ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering and Technology (ICIET'14)

On 21st & 22nd March Organized by

K.L.N. College of Engineering, Madurai, Tamil Nadu, India

An Efficient Dynamic Indexing and Metadata Model for Storage in Cloud Environment

S. Anjanadevi, D. Vijayakumar , Dr. K .G. Srinivasagan

PG Scholar, Dept of Computer Science and Engineering, PG National Engineering College, Kovilpatti, Tamilnadu, India.

Assistant Professor, Dept of Computer Science and Engineering , National Engineering College, Kovilpatti, Tamilnadu, India.

Professor & Head Dept of Computer Science and Engineering , National Engineering College, Kovilpatti, Tamilnadu, India.

Abstract - Cloud computing is an emerging, computing paradigm in which tasks are assigned to software, combination of connections, and services accessed over a network. This connections and network of servers is collectively known as the cloud. Instead of operating their own data centers, users might rent computing power and storage capacity from a service provider, paying only for what they use. Cloud storage involves the delivery of data storage as service including database. If the data is stored in cloud, it must provide the data access and heterogeneity. With the advances in cloud computing it allows storing of large number of images and data throughout the world. This paper proposes the indexing and metadata management which helps to access the distributed data with reduced latency. The metadata management can be improved for a file system in large scale applications. When designing the metadata, the storage location of the metadata and attributes is important for the efficient retrieval of the data. Indexes are used to quickly locate data without having to search over every location in storage. Based on these two models, the data can be easily fetched and the search time was reduced to retrieve the appropriate data.

Keywords - Metadata, Cloud Storage, Index, Google File System, Hadoop Distributed File System

I.INTRODUCTION

Cloud Computing is an emerging technology which categorize and schedule service based on the
Internet. The concept of cloud computing includes the Web 2.0, web infrastructure, virtualization technologies and other emerging technologies. With the cloud computing technology, users use a variety of devices with laptops, smart phones, PCs, PDAs to access storage, and application-services offered by cloud computing providers. Some advantages of the cloud computing technology include cost savings, high availability, and scalability.

In Cloud, there are different types of cloud services. They are Infrastructure as a service, Software as a service (Saas), and Platform as a service (Paas). In Iaas, the clients control and manage the systems in terms of the applications, storage, and network connectivity, operating systems, but do not they control the cloud infrastructure. Example: Amazon EC2. In Saas, clients purchase the ability to access and use an application or service that is hosted in the cloud. Example: Salesforce.com. In Paas, clients purchase access to the platforms, enabling them to use their own software and applications in the cloud. Example: Google AppEngine. There are different types of deployment models in cloud. They are Private cloud, Public cloud, Hybrid cloud. In private cloud, the cloud infrastructure has been organized, maintained and operated for a specific organization. The operation may be internal or with a third party on the premises. In Public cloud, the cloud infrastructure is available to the public on a commercial basis by a cloud service provider. In Hybrid cloud, it can be a combination of private and public clouds that support the requirement to maintain some data in an organization.

The primary uses of cloud computing is for data storage. Cloud storage is one of the services which provide storage resource and service based on the remote servers. Cloud storage is a model of online storage where data is stored on various third party servers. It will be able to provide storage service at a lower cost and more reliability and security. The advantage of cloud storage is it enables users at any time access data. There are different cloud storage systems. Some have focus on storing Web email messages or digital pictures. Others are available to store all methods of numerical data. Some cloud storage systems are small processes, but others are so large that the physical equipment can fill up an entire warehouse. The services that provide cloud storage systems are called datacenters.

The HDFS is one of the storage systems. The Hadoop distributed file system (HDFS) is a distributed, scalable file system written in Java for the Hadoop framework. Each node in a Hadoop instance usually has a single namenode which is a cluster of datanodes form the HDFS cluster. HDFS stores large files in the range of gigabytes to terabytes across several machines. It succeeds reliability by replicating the data across multiple hosts, and requires RAID storage on cloud. The replication data is stored on three nodes. HDFS added the high-availability capabilities and allowing the main metadata server the NameNode to be failed over manually to a backup in the event of failure.

This paper proposes the design of cloud storage system which focus on Metadata organizer, Level indexer and PC cluster based storage system in HDFS. The rest of this paper is organized as follows. Section 2 refers to the Related works. In section 3 presents Cloud Storage Architecture. In section 4 describes Evaluation and Analysis of the model. Finally, section 5 is the Conclusion and References.

