Implementation of a PC-Cluster based Storage System for Cloud Storage

Tin Tin Yee, Thinn Thu Naing
University of Computer Studies, Yangon, Myanmar
E-mail: tintinyee.tty@gmail.com, ucsy21@most.gov.mm

Cloud computing is an emerging computing platform and service mode, which organize and schedule service based on the Internet. Cloud storage is one of the services which provide storage resource and service based on the remote storage servers through the Internet to the clients. Design and architecture of cloud storage system plays a vital role in cloud computing infrastructure in order to improve the storage capacity as well as cost effectiveness. One of the challenges of cloud storage system is difficult to balance the providing huge elastic capacity of storage and investment of expensive cost for it. In order to solve this issue, this paper proposed the low cost PC-Cluster based storage system that can be activated to store large amount of data and provides cost-effective machine. In this proposed system has lower implementation cost and store large amount of data but major bottleneck is I/O operation for data processing. To address this need, block placement and Optimal Binary Search Tree (OBST) algorithms are proposed for improve performance of PC cluster based storage system for cloud storage.

1. Introduction

Cloud computing is a consequence of economic, commercial, cultural and technologies conditions that have combined to cause a disruptive shift in information technology towards a service-based economy. Cloud computing offers more opportunity to creative and ambitious people by lowering up-front costs to start new businesses. Many cloud computing service providers have been continuously made efforts to cut down expenses of system maintenance by developing energy-efficient and cost-effective infrastructure and platform software. In addition to the technologies reducing the maintenance costs, it is necessary to reduce a significant up-front investment to build the cloud computing service.

From the end user point of view, cloud computing services provide the application software and operating system from the desktops to the cloud side, which makes users, be able to use anytime from anywhere and utilize large scale storage and computing resources. In cloud computing, disk storage is the one of the biggest expenses. There is strong concern that cloud service providers will drown in the expense of storing data, especially unstructured data such as documents, power points, PDFs, VM Images, multimedia data, etc. Cloud service providers offer huge capacity cost reductions, the elimination of labor required for storage management and maintenance and immediate provisioning of capacity at a very low cost per terabyte. Therefore, the storage and computing on massive data are major key challenge for a cloud computing infrastructure. In this paper, PC cluster based storage system is configured to be activated to store large amount of data and is implemented with Hadoop Distributed File System (HDFS).

Hadoop distributed data storage management is the best large scale data processing solution. It stores each file as a sequence of block; default block size is 64MB and all blocks in a file are the same size except the last one. The blocks of a file are replicated for high availability and fault tolerance. The proposed system of replication policy is random replica placement policy that replicas of a block are placed at random on any of the machines in the entire cluster. The default replica factor for single data block is three. Although, this replication management don’t take into account

disk space utilization and data access cost, therefore data will not always be placed fairly and uniformly to storage node and will became unbalance. In view of this issue, this paper proposed block placement algorithm by determining available disk space utilization and predicting network traffic situation of each node and choose optimal node using Optimal Binary Search Tree (OBST) to improve data access performance. The contributions of this paper are: (1) introducing PC-Cluster based storage system for cloud storage (2) presenting the block placement algorithm which is more useful than the original one; (3) describing I/O traffic management using OBST (4) based on these methods, designing, implementing, and evaluating on proposed storage system.

The rest of this paper is organized as follows. In section 2 describes cloud storage and related works. In section 3 discusses how to build physical topology of PC-Cluster based storage system and I/O traffic management in section 4. Finally, in section 5 discusses the experimental environment and section 6 is the conclusion and future work.

2. Cloud Storage and Related Works

Nowadays, a huge variety of cloud storage systems is available, all with different functionality, optimizations and guarantees. Hussam et al. [4] described RAID (Redundant Arrays of Inexpensive Disks)-like techniques at the cloud storage and showed that reduce the cost and better fault tolerant. Amazon S3 provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the cloud. It gives any developer access to the same highly scalable, reliable, secure, fast, inexpensive infrastructure that Amazon uses to run its own global network of web sites. The service aims to maximize benefits of scale and to pass those benefits on to developers [1]. André et al. [5] presented analytical considerations on the scalability of storage clusters and presents a storage cluster architecture based on peer-to-peer computing that is able to scale up to hundreds of servers and clients. It used Internet SCSI (iSCSI) as inter-connect protocol. This storage cluster environment is implemented and tested on a Linux based HPC-cluster. Yong et al. [6] proposed a storage cluster architecture CoStore using network attached storage devices. In CoStore the consistency of a unified file namespace is collaboratively maintained by all participating cluster members without any central file manager and designed the data anywhere and metadata at fixed locations file system layout for efficiency and scalability in a CoStore cluster.

