An Efficient Approach to reducing the memory overhead for handling large number of small size files in HDFS with Big Data

[1]Rajesh R Savaliya [2]Dr. Akash Saxena

[1]Research Scholor, Rai University, Vill. Saroda, Tal. Dholka Dist. Ahmedabad, Gujatat-382 260

[2]PHD Guide, Rai University, Vill. Saroda, Tal. Dholka Dist. Ahmedabad, Gujatat-382 260

Abstract :*Big-Data handling is the supper and most demand now day in the world of software development. Hadoop is the open source software framework for distributed file processing of big data in the cluster on commodity hardware. This framework is the most powerful to store the big data. NameNode is the component of HDFS which is used to store the metadata or all files, directories and blocks. HDFS specially design to handle the Big size file, but this frame work is not properly handle massive number of small size files. This paper introduce we propose the how memory overhead of the NameNode for data storage will be reduce with respect to storing large number of small size files in HDFS. These propose approach will more useful for understanding how memory consumption and NameNode workload will be reduce in Hadoop Distributed File System.*

KEYWORDS: **BIG-DATA, HADOOP, HDFS, NAMENODE, DATANODE, META-DATA.**

1. INTRODUCTION

Big-Data is the new key word used to express large volume of structured, unstructured as well as semi-structured data. Those big data handle by the open source hadoop framework [1]. It is used to handle the large size of data files, and there for it suffers from storing and handling large number of small size files with respect to different application areas like e-learnig, e-business, energy, social network, biology, etc[4]. All the storage control of the data in hadoop is done by the NameNode. NameNode is used to create and handle the metadata management for each file, directories and blocks. Each file, directories and blocks act as one object in HDFS. For Each object, it required the 150 bytes memory to store the metadata for each. HDFS divide the data into the block of 128MB by default [7]. And NemeNode store the metadata for that blocks and DataNode store the actual DataNode

2. HADOOP DISTRIBUTED FILE SYSTEM (HDFS)

Hadoop Distributed File system is the open source framework to handle massive size data file. Hadoop Distributed File system is the design by Apache [6]. So many applications implemented with the help of the concepts of HDFS, such as social media like Whatup, Facebook, Amazon. Hadoop Distributed File system which is used to maitain large data size file. HDFS mainly divided into two sections such as NameNode and DataNode [2].

