# Rethinking Deduplication in Cloud: From Data Profiling To Blueprint

Chulmin Kim, Ki-Woong Park
CORE Lab., KAIST
Daejeon, Korea
{cmkim, woongbak}@core.kaist.ac.kr

KyoungSoo Park
NDSL, KAIST
Daejeon, Korea
kyoungsoo@ee.kaist.ac.kr

Kyu Ho Park
CORE Lab., KAIST
Daejeon, Korea
kpark@ee.kaist.ac.kr

***Abstract*—Cloud storage system is becoming the substantial component of the cloud system due to emerging trend of user data. Different from other computing resources, storage resource is vulnerable to the cost issue since the data should be maintained during the downtime. In this paper, we investigate the benefit and overhead when deduplication techniques are adopted to the cloud storage system. From the result, we discuss several challenges across the cloud storage. Furthermore, we suggest the cloud storage architecture and the deduplication engine to optimize the deduplication feature in the cloud storage system. We expect that our suggestions reduce the cloud storage system cost efficiently without performance degradation of data transfer.**

## I. Introduction

Cloud computing is an important transition and a paradigm shift in the Internet technology. In recent years, emerging cloud services, including mobile office, web-based storage service, and content delivery service have become popular. Examples include GoogleDocs [1], DrobBox [2], and Amazon S3 [3]. It allows users to store data on the networked storage of service provider. The pay-per-use model of the cloud storage services brings significant savings for users. It offers a flexibility and scalability in terms of capacity and performance.

Amazon EC2 and S3 [3] service is the most popular cloud service in the world. As shown in Fig. 1, overall storage resources of the cloud service provider can be divided into two sections. They are the VM image repository and user data repository; it provides users with storages in two ways. At the initiation time of a user's virtual machine (VM), the local storage containing OS image and essential system files is assigned to the VM as a booting partition. The storage can be backed up into a VM image repository when the VM instance is turned off or snapshotted. If the user requests an additional data space to the cloud provider, the provider will rent the space of user data repository owned by the provider.

From the service provider's perspective, the ability to utilize storage efficiency technologies such as deduplication is a critical metrics to provide a qualified services in a cost efficient manner. It is because servicing storage space can be cost burden. While CPU and memory is only needed during the runtime, storage space should be maintained even though the user's VM status is turned off. For this reason, the data deduplication technology which saves storage resource has been highlighted by the cloud service provider. The deduplication in storage area is not the new concept. However, the

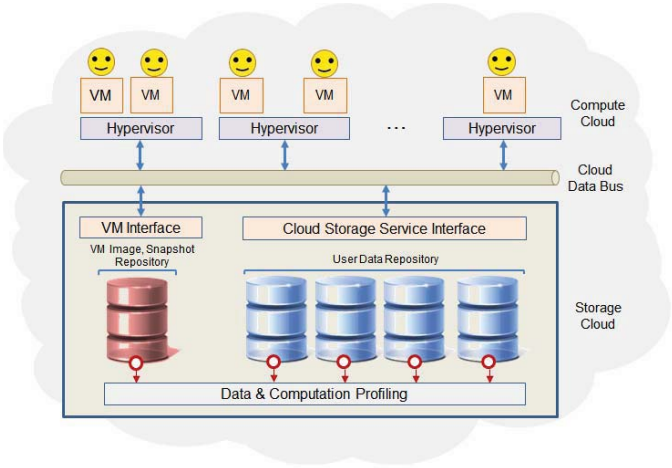


Fig. 1. Overall Storage Architecture of the Cloud Computing Environment

usage so far are limited to backup systems such as the bank system. Data deduplication is an effective way of enabling high-performance and efficient storage system especially when used in cloud computing environment. It allows the nominally separate system images (OS, system files) for each VM into a single storage space. By reducing the amount of storage needed, deduplication can save other resources such as the energy consumption, physical volume, and cooling needs to store the data.

As a step to adopt the deduplication in the cloud service, we investigate how much benefit can be obtained from the deduplication. Comparing the benefit with the overhead needed for the deduplication, we discuss the challenges of current deduplication techniques to be applied in the cloud storage system. We insists that the emerging hardware components such as flash-based solid state drives (SSD), general-purpose computing on graphics processing units (GPGPU), and multi-core CPU offer a new possibility to realize a high-end and cost-efficient deduplication system. Consequently, we intensively discuss the future direction of deduplication system architecture for better deduplication in the cloud service.

