The Efficient Multi-Tired Storage System on Private Cloud

Cho Cho Khaing, Thinn Thu Naing
University of Computer Studies, Yangon
chokhaing.ck@gmail.com,ucsy21@most.gov.mm

Abstract

The widespread popularity of Cloud computing as a preferred platform for the deployment of web applications has resulted in an enormous number of applications moving to the cloud, and the huge success of cloud service providers. The distributed file system plays a very important role in cloud computing environments as well as it is necessary to store and manage data on low cost storage servers in which storage space can be reduced. Today's businesses are challenged with managing and protecting the ever-increasing amounts of data that is created on a daily and even hourly basis. To counteract this issue, this system proposes a method of Multi-tired storage system. The advantage of Multi-tired storage system can control and manage service levels to get better storage costs in different types of application environment.

1. Introduction

IT organizations worldwide are dealing with the tremendous growth of data. With the growth of capacity comes the complexity of managing the storage for that data. The data growth is coming from a wealth of data-intensive applications (e.g., business analytics), expanding use of high-performance computing (e.g., financial services and life sciences), collaboration and Web 2.0 applications, and content-rich data (e.g., digital images or video). In this data-intensive environment, IT managers need to optimize the capacity and performance of storage systems while working to reduce complexity and lower costs.

In addition to the continued growth in capacity, the accelerated use of virtual servers and desktops is rapidly altering the storage landscape. IT organizations worldwide are turning to virtualized environments to improve datacenter flexibility and scalability. This in turn drives implementation of networked storage solutions, which can create new pressures on storage performance as I/Os that were previously more distributed are aggregated into a smaller number of host interconnects. There are also implications for organizations' data protection processes and architectures to ensure that every virtual server is protected and that the storage has the same flexibility and resiliency as the virtualized server environment.

Storage efficiency has been a goal of primary storage systems managers and an area of storage innovation for some time. Disk storage system techniques such as thin provisioning, space-efficient snapshots, automated tiering, and virtual storage management have all been developed to help IT managers improve storage system utilization and efficiency[9]. More recently, data deduplication and compression technologies for primary storage have been gaining attention in the quest for improved storage efficiency. Deduplication is one of the few IT solutions that cuts costs quickly and improves service levels. With it, organizations can reduce storage expenses without compromising data protection.

This paper presents the efficient storage system in private cloud. This approach has been designed to use over virtualized storage system. The idea of the proposed system is to exploit the use of virtualization technology to avoid unnecessary storage purchases and reduce storage space for addressing large volumes of data handling problem.

The rest of this paper is organized as follows. Section 2 describes the related work. In section 3, cloud computing and virtualization is described. Section 4 discusses data deduplication system. Section 5 the proposed system is presented. Finally section 6 concludes the paper.

2. Related Work

Much work on deduplication focused on basic methods and compression ratios, not on high throughput.

Early deduplication storage systems use file-level hashing to detect duplicate files and reclaim their storage space [4,7,14]. Since such systems also use file hashes to address files. Some call such systems content addressed storage or CAS. Since their deduplication is at file level, such systems can achieve only limited global compression.

Venti removes duplicate fixed-size data blocks by comparing their secure hashes [12]. It uses a large on-disk index with a straightforward index cache to lookup fingerprints. Since fingerprints have no locality, their index cache is not effective. When using 8 disks to lookup fingerprints in parallel, its throughput is still limited to less than 7 MB/sec. Venti used a container abstraction to layout data on disks, but was stream agnostic, and did not apply Stream-Informed Segment Layout.

To tolerate shifted contents, modern deduplication systems remove redundancies at variable-size data blocks divided based on their contents. Manber described a method to determine anchor points of a large file when certain bits of rolling fingerprints are zeros [10] and showed that Rabin fingerprints can be computed efficiently. Brin et al. [6] described several ways to divide a file into content-based data segments and use such segments to detect duplicates in digital documents. Removing duplications at content-based data segment level has been applied to network protocols and applications [8, 3, 11, 13] and has reduced network traffic for distributed file systems [2,5]. Kulkarni et al. evaluated the compression efficiency between an identity-based (fingerprint comparison of variable-length segments) approach and a delta-compression approach [1]. These studies have not addressed deduplication throughput issues.

3. Cloud Computing and Virtualization

3.1. Storage for Cloud Computing

For cloud computing boot images, cloud storage is almost always offered via traditional block and file interfaces such as iSCSI or NFS. These are then mounted by the virtual machine and attached to a guest for use by cloud computing. Additional drives and file systems can be similarly provisioned. Of course cloud computing applications can use the data object interface as well, once they are running.

