File Replication and Consistency Maintenance in the Hadoop cluster using IRM Technique

Ashish R. Varma, Dr. A. K. Shrivastava

***Abstract*— Storage systems are very important and basic building blocks for cloud computing technology. High performance storage servers are the better solution for cloud storage; the implementation of storage system at very low cost is a big issue. To solve this problem, the very efficient and cost effective way of storing data is to form the cluster of commodity computers. Hadoop Distributed File System (HDFS) is freeware software whose source code is freely available for cloud and it can be deployed in low-cost hardware. Cloud Storage system using the cluster computing can be build with HDFS by making optimized replication management scheme. Hadoop maintains multiple Replicas of each data file. Hadoop is using the Triplication (three replica of each block) policy to replicate the data within the cluster by using this policy the Hadoop guarantees data availability and fault tolerance.**

We have observed there is some large amount of data is located within the clusters which is not being used frequently and I came across the point that maintaining the three replicas of such file is not much important. It leads to poor utilization of storage space. I have read several papers of other file system and decide to implement new way of the replication called " IRM for cloud" which will create the replica of only those file that are to be needed very frequently and keeping the replica on the requesting node may decrease access time and guarantees the full utilization of the storage space. This system provides optimal replica number as well as minimizes the access time required for the nodes to access replicas by placing the data blocks at the requesting node.

Index Terms*—HDFS, Cloud Computing, Replication Management

I. INTRODUCTION

Cloud computing has become an important computer technology, and many of the people are expecting that cloud

Computing will change the information technology (IT) processes and the IT field. With the cloud computing, people are using several services provided by the cloud on their personal devices like computers, laptops, Smartphone and iPods. Services provided by the cloud are storage, applications, and data over the internet. Advantages of using the cloud computing are it minimizes the cost, provides high availability and easily scalable. Many large companies such as Microsoft, Google and Amazon have been building large data centers from last few years.

These data centers are Providing CPU, networking and storage services at a very low cost. This allows very small scale companies to switch over the cloud.

Cloud storage (or data storage as a service) is the storage service provided by the cloud as Infrastructure as a Service (IaaS). Within this service, storage is provided as per the demand. The cloud storage hides the complexity of the existing infrastructure and provides the user very simple storage. It hides from the user whether the data is stored at local computer or whether it is present at the remote machine. Cloud storage infrastructure is using large set of the commodity computers within the cluster by using the distributed file system.

Nowadays the large storage system is required by most of the organizations therefore their interest in the cloud storage provider has been increased. Therefore, the number of cloud storage providers are emerging day by day, also the quantity of such storage providers are also increasing daily. Because developing such large storage system is not possible for small scale companies and cloud storage providers are providing storage services a very low cost to meet the needs of storage user.

Cloud computing can merge physical and logical resources to provide computing services such as Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). Most of the cloud storage providers are using the large servers at datacenter, maintaining such big servers is very costly and reliable. That is storing data and managing this data at the low cost is a big issue for small and large companies. Most of the peoples are come up with different techniques to provide the solution. But these solutions are also costly and very difficult to implement and maintain. Recently the cloud storage providers are using the cluster of commodity computers to solve the above discussed problem.

II. TRADITIONAL APPROACH VS HADOOP'S APPROACH

Traditional Approach: In a traditional approach an enterprise will get a powerful computer and it will feed whatever the data is available to this computer and the computer performs a good job until a certain point. A point will come when this computer will not be able to do processing anymore because it is not suitable and the big data is growing and growing so traditional approach has its limitations when it comes to the big data.

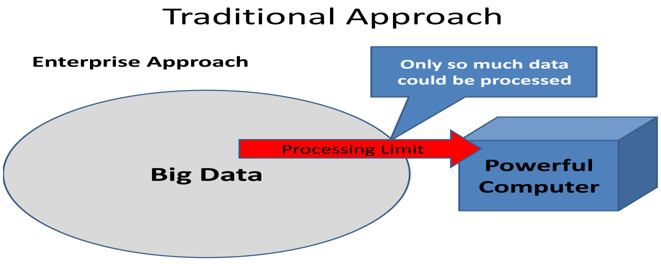

Fig.1. Traditional Approach

Hadoop's Approach:

Hadoop takes a very different approach than the enterprise approach, it breaks the data into smaller pieces and that's why it is able to deal with big data. Breaking the data into similar pieces is good idea but then how are you going to perform the computation? So it breaks the computation as well and sends each piece of computation to the each piece of data so that all computations will be finished in equal amount of time. When all this computations are finished then the results are getting combine together and this is send back to the application as a combined overall result.

