Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 10, October-2016

A Review on an Efficient Approach to Manage Small Files in Distributed File Systems

Aakash Patil, Ganesh Sagare, Kunal Saraf and Prof. S.A.Ahirrao.
UG Students
Computer Engineering,
Sandip Institute of Engineering and Management, Nashik.
Assistant Professor, Computer Engineering, Sandip Institute of Engineering and Management.

Abstract: **- Nowadays, to manage excessive number of small files is became a challenge in Distributed File System. Currently, the combined block storage technique is used to store the files this technique is used in existing system such as Extfs and Xfs. This technique is liable to inefficiency when accessing files randomly. We present the proposed system to manage small files which is based on simple metadata and storage architecture.**

Our system focuses on replacing the existing system drawbacks in Data servers that used to store excessive number of small files and retrieval of files in a better way. We designed new metadata structure which will decrease the size of original metadata that will help to increase the speed of file accessing.

Keywords: ***Information System, Information Storage And Retrieval, Indexing Methods., Content Analysis., Computing Methodologies, Document Processing, Various Types of Files.***

1. INTRODUCTION:

- *What is Metadata ?*

Metadata is consist of data related data that means in file system metadata contains the information which is helpful to search the files in file systems for eg. Address of the file, size of the file, modified date of updated information etc.

Nowadays, Everyone is using social networking and e-commerce websites for communication and purchasing purpose by considering the usage of the websites which required to store the data which is small in size then there is the difficulty in storing and retrieving the files which are smaller in size and the number of this files are bulk because of many users are frequently uploading or modifying the data in the storage space.

So, the managing this small files is became a problem in distributed file system because of the metadata generated by the files is bigger in size. In some cases the files are rarely modified or updated and the size of this file is in between 1kb's to 10kb's such as pictures, text etc. uploaded on social networking and e-commerce websites in daily or timely basis.Distributed file system is based on storing and accessing files based on simple client-server architecture. In distributed file system all data iscopied and placed on the different data servers and the information about the data is stored in which are then connected in network [8].

A client or user searches the file using metadata server other than the using the actual location of that file the same process is used in existing system, client request the file which is stored in a distributed file system by using two phases.

1.Client sends the query containing about the data needed to the metadata server and gets the IP address of data server which stores the target file.

2.In next phase connection between data server and user is established and granted for fetching the data file.

- *Existing System of Distributed File System:*

Global File System (GFS) and Hadoop Distributed File System (HDFS) it uses divide block storage technique to manage files which are big in size in this big files are divided into the different fixed size data blocks usually they are of 64MB in size. Most of the files are small files the amount of stored files increases rapidly this result in generating excessive amount of metadata on the metadata server here it causes low performance in accessing files stored on the data server.

Nowadays, combined block storage is suitable technique to process large amount of small files in this technique small files are combined together in big data blocks to minimize the amount of metadata on metadata server. It is used in existing file system like Ext4 and Xfs and has a low performance to process excessive number of small files[7].

- *Why we are shrinking the size of metadata?*

In our proposed system the main reason behind shrinking the size of metadata is in DFS when we are storing the file, the size of its metadata is big in size because of it contains every attributes as discussed earlier. Because of these the accessing speed of a particular file takes more time. In our system the metadata will contain only two things that are size of the file andphysical address of that file so that accessing speed can be increased.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 10, October-2016

2. SYSTEM ARCHITECTURE:

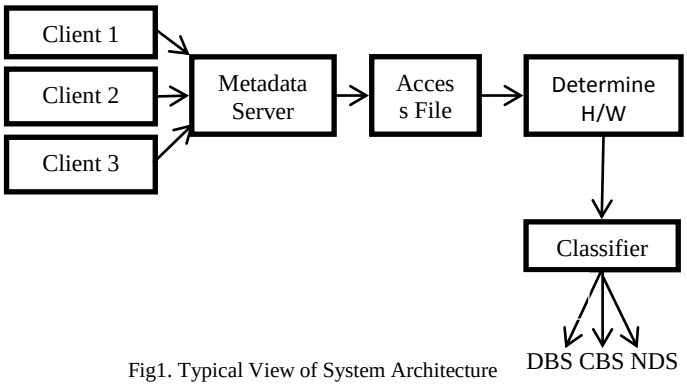

Fig1. Typical View of System Architecture

The system architecture defines the flow of accessing the files from the Distributed File System.

- Clients are the actual users who query the required file to obtain the contents of data.
- Metadata server contains the metadata (contains file attributes) of the files.
- Access File is the file which is requested by the client.
- The component determine hardware configuration detects the system hardware configuration of the client side system that will decide the client can access or cant access the data from the distributed file system.
- Classifiers classifies the data types in three ways:
- DBS(Divided Block Storage)
- CBS(Combined Block Storage)
- NDS(NoSQL Database System)

Combined Block Storage System is used to store the small files in the server. In these all small files are combined and stored in one block.

