Hadoop based file system revisions with derby and Virtual Local File System Models

Rama Naga Kiran Kumar. K *[1], Dr. Ramesh Babu. I [2]

*[1]Research Scholar, Dept of Computer Science & Engineering, Acharya Nagarjuna University, Guntur, Andhra Pradesh, India.

[2] Professor, Dept of Computer Science & Engineering, Acharya Nagarjuna University, Guntur, Andhra Pradesh, India.

[1]E-mail: karasala_be@hotmail.com. [2]E-mail: rinampudi@outlook.com.

Abstract

We believe that this work addresses the challenges in the Hadoop file system management in a novel methodology by introducing the concepts of Derby model and Virtual Local System Model (VLFSM).

Keywords: *Hadoop File System, Derby Model, Virtual Local File System, Eco system tools, Map Reduce.*

1. Introduction

Hadoop File System inspired from Google File System follows a typical allocation of memory based on blocks. The typical allocation consists of Text Input Format if we want to allocate the default then we can make use of this format. The user may not always know the input data format in that case the preferable method of file allocation is Text Input Format.

job.setInputFormatClass (TextInputFormat.class) the code snippet helps the developer to handle the input with default methodology.

If the input file format is familiar to the developer then the preferable format is (Key, Value) format. Sometimes there may be a possibility of handling an input file which is in the form of images, then the supported file handling format by Hadoop is Sequence Format. The Sequence Format helps us to convert the image data into encrypted format and then the processing will start by Hadoop engine.

The file system in Hadoop is the base point for all the other eco system tools, running the Jobs and scripts with the ultimate memory usage is from a versatile storage model of Hadoop , the effective utilization of HDFS leads to usage of entire eco system in effective and efficient manner.

The organization of the paper follows like in section II the issues in the existing file system are described, In Section III the proposed solution to the issue is mentioned, In Section IV the conclusion of the work is described.

2. Research Issues in Hadoop File System Usage

As mentioned earlier the backbone of the eco system in Hadoop Frame work is HDFS, the eco system handles various aspects such as import/export of the structured data, importing of the unstructured data like twitter data or any other social media information.

The usage of NOSQL to avoid the key concepts such as primary key or referential integrity kind of the relationships, similarly the concept of normalization is also not there in NOSQL so as to allow the user to have the simplest and huge data storage according to the requirements.

The user familiar with data warehouse and SQL kind of the back ground can develop the query based implementations and they can run the queries with local and Hadoop based file in the form of load data local inpath and load data inpath respectively.

The user familiar with scripting can make use of the scripting with lot of customizations in loading and storing of the data along with user defined functions.

In all the above-mentioned aspects the common problem is allocation of the memory in the block based storage.

Consider the scenario of allocating the 600MB file with replication factor of 3 with the default storage of 128MB. The simple allocation followed by HDFS is 5blocks of 128MB so which leads to the storage of 5 block each 128MB leads to 640MB allocation and with replication of 3 the amount becomes 1920MB which simply holds 120MB without usage.

The same scenario with replication factor 5 with default storage of 128 MB, requires 3200MB which is almost double to replication 3.

Table1: Hadoop Block Allocation Context

Memory Required	Memory Allocated	Blocks	Replication	Observation	
1800 MB	1920 MB	5	3	120MB Allocation	Excess
3000 MB	3200 MB	5	5	200MB Allocation	Excess

The above scenario is so simple but in real time the allocation consumes more amount of memory based on the replication factor.

The other issue in the research of Hadoop environment is static replication factor which is not significant in the real time scenario. The user is working on some sentiment analysis where he wants to have the storage in 2 locations only, so the replication may be 2 in this case.

The user who works for a banking sector to identify the customers who are in need of loan such as personal and home loan may want to replicate the same file in different locations of major cities in India. In this case the replication factor is at least 10(for example).

