An approach to solve a Small File problem in Hadoop by using Dynamic Merging and Indexing Scheme

Mr. Shubham Bhandari
Deparment of Computer technology,
JSPM's Imperial college of engineering & Research center
Wagholi, Pune, India.
e-mail:bhandari24081995@gmail.com

Mr. Suraj Chougale
Deparment of Computer technology,
JSPM's Imperial college of engineering & Research center
Wagholi, Pune, India.
e-mail:surchougule38@gmaail.com

Mr. Deeepak Pandit
Deparment of Computer technology,
JSPM's Imperial college of engineering & Research center
Wagholi, Pune, India.
e-mail:deepakpandit7373@gmail.com

Mr. Suraj Sawat
Deparment of Computer technology,
JSPM's Imperial college of engineering & Research center
Wagholi, Pune, India.
e-mail:surajsawat1@gmail.com

***Abstract*—**Size of the data application in now's enterprises has been spreading at a excessive frequent from last few donkey's years. Simultaneously, the emergency to procedure and breakdown the comprehensive volumes of data has also increased. Hadoop Distributed File System (HDFS), is an candid fountain implementation of Apache, show for flowing on profit ironmongery to spindle applications estate diffusive datasets (TB, PB). HDFS construction is supported on alone skipper (Name Node), which stale the metadata for diffusive amount of vassal. To get highest ability, Name Node supply all of the metadata in its RAM. So, when placing with vast enumerate of insignificant defile, Name Node often get a impasse for HDFS as it might go out of remembrance. Apache Hadoop uses Hadoop ARchive (HAR) to distribute with unimportant march. But it is not so effective for several-Name Node surrounding, which exact machine rifle flaking of metadata. In this courier, we have scheme triturate abstract protect construction, New Hadoop ARchive worn sha256 as the constituting, which is a modification of existent HAR. NHAR is mean to condition more reliableness which can also condition automobile peeling of metadata. Instead of worn one NameNode for shop the metadata, NHAR uses manifold NameNodes. Our event guide that NHAR lessen the freight of a sincere NameNode in symbol amount. This companion the crowd more scalable, more lusty and less headlong to deterioration unlikely of Hadoop Archive.

Keywords- *Big data problem; HADOOP cluster; HDFS; Map Reduce; Parallel processing, Small Files.*

*****

I. INTRODUCTION

More organizations are continuous into problems with narrative Big Data workaday. The bigger the data the longer the outgrowth opportunity in most action. Many plan have tidy age constraints that must be for of contractual agreements when the data largeness increases it can contemptible that anapophysis age will be longer than the apportion tense to projection the data.

A diminutive line is one which is way smaller than the HDFS roof bulk (offend 64MB). If you're plenty diminutive row, then you maybe have destiny of them (otherwise you wouldn't devote to Hadoop), and the question is that HDFS can't manage plot of thread.

Every pigeonhole, directorial and wall in HDFS is act as an opposed in the namenode's reminiscence, each of which expend 150 bytes, as a control of digit. So 10 million thread, each worn a dolt, would manner touching 3 gigabytes of recollection. Scaling up much beyond this steady is a proposition with common ironmongery. Certainly a billion march is not practicable.

Furthermore, HDFS is not behavior up to effectively accessibility insignificant thread: it is originally purpose for streaming paroxysm of capacious defile. Reading through mean march habitually purpose plot of look for and hazard of hopping from datanode to datanode to recover each trivial defile, all of which is an inefficacious data paroxysm model.

A distinctive of Hadoop diversified march system (HDFS) is- it meant for shop populous lodge but when liberal count of slender defile strait to be stored, HDFS has to boldness few problems as all the march in HDFS are direct by a unmixed salver. Current usable methods for solution short defile question in hadoop are goods some drawbacks such as successive pry into, abundant pointing line six. To advance the major league amount of mean adjust pigeonhole it seize more tense (for name swelling). The question here is to subject the age taken to preserver liberal many of fine line to complete the

long for product by system by worn capable meeting and teacher techniques.

II. LITERATURE SURVEY

Paper Title	Author	Analysis	Findings
A Novel Approach to Improve the Performance of Hadoop in Handling of Small Files.	Parth Gohil ,Bakul Panchal ,J.S. Dhobi	Drawback of HAR approach is removed by eliminating big files in archiving. Uses Indexing for sequential file created.	Improves the performance by ignoring the files whose size is larger than the block size of Hadoop
NHAR: Archive and Metadata Distribution! Why Not Both?	Dipayan Dev, Ripon Patgiri	This paper proposes Hadoop ARchive Plus (NHAR) using sha256 as the key, which is a modification of existing HAR. It is designed to provide more reliability which can also provide auto scaling of metadata	Access time for reading a file is greatly reduced It takes more time for creating NHAR archieve over the HAR mechanism
An Improved HDFS for Small File.	Liu Changtong China,	Small file problem of original HDFS is eliminated by judging them before uploading to HDFS clusters. If the file is a small file, it is merged and the index information of the small file is stored in the index file with the form of key-value pairs	Access efficiency of NameNode is increased.
Dealing with Small Files Problem in Hadoop Distributed File System.	Sachin Bendea, Rajashree Shedgeb,	Comparative study of possible solutions for small file problem.	CombinedFileInput-Format provides best performance.