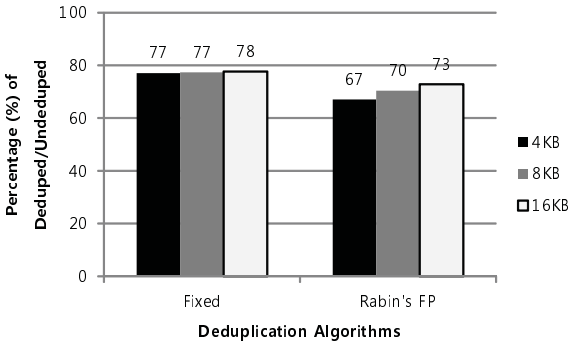


Fig. 2. Deduplication Efficiency in VM Image Repository

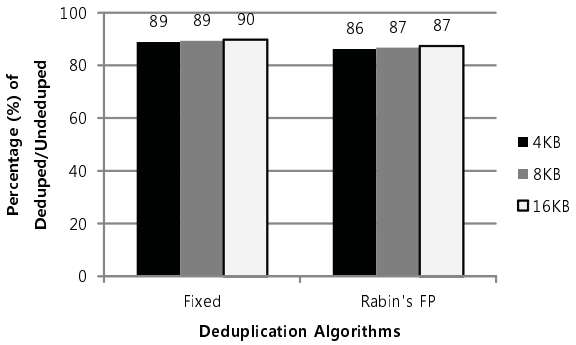


Fig. 3. Deduplication Efficiency in User Data Repository

## II. Preliminary Experiments for Deduplication in Cloud Storage System

### A. Deduplication in Cloud Storage System

The real implementation of the cloud storage system would be much more complex than the system shown in Fig. 1 since the system is enormous size and each disk is connected through the network. We simplified the cloud storage system since our focus is on the high-level data layout for the deduplication [4]–[6]. Briefly reminding, the deduplication removes the identical data block write. If there is the identical data block already in the storage, the deduplication filesystem memorizes where the block address is instead of writing the block in the vacant disk space. To find the identical data block in the entire storage, the deduplication technique maintains the hash values of data blocks in the storage and manages them in the index table. Whenever a write request arrives, hash value of the data to be written is calculated and the hash value table is scanned to find the identical hash value of the data.

There are a lot of deduplication techniques depending on the algorithms chunking the data blocks to the deduplication chunks, average deduplication chunk size, and so on. In this paper, we choose Fixed Block [7] and Rabin's Fingerprint [8] which are the most well known algorhtms as the representatives. Fixed Block algorithm literally uses a fixed size block as a unit of the deduplication while Rabin's Fingerprint uses variable block size. Varying the average chunk size and the chunking algorithms, we measured the deduplication efficiency and overhead in two regions (VM image repository, User data repository) of the cloud storage system, separately. We used that the ratio of the deduplicated storage size to the undeduplicated storage size to show the deduplication efficiency. Low value of the ratio indicates more efficient deduplication.

### B. Expected Efficiency of Deduplication

We have done several preliminary experiments in this cloud storage system to measure the various effects of deduplication. Since the objective of each storage region is different, trends of the experiments with same deduplication parameter also show some differences depending on the storage region.

We measured deduplication efficiency within a filesystem itself, first. Filesystems with Linux OS images represents the data in the VM image repository, and filesystems without any OS image represents the data in the user data repository. For each cloud storage part, 3 filesystems are examined in total and the averaged results are shown in Fig. 2, 3. The measurement is the deduplication efficiency we defined previously. Above all, the efficiency differs in between storage areas. In case of the result of 4KB chunk size and Fixed Chunk Size algorithm, the VM image repository shows 77% deduplication ratio and the user data repository reports 89%.