II.RELATED WORK

The paper proposed by sai Wu et al [11], describes an indexing framework for the Cloud system based on the structured Overlay. The indexing framework reduces the amount of data transferred inside the Cloud and facilitates the deployment of database back-end applications. The Cloud system dynamically allocates computational resources in response to customer's resource reservation requests and in accordance with Customer's predetermined quality of service.

A RAMCloud Storage System (RCSS) was proposed by YifengLuo et.al [7]. It is used to allow efficient random read accesses in cloud environments. Based on the HDFS, RCSS adds the available memory resources in an HDFS cluster to form a cloud storage system, which backs up all data on HDFS managed disks, and gets data from disks into memory. RCSS can achieve high I/O performance with low latency and high throughput for random read accesses.

Cloud storage system is difficult to balance the providing huge elastic capacity of storage and investment of expensive cost for it. To solve this issue in cloud storage infrastructure, low charge PC cluster based storage server was proposed by Tin TinYee et.al [9]. PC cluster based storage server provides the hardware and software devices for large amount of data storage. The framework of Cloud based Cluster PC System (CCPS) consists of three layers. They are web based application services layer, Hadoop Distributed File System (HDFS) layer and PC cluster layer. The web based application layer offers interface to store their applications such Virtual Machine (VM) images, dataset and multimedia data, etc.

HDFS does not perform well for large numbers of small size files. Bo Dong et.al [12], examine the reasons of small file problem of HDFS: (1) large numbers of small files force burden on NameNode of HDFS; (2) correlations between small files are not considered for data placement and (3) There is no optimization mechanism is provided to improve I/O performance. According to file correlation features, files are classified into three types: logically related files, structurally-related files, and independent files. Based on these, an optimized approach is designed to improve the storage and access efficiencies of small files on HDFS. File merging and prefetching scheme is used for structurally related small files, whereas file grouping and prefetching scheme is used for managing logically-related small files.

A W Leung et.al [14] describes the Spyglass, a file metadata search system that achieves scalability by exploiting storage system properties. By designing using an index that exploit storage properties, Spyglass is able to achieve significantly better query performance than existing solutions, while using less disk space.

Gurmeet Singh et.al [13] describes that the data intensive applications produce and analyze petabytes and terabytes of data that are distributed in millions of files or objects. To manage these large data sets, metadata needs to be managed. There are various types of metadata that are specialized for particular types of metadata cataloguing and discovery. In this paper, they present the design of a Metadata Catalog Service (MCS) that provides a mechanism for storing and accessing descriptive metadata and allows users to query for data items based on desired attributes.

III.CLOUD STORAGE ARCHITECTURE

Cloud storage infrastructures are focused on providing users with easy interfaces and high performance services. The main objective of this paper is to propose efficient cloud storage for the fast access of data to minimize the search time of consumer This architecture describes the user interface configuration is used to connect to indexing and metadata to determine the storage, which is described in Fig. 1. The cloud is setup in Eucalyptus cloud platform. Eucalyptus is free and open-source computer software for building private cloud computing environments. Eucalyptus enables pooling compute, storage, and network resources that can be dynamically scaled up or down as application workloads change. Eucalyptus consists of five components. They are Cloud Controller (CLC), Walrus, Cluster Controller (CC), Storage Controller (SC), and Node Controller (NC).

The Cloud Controller (CLC) handles the incoming requests and it acts as the administrative interface for cloud management. The CLC accepts user API requests and manages the compute, storage, and network resources. Walrus is providing a mechanism for storing and accessing virtual machine images and user data. Walrus can be accessed by end-users, whether the user is running a client from outside the cloud or from a virtual machine instance running inside the cloud.