Bo et al. [7] presented access latency caused by reading from Hadoop Distributed File System constitutes the major performance bottleneck of processing user request for Hadoop based Internet applications. In this paper introduces a two-level correlation based file prefetching approach including metadata prefetching and block prefetching to improve performance by reducing access latency. Futhermore, four placement patterns to store prefetched data are presented to achieve trade-off between performance and efficiency of prefetching. Jiyi et al. [12] introduced cloud storage reference model. This model designed scalable and easy to manage storage system but aren’t designed to be high performance. This paper presented the key technologies, several different types of clouds services, the advantages and challenges of cloud storage. Xu et al. [14] proposed new cloud storage architecture based on P2P which provide a pure distributed data storage environment without any central entity. The cloud based on the proposed architecture is self-organized and self-managed and as better scalability and fault tolerance using Distributed Hash Table approach.

3. PC-Cluster based Storage System Design

In this section describes how to build physical topology of PC-Cluster based storage system for cloud storage. The system architecture is shown in figure 1.The overall system architecture of PC-Cluster based storage system consists of three layers which are web based application services layer, Hadoop Distributed File System (HDFS) layer and PC-Cluster layer.

The web based application service layer provides interface for the users whose can store and access their own applications data such as Virtual Machine (VM) images, datasets and multimedia data and so on. The HDFS layer supports the file system for PC-Cluster layer. HDFS is as a user-level file system in cluster which exploits the native file system on each node to store data. The input data are divided into blocks, typically 64 megabytes and each block is stored as a separate file in the local file system. HDFS is implemented by two services: one NameNode and many DataNode. The NameNode is responsible for maintaining the HDFS directory tree, and is a centralized service in the

cluster operating on a single node. Clients contact the NameNode in order to perform common file system operations, such as open, close, rename, and delete. NameNode does not store HDFS data itself, but rather maintains a mapping between the HDFS file name, a list of blocks in the file, and the DataNode(s) on which those blocks are stored [3]. The PC-Cluster layer provides to store large amount of data.

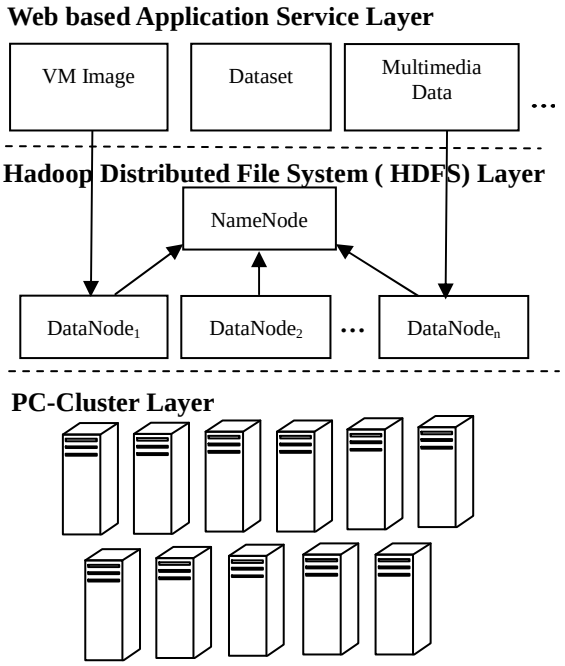

Figure 1: PC-Cluster based Storage System Architecture

3.1 PC-Cluster

Nowadays, many organizations need terabytes storage systems but are very expensive and require higher degree of skills for their operations and maintenance. Usually a desktop PC contain more than 100 GB Hard Disk Drive (HDD), at least 256 MB or greater RAM and 2GHz or higher processor. The typical installation of an operating system and other software installation do not use more than 20 GB of HDD storage. This leaves on the average about 80% of the storage space to be unused. Therefore, this paper proposed PC-Cluster based storage system that is inexpensive and easy to maintain.

