A Scalable PC Cluster-based Cloud Storage System with Binary Weighted Tree Approach

Julia Myint, Thinn Thu Naing
University Of Computer Studies, Yangon
juliamyint@gmail.com, thinnthu@gmail.com

Abstract

The need and use of scalable storage on cloud has rapidly increased in last few years. Organizations need large amount of storage for their operational data and backups. To address this need, high performance storage servers for cloud computing are the ultimate solution, but they are very expensive. Therefore we propose efficient cloud storage system by using inexpensive and commodity computer nodes. These computer nodes are organized into PC cluster as datacenter. Data objects are distributed and replicated in a cluster of commodity nodes located in the cloud. In the proposed cloud storage system, the PC cluster is logically organized as a binary weighted tree. It supports the weighted allocation of data objects and scalability. Moreover, it can be seen from theoretical analysis that the proposed system is able to scale and can adjust replica numbers according to failure probability and availability.

1. Introduction

As the latest development of the distributed storage technology, cloud storage is the product of integration of distributed storage and virtualization technologies. It is a method that allows user to use storage facilities available on the Internet. In the other words, cloud storage is simply the delivery of virtualized storage on demand. The formal term for on demand cloud storage services is Data Storage as a Service (DaaS) [11].

This time, however, the economic situation and the advent of new technologies have sparked strong interest in the cloud storage provider model. Cloud storage providers deliver economics of scale by using the same storage capacity to meet the needs of storage user, passing the cost savings to their storage.

Many cloud service providers use high performance storage server as datacenter which is very expensive and reliable. However, storing information and managing its storage in a limited budget is a critical issue for a small business as well as for large enterprises. Vendors come up with different solutions day by day but these solutions are very expensive and hard to maintain.

In this paper, we propose a solution that addresses the above mentioned problems. Nowadays, a standard desktop PC has enormous computing and storage capacity. Usually, a standard PC contains more than 40 GB hard disk drive (HDD), 256 MB RAM and 2 GHz or higher processor. If the available storage capacity of these PCs is combined together, then a PC cluster containing 20 PCs with above mentioned specifications can provide 20x40 =800 GB of storage capacity.

In cluster technology, many computers or storage nodes are connected together to provide high capacity storage server. But storage nodes are connected in a cluster may result in obvious load balancing problems and scalability problems.

This paper proposes the design of cloud storage system which utilizes a PC cluster consisting of computer machines (PCs) operating in our university. This solution is very cost

effective because any organization or university can utilize this system over their existing resources (desktop machines) without purchasing any extra hardware or software components.

The rest of this paper is organized as follows. In the next section, we discuss the related papers with our paper. In section 3, we present overview of proposed framework. In section 4, we demonstrate infrastructure of proposed cloud storage on PC cluster. And then we describe the replication management in section 5. Then we analyses theoretically our model in section 6. Finally, we conclude our paper.

2. Related Work

There are large amount of researches in the design of cloud storage. Many of these researches are file system based storage system such as GFS[8] and HDFS[2]. These architectures are master-slave routing paradigm. In those storage systems, replication management is performed by using default replica number.

Hoang Tam Vo and et al. [9] proposed ecStore, a highly elastic distributed storage system with range-query and transaction support. The design of ecStore is based on distributed index structure called BATON (Balanced Tree Overlay Network) [5]. Despite the effect of single master node failure, it is organized into peer-to-peer virtual network structure on storage nodes. ecStore also applied data migration technique to balance the load of storage node and it needs to restructure the BATON whenever load unbalanced. Other cloud storage systems that used data migration for load balancing are Dynamo [4], Pnuts[3] and Cassandra[6].

Replication management has been active research issue in storage system. Qingsong Wei [10] proposed a cost-effective replication management scheme for cloud storage cluster. In that paper, blocking probability is used as a criterion to place replicas among data nodes to reduce access skew, so as to improve load balance and parallelism. In [12], cloud storage is designed based on hybrid of replication and data partitioning. Two level DHT approach is proposed for widely distributed cloud storage.

In this paper, we propose a scalable cloud storage system with the support of availability and load balancing over PC cluster. In order to achieve this goal, we manage replication strategy for fault tolerance and availability.

3. System Overview

The proposed system architecture consists of three layers, i.e., PC cluster, cloud storage infrastructure and application layer. The PC cluster layer provides the hardware and storage devices for large scale data storage. The application layer provides interfaces for clients who store their files, databases, and multimedia data and so on. The second layer is the main focus of our work consisting of replication layer and virtual infrastructure of machines to support cloud storage effectively.