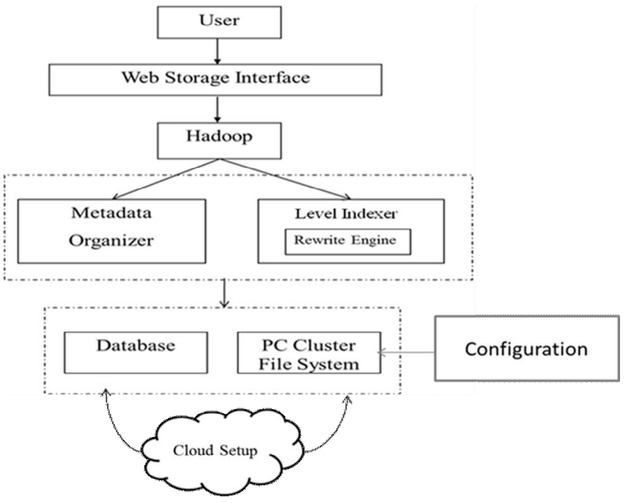


Fig. 1 Cloud Storage Architecture

The Cluster Controller (CC) acts as the front end for a cluster within a Eucalyptus cloud and communicates with the Storage Controller (SC) and Node Controller (NC). The SC communicates with the Cluster Controller (CC) and Node Controller (NC) within the distributed cloud architecture and manages Eucalyptus block volumes and snapshots to the instances within its specific cluster. The Node Controller (NC) hosts the virtual machine instances and manages the virtual network endpoints. The NC downloads and caches images from Walrus as well as creates and caches instances.

Eucalyptus user console has the following systems: The dashboard is the starting point for using the Eucalyptus User Console. From the dashboard, users can access landing pages for instances, storage items (volumes and snapshots), and networking and security objects (key pairs, security groups, and IP addresses). An instance is a virtual machine which is essentially an operational private computer that contains an operating system, applications, network accessibility, and disk drives. A Manage Instances allows users to view a list of instances, create new instances, and connect to, reboot, terminate, and delete instances. Users can also associate/disassociate IPs from instances, attach/detach volumes, and use an instance as a model to create an auto scaling launch configuration, view logs, and stop/start EBS-backed instances. The Storage allows users at how many storage objects are running and access the various storage object management screens. Manage Volumes allows users to view a list of your volumes, create new volumes, attach and detach volumes to a running instance, and delete volumes. Manage Snapshots allows users create a volume from a snapshot and register a snapshot as a machine image that can then be launched as an instance. Manage Images allows users to view a list of images, create auto scaling launch configurations from an image, and launch instances from an image.

/change profile is to display the search result of the current user. In the search result page, user can view or change your identity's profile. View access key is to displays the search result of the current user's access key. Change password is The Eucalyptus Administrator Console has the following areas: View to displays a dialog to change password. Download new credentials is to downloads the current user's credential package in a zip file. A Eucalyptus Machine Image (EMI) is a special type of pre-packaged operating system and application software that Eucalyptus uses to create a virtual machine instance. An EMI contains all necessary information to boot instances of your software. An EMI can contain whatever software you want to meet your needs.

The Hadoop distributed file system (HDFS) is a distributed, scalable file system written in Java for the Hadoop framework. Before installing the hadoop, create a designated hadoop user on the system and install the Preliminary Software. After that install, format and run the Hadoop.

Running Hadoop

First, start up the DFS. This will start up a TaskTracker, JobTracker, and DataNode on the machine.

```
$:~ hadoop-*/bin/start-all.sh
$:~ hadoop-*/bin/hadoop dfs -copyFromLocal hadoop-*/conf input
$:~ hadoop-*/bin/hadoop dfs -ls
Found 1 item
/user/hadoop/input   <dir>     2008-09-11 13:33   rwxr-xr-x   hadoop supergroup
```

To compile the code. cd into the hadoop-*/ directory and do:

```
$:~ ant
```

This will compile the example programs found in hadoop-*/src/examples

Now, run the example program on the conf program as input.

```
$:~ hadoop-*/bin/hadoop jar hadoop-*-examples.jar grep input output 'dfs[a-z.]+'
```

If this works, the pop up will displayed.

```
08/09/13 20:47:24 INFO mapred.FileInputFormat: Total input paths to process : 1
08/09/13 20:47:24 INFO mapred.JobClient: Running job: job_200809111608_0033
08/09/13 20:47:25 INFO mapred.JobClient:  map 0% reduce 0%
08/09/13 20:47:54 INFO mapred.JobClient:  map 44% reduce 0%
 and so on
```

The last step is to check if you have output!