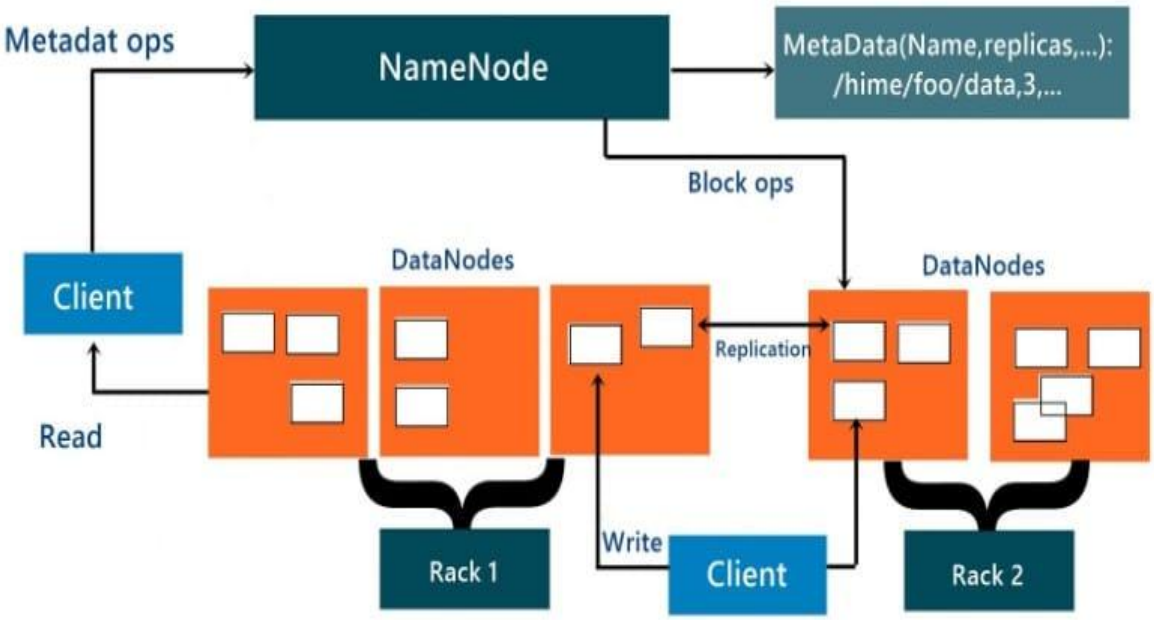

Figure 1: HDFS Architecture [3].

2.1 NameNode:-

The NameNode component is used to manage and store the metadata of of data file stored in the DataNode in HDFS[5]. NameNode is also used to maintain heartbeat means maintain the link between NameNode as well as DataNode. NameNode maintain the access and storage control of data file. NameNode also maintain the Metadata for replication. By default replication factor is three. NameNode take the 150 bytes memory size to store the metadata for each objects [8].

2.2 DataNode:-

The DataNode is used to deal with the storage of large amount of data file in the freely available datablocks in the HDFS cluster. DataNode is also used to store replication factor in different block available in the cluster of HDFS. DataNode also supports NameNode component to maintain the heartbeat between them.

HDFS especially design to handle the Big size file, but this frame work is not properly handle massive amount of small size files, so we provide the following proposed approach to resolve the large number of small size storage problem with metadata creation and management for each small size files by the NameNode.

3. OPTIMIZE STRATEGY FOR STORAGE OF SMALL SIZE FILES WITH RESPECT TO REDUCETHE OVERHEAD AND MEMORY CONSUMPTION OF THE NAMENODE IN HDFS

In this propose optimize strategy for storage of small size files with respect to reduce the overhead and memory consumption of the NameNode in HDFS. In this propose strategy divided in two sections which are File Processing Unit and File Merging Unit.