3.1.1 System Overview of PC-Cluster based Storage System

PC-Cluster is a collection of computer nodes, which is interconnected by a high-speed switching network, all nodes can be used individually or collectively as a cluster. PC-Cluster based storage system tries to transfer from the cluster computing to storage server. The storage system uses inexpensive PC components. The large files can be stored by striping the data across multiple nodes. In PC-Cluster consists of one NameNode as server and many DataNodes as clients. PC-Cluster based storage system uses HDFS (Hadoop Distributed File System) to store data in the collection of the nodes. Each node has its own memory, I/O devices and operating system. The nodes are physically separated and connected via a LAN.

In PC-Cluster based storage system consists of a NameNode and many DataNodes. NameNode manages the whole PC-Cluster and maintains the metadata of HDFS that contains the information of blocks, the current locations of blocks, and monitoring the states of all nodes in the cluster. NameNode is recorded any changes to the file system namespace such as opening, closing, renaming files and directories. It also determines the mapping of blocks to DataNodes. The DataNodes store the physical storage of the files. DataNodes also perform block creation, deletion and replication upon instruction from the NameNode. In cluster, files are divided into blocks that are stored as independent units. Each block is replicated to a small number of separate machines (typically 3) for fault tolerance. In PC-Cluster layer, each individual machine of a cluster is referred to as a node. This system is based on client-server architecture and consists of one server node (NameNode) and

many clients nodes (DataNodes) in order to make them work as a single machine. NameNode has two networks cards and one is connect to the front-end infrastructure using gigabyte switch. The PC-Cluster is the use of multiple computers, typically PCs which was built 1.5 GHz Pentium P4 processors, 80 GB Hard disks and 512 MB RAM and running with Ubuntu OS. These PC are connected by a gigabyte switch that can create illusion of being one machine. The proposed PC-Cluster based storage system utilizes the existing PC machines in our university without purchasing any extra hardware and software components. Therefore, this storage system is very cost effective architecture.

4. I/O Traffic Management

In this section presents the I/O traffic management scheme for PC-Cluster based storage system. First, we describe the block placement algorithm and then introduce Optimal Binary Search Tree (OBST) to manage I/O traffic on the proposed system.

A. System Model

PC cluster based storage system is composed of N independent heterogeneous machines. There is exactly 1 NameNode and the remaining N-1 machines are DataNodes DN= {DN_1, DN_2,…..,DN_n}. Let file F is divided into blocks denoted as B= {b_1, b_2, …, b_n} and stored into different DataNodes. NameNode maintains B different blocks of metadata, where n is the total number of metadata.

B. Block Placement Algorithm

In this paper, block place algorithm is used to store block replicas among DataNodes to improve load balance and to utilize storage space and defines definitions as follows:

Definition 1: let N_b be the number of data block for file, D_s be the size of file F and block size is denoted by B_s. So, we get

$$N_b = D_s / B_s \quad (1)$$

Definition 2: Let T_D be the total disk space of each DataNode, U_B be the used disk space of each DataNode and A_D is denoted by the available disk space of each DataNode. So, we get

$$A_D = T_D - U_D \quad (2)$$

Definition 3: Let ***Throughput*** of DataNode DN= {DN_1, DN_2, …., DN_n} is calculated as the following equation.

$$Throughput = D_s / (date\ transfer\ time - latency) \quad (3)$$

Algorithm 1: Block Placement Algorithm

Input: DataNodes **DN= { DN_1, DN_2,….DN_n}** , replication factor **r**, Number of Data Block N_b

Output: DataNodes **DN** List

1. **for** each file F **do**
2. Calculate $N_b = D_s / B_s$
3. **end for**
4. **for** each DataNode **DN do**
5. Calculate $A_D = T_D - U_D$
6. Calculate **Throughput = data size / (date transfer time-latency)**
7. **end for**
8. i=0
9. **while** (i<=r)
10. DN= find the largest A_D in the DN && efficient throughput
11. i= i+1
12. **end while**
13. return **DN**

This algorithm is applied to available disk space and throughput of each DataNode that dynamically predict network resources located at different sites of DataNodes using Network Weather Service (NWS) [13]. In this paper, NameNode uses B-tree indexing data structure to keep DataNodes in descending order using their available disk space as the key. When we want to find DataNodes to place block, the Name node quickly searches the B-tree and return DataNodes list. The available disk space of each DataNode is calculated locally in DataNode side and updated to the NameNode periodically.