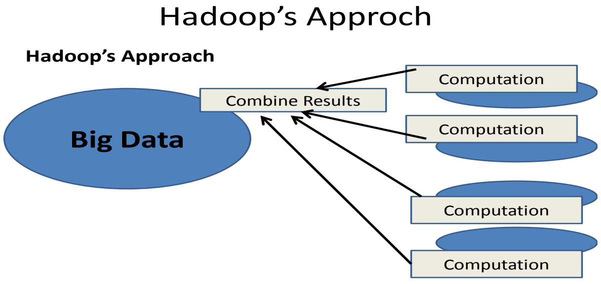

Fig.2. Hadoop's Approach

III. WHY INTEGRATED FILE REPLICATION IS IMPORTANT?

Hadoop provides us the method of accessing this data very rapidly as compared to other system named as MapReduce technique. By using the MapReduce we can access this data easily at very high access rate. Therefore using the Hadoop with the combination of MapReduce is the best way to deal with the big data. In the HDFS Hadoop is maintaining the Replica of each data file? Hadoop is using the Triplication (three replica of each block) policy to replicate the data within the cluster. By using this policy the Hadoop guarantees data availability and fault tolerance. Hadoop is using the distributed file system for storing data. This distributed file system is relay over the cluster of computers. The cluster is the group of computers which is located apart from each other. Cloud is using HDFS for storage purpose.

There are large amount of data are collected within the cluster and these data are stored on the individual computers called nodes. The data is kept in distributed manner to provide data availability and fault tolerance. Within these large data sets some of the data is used very frequently while some data is never used. Hadoop is maintaining three copies of each data that is Hadoop distributing the file equally within the cluster. This triplication policy of the HDFS is some time giving nice performance but sometime it is getting failed when there are more than three requests for the same file block. At this time the blocks of the required file is feeling less while the unrequested data blocks are appeared in large size. Here this non accessed block is appearing as wastage of space and wastage of resources. This type of replicas of blocks are degrading the system performance while checking for the requested blocks and making the MapReduce Process little slower. We have observed that, there is some large amount of data is located within the clusters which is not being used frequently and I came across the question that maintaining the three replicas of such file is really important? Or it is just wasting the space and making the system slower.

I have read several papers of other file system which is used in the peer-to-peer file system and found the idea of maintaining the more number of copies of only those data which is really required in large number of replica copies. So I am proposing the technique, it is not an algorithm because we are not following the steps. We are just interested to create the replica of only those file that are required very frequently. So what we have to do is that we have to keep track of the record that which file is accessed by which node and how many number of times. By finding the access rate of the file we can find the popularity rate of the file. And we replicating only those files on the several nodes whose popularity is higher that only those file that are accessed several times.

IV. BIG DATA: ARCHITECTURE AND APPROACH

In their work of big data study Kapil Bakshi has discussed about the big data and complications arise to accommodate big data in the cloud that is in the cluster which is used by the cloud to store large size of data. They have provided the architecture and approach of storing large size data (Big data). As the data develops in the companies, new techniques and approaches need to be adopted. Their paper focuses on the unstructured looks of data analytics as a case study, some of the key projects in Apache Hadoop. They also describe the basic principles of relational database management systems (RDBMS) and their usage for traditional big data sets in data warehousing, decision support, and analytics. The paper then brushes up non-relational big data methodology such as distributed/shared-nothing architectures, horizontal scaling, key/value stores, and eventual consistency. Some part of the paper distinguishes between structured versus unstructured data. The paper describes various building blocks and methods for Map Reduce and HDFS, HBase and their implementation in an open source Hadoop framework. Their paper centralizes on the infrastructure planning (compute, network, and storage systems), and reviews Hadoop design criteria and implementation considerations. Hadoop consists of many methodologies, including MapReduce, which interact with the infrastructure elements while studying data. There paper reviews performance considerations and describes applicable benchmarks with a Hadoop analytics cluster. In conclusion, the paper reviews the options to hosting an analytics cluster in a public cloud.

