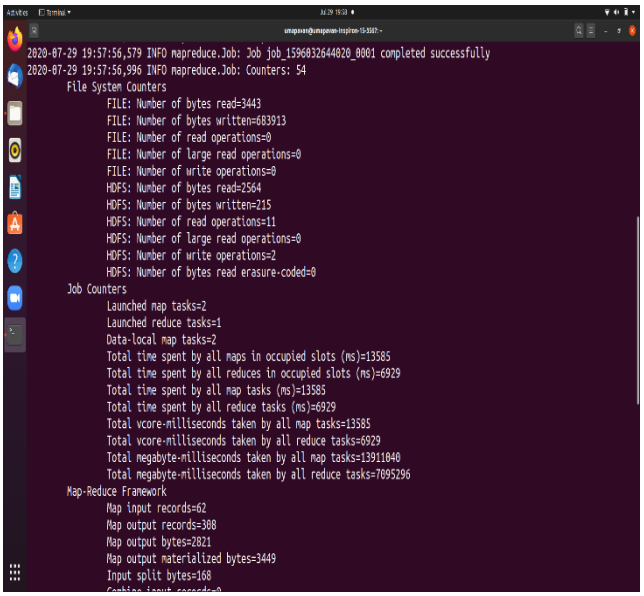


Figure 2: Processing of the .jar file in Map Reduce (MR) along with other parameters.

The main observation here is due to the file access from the HDFS there is need of various parameters and traversing of the data from LFS(such as .jar file and JDK and Hadoop API support) to HDFS(Input file/Directory and output directory) require more support and other configurations in the dealing of the processing.
So, the proposed methodology mainly deals with two important considerations.

- The usage of Local file System in the file storing from the external source like mysql,FB data or Twitter Data according to figure1 it is very evidential that the LFS took less time when compared with HDFS import so the claim is to make use of LFS for storing the external files.
- The second and most required revision is usage of LFS for not only .jar and Java and Hadoop API as the complete build path we are performing in LFS along with Mapper, Reducer and Driver logics.
- The parameters like input file or directory and output directory shall make use of LFS for the log file and other temporary file information rather than every time switching to HDFS.

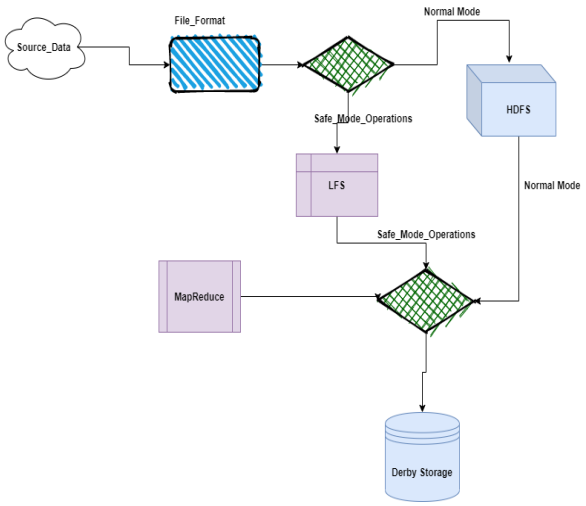


Figure 3: Unified Model of Hadoop Storage and Processing.

The architecture we are proposing is the heart of the work, the main theme of the architecture is to resolve the additional configuration space requirements usage by Hadoop framework, which in turn solves the time requirements as we are providing the alternative memory maps and based on the same configurations and with reasonable changes in the Hadoop configuration files one can achieve the expected phenomenon of reducing space and time parameters in the File System usage of HDFS.
The various components placed in the architecture involves HDFS,MR, other eco system tools (as data base users can go with Hive/Scripting users can go with Pig Latin etc.,) and simple import and export tools like Sqoop and Flume so as to handle structured and unstructured kind of the source data. The scope of the work is not in our current research work but creates an impact in the community of the researcher to come up with various ideas in the field of Hadoop and Big data.

IV. CONCLUSION

The current article described the working of HDFS and MR in the context of huge amounts of the data. We have estimated the time and space parameters required to perform the storage with local file system and HDFS.