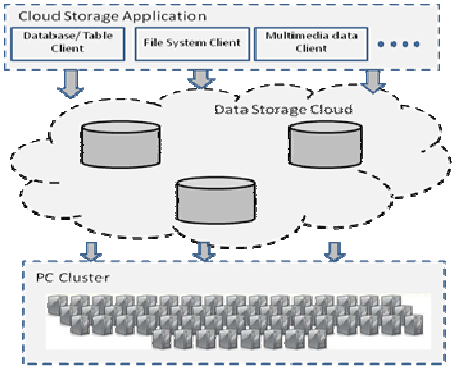


Figure 1 . The system framework of cloud storage on PC cluster

4. Cloud Storage Infrastructure

Cloud storage infrastructure is deployed in PC cluster. Since we apply PC nodes as cloud storage server instead of high performance storage server, there are many challenges for that such as

- Fault-tolerance
- Load balancing
- Availability

- Scalability
- Faster access to the distributed files
- Automated data partitioning and replication

Among these problems, we propose an approach to scalable cloud storage infrastructure as shown in figure2.

The designed storage system is based on a logical organization of PC cluster for cloud storage instead of a single storage server to achieve scalability.

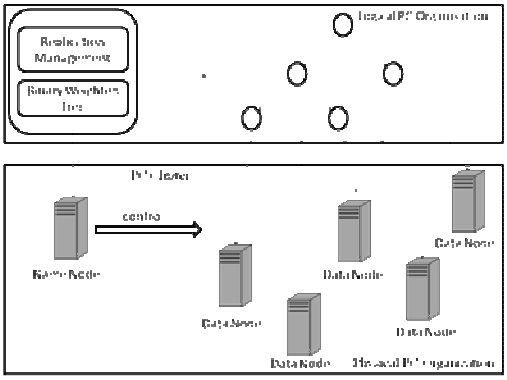


Figure 2 . Overview of proposed cloud storage infrastructure

4.1. Physical PC Organization

The PC cluster consists of a single Name node and a number of Data nodes, usually one per node in the PC cluster, which manages storage attached to the nodes that they run on.

In the proposed system, the Name node manages the whole PC cluster by organizing a logical structure over physical organization. It also executes file system namespace operations like opening, closing, and renaming files and directories. It also determines the mapping of blocks to Data nodes. The Data nodes are responsible for serving read and write requests from the clients. The Data nodes also perform block creation, deletion and replication upon instruction from the Name node.

4.2. Logical PC Organization

The logical PC organization is managed by a single Name node. It organizes a number of Data nodes as binary weighted tree. Binary tree offers more flexibility in the ways in which a system can be reorganized. Each node in the binary tree has its own weight which is the total capacity of storage. For example, if the capacity of drive in PC is 80 gigabytes, the initial weight of that PC node is 80.

Each node in the tree knows the total weight and the total number of nodes in the left and right sub-tree. A tree illustrating result of this process for a system with nine PCs is shown in figure 3.

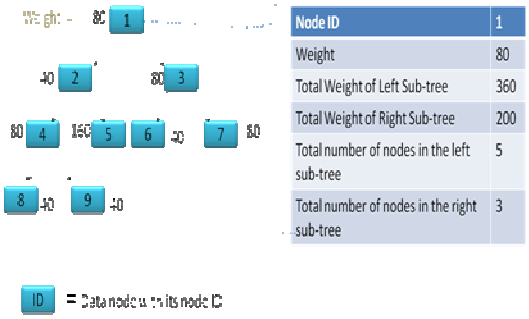


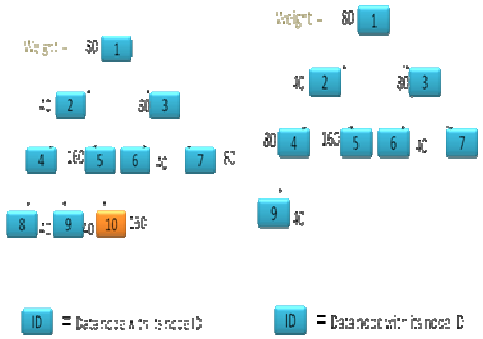
Figure 3 . A binary weighted tree for a cluster with nine PCs

4.3. Adding and Removing PCs

As the storage system grows and changes, the Data nodes are needed to add or remove. Therefore, the proposed system is designed to make this task properly.

Adding a Data node to the storage system requires a minimal amount of administrative work. The new Data node contacts the Name node to inform about it. The Name node only needs to reorganize the logical PC structure without affecting the whole storage system. For example, if a Data node which has 160 GB hard disk drive is added to the PC cluster resulted from figure 3, it informs the Name node. And then, the Name node reorganizes the binary weighted tree as shown in figure 4. Similarly, if a Data node with ID 8 is removed from the PC cluster shown in figure 3, it informs the Name

node. After that, the Name node shifts Data node 9 to the left in the logical PC structure. The resulted figure is illustrated in figure 5.