Divided Block Storage is used to store the large files. In these approach the large files are divided into small files usually these files are stored in different memory blocks of 64kb.

NoSql Database System these system is used to store the bit level data.

In existing system the classifiers are used to locate the files in above three different locations DBS contains the large files, CBS contains the small files and NDS contains the byte level files so because of the three approach the time complexity is increased so in proposed system we are combining these three techniques in a single storage architecture that will help to increase the performance of the accessing time and reduces the time complexity.

3. LITERATURE SURVEY

Granrt Mackey,Saba Sehrish,Jung Wanvg [1] (Granrt Mackey, 2009):In this paper it is given about to improve metadata management for small files in HDFS. This scheme is based on the assumption that each client is assigned quota in file system for the SPACE as well AS NUMBER OF FILES. The compression method "harballing" provides by hadoop is used.The problem occurs in this paper is that as the fixed quota is assigned for the space and number of files. Two cases occur that is

1. User reaches the limit on number of files but the space is still underutilization.

2. User reaches the limit on space but, number of files is underutilization. In above both cases user will not be allowed to proceed. Because of inability to create new files in respective directory.

Qinqin He,Zhanhuai Li,Bo Wang,Huifeng Wang,Jian Sun:(Qinqin He, 2011)[2]In this paper the scientist has given about how to enhance system's performance and how to optimize system under the different configuration and to. In this paper they discussed that how to configure lusture can play its optimal performance and implements specific comparison testing respectively from network performance, lusture file system setting and application. Infiniband networks architecture of the application and given full play the advantages of lusture file system. Set of smaller bar lusture system gives the better aggregate performance, Lusture compartmentalization also greatly affect system performance. The size of read-write block should be accessed between 2MB-4MB.How to integrate the underlying storage optimization lusture in lusture file system and to organize the stored data in better way are the future work.

Randolph Y Wang,Thomas E Anderson:(Randolph Y Wang, 1993)*[3]* In 1993 generation of file system an inadequate in facing challenges of wide area networks and massive storage. XFS is a prototype file system developed to explore the issues brought about by these technology advances. It organizes hosts into a hierarchical structure so, locally within the cluster of workstation can be better exploited. XFS achieve better performance and ability then current generation network file system runs in wide area.

Difficulties are flow to deal with machines that do not respond to ownership revolves due to crashes or network partition. Another difficulty is backup when storage is concentrated on several gigabytes of disk on few server disaster recovery can be accomplished by rolling the whole world back to take. Security concern is another difficulty ,because of ownership is not defined and avability of the file is open source and any untrustworthy owner can gain access to the files.

S. Anjanadevi,D. Vijaykumar. Dr. K. G.Shrinivasan:(S. Anjanadevi, 2014) *[4]* Cloud computing is an emerging computing model wherein the tasks are associated to software, combination of connection and service accessed over network. They proposed the indexing and metadata management which help to access the distributed data with reduced latency. The metadata management can be enhanced for large file system application. Designing the metadata and attributes is important for efficient retrieval of the data.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 10, October-2016

There has been a significant advancement in the area of cloud computing. However, the knowledge about the storage in cloud is not yet discovered properly. Thisresearch project is helpful in acquiring the storage with index and metadata. It will help to minimize the search time of consume and fast access of the data.

Xian Tao, Liang Alei(Xian Tao, 2014) *[5] :*small file access management based on GlusterFS is a strategy to optimize small files reading and writing performance on traditional distributed file system. Traditional file system like Glusterfs stores data within local fie system, which shows the bottleneck on file metadata lookup in this research the redesigning of metadata structure by merging small files into large file thus to reduce size of metadata inside main memory and implement the whole strategy on glusterfs.

But there are some disadvantages of this work because of only application to the GlusterFSb based platforms while there are more distributed systems are avalible. Implementation cost is more because of extreme workload and might compromise original good feautures of the distributed file system

Tao Wang, Shilong Yao, Lian Xiong, Xin gu(Tao Wang, 2015) *[6]:*HDFS,DFS are adopted to support cloud storage and are designed for optimizing large file access but unfortunately the problem of massive small files is neglected and seriously restricts the performance of DFS.To improve and even solve the small files problem in this research user task access is defined. The co-relation among the access task, application and access fie are constructed by improving PLSA and researchobject is transformed from file level to task level

Hence, improving performance the strategy used to merge small files in terms of access task and select prefetching targets based on the transition probability of the task.

The problems are rise when the study of user access task in cloud storage due to the performance of the proposed strategy is influenced by the precision of task identification.