C. How to reduce I/O Cost?

In this section describes to reduce I/O cost by selecting optimal DataNode using Optimal Binary Search Tree (OBST) algorithm from given DataNodes list to improve access performance and defines definitions as follows:

Definition 4: Given a sequence of DataNode DN = {DN_1,DN_2,..,DN_n} of n distinct DataNodes and a set of probabilities p= <p_1,p_2,..,p_n> for searching the data block B in DN and q= <q_1,q_2,..,q_n> for unsuccessful searches.

$$\sum_{i=1}^{n} p_i + \sum_{i=1}^{n} q_i = 1 \quad (4)$$

Algorithm 2: Optimal Binary Search Tree

Input: DataNodes **DN= { DN_1, DN_2,…., DN_n** } of n distinct DataNodes , a set of probabilities p=<p_1,p_2,..,p_n> for searching the data block B in DN and q=<q_1,q_2,..,q_n>for unsuccessful searches.

Output: Optimal binary search tree for **DN**

1. **for** each DataNode **DN_i do**
2. weight = q_i
3. cost = 0
4. **end for**
5. **for** L=1 **until** DN **do**
6. **for** i=0 **until** DN-L **do**
7. j = i+L
8. $weight_{ij}$= $weight_{i,j-1}$ + p_j+q_j
9. Let m be a value of k , i<k≤ j , for which $c_{i,k-1}$+c_{kj} is minimum
10. $cost_{ij}$= $weight_{ij}$+ $cost_{i,m-1}$ + $cost_{mj}$
11. r_{ij}= DN_m
12. **end for**
13. **end for**

D. Implementation in PC-Cluster based Storage System

PC-cluster based cloud storage system is designed for reliably storing very large files across distributed commodity machine in a large cluster. It stores each file as a sequence of block; default block size is 64MB and all blocks in a file are the same size except the last one. The blocks of a file are replicated for high availability and fault tolerance. The default replica factor for single data block is three. The proposed algorithms can be used to determine block placement for balance workload and reduce data access time. The figure 2 shows the framework of data store operation and data access operation on PC-Cluster based storage system.