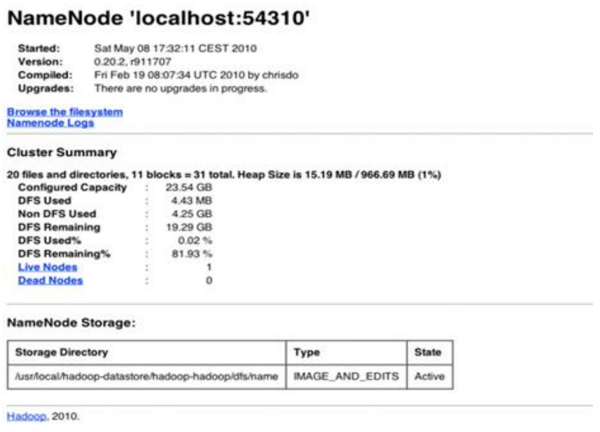
```
$:~ hadoop-*/bin/hadoop dfs -ls output
Found 2 items
/user/hadoop/output/_logs <dir>  2008-09-13 19:21 rwxr-xr-x hadoop supergroup
/user/hadoop/output/part-00000 <r 1> 2917 2008-09-13 20:10 rw-r--r-- hadoop supergroup
```

To check the actual contents of the output do:

```
$:~ hadoop-*/bin/hadoop dfs -cat output/*
```

Alternatively, you can copy it to local disk and check/modify it:

```
$:~ hadoop-*/bin/hadoop dfs -copyToLocal output myoutput
$:~ cat myoutput/*
```



NameNode 'localhost:54310'

Started:	Sat May 08 17:32:11 CEST 2010
Version:	0.20.2, r911707
Compiled:	Fri Feb 19 08:07:34 UTC 2010 by chrisdo
Upgrades:	There are no upgrades in progress.

Browse the filesystem
Namenode Logs

Cluster Summary

20 files and directories, 11 blocks = 31 total. Heap Size is 15.19 MB / 966.69 MB (1%)

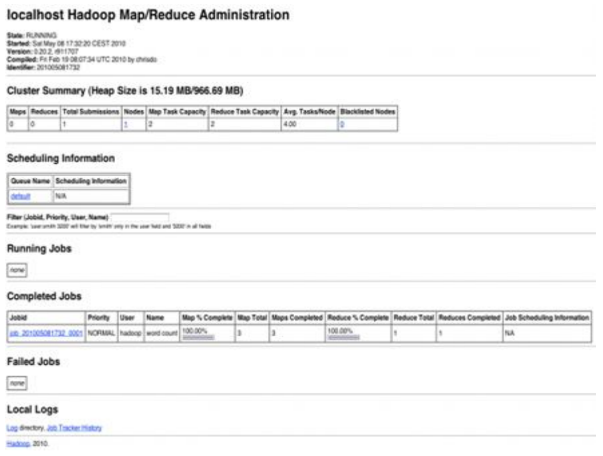
Configured Capacity	:	23.54 GB
DFS Used	:	4.43 MB
Non DFS Used	:	4.25 GB
DFS Remaining	:	19.29 GB
DFS Used%	:	0.02 %
DFS Remaining%	:	81.93 %
Live Nodes	:	1
Dead Nodes	:	0

NameNode Storage:

Storage Directory	Type	State
/usr/local/hadoop-datastore/hadoop-hadoop/dfs/name	IMAGE_AND_EDITS	Active

Hadoop, 2010.

Fig. 2 Namenode Web Interface(HDFS layer)



localhost Hadoop Map/Reduce Administration

State: RUNNING
Started: Sat May 08 17:32:20 CEST 2010
Version: 0.20.2, r911707
Compiled: Fri Feb 19 08:07:34 UTC 2010 by chrisdo
Identifier: 201005081732

Cluster Summary (Heap Size is 15.19 MB/966.69 MB)

Maps	Reduces	Total Submissions	Nodes	Map Task Capacity	Reduce Task Capacity	Avg. Tasks/Node	Blacklisted Nodes
0	0	1	1	2	2	4.00	0

Scheduling Information

Queue Name	Scheduling Information
default	N/A

Filter (Jobid, Priority, User, Name)
Example: 'user:smith 3200' will filter by 'smith' only in the user field and '3200' in all fields

Running Jobs

none

Completed Jobs

Jobid	Priority	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Complete	Reduce Total	Reduces Completed	Job Scheduling Information
job_201005081732_0001	NORMAL	hadoop	word count	100.00%	3	3	100.00%	1	1	NA

Failed Jobs

none

Local Logs

Log directory, Job Tracker History

Hadoop, 2010.