V. Integrated Data Replication & Consistency Maintenance in P2P SYSTEM

Haiying (Helen) Shen proposed a technique called IRM: Integrated Data Replication & Consistency Maintenance. Integrated Data Replication & Consistency Maintenance is the technique proposed in peer to peer architecture. It is based on Decentralized Data Replication Management. It Replicates File as per Request Query. In its peer to peer architecture it Provides Autonomy to Node, and performs the great job of Reducing Unnecessary Replication

In peer-to-peer file sharing schemes, IRM are widely used methods for high system operation. Most file replication techniques rigidly define replica nodes, heading towards to low replica utilization, unwanted replicas and hence excess consistency maintenance overhead [3]. Most consistency maintenance methods disseminate update messages, heading to inefficient file update and hence high chances of outdated file response. This paper is presenting a new mechanism IRM that combines the two techniques in a sophisticated and consonant manner. It achieves highly effective results in file replication and consistency maintenance at a very less cost. Instead of passively assuming replicas and modifies, each node ensures file replication according to the current update rate of the file, and neglects unwanted file replications and updates. A simulation output shows the strength of IRM in comparison with other techniques.

It reduces the unwanted replica overhead and improves the storage efficiency. The traffic of the P2P architecture has increased because of the videos and audios. The study is also showing that to access this file is very frequent as compared to other. Such files can discharge the capacity of a node, heading to detained response.

The core idea behind the IRM is to do a file creation as per the file query and update rate. When the node is came to know about some of the replica has been accessed very frequently or itself is accessing very frequently it finds such replicas and creates the copy of it near and ask to the owner of the file about updating with date to avoid consistency overload.

The proposed replication algorithm can provide better results in the query efficiency and overhead in file replication. It actively finds time-varying file popularity and interest of the node and actively creates the replica node. It concentrates on two major problems that are where to replicate the file so that accessed rate should be less and replica can be utilized fully. Second is how to minimize the consistency maintenance overhead required by unpopular replicas?

Hadoop is providing the way of doing all this with its Hadoop Distributed File System (HDFS) and computation model called MapReduce. HDFS is creating replicas of data to ensure availability and reliability of data but it creates the three numbers of replicas for each data block hence we require triple size of data to store single block. It occupies so many space of the Hadoop cluster and increases the consistency maintenance overhead. Again it is creating the same number of replicas for all kind of data blocks including popular and unpopular data blocks. So to maintain the storage capacity and consistency is becomes a challenge. This usually observed that system has to sacrifice the storage capacity and performance to achieve high availability. When we store the data in the HDFS we required triple size of free space. The cost of this triplication policy of the Hadoop is very expensive.

For this purpose, we put forward the innovative approach “IRM: Integrated file replication and consistency maintenance”. It is secure and effective to maintain the number of replicas as per the demand and popularity and minimizes the replica consistency overhead.

There is need of some modern replication policy which will create the replicas as per popularity of the file. There should be large number of replicas for the file which is accessed very frequently. And less number of replicas which is accessed rarely. Again if we placed this replicas at the proper node so we can achieve full utilization of the replicas and decrease the network traffic and improves the overall performance of the system.

VI. Objectives of work

Our new replication technique called “IRM for the cloud” is efficient and reliable. It has accomplished space requirements and is fast enough for most widely used cloud environments. IRM for cloud storage technique limits the added time cost for accessing the file in the distributed file system such as HDFS and at the same time improves the efficiency of the storage in the cluster by keeping proper number of replicas on the nodes.

By changing the replication factor of the system we can minimize the replica overhead but harm to the reliability factor. But by replicating the data blocks of only that file which is really popular and being access more frequently than other files we can increase the reliability and availability of the popular file. Hence freeing the space consumed by the non accessed file to keep their three replicas of the blocks. In short we are maintaining more number of replicas for the popular file and minimizing the replicas of unpopular files. So the proposed scheme replicates only that file blocks which is being accessed very frequently and which is really required by the nodes. Note that it is unnecessary to replicate unpopular file which not even accessed single time and not being used by any node. We are just keeping the watch on the files and counting their access. We are referring the access count of the file as the popularity rate for the file. As the access count number is increased for the particular file its popularity is also getting updated, when it reaches up to the marks of being replication this file blocks are replicating and untouched file that is unpopular file which is not being access more enough is remains as it is. Again by taking the idea of IRM used in the P2P environment. We are choosing the very good node for placing the replica. The definition of very good node is a node which is accessing that file more number of time that is by keeping the file blocks on such node we can decrease the network traffic and the latency period required to access the file. Because it is not a big issue if the nodes are located in the short geographical area but when we are coming to the large cloud services such as Facebook, Amazon whose servers are scattered across the several countries then this access time is more. So every time accessing the file from the node of remote country require more time as compared to access the same file from our own country node.