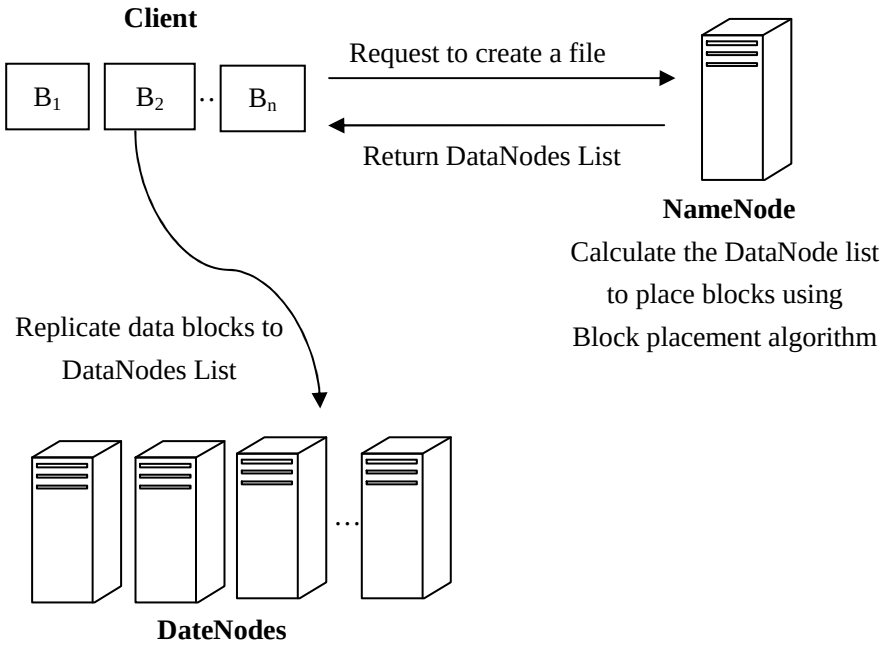

Figure 2 (a). Data Store Operation

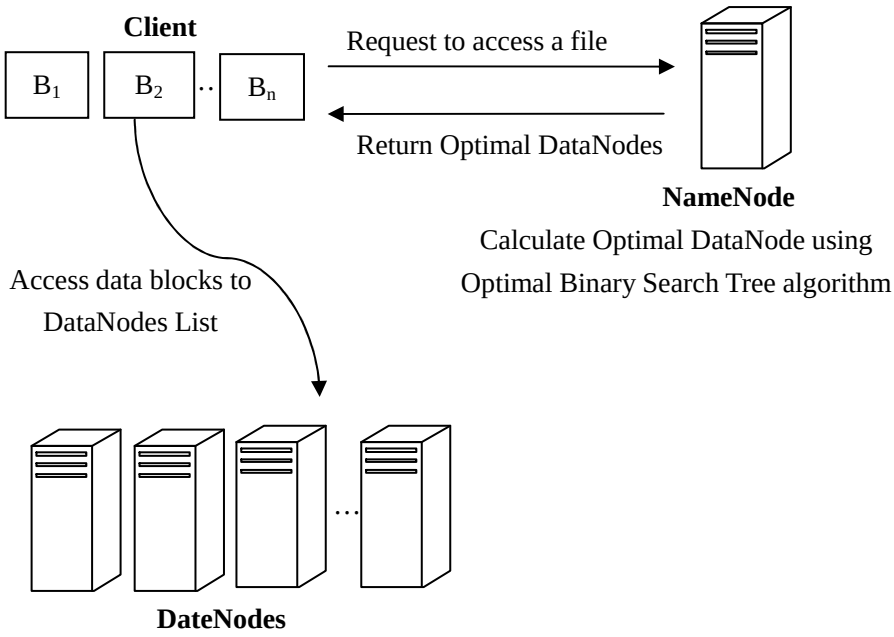

Figure 2 (b). Data Access Operation

Figure 2: Framework of Data Store and Access Operation on PC-Cluster based Storage System

In data store operation, when clients send their data store in this storage system, firstly NameNode calculate the DataNodes list depend on replication factor to place data fairly and uniformly to DataNodes using block placement algorithm. And then, NameNode send DataNodes list on client for store their data. In data access operation, when clients request their data access from this storage system, NameNode find optimal DataNode whose expected I/O cost is smallest from list of DataNodes whose stored the data using dynamic programming approach with Optimal Binary Search Tree algorithm to improve data access time.

An example illustration result of data store operation with five DataNodes is shown in figure 3. In this example, assume that the total disk space of each DataNode T_D is 80 GB and throughput of each DataNode has same. When a new data B_1 is added, NameNode is selected DataNode list DN_2, DN_1 and DN_5 by using block placement algorithm and the DataNodes list is returned to the client.