Fig 3. Jobtracker Web Interface(Mapeduce layer)

A user interface is the interaction between users and cloud. A cloud storage system wants data server connected to the Internet. A client sends copies of files to the data servers, which will records the information. When the client needs to retrieve the information, they access the data server through a Web storage interface. The server then sends the files back to the client or allows the client to access and manipulate the files on the server itself. Databases are created to function large quantities of information by inputting, storing, retrieving, and managing that information. The high performance storage server provides expensive cost. To address this problem, the efficient cloud storage system is implemented with low cost and nodes that are organized into PC cluster based data center.

A PC cluster is a collection of computer nodes, all nodes can be used individually or collectively as a cluster. In PC cluster, the large files can be stored by striping the data across multiple nodes. PC cluster consists of one NameNode as server and many DataNodes as clients. PC System uses HDFS (Hadoop Distributed File System) to store data in the collection of the nodes. NameNode manages the whole PC cluster and maintains the metadata of HDFS that contains the current locations of blocks, monitoring the states of all nodes in the cluster and the information of blocks. NameNode is recorded any changes to the file system namespace such as opening, closing, renaming files and directories. It also defines the mapping of blocks to DataNodes. The DataNodes store the physical storage of the files. DataNodes also perform block creation, replication and deletion upon instruction from the NameNode. In PC cluster, files are divided into blocks that are stored as independent units. Each block is copied to a small number of single machines for fault tolerance.

An index is a data structure that improves the speed of data retrieval operations on a database and the use of more storage space to maintain the extra copy of data. Indexes are used to rapidly locate data without having to search every row and every time a database is accessed. Index could provide the data pointer that specified to the data value set which is stored in specified column and then order these pointers according to the specified sort order. The first column contains a copy of the primary or candidate key of a table and the second column contains a set of pointers holding the address of the disk block where that particular key value can be found.

The metadata file is the major operation in searching files in cloud storage systems. The metadata refers to high level information about who is using the content, how the content is being used, and when multiple pieces of content are being used. It also identifies the number of partitions and the number of levels of the stored file. Before a file is retrieved the related metadata is to found and permission to be accessed to be obtained. Metadata is managed separately by one or more specialized metadata servers. To distribute metadata among multiple servers some techniques are used like Sub-Tree Partitioning, Hashing technique, Consistent Hashing and Dynamic Hashing.

In sub tree partitioning, namespace is divided into many directory sub trees, each of which is managed by individual metadata servers. The metadata may not be evenly distributed, and the computing and transferring of metadata may generate a high time and network overhead. In hashing, metadata can be distributed uniformly among cluster, but the directory locality feature is missing. It has slow directory operations such as listing the directory contents and renaming directories. In Consistent hashing, the output range of the hash function is treated as a ring. Not only the data is hashed, but also each node is hashed to a value in the ring. In Dynamic hashing, the metadata can be distributed uniformly among cluster, it works dynamically and the directory locality feature will not lose. Thus the efficiency is high than the other three technique.

IV.PERFORMANCE EVALUATION

In Table I, the comparison of metadata techniques is defined with various attributes such as efficiency, reliability, scalability, elasticity and load balancing. In Fig. 4, the efficiency, reliability, scalability, elasticity and load balancing refers to 1,2,3,4 and 5 in the graph respectively. In Dynamic Hashing technique the efficiency is high as compare to other three techniques. Thus it dynamically works in the distribution of metadata and routing of metadata request is effectively done using dynamic hashing which leads to improvement in system reliability and elasticity. In this the data bucket is divided into equal size which is evenly distributed over the NameNodes which leads to efficient load balancing because of even distribution.

TABLE I. Comparison of Metadata Techniques

	Consistent Hashing	Hashing	Sub-Tree Partitioning	Dynamic Hashing
Efficiency	Medium	Low	Medium	High
Reliability	Medium	Low	Low	High
Scalability	Moderately Scalable	Moderately Scalable	Moderately Scalable	Highly Scalable
Elasticity	Medium	Medium	Low	High
Load Balancing	Yes	Yes	No	Yes

In consistent technique the data bucket is not correctly divided into equal size which leads to less load balancing. Thus the elasticity and scalability is moderate which leads to less efficiency. In Hashing technique the efficiency is low since hashing destroys the locality of metadata which causes the opportunity to prefetching and storing the metadata in bucket. In hashing the hash function uses the search-key of the bucket. This search key is unique. Due to the uniqueness of search key in hashing dependency is generated which leads to low reliability and low elasticity. In Sub-Tree partitioning the performance is medium compare to hashing technique. Thus reliability and elasticity decreases.