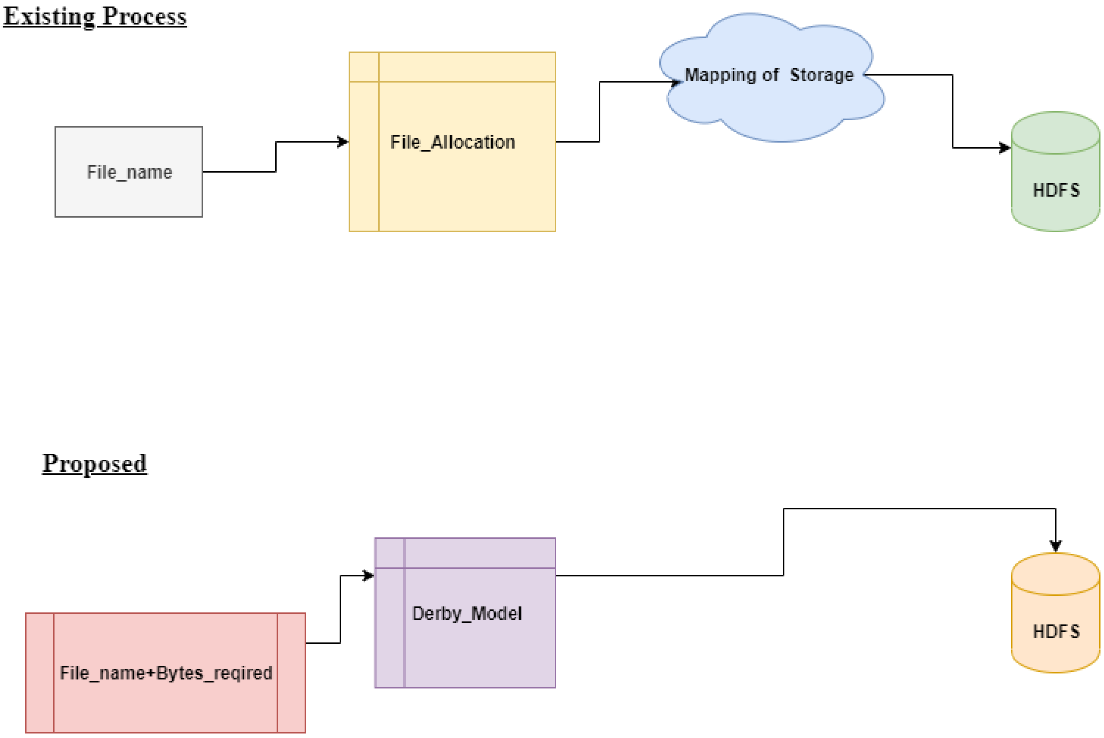
So based on the scenario the architecture must support the notion of dynamic replication factor assignment according to the scenario of the use case.

3. Experimental Setup:

To perform the big data context storage and processing contributions we are making use of Amazon EC2 and EMR usage so as to connect with high end processor. The machine is configured with Hadoop eco system tools like HDFS, Map Reduce along with R base and R-studio so as to perform the aspects of the big data and Machine learning kind of the observations.

4. Proposed Method for the Identified Issues

The existing method of allocation requires the input file and it maps the storage to HDFS which leads to wastage of memory based on the blocks and replication factors.


Existing Process

Proposed

Figure 1: Proposed Derby Model to handle the allocation.

The outcome of this module is to minimize the unused blocks with the intelligent method of tracking the units and we believe that the process yields the best usage of the memory which in turn improves the performance in case of memory intensive based applications. The second contribution is the local memory usage of the machine, in general the Hadoop Map Reduce model is completely depends on HDFS usage and the proposed method here is assisting some of the internal tasks of Map Reduce with the usage of local file system and we have coined the module as **Virtual Local File System Model (VLFS).**

The advantage of this proposed approach is the usage of the local system effectively while performing the Map Reduce model and some of the possibilities in the handling of MR aspects such as Mapper and Reducer configuration details in the form of log files can be redirected to local file system which greatly reduces the context switch of the HDFS every time.

The main purpose of this model is to greatly minimize the context switch timing and effective usage of HDFS and Local file systems which is not focused so far in the literature of the Hadoop frame work research.