(a) **(b)**

Figure 4 . A binary weighted tree (a) after adding a new data node and (b) after removing data node 8

5. Replication Management

The placement of replicas is critical to the storage system for data availability and fault tolerance. A good replica placement policy should improve data reliability, availability and network bandwidth utilization. Therefore, we focus replication strategy to achieve these goals.

5.1. System Model

The cloud storage system is implemented with PC cluster system. It is composed of N independent heterogeneous nodes storing of M different blocks $b_1,b_2,\ldots,b_M$. We model failure probability of data node S_i as f_i, the optimum replica number of a data block as R_{opt}. The availability of the system, α is defined as the fraction of time that the system is available for serving user requests. The system can be in one of the two states: the normal state or the idle state.

In normal state, at least one data node is available for serving user requests. However, idle state, P_0, is the state in which all data nodes containing the data block b_j have failed.

5.2. Degree of Replication

The goal of replication is to increase reliability and availability by keeping the data accessible even when failures occur in the system. It is clear that the reliability of a system will generally increase as the number of replicas or the degree of replication increases since more replicas will be able to mask more failures. However, it is a key issue how the degree of replication will affect system availability [14].

When replica number reaches a certain point, the file availability is equal to 1, and adding more replicas will not improve the file availability any more. The lower the node failure ratio, the less replica number it requires for file availability to reach 1. Therefore, we can only maintain minimum replicas to ensure a given availability requirement [14].

With replica number increasing, the management cost including storage and network bandwidth will significantly increase. In a cloud storage system, the network bandwidth resource is very limited and crucial to overall performance. Too much replicas may not significantly improve availability, but bring unnecessary spending instead.

With attention to this, we developed a model to express availability as function of replica number which is adapted from primary site approach described in [14]. This model is used to determine how much minimal replica should be maintained to satisfy availability requirement. Since the availability of the system is opposite of the idle state of the system, we get

$$\alpha = (1 - P_0) \quad (1)$$

The idle state of the system P_0 means that all replica nodes have failed with failure probability f_i. That is

$$P_0 = f_1 \, x \, f_2 \, x \ldots\ldots x \, f_{Ropt} = \Pi_{i=1}^{Ropt} f_i \quad (2)$$

Therefore, if the availability probability α and failure probability f_i is known, we get the optimum replica number R_{opt} by using the following equation such as:

$$\alpha = (1 - \Pi_{i=1}^{Ropt} f_i) \quad (3)$$

According to the equation (3), Name node calculates optimum replica number R_{opt} to satisfy expected availability with average data node failure probability f_i . In case of data node failure

and current replica number of a block is less than R_{opt} , additional replicas will be created into data node cluster.

5.3. Replica Placement Algorithm

After determining how many replicas the system should maintain to satisfy availability requirement, we will consider how to place these replicas efficiently to maximize performance and load balancing. In this paper, binary weighted tree is used to place replicas among data nodes.

Whenever a new block is added to the PC cluster, highly available nodes are selected by using binary weighted tree and the list of node id to store the replicas is returned to the client. According to the list, the client stores the data block on PC cluster. The notations used in the proposed algorithm are described in table 1 and the proposed replica placement algorithm is shown in figure 6. The Name node applies the proposed replica placement algorithm repeatedly for the number or replicas resulted from availability model.

Table 1 . Symbols used in replica placement algorithm

Symbol	Description
N_L	The total number of nodes in the left sub-tree
N_R	The total number of nodes in the right sub-tree
W_L	The total available capacity in the left sub-tree
W_R	The total available capacity in the right sub-tree
W_{root}	The available capacity of root node
Avg_L	The average capacity of left sub-tree
Avg_R	The average capacity of right sub-tree

```
ReplicaPlacementAlgo(rootnode)
If rootnode has no sub-tree
        Return nodeId of rootnode
Else
      AvgL ← WL / NL
      AvgR ← WR / NR
     If Wroot >AvgL>AvgR or Wroot > AvgR> AvgL
                Return nodeId of rootnode
      Else If  AvgL > AvgR
          Relay the block to the left subtree
          ReplicaPlacementAlgo(left-subtree)
      Else
          Relay the block to the right subtree
          ReplicaPlacementAlgo(right-subtree)
      End If
End If
```

Figure 6 . Replica placement algorithm

6. Theoretical Analysis

In this paper, we propose cloud storage system on PC cluster that is scalable and able to balance the storage loads of the cluster. In section 5, we demonstrated availability model as a function of failure probability and binary weighted tree to place replicas for balancing storage loads of PC nodes.

6.1. Scalability

A single server based storage system does not effectively meet the need of users at the terabyte-scale. Users need voluminous storage on cloud in order to properly manage and use their growing data. This paper presents PC cluster based cloud storage system to address this need. Adding and removing storage nodes to the system do not need the whole storage system to change. It only requires a minimum amount of administrative work. The new Data node contacts the Name node to reorganize logical PC cluster. Moreover, if users need additional storage on PC cluster, additional PC that has hard drive needs to be added to the PC cluster without affecting the whole PC cluster system.