The cloud storage has two parameters. First, the load intensity is specified that in this case is the rate at which files arrive. Second, we specify the service demand is also specified that is the average service requirement of a file. The average response time of the system with different arrival rate are shown in Fig. 5.

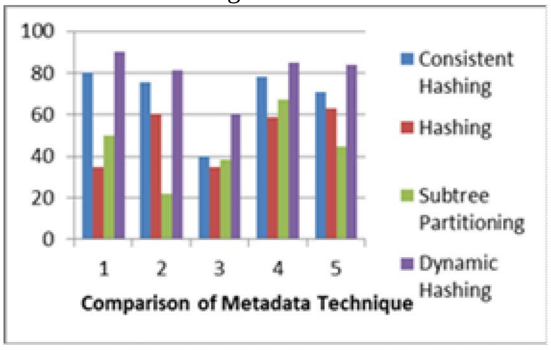


Fig 4. Comparison of Metadata Techniques

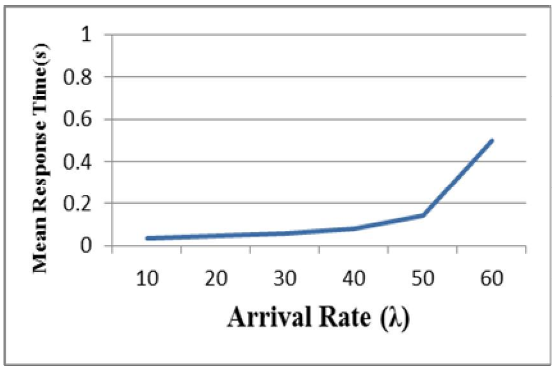


Fig 5. Mean Response Time

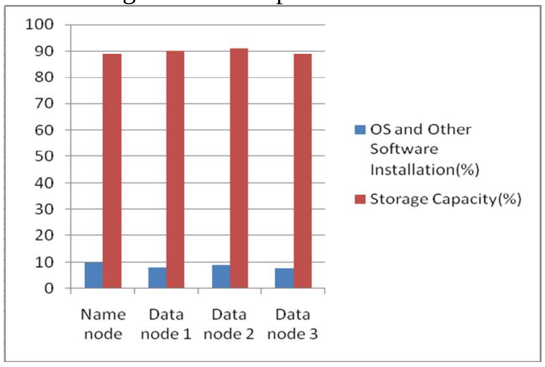


Fig. 6. Average Storage Capacity of PC cluster based Storage Server

In our system configuration 3 DataNodes are used and consumed about 50 GB Hard disks storage of each PC for installation of an operating system and other software installation.

The available storage capacity of these PCs is shared together, and then it can provide 3 x 50 = 150 GB of storage capacity. The storage capacity remains unused and can be utilized if shared to store huge amount of data. If the system configuration used the number of 20 DataNodes, then the available storage capacity can provide 20x50=1000 GB. Therefore, storage capacity of the server can be increased depending on the number of node in PC cluster. The average storage capacity of PC cluster based storage server is shown in Fig. 6.

In this paper, we have adopted dynamic hashing technique to distribute the metadata evenly among the metadata server. As compare to other techniques dynamic hashing evenly distributes load to the server and consumes less memory and provides high efficiency in HDFS.

V.CONCLUSION

There has been a significant advancement in the area of cloud computing. However knowledge about the storage in cloud is not yet discovered. This project is helpful in acquiring the storage with index and metadata and the PC cluster process in HDFS, which will help to minimize the search time of consumer and fast access of the data. This will guarantee the minimization of the search time of consumer.

REFERENCES

[1] S. Ghemawat, H. Gobioff, S. Leung. “The Google file system,” In Proc. of ACM Symposium on Operating Systems Principles, Lake George, NY, Oct 2003, pp 29–43.

[2] Konstantin Shvachko,et al., “TheHadoop Distributed File System,” Mass Storage Systems and Technologies (MSST), IEEE 26th Symposium on IEEE, 2010, http://storageconference.org/2010/Papers/MSST/Shvachko.pdf.