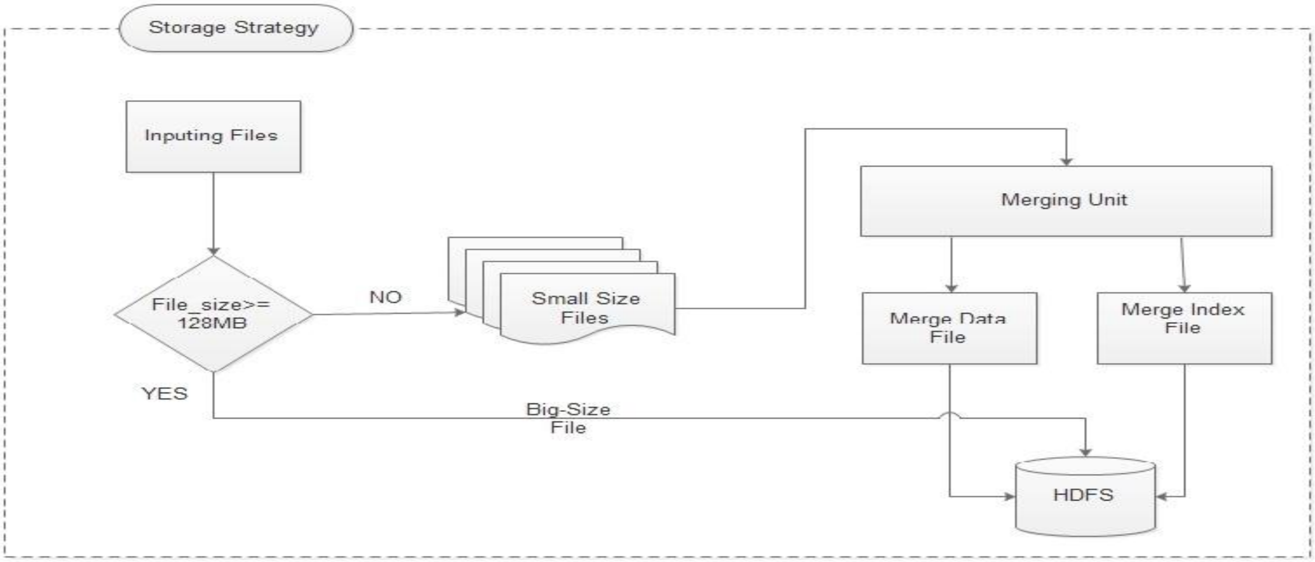

Figure 2: Proposed Optimize File Storage Strategy by NHDFS.

3.1 File Processing Unit:-

File Processing Unit takes files as an input from the user. File processing unit judges that inputted files are of “Big-Size file” or ”Small Size File”. If file size is greater than 128MB, those files are count as Big Size Files. If file size is less than 128MB, those files are count as Small Size Files. If file is a Big Size File then it will be directly store in HDFS while file is small size then small files will be forwarded to the File merging Unit.

3.2 File Merging Unit:-

File merging Unit will received the files send by the file processing unit. After the receiving small size files merges into Data merge files. According to merging merge index file is also created. After the completion of merging process merge data file and merge index file forwarded to storage into the HDFS.

4. RESULT AND DISCUSSION

Each files, blocks as well as directories count as one object in HDFS. Name Node creates the metadata in the memory for each and every object. Name node occupies 150 bytes in the memory for metaData.150 bytes used for metadata storage by the Name Node. HDFS divides the data files entered by the users into different data blocks of128 MB. And the name node stores the Namespaces for each data block.

For the storage calculation, we resolve the storage memories usage for three parameters of files as a 10000, 20000 and 30000 number of small size files correspondingly. The memory used by the Name Node in the MB size using the Original HDFS approach with Propose NHDFS(Novel Hadoop distributed file system) Approach.

Sr.No	No. Of Small Size Files	Total size for Small Files Together	Memory used by Original HDFS	Memory used by Propose NHDFS
1	10000	1GB	1.43 MB	0.0011444 MB
2	20000	1GB	2.86 MB	0.0011444 MB
3	30000	1GB	4.29 MB	0.0011444 MB

Table 4.1 Evaluation of Memory used by HDFS and NHDFS

In the table 4.1 we easily reached the amounts of memory consumption by the original HDFS approach and Proposed NOHDF approach. We used the parameters as 10000, 20000 and 30000 small files for estimation of memory used prerequisite with the total size of small files with of 1GB files size together. The on tops of parameters are used to estimate the memory uses by the Name Node in original HDFS with projected NOHDFS.

In follow the Figure 4.1 we create graph for memory used by name node to store the metadata base on top of table 4.1.