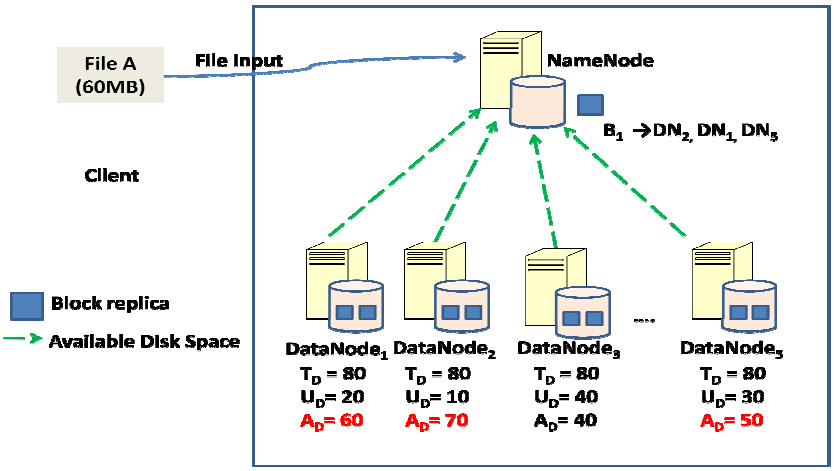

Figure 3: Example illustration of Block Placement Algorithm with five DataNodes

Client's data is stored in DN_2, DN_1 and DN_5. For data access operation, NameNode need to choose optimal DN using Optimal Binary Search Tree algorithm for a given DataNodes list DN_2, DN_1 and DN_5 to improve access performance. We assume that with equal probabilities of p_n and q_n for all DataNodes, we get optimal DataNode is DN_1.

5. Experimental Environment

The test platform is built on a cluster with one NameNode and five DataNodes of commodity computer. All nodes are interconnected with 1 Gbps Ethernet network. In each node, Ubuntu server 10.10 with the kernel of version 2.6.28-11-server is installed. Java version is 1.6.0 and Hadoop version is 0.20.2. The size of HDFS blocks is 64 MB and the number of replicas is set to three. The system configuration is shown in Table 1. During the experiments, installation of Ubuntu operating system uses 9.8125% and the remaining 90.1875% of the storage space to be used PC- Cluster based storage system.

Table 1. PC-Cluster based Storage System Configuration

System	Process Configuration	Other Information
NameNode	Pentium® Dual-Core	Processor: Pentium® Dual-Core CPU T4200 @2.00GHz Memory:1GB RAM Storage: 250GB HDD
DataNodes	Pentium P4	Processor: 1.5GHz Pentium P4 Memory:512 MB Storage:80GB HDD

In the system configuration used the number of five DataNodes and consumed about 10 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 combined together, and then can provide 5 x 70 = 350 GB of storage capacity. The storage capacity remains unused and can be utilized if combined to store

huge amount of data. If our system configuration used the number of 20 DataNodes, the available storage capacity can provide 20x70=1400 GB. Therefore, in the storage server, storage capacity can be increased depending on the number of PC node in PC-Cluster. The average storage capacity of PC-Cluster based storage system is shown in figure 4. Moreover, 100 MB, 100 MB and 1000 MB of data files are stored to the cluster. The results of experiment are shown in figure 5.

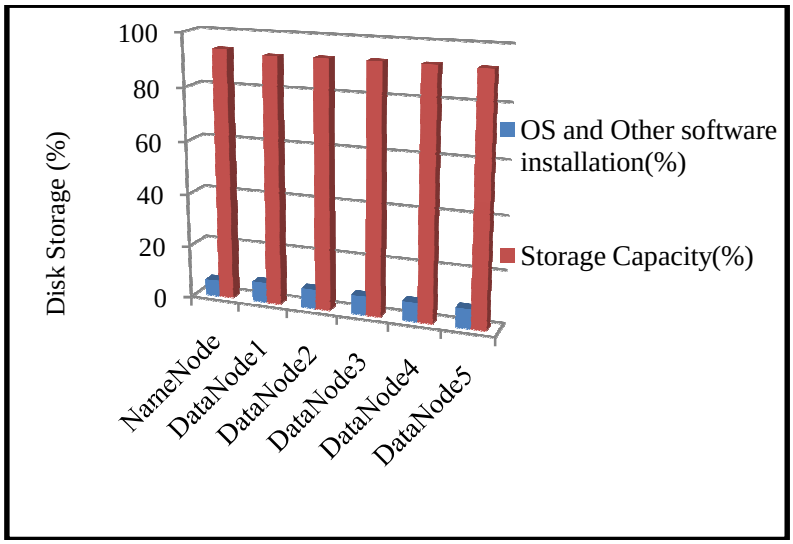

Figure 4. Average Storage Capacity of PC-Cluster based Storage System

According to the result of figure 4, the OS and other software installation used about 10% of the disk storage and 90% of the storage capacity can be used in PC cluster based storage system.