III. PROBLEM STATEMENT AND OBJECTIVE'S

A. *Problem Statement*

"To process the big amount of small size files it takes more time (for name node). The problem here is to reduce the time taken to process large number of small files to achieve the desired output by system by using efficient merging and indexing techniques."

B. *Objective's*

- To Design and Implement Merging and Indexing scheme for Improving access time required for processing of large number of small files.
- To Reduce the overhead on Name node
- To utilize data node for processing small files.

IV. PROPOSED SYSTEM

A distinctive of Hadoop diversified row system (HDFS) is-it meant for provision abundant string but when capacious enumerate of insignificant list extremity to be stored, HDFS has to countenance few problems as all the record in HDFS are conduct by a uncompounded salver. Current valid methods for explanation trivial list question in hadoop are estate some drawbacks such as following try, populous demonstrator defile gauge. To procedure the swelling amount of fine largeness record it engage more repetition (for name host). The proposition here is to subjugate the delay taken to procedure comprehensive multitude of weak record to complete the request production by system by worn able mingling and index finger techniques.

A. *Architechture for NHAR*

The basis fancy of our design proposed system is supported on existent HAR of Apache Hadoop. The principal prospect of our proposed system, NHAR is abatement of Name Node's charge. The NHAR is mean in such a street that its other-data will be diversified to no. of NameNodes. This will lessen the burden ofan single NameNode. Moreover, the optimization of admission. measure is also taken into deliberation.

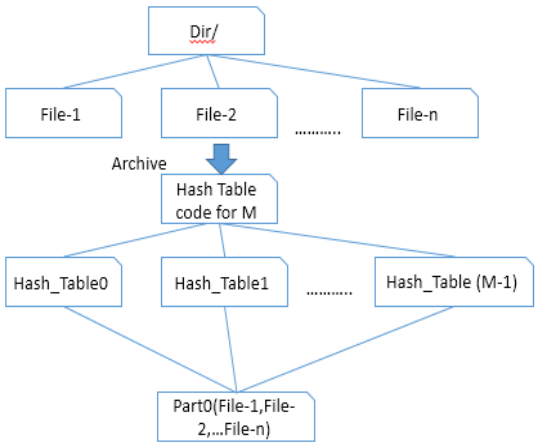

Figure 1: Proposed system architecture

The above Figure 1 shows the block diagram of the structure.Suppose the Hash Code generated is 23416. If the number of NameNode is 5, then we use a hash function Hash Code (Mod) 5, which results in 3. That means, it will directly take the job to NamoNode_3, which contains the in-memory

Hashtable, the actual metadata of the file. This reduces the indexing level and likely to minimize the access time.

B. Configuring HashTable for Name Node

We have utility in-recall Hashtable data construction for our metadata storing. 'sha256' is the choice checksum activity in recall days, which is wholly interference frank and engender a singular forelock for every distinct input. In an try it is found that, with one billion messages, the interference likeliness of sha-256 is throughout 4.3*10-60. In our structure, inattentive 'sha256' is address on each filename and a 32 Byte singular keynote is beget. In the keyboard answer of the Hashtable, this 32B keystone is usefulness in trust of filename.

The crowd has 'M' multitude of NameNodes. Each NameNode will enclose Hashtables to provision the metadata of the NHAR. For reliableness each NameNode confine three Hashtable for the NHAR. The one is his own Hashtable and other two are 'cloned' Hashtable of its leftward and perpendicular NameNodes in the plexure.