[3] P. H. Carns, W. B. Ligon III, R. B. Ross, and R. Thakur. “PVFS: A parallel file system for Linux clusters,” in Proc. of 4th Annual Linux Showcase and Conference, 2000, pp. 317–327.

[4] M. Satyanarayanan, J. 1. Kistler, P. Kumar, etai, "Coda: A highly available file system for a distributed workstation environment", IEEE Transactions on Computers, 1990, pp. 447-459.

[5] Y.Zhu, H.Jiang, J.Wang, and F.Xian, "HBA: Distributed Metadata Management for Large Cluster-Based Storage Systems", IEEE Trans. Parallel and Distributed Systems, June 2008, vol.19, no.6, pp.750- 763.

[6] S. A. Weil, S. A. Brandt, E. L. Mille, "Ceph: A Scalable, High-Performance Distributed File System", In Proceedings of the 7th symposium on Operating systems design and implementation, 2006, pp. 307-320.

[7] YifengLuo, SiqiangLuo, JihongGuanb, Shuigeng Zhou, “A RAM Cloud Storage System based on HDFS: Architecture, implementation and evaluation”, Elsevier (The Journal of Systems and Software), 2013, vol.86, pp.744 – 750.

[8] Bo Dong, QinghuaZheng, FengTian, Kuo-Ming Chao, Rui Ma, RachidAnane, “An optimized approach for storing and accessing small files on cloud Storage ”, Elsevier (Journal of Network and Computer Applications), 2012, vol.35, pp.1847–1862.

[9] Tin Tin Yee and Thinn Thu Naing, “PC-cluster Based Storage System Architecture For Cloud Storage”, International Journal on Cloud Computing: Services and Architecture,Vol.1, No.3, November 2011, pp.117-128.

[10] MrudulaVarade,Vimla Jethani, “Distributed Metadata Management Scheme in HDFS”, International Journal of Scientific and Research Publications, Volume 3, Issue 5, May 2013.

[11] Saiwu (2009), “An indexing framework for effective Retrieval on the cloud”, IEEE Computer Society Technical Committee on Data Engineering, Copyright IEEE, pp.1-8.

[12] Bo Dong, QinghuaZheng, FengTian, Kuo-Ming Chao, Rui Ma, RachidAnane (2012), “An optimized approach for storing and accessing small files on cloud Storage”, Elsevier (Journal of Network and Computer Applications), vol.35, pp.1847–1862.

[13] Gurmeet Singh, ShishirBharathi, Ann Chervenak, Ewa Deelman, Carl Kesselman, Mary Manohar, Sonal Patil, Laura Pearlman, “A Metadata Catalog Service for Data Intensive Applications”, Proceedings of the ACM/IEEE SC2003 Conference (SC’03).

[14] A W Leung, M Shao, T Bisson, S Pasupathy and E L Miller, “High-performance metadata indexing and search in petascale data storage systems”, Conference Series 125(2008) 012069.

[15] Abhishek Verma, Shivaram Venkataraman, Matthew Caesar, Roy Campbell, “Efficient Metadata Management for Cloud Computing applications”, University of Illinois at Urbana-Champaign.

[16] Jin Xiong, Yiming Hu, Senior Member, Guojie Li, Rongfeng Tang and Zhihua Fan, “Metadata Distribution and Consistency Techniques for Large-Scale Cluster File Systems” , IEEE Transactions On Parallel And Distributed Systems, pp.1-15.

[17] Qian Wang, Cong Wang, “ Enabling Public Auditability And Data
Dynamics For Storage Security In Cloud Computing”, IEEE transactions on parallel and distributed systems, May 2011, vol. 22, pp.847-859.

[18] LI Wen-juan, “A Novel Scalable Architecture of Cloud Storage System for Small Files based on P2P”, IEEE transactions on cluster computing,2012,vol.17,pp.41-47.

[19] Da Xiao, “Multiple-File Remote Data Checking for cloud storage”, Elsevier (computers & security), 2012,vol.31,pp.192-205.

[20] Lanxiang Chen, “Data dynamics for remote data possession checking in cloud storage”, Elsevier(Computers and Electrical Engineering), 2013,vol 13,pp.1121-1133.