3.2. Server Virtualization

Virtualization is a method of running multiple independent virtual operating systems on a single physical computer. This approach maximizes the return on investment for the computer. The term was coined in the 1960s in reference to a virtual machine (sometimes called a pseudo-machine). The creation and management of virtual machines has often been called platform virtualization. Platform virtualization is performed on a given computer (hardware platform) by software called a control program. The control program creates a simulated environment, a virtual computer, which enables the device to use hosted software specific to the virtual environment, sometimes called guest software. The guest software, which is often itself a complete operating system, runs just as if it were installed on a stand-alone computer. Frequently, more than one virtual machine is able to be simulated on a single physical computer, their number being limited only by the host device's physical hard-ware resources. Because the guest software often requires access to specific peripheral devices in order to function, the virtualized platform must support guest interfaces to those devices. Examples of such devices are the hard disk drive, CD-ROM, DVD, and network interface card. Virtualization technology is a way of reducing the majority of hardware acquisition and maintenance costs, which can result in significant savings for any organization.

4. Data Deduplication

Data deduplication has become an important storage technology in the past few years. Storage solutions, either based on deduplication or with deduplication as a feature, are now available across the entire spectrum of storage offerings from many vendors, large and small. Data deduplication identifies and eliminates redundant data. It can be performed at the file, block, or byte level. The opportunity to find and eliminate redundancy becomes greater with more granular examination. In secondary storage processes, such as backup, data is initially seeded on the secondary storage device and all subsequently written data is examined for redundancy. Replicate data is not stored twice; instead, a pointer to the stored duplicate data is written (which takes up significantly less space).

4.1. Hash-Based Approach

A hash-based de-duplication system consists of three components - file chunking, hash value generation, and redundancy detection. When a new file arrives, the de-duplication system breaks entire file into many small blocks as know as chunks. Then the system uses hash algorithm, like Secure Hash Algorithm - 1 (SHA-1) to generate the unique signatures to present these chunks. Now, the de-duplication system can more easily and accurately compare these signatures with its database or lookup table to identify these small chunks are new data or redundant data. During the comparisons, if the system cannot find any signature matching in its lookup table or database, then the system assumes these chunks are new data and stores them. If the system detects a signature matching, then it creates a logical reference to the duplicate data whose copy was already stored in database.

4.2. Block-Level Deduplication

Block-level deduplication (fixed and variable) requires more processing overhead but allows for better deduplication of files that are similar but slightly different. All blocks are shared except for the different ones. This approach is very useful with virtual machine images, for example, which mostly include a large copy of the guest operating system with some blocks that are unique to each virtual machine.

5. Proposed System

The proposed system for the efficient data storage is as shown in Figure 1.

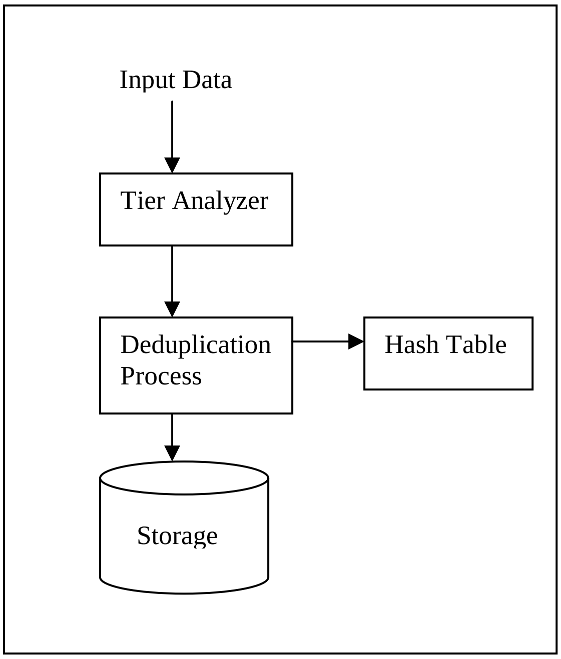

Figure 1. System Architecture of the proposed system

5.1. Tiered Storage Service Levels

The Tier 1 storage offers quick response times and fast data transfer rates. It is a great solution for organizations that need to effectively store high-performance data that demands high availability. Services of Tier 1 can be viewed from any cloud member. Tier 1 services such as notice board, news and calendar service. These service levels do not need to provide security system. Some of the Tier 2 and Tier 3 services have limited. Only authorized person can access it. This services need to support security policy. Tier 2 services are photo sharing, comments, prints, download ,upload, appointment with supervisors. Tier 3 services such as software access service, mail service and multimedia.