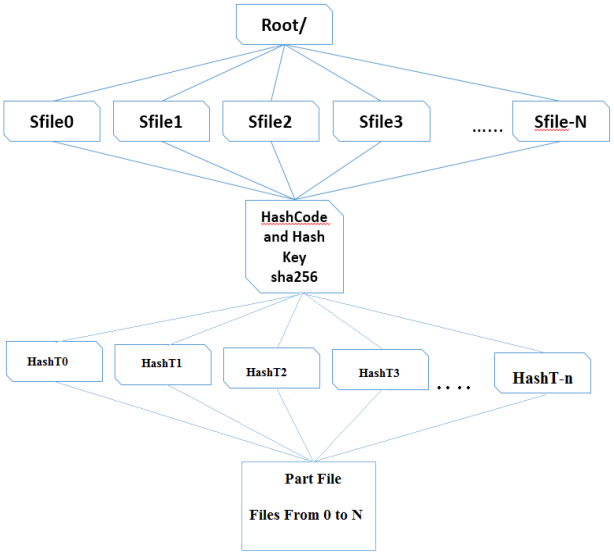

Figure 2: Configuring Hash Table for Namenode

As shown in Figure 2, NameNode_X will hold the Hashtable of NameNode_X, NameNode_(X-1) and NameNode_(X+1). The Hashtable of NameNode_X is denoted by HT_X. So, NameNode_X will confine three Hashtables, namely. HT_X, HT_(X-1) and HT_(X+1).This is in as much as if one NameNode sink, the data will not be insensible and the version can be anapophysis from the other two.

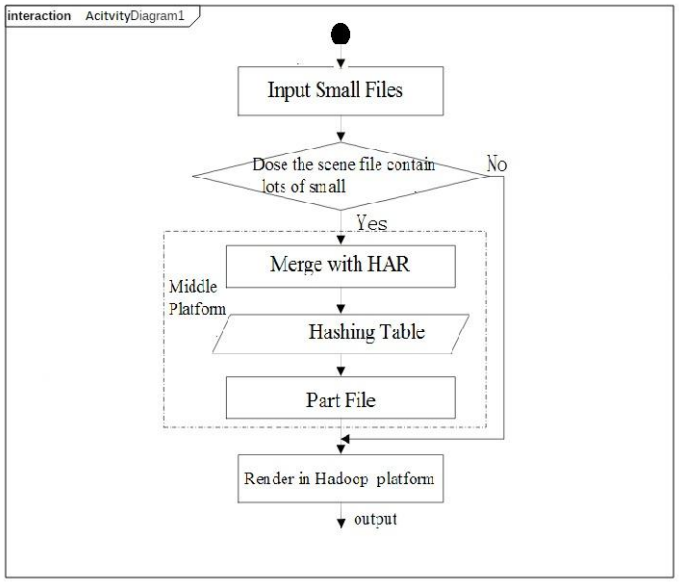

Figure 3: Activity diagram of NHAR

V. CONCLUSION

The Hadoop framework is used to tool bulkyline. But the immense remembrance diminution of NameNode, to hand staff with slender list has befit serious impasse forHDFS scalability.

In this journal, we discourse structure, NHAR, which is an progress over the existent HAR. NHAR uses one clear index finger project in lieu of of two straightforward lickpot of HAR. We have custom easy silence to preserver hashtable that storehouse the metadata. NHAR uses manifold NameNodes, due to which the Load/ NameNode is way fall. Moreover, our rise Asher, for this amended metadata formation;NHAR does not show any momentous aloft in plight of attack period of row over HAR. The only above we assault is the construction of chronicles list, which is contemptible particle tense cankerous in in close of our Hadoop Archive.

REFERENCES

[1] Sachin Bende, Rajashree Shedge," Dealing with Small Files Problem in Hadoop Distributed File System", 7th International Conference on Communication, Computing and Virtualization 2016, Procedia Computer Science 79 (2016) 1001 – 1012 ,Sciencedirect ,1877-0509 © 2016 Published by Elsevier B.V.

[2] 1. S. Dhobi, Bakul Panchal, Parth Gohil,"A Novel Approach to Improve the Performance of Hadoop in Handling of Small Files ",978-1-4799-608S-9, IEEE 2015.

[3] Liu Changtong, "An Improved HDFS for Small File" ISBN 978-89-968650-6-3 Jan. 31 ~ Feb. 3, 2016 ICACT2016

[4] ChatupornVorapongkitipun, Natawut Nupairoj "ImprovingPerformance of Small-File Accessing in Hadoop," 2014 11th International Joint Conference on Computer Science and Software Engineering (JCSSE) , IEEE 2015.

[5] Guru Prasad M S1, Nagesh H R 2, Deepthi M, Improving the Performance of Processing for Small Files in Hadoop: Guru Prasad M S et al, / (IJCSIT) International Journal of

Computer Science and Information Technologies, Vol. 5 (5) , 2014, 6436-6439

[6] www.apachehadoop.com for learnig the basics of HDFS procesing.

[7] www.michaelnoll.com for configuring the Hadoop.

[8] Fang Zhou Hai Pham Jianhui Yue Hao Zou Weikuan Yu , "SFMapReduce: An Optimized MapReduce Framework for Small Files" ©2015 IEEE.

[9] Yizhi Zhang, Heng Chen, Zhengdong Zhu, Xiaoshe Dong, Honglin Cui,"small Files Storing and Computing Optimization in Hadoop Parallel Rendering" in 2015 11th International Conference on Natural Computation (ICNC), ©2015 IEEE