There is clear observation that the time required to perform the storage with LFS took less time when compared with HDFS. The other experiment we have done is performing the MR logic so as to execute the parallel and distributed usage of the HDFS. The observation here is the MR process uses the LFS for .jar, jdk and Hadoop API usage and for the input file and output directory the HDFS. So the proposed method of unified approach recommends the usage of LFS to reduce the time and space requirements of MR processing. The article gives the exposure of file system usage in the context of LFS and HDFS, as the time and space requirements of HDFS are devoted to configuration and other activities of the import /export /parallel processing of the given data and once the data is processed again the data is pushed to distributed storage with additional parameters. The article has given the possibilities of utilizing the space of the LFS whenever there is possibility of avoiding the HDFS heavy configuration and context switch with the services. The safe mode option of the file system and processing of the Hadoop Framework is best suitable for the pilot projects or the projects that need the testing of the performance of the cluster. The safe mode allows the cluster to perform the same activities like view of the HDFS file system, and perform the MR tasks with light weight input and the output of the work can be estimated in the LFS directory which in turn reduce the consumption of the resources and saving the time parameter. We believe that the work greatly helps us to improve the time efficiency and reduce the space usage by typical Hadoop frame work, and the Unified framework is a revolutionary attempt in the research of the Hadoop File system along with parallel and distributed aspects of Map Reduce.

REFERENCES

1. Ivanliton Polato, “A Comprehensive view of Hadoop Research- A Systematic Literature Review, Volume 46,November 2019,PP:1-25.
2. Konstantin Shvachko,” The Hadoop Distributed File System” IEEE 2010.
3. Mohd Rehan Ghazi, “Hadoop, MapReduce and HDFS: A Developers Perspective” Procedia Computer Science ,2015.
4. Wu Jun ,” Study of New Materials Design based on Hadoop”, MATEC Web of Conference 61,07016(2016).
5. Himi Egemen Ciritogulu, “A Heterogeneity –aware replica deletion for HDFS” Journal of Big data, October 2019.
6. Sujit Roy,”Hadoop Periodic Jobs Using Data Blocks to Achieve Efficiency”, Indian Journal of Research in Compueter Science and Information Technology,Vol:3, Issue:3,2018.
7. Ronald C.Taylor, “ An Overview of the Hadoop/Map Reduce/HBase/ framework and its current applications in bioinformatics”,BMC Bio Informatics,2010.
8. Jason.C.Cphen, “Towards a Trusted HDFS storage platform”,ACM Digital Library,Vol:19,No:3,July 2014.
9. Aibo Song, “A memory-Based Hadoop Framework for Large Data Storage”,Resource management in Virtaulized Clouds,Hindawi publishers,Volume 2016
10. Tafiq Hassanin, “Severly Imbalanced Big data challenges : Investigating Data Sampling approaches”,Springer, 30 November 2019.
11. Konstantin, The Hadoop Distributed File System,”Symantic Scholar.org,October 2013.
12. D.Borthakur ,”The Hadoop Distributed File System Architecture and Design”, 2017,Apache.org. [13].H.Liao, ”Multi-Dimensional index on Hadoop Distributed File System” 2010, IEEE explore ieee.org. [14].J.Zhang “A Distributed Cache for Hadoop Distributed File system in real-cloud services,ACM 2012. [15].S.Jin,”Design of trusted file System based on Hadoop,2012, Springer.
13. UmaPavan Kumar,”Integration of Hadoop and IOT for better analytics” TEST Engineering and Management,February 2020.
14. UmaPavan Kumar,” Various Issues in Hadoop Distributed File System, Map Reduce and Future Research Directions, International Journal of Pure and Applied Mathematics.
15. Rama Naga Kiran Kumar,”Hadoop Based File System Revision with Derby and Virtual Local File System Models., International Journal of Advanced Science and Technology, Vol.29, June 2020.

AUTHORS PROFILE

K Rama Naga Kiran Kumar completed B.E(CSE), M.Tech(CSE) and pursuing Ph.D. (CSE) from Acharya Nagarjuna University. Areas of interest in AI, Data Mining, Machine Learning, Software Engineering.

Dr. Ramesh Babu I, done his B.E (ECE), M.E (Computer Engg), Ph.D. (CSE) in ANU. His areas of interest in Image Processing, Computer Graphics, AI, Data Mining, Machine Learning, Software Engineering