5.2. Tier Analyzer Process

```
Procedure Tier Analyzer Process
BEGIN
Set (tier1, tier2 and tier3) data
I=Input data
B: the array to store the input data and tier
   number
 For k=tier 1 to tier 3 do
   If I match k then
      Set B= (I, k)
      Deduplication process (B)
      Store data according to tier number
      Exit
    End if
  End for
  END
```

Figure 2. Algorithm of Tier Analyzer Process

5.3. Deduplication Process

```
Procedure Deduplication Process
BEGIN
While (input data)
**Divide** input data into block
**Generate** hash value (block_id)
**Find** the redundant data using hash value
End while
END
```

Figure 3. Algorithm of Deduplication Process

6. Conclusion

Data will continue to grow for IT managers, and they will continue to look for every opportunity to improve their storage utilization and efficiency. This proposed system is expected to be reduced hardware costs and management overhead. Multi-Tiered Storage System will control and manage different types of service levels as well. By using Deduplication process will save space, improve our data protection and increase speed. Reducing the amount of data redundancy will make backups more efficient and quicker as well. Thus, Network bandwidth will also improve after de-duplication.

6. References

[1] P. Kulkarni, F. Douglis, J. D. LaVoie, J. M. Tracey: “Redundancy Elimination Within Large Collections of Files”. In Proceedings of USENIX Annual Technical Conference, pages 59-72, 2004.

[2] N. Jain, M. Dahlin, and R. Tewari. TAPER: “Tiered Approach for Eliminating Redundancy in Replica Synchronization”. In Proceedings of USENIX File And Storage Systems (FAST), 2005.

[3] C. P. Sapuntzakis, R. Chandra, B. Pfaff, J. Chow, M. S. Lam, and M. Rosenblum.” Optimizing the migration of virtual computers”. In Proceedings of USENIX Operating Systems Design and Implementation, 2002.

[4] A. Adya, W. J. Bolosky, M. Castro, G. Cermak, R. Chaiken, J. R. Douceur, J. Howell, J. R. Lorch, M. Theimer, and R. P. Wattenhofer.” FARSITE: Federated, available, and reliable storage for an incompletely trusted environment”. In Proceedings of USENIX Operating Systems Design and Implementation (OSDI), December 2002.

[5] Athicha Muthitacharoen, Benjie Chen, and David Mazières. “A Low-bandwidth Network File System”. In Proceedings of the ACM 18th Symposium on Operating Systems Principles. Banff, Canada. October, 2001.

[6] S. Brin, J. Davis, H. Carcia-Molina. “Copy Detection Mechanisms for Digital Documents(weblink)”. 1994, also lso in Proceedings of ACMSIGMOD, 1995.

[7] P. Kulkarni, F. Douglis, J. D. LaVoie, J. M. Tracey: “Redundancy Elimination Within LargeCollections of Files”. In Proceedings of USENIX Annual Technical Conference, pages 59-72, 2004.

[8] N. T. Spring and D. Wetherall. “ A protocol-independent technique for eliminating redundant network traffic”. In Proceedings of ACM SIGCOMM, pages 87--95, Aug. 2000.

[9] S. Scully and W.Benjamin: “Improving Storage Efficiencies with Data Deduplication and Compression” May 2010.

[10] Udi Manber.” Finding Similar Files in A Large File System”. Technical Report TR 93-33, Department of Computer Science, University of Arizona, October 1993, also in Proceedings of the USENIX Winter 1994 Technical Conference, pages 17–21. 1994.

[11] J. C. Mogul, Y.-M. Chan, and T. Kelly. Design, implementation, and evaluation of duplicate transfer detection in HTTP. In Proceedings of Network Systems Design and Implementation, 2004.

[12] S. Quinlan and S. Dorward, Venti: “A New Approach to Archival Storage”. In Proceedings of the USENIX Conference on File And Storage Technologies (FAST), January 2002.

[13] S. C. Rhea, K. Liang, and E. Brewer.” Value-based web caching”. In WWW, pages 619–628, 2003.

[14] N. Tolia, M. Kozuch, M. Satyanarayanan, B. Karp, A. Perrig, and T. Bressoud. “Opportunistic use of content addressable storage for distributed file systems”. In Proceedings of the 2003 USENIX Annual Technical Conference, pages 127–140, San Antonio, TX, June 2003.