6.2. Availability

In this section, we evaluate our availability model by demonstrating the relationship between failure probability and replica number as illustrated in figure 7. From the figure, it can be said that the availability model can adjust the replica number according to the failure probability and expected availability.

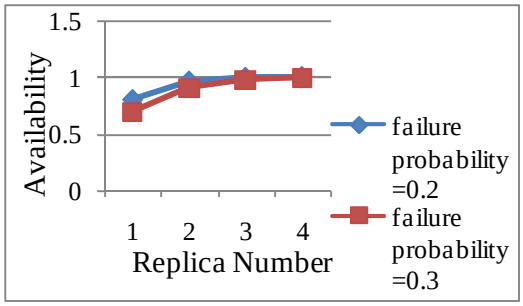


Figure 7 . Availability comparison of failure probability of 0.2 and 0.3

6.3. Balancing of Storage Load on PC Cluster

In order to achieve the load balance of the PC cluster, we apply binary weighted tree over physical PC cluster. The proposed replica placement over the binary tree structure makes the storage loads of the nodes to be balanced initially when the data block is added. Theoretically, searching the highly available node in the binary tree has time complexity of O(*log* N) where N is the number of nodes in the PC cluster.

6.4. Cost Effectiveness

The proposed system utilizes inexpensive computer machines as storage server of the cloud. It saves the cost of high performance storage server. Besides this, optimum replication strategy only needs minimum replica number to achieve expected availability. Since it eliminates unnecessary replication of data block, the storage cost is saved more efficiently.

7. Conclusion

In this paper, we presented a scalable cloud storage infrastructure on PC cluster. As the system is designed for a storage system using inexpensive computer machines, these machines may fail down which can be slow down and longer data retrieval time. Consequently, we propose binary weighted tree based logical PC organization in which PCs can be added or removed without failing the whole system to achieve scalability. In the future, we intend to analyze and enhance the performance of the system and believe that our system is cost effective and efficient utilization of computer machines.

References

[1] Amazon-S3,http://www.amazon.com, 2009.

[2] D. Borthakur. “The Hadoop Distributed File System: Architecture and Design”. *The Apache Software Foundation*, 2007.

[3] B. F. Cooper, R. Ramakrishnan, U. Srivastava, A. Silberstein, P. Bohannon, H.-A. Jacobsen, N. Puz, D. Weaver, and R. Yerneni. “Pnuts: Yahoo!’s Hosted Data Serving Platform”,*In VLDB*, 2008.

[4] G. Decandia, D. Hastorun, M. Jampani, G. Kakulapati,A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall and W. Vogels. “Dynamo: Amazon’s Highly Available Key-value Store”, *In SOSP*, 2007.

[5] H. V. Jagadish, B. C. Ooi, and Q. H. Vu,”BATON: A Balanced Tree Structure for Peer-to-Peer Networks”. In VLDB, 2005.

[6] A. Lakshman, P. Malik, “Cassandra - A Decentralized Structured Storage System”, *ACM SIGOPS Operating Systems Review, Volume 44 Issue 2*, April 2010.

[7] A. Sage. A. W. Scott . L.B. Ethan and et.al., “Ceph: A Scalable, High-Performance Distributed File System”, *Proceeding of 7th conference on operating system design and implementation (OSDI’06)*, November 2006.

[8] S. Ghemawat, H. Gobioff, S. T. Leung, “The Google File System”, *Proceedings of 19th ACM*

Symposium on Operating Systems Principles(SOSP 2003), New York USA, October, 2003.
[9] H. T. Vo, C. Chen , B.C. Oo, "Towards Elastic Transactional Cloud Storage with Range Query Support", *International Conference on Very Large Data Bases* , Singapore, September 13-17 2010.
[10] Q. Wei and et. al., "CDRM: A Cost-effective Dynamic Replication Management Scheme for Cloud Storage Cluster", *IEEE International Conference on Cluster Computing*, 2010.
[11] J. Wu1 and et. al., "Recent Advances in Cloud Storage", *Proceedings of the Third International Symposium on Computer Science and Computational Technology(ISCSCT '10)*, China, 14-15,August 2010, pp. 151-54.
[12] Y. Ye and et. al., "Cloud Storage Design Based on Hybrid of Replication and Data Partitioning", *16th International Conference on Parallel and Distributed Systems*, 2010.
[13] B. Furht, A.Escalante, *Handbook of Cloud Computing*, ISBN 978-1-4419-6523-3, Springer Science+Business Media, LLC 2010.
[14] P. Jalote, *Fault Tolerence in Distributed Systems*, ISBN 0-13-301367-7,A Pearson Education Company, 1994.