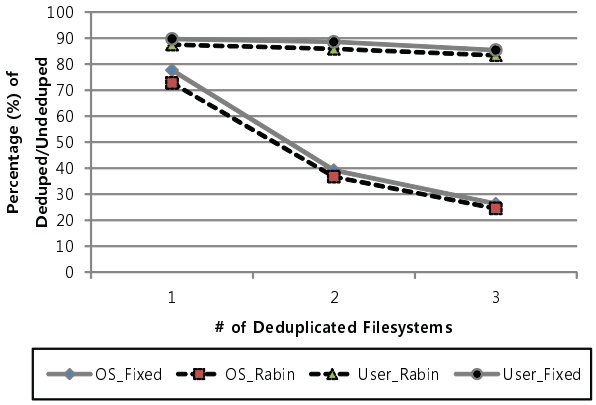


Fig. 4. Deduplication Efficiency Varying the number of Filesystems in a deduplication set

Depending on chunk unit sizes and algorithms of deduplication, the efficiency also varies slightly. When chunk size gets smaller or Rabin's fingerprint is used instead of fixed chunk size, it shows better deduplication efficiency. While this trend appears in both experiments similarly, the amount of changes is limited within 6% which is small.

We measured the deduplication efficiency varying the number of filesystems in a deduplication set. Multiple filesystems in the set means that there are more chances for data blocks to be deduplicated. Especially for the VM image repository, each filesystem might have similar or same files and data blocks if the type of OS is same. In our experiments, filesystems for the VM image repository commonly contains Linux OS. As shown in Fig. 4, the deduplication efficiency of the VM image

TABLE I
COMPRESSION OF USER DATA STORAGE

| | Original Data Size | Compressed Data Size | Comp. Rate |
|---|---|---|---|
| Filesystem 1 | 69,797,949 | 56,989,893 | 82% |
| Filesystem 2 | 171,938,885 | 150,690,143 | 88% |
| Filesystem 3 | 89,323,298 | 49,671,737 | 56% |
| Total | 331,060,132 | 257,351,773 | 78% |

repository gets better following the number of filesystems in a deduplication set. In contrast, filesystems in the user data repository contain typical user data such as video files, music files, documents and so on. Possibility of the duplicated appearance of these files in multiple filesystems is relatively low. Thus, there is not much improvement of the deduplication efficiency for the user data repository as illustrated in Fig. 4.

Additionally, we compared the deduplication efficiency with the compression which is another storage saving technique. Results in Table. I illustrates the compression efficiency of the filesystems represented the user data repository in the previous experiments. In average, it shows better efficiency than that of the deduplication since the compression uses much smaller size of data to detect the similarity even though it does not scan all the data. In the filesystem with very bad deduplication efficiency, the compression can be another option for the cloud storage.

### C. Expected Overhead of Deduplication

To measure the overhead of the deduplication, we first count the number of hash index entries which build the deduplication table. Since the deduplication process should scan the entries to find the matched block regardless of the possibility of the matching, it can be one of the performance metric of the deduplication. We measured all index counts per experiment done previously. In common, the number of index counts is proportional to the data size after the deduplication which is appeared in the form of the deduplication efficiency in Fig. 2, 3. If the deduplication efficiency is good, the benefit includes not only space saving, but also the reduced deduplication overhead.

In addition, we measured how much computation power is needed to process the deduplication. Since a hash operation per data block write should be done for the deduplication, the computation burden is higher than that of general filesystems. Using oprofile(reference), the number of clock cycles consumed for the hash operation is profiled when files are written into ZFS deduplication filesystem [9] by bonniee++ benchmark [10]. We can see that the hash function in the crypto library needs about half of entire clock cycles consumed for the data transfer in Table. I.

TABLE II
OPROFILE RESULTS OF CLOCK CYCLE EVENTS

| Symbol Name | Event Count | Ratio |
|---|---|---|
| libcrypto.so.0.9.8e | 1,438,001 | 52% |
| libc-2.5.so | 582,005 | 21% |
| libpthread-2.5.so | 291532 | 11% |
| zfs-fuse | 269,732 | 10% |
| bonnie++ | 110,031 | 4% |
| librt-2.5.so | 23,421 | 1% |
| libfuse.so.2.8.5 | 17,490 | 1% |
| oprofiled | 15,082 | 1% |

## III. DISCUSSION

The cloud providers want to save the space while the deduplication does improve or not harm the storage throughput. To mitigate the desire of the cloud providers, cloud system has several interesting challenges across various topics from overall storage architecture to the hardware specification of the deduplication engine in the cloud. We discuss the challenges in detail through this section.

From the preliminary experiments in Section 2, we have learned 5 kinds of lessons elaborated in the below:

1) OS Storage brings more deduplication efficiency.
2) Aggressive deduplication brings few improvement in general.
3) More Filesystems in a deduplication set brings more deduplication