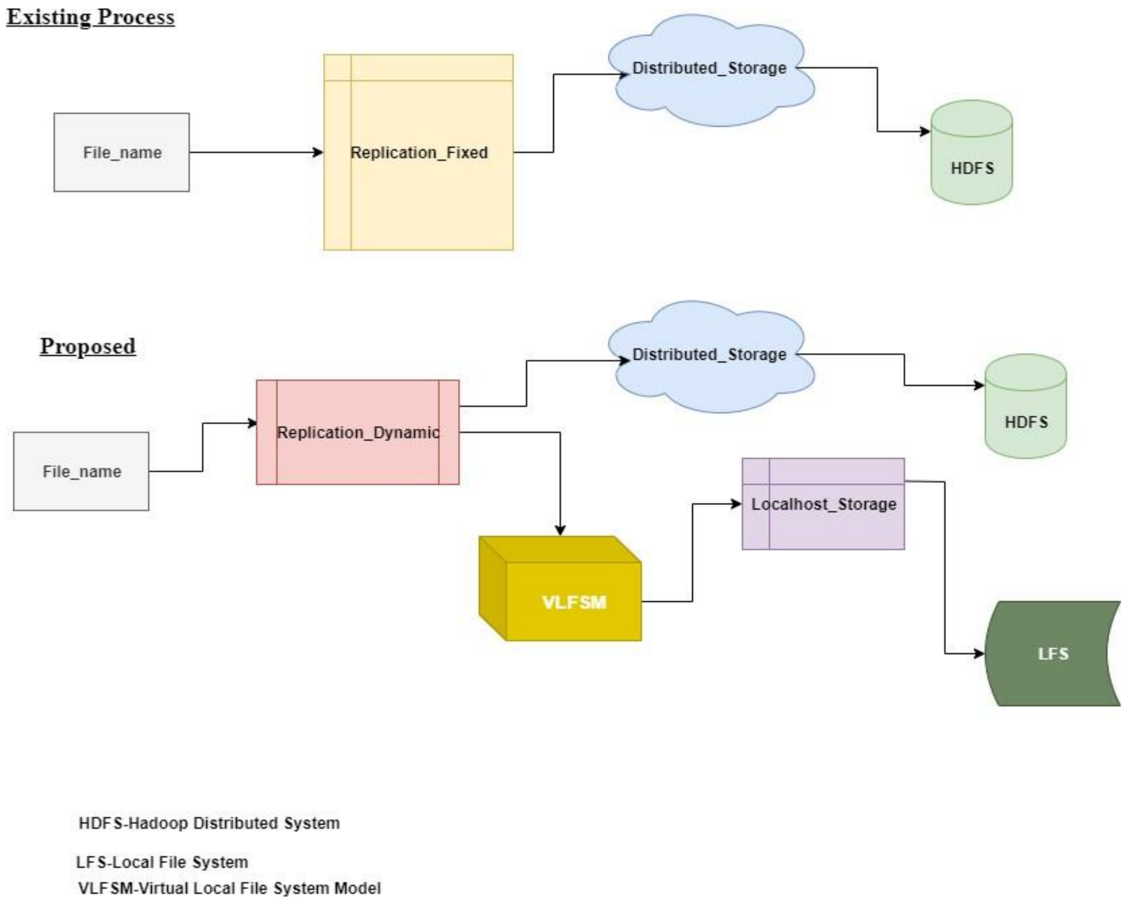


Figure 2: Proposed VLFSM Model to utilize local and HDFS file System.

5. Conclusions

The work deals with the usage of file systems in Hadoop frame work in efficient and effective Manner. The work proposed two facets like Derby model which focus on the usage of the file allocation and address the problems in existing works. The second facet is Virtual Local File System Model which provides the way of dealing with replication of distributed file system with fixed and dynamic models of Hadoop Configuration. We believe that the work provided a landscape to the researchers in the area of Hadoop framework research aspects.

References

[1] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters", Proceedings c. of the 6th Conferece on OS Design and Implementation San Francisco CA, Dec. 2004.

[2] A. Gates, O. Natkovich, S. Chopra, P. Kamath, S. Narayanam, C. Olston, et al., "Building a High-Level Dataflow System on top of MapReduce: The Pig Experience", data bases proceedings, vol. 2, no. 2, pp. 1414-1425, 2009.

[3] M. K. McKusick and S. Quinlan, "GFS: Evolution on Fast-forward" in ACM Queue, New York, NY, vol. 7, no. 7, August 2009.

[4] O. O'Malley and A. C. Murthy, "Hadoop Sorts a Petabyte in 16.25 Hours and a Terabyte in 62 Seconds", May 2009.

[5] K.UmaPavan, , various issues in Hadoop file systems,IJPAM, Volume 120 No. 6 2018, 4441-4451.

[6] Uma Pavan Kumar Kethavarapu, “Various Computing models in Hadoop eco system along with the perspective of analytics using R and Machine learning”, Vol. 14 CIC 2016 Special Issue IJCSAI, PP-17-23.

[7] Uma pavan kumar kethavarapu, “The Ten Ingredients of Data Base Systems for Improving Performance and Their Review Leading to Research Problems”, IFRSAs International Journal Of computing—Vol2—issue 2April 2012, 409-415.

[8] S. Madden, “From Databases to Big Data.”, Internet Computing., vol. 16, no.3, 2012.

[9] PZikopoulos, ,Scope of big data: Analytics for enterprise and streaming data. , 2015.

[10] AMcAfee, E. Brynjol, T.Davenport, J.Patil, , “Big data”, Revolutionary. Harvard Business Review, vol.90, no. 10, pp. 6167, 2012.

[11] RAppu swamy, and ARowstron,“Scaleup vs Scaleout for Hadoop: , in Proceedings of the 4thannual Symposium on Cloud Computing,2013, p. 20.

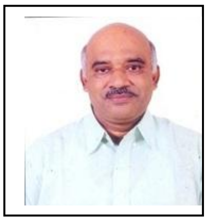
[12] Chun PChen and C. Zhang, “Data intensive applications, challenges,techniques and technologies: A survey on Big Data”, Information . Scienece., vol. 275, pp. 314347,2014.

[13] J. Y. Monteith, J. D. McGregor, and J. E.Ingram, “Hadoop and its Evolving Ecosystem.”, in IWSECO@ ICSOB, 2013,pp. 5768.

Authors

Rama Naga Kiran Kumar K completed his B.E(CSE) from University of Madras, M.Tech(CSE) from JNTUH and pursuing Ph.D. (CSE) from Acharya Nagarjuna University. His area of interest is 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 area of interest is in Image Processing, Computer Graphics, AI, Data Mining, Machine Learning, Software Engineering

.