Songling Fu, Liagang He, Chenlin Huang and Keni Li:(Songling Fu, 2015) *[7]* The processing of massive number of small files is challenge in the design of distributed file system currently the block-storage is used it causes inefficiency when accessing small files. iflatLFS is used to manage small files which are based on metadata scheme and flat storage architecture. The new metadata design is proposed in this research which occupies only fraction of metadata size based on traditional file system. Thus, the hybrid storage architecture improves the performance but implementing workload of this system is more and developing the system in actual world has a lot of work and analysis for managing small files.

4. MATHEMATICAL MODEL

Where,
Let the system is decided by,
S= {D,CS,AF,MS,C}

D:Data (Text,Image,Video)

*CS: Client Search:*Request for required data such as text, images, video audio Retrieve the data from server.

*MS: Metadata Server:*Queries locally for id of data block IP of all of data server Retrieve id of data block IP address of data server to client.

AF: Access File: Files which are requested by client such as text, multimedia files

C: Classifier: Classifies the file into different data blocks (Combined block, divided block, No SQL block)

*DB: Database:*Contains different type of data blocks (Combined block, divided block, No SQL block)

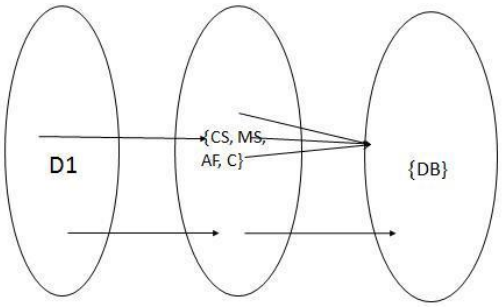

Fig 2. Venn Diagram of Mathematical Model

5. CONCLUSION:

In Proposed system will be focusing on how to constructefficient distributed file systems, one of the challenges is to optimize the storage and access of massive numbers of small files for Internet based applications. Previous work mainly focuses on reducing the problems in traditional files systems, which generate too much metadata and causes lack of file access performance on data servers. We focus on optimizing the performance of data servers in accessing massive numbers of small files and present a proposed system which directly accesses raw disks and adopts a simple metadata scheme and a flat storage architecture to manage massive numbers of small files. New metadata generated by our system consume only a fraction of total space used by the original metadata based on traditional file systems.

In this, each file access needs only one disk operation except when updating files, which rarely happens. Thus the performance of data servers and the whole DFS can be improved greatly. This paper finally proposes a hybrid storage system to integrate different storage systems, each of which represents a better solution for different ranges of data sizes.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 5 Issue 10, October-2016

REFERENCES

[1] Grant Mackey, Saba Sehrish, Jun Wang, "Improving metadata management for small files in HDFS," in Proc. IEEE Int. Conf. Cluster Computer. Workshops, New Orleans, LA, USA, Sep. 2009, pp.[1-4]

[2] Qinqin He Research On Cloud Storage environment File System Performance Optimization Department of Computer Science Northwestern Polytechnic University, Xi'an China, International Conference of Information Management, Innovation Management and Industrial Engineering, 2011, pp[78-83]

[3] RandolphY. Wang and Thomas E. Anderson{ rywang,tea} XFS: A Wide Area Mass Storage File System @cs.berkeley.edu Computer Science Division University of California Berkeley, CA 94720, 1993, pp[71-78]

[4] S. Anjanadevi, D.Vijayakumar, Dr. K .G. Srinivasagan PG Scholar, Assistant Professor, Professor & Head Department of Computer Science and Engineering – PG National Engineering College (Autonomous), Kovilpatti, India "An Efficient Dynamic Indexing and Metadata Based Storage in Cloud Environment" , 2014, International Conference on Recent Trends In Information Technology, pp[78-84]

[5] .Xie Tao, Liang Ale Sc]hool of Software Engineering Shanghai Jiao Tong University Shanghai, China Foxterran@163.com, liangalei@sjtu.edu.cn, "Small File Access Optimization Based on GlusterFS", 2014, International Confernence on Cloud Computing and Internet of Things, pp[101-104

[6] Tao Wang, Shihong Yao, Zhengquan Xu*, Lian Xiong, Xin Gu, XipingYangState Key Laboratory for Information Engineering in Surveying, Mapping and Remote Sensin Wuhan University Wuhan, China wangtao.mac@gmail.com "An effective strategy for improving small file problem in distributed file system", 2015, 2nd International Conference on Information Science and Control Engineering, pp[122-126]

[7] Songling Fu, Liagang He, Chenlin Huang and Keni Li: "The processing of massive number of small files is challenge in the design of distributed file system.", 2015, IEEE Transcations on Parellel and Distributed System, pp[3433-3447]

[8] https//:www.niso.org/publications/press/understandingmetadata