VII. Proposed System

In this project, we propose a new and very novel approach of replica creation and maintenance in the Hadoop Distributed File System. This approach is developed to overcome from the several disadvantages of existing Hadoop Distributed File System discussed previously in the existing system section, existing replica placement policy suffers from several point view. It maintains the same number of replicas for all the files. It does not consider file popularity for creating replica. Even Hadoop Distributed File System places the replica randomly on the nodes and creates the replica node without considering any factors of node. In the search of data availability and fault tolerance system HDFS creating three replicas of each file blocks and keep it on the different racks to make the system fault tolerance. But by doing such it makes lots of replicas for the file which is not being accessed and it makes shortfall for the file blocks which is accessed very frequently. That is there is a need to develop the system which creates the replicas as per their need and as per their popularity because creating such large number of replicas of large file consumes very large space and making the storage system of cluster inefficient to store large size of data. I have read several papers of the replication strategies and I came across the point that within the P2P system there is IRM approach is proposed which is considering the file popularity rate before making the replicas for the file, it finds the popularity of the file and replicate the file as per their popularity along with the popularity it finds the better replica node to place replica so that utilization of the file replica should be hundred percent. By placing the replicas of the file for which the node is requesting very frequently on that node we can decrease the network traffic and also the access time required for accessing that file from the remote node. This access time later on appears as a latency period and degrades the system performance hence by doing so we can increase the overall performance of the system.

Again there are some of the files that can be accessed by the single node but some of the files are accessed by the several nodes concurrently and if the count of replica is less than the requesters there will be a competition within the nodes to accessed replicas and in some of the cases this may cause the system failure.

We are proposing a very novel technique of placing the replica in the HDFS cluster. This provides the full utilization of replicas by placing the replica at the requesting node and also decreases the access latency required to access the file blocks from remote node also minimizes the network traffic and improves the network bandwidth.

Advantages

- ➢ The proposed system implements replication management which reduces replica consistency overhead.
- ➢ It provides high availability of data according to the type of data.
- ➢ It provides high reliability of data.
- ➢ It guarantees the full utilization of replicas.

In this work, we shall study various approaches which are followed in realizing this system. We will further study factual aspects which can be made use of in designing and developing an efficient replication technique with Hadoop Distributed File System. The important consideration made in this thesis is of using the existing standard methods and have developed the innovative technique with different functionalities.

VIII. SYSTEM IMPLEMENTATION

The proposed cloud storage system is implemented with Hadoop storage cluster (HDFS). HDFS is an open source storage platform and designed to be deployed in low-cost commodity computers. Therefore, it is suitable to implement and evaluate the proposed system. In HDFS, the replication factor is changeable. However, HDFS does not provide any policy to determine the replication factor. In proposed replication management, optimum replica number is determined before data is stored to storage servers and the optimum replica number can recover not only failure probability but also expected availability.

The tested cluster consists of three homogeneous nodes, whose parameters are summarized in Table III. All nodes are connected with 1 Gbps Ethernet network. On each node, Ubuntu 64 Bit Operating system is installed. Installed Java version is 1.6.2 and Hadoop version is 0.20.2. The size of HDFS block is 64 MB.

Instead of passively replicating files in the HDFS file system on several Datanodes, IRM harmonically integrates file replication and consistency maintenance by letting each node autonomously determine the need of file replication and replicas creation is based on query rate for the file. The basic idea of IRM is to use file query update rate to direct file replication and consistency maintenance. When node itself queries a file frequently, placing a replica in the node can improve the query efficiency and make a full use of replicas.

IRM file replication places replicas in frequently visited nodes to guarantee high utilization of replicas and meanwhile reduce underutilized replicas and overhead of consistency maintenance.

It optimally chooses source files to replicate and replicate them in physically close nodes and tries to minimize the number of replicas. To optimally choose the source files, it considers file visit rate. An unpopular file does not need to have many replicas since the replicas would have low probability to be visited.