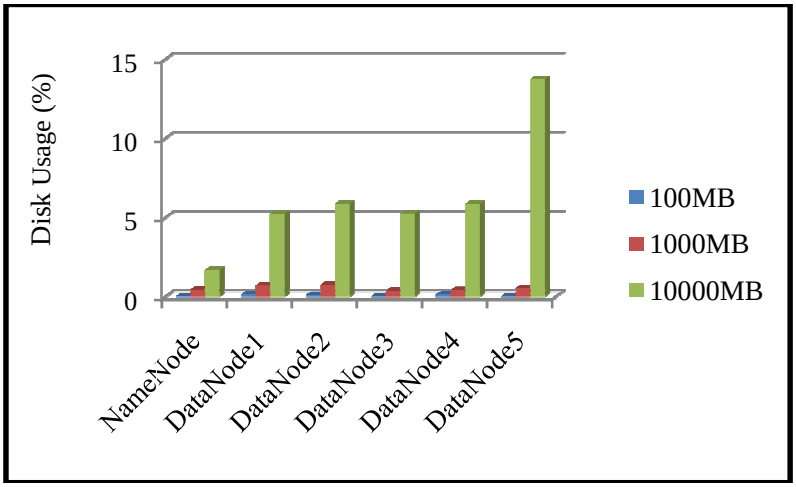

Figure 5. Average Disk Usage of PC-Cluster based Storage System

The figure 5 shows the distribution of disk usage when 100 MB, 1000 MB and 10000 MB of data are stored in the cluster. In this figure used replication policy is random replica placement policy and replication factor is three. According to the result of figure 5, we can see that data don't always be placed fairly and uniformly to storage node and became unbalance workload. Therefore, the proposed algorithms can apply storage utilization and load balancing of the data storage nodes and improve the data access performance.

6. Conclusion and Future Works

In this paper, design and architecture of PC- Cluster based storage system is configured that

utilizes inexpensive computer machines as storage system of the cloud. It can be used to store large amount of data and cost effective. As can be seen from experimental results, the storage can be utilized more than 90% of storage space. As future plan, to investigate more effective solutions and evaluation results for improve I/O performance of PC-cluster based storage system.

References

[1] Amazon S3. http://aws.amazon.com/s3/

[2] Eucalyptus. http://open.eucalyptus.com

[3] Hadoop Distributed File System. http://hadoop.apache.org/core/docs/current/hdfs_deign.html

[4] A. Hussam, P. Lonnie, W. Haki , "RACS: A Case for Cloud Storage Diversity", In Proceedings of the SoCC'10, 2010.

[5] B. André and E. Sascha, "Inter-node Communication in Peer-to-Peer Storage Clusters", In Proceedings of 24th IEEE Conference on Mass Storage Systems and Technologies, 2007.

[6] C.Yong, M.Lionel and Y.Mingyao, "CoStore: A Storage Cluster Architecture Using Network Attached Storage Devices", In Proceedings of the Ninth International Conference on Parallel and Distributed Systems, 2002.

[7] D.Bo, Z.Xiao, Z.Qinghua, J.Lirong, L.Jian, Q.Jie and L.Ying, "Correlation based File Prefetching Approach for Hadoop", In Proceedings of 2nd IEEE International Conference on Cloud Computing Technology and Science,2010.

[8] G. Garth, "Cloud Storage and Parallel File Systems", In Proceedings of SNIA Storage Developer Conference, 2009.

[9] H.Jiawei, K.Micheline, "Data Mining Concepts and Techniques"

[10] L.Lin, L.Xuemin, J.Hong, Z.Yifeng, " AMP: An Affinity-based Metadata Prefetching Scheme in Large-Scale Distributed Storage Systems", Technical Report, Novermber, 2007.

[11] P. Bo, C. Bin and L. Xiaoming, "Implementation Issues of A Cloud Computing Platform", Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 2009.

[12] W. Jiyu, Z. Jianlin, L. Zhije, J. Jiehui, "Recent Advances in Cloud Storage". In Proceedings of the Third International Symposium on Computer Science and Computational Technology, 2010, pages 151-154.

[13] W.Rich, " Dyanmically Forcasting Network Performance Using the Network Weather Service", Technical Report, January, 1998.

[14] X.Ke, S.Meina, Z.Xiaoqi and S.Junde, "A Cloud Computing Platform Based on P2P". In Proceedings of IEEE, 2009.