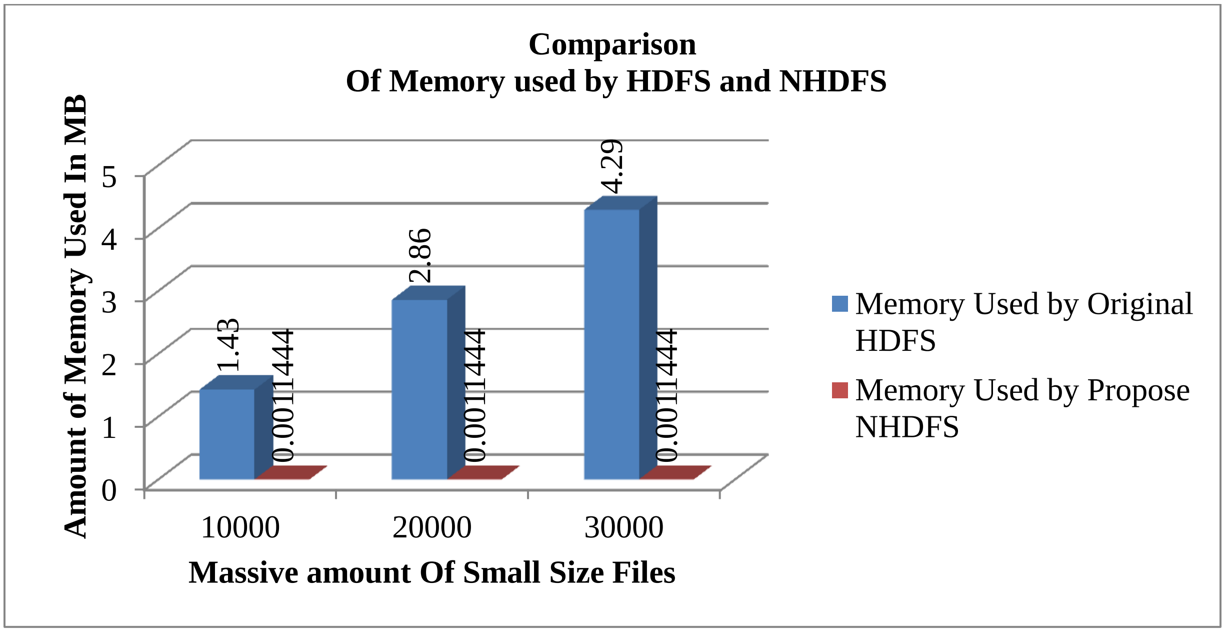

Figure 4.1 NmaeNode memory utilization graph based on table 4.1

5. CONCLUSION

Hadoop distributed file system is the great and world widely popular framework to work with the massive size of file. We uses the hadoop to manage the massive size file. Currently one major research demand is to include the working support of large number of small files in HDFS. As we discuses on the top section is that HDFS cannot working massive amount of small files as wall. In order to solve this research problem, we study and analyze different types of available HDFS system and we also design optimize storage strategy based on HDFS and based on that we propose the novel and optimized hadoop distributed file system (NHDFS) architecture to handle the small size files in HDFS. And also projected approach reduce memory overhead of the NameNode with respect to store the Metadata.

6. References

[1] http://hadoop.apache.org.[Accessed: Oct. 11, 2018]

[2] http://en.wikipedia.org/wiki/Big_data .[Accessed: Oct. 19, 2018]

[3] https://intellipaat.com/tutorial/hadoop-tutorial/hdfs-overview/.[Accessed: Oct. 19, 2018]

[4] Yingchi Mao,Bicong Jia et.al, "Analyzing Optimization Scheme for Small Files Storage Baseds On Hadoop Distributed File System."International Journal of Database Theory and Application , PP. 241-254, 2015.

[5] Shreikrishna Utpat, K.A Dehamane et.al, "An optimized Storing and Accessing Mechanisam for Small Files on HDFS", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 5, Issue-1, PP. 673-677,1 January 2015.

[6] Bharti Guptu, Rajender Nath and Girdhar Gopal, "A novel Technique to Handle Small Files With Big-Data Thequenology", International Journal of Scientific & Engineering Research, Vol. 7, Issue-12, PP. 48-52, December 2016.

[7] Monica B.Bisane, Pushpanjali M et.al, "Improving Access Efficiancy of Small Files in HDFS", International Journal of Scientific & Engineering Research, Vol. 7, Issue-2, PP. 68- 72 February 2016.

[8] Guru Prasad M S, Nagesh H R and Swatithi prabhu, "An Efficient Approach to optimize the Performance of Massive Small Files in Hadoop MapReduce Framework ", International Journal of computer Science and Engineering, Vol. 5, Issue-6, PP. 112- 120 June 2017.

Authors' Profile

[1]Mr. Rajeshkumar Rameshbhai Savaliya from Ambaba Commerce College, MIBM & DICA Sabargam and master degree in Master of science and Information Technologies(M.Sc-IT) from Veer Narmad South Gujarat University.Rajesh R Savaliya has teaching as well programming experience and PHD Pursuing from RAI University.

Co-Authors' Profile

[2]Dr. Akash Saxena PhD Guide from Rai University

https://intellipaat.com/tutorial/hadoop-tutorial/hdfs-overview/