We are placing the replica as per the node request. We are selecting the node to place replica which is requesting for the file block more frequently. We have set the access request maximum count is three and if the node is accessing the same file block fourth time then all file blocks are replicated to that node and new replica node is created as per the demand. We are selecting a node as per the request and hence by keeping the file on that node we can guarantee better utilization of replicas. If the replica is available on the same node the access time for that file is really improved because all the network traffic and latency is discarded if the file is downloading from the same data node. We have shown the result from the simulation the time required to access the file from several nodes in milliseconds.

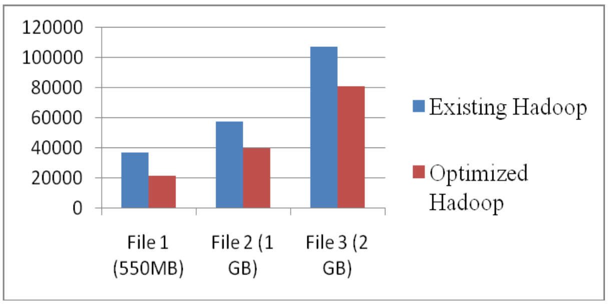

Fig.3. Access time from Master

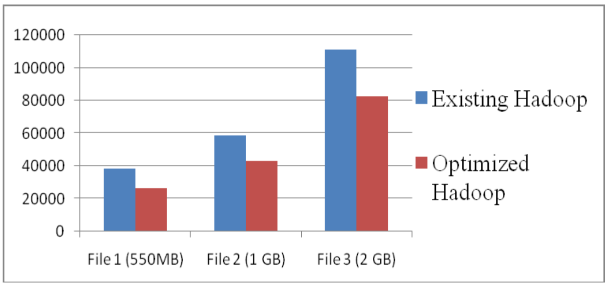

Fig. 4. Access time from Slave 1

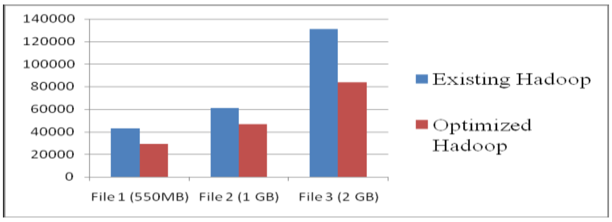

FIG. 4. ACCESS TIME FROM SLAVE 2

IX. CONCLUSION AND FUTURE WORK

We have presented a novel approach to prevent excessive number of replicas. By using the MapReduce programming model and Hadoop Distributed File System in the cluster of commodity computers, we have implemented very cost effective way of storing data in the cloud with less number of replicas maintenance overhead by replicating only popular files. Also creating the replica on the requesting node we can increase the locality of data and minimize the access time required to access the file. It decreases network traffic and improves the performance of the system considerably.

Replication is the important part of the Hadoop Distributed File System by which the Hadoop can provide availability and reliability but creates more unwanted replicas and consumes more storage capacity that is triple of the size of data. We have introduced a new replication technique which will create replicas as per the demand and as per the popularity, how frequently file can be accessed.

In future, we are interested to extend this project for large cloud storage services**.** This approach can be extending as per need of big data applications. It integrates data analyzing techniques. This application can be designed in such a way that we can change contents of files at block level.

REFERENCES

1. Konstantin Shvachko, Hairong Kuang, Sanjay Radia and Robert Chansler *"The Distributed File System" IEEE 2010.*

2. Kapil *Bakshi "Considerations for Big Data: Architecture and Approach" Proc. IEEE Symp. Mass Storage Systems and Technologies, 2012.*

3. Haiying (Helen) Shen *"IRM: Integrated File Replication and Consistency Maintenance in P2P System" IEEE Transactions on Parallel and Distributed Systems, Vol. 21, No. 1, January 2010.*

4. Zacharia Fadika, Elif Dede, Jessica Hartog and Madhusudhan Govindaraju *"MARLA: MapReduce for Heterogeneous Clusters" 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2012.*

Ashish R. Varma. M.Tech student at Institute of Engineering & Technology, Alwar (Raj.) Rajasthan Technical University, Kota (Raj.)

Prof. (Dr.) A. K. Srivastava Ph.D. (AI), TIFR MUMBAI
Principal General, IET Group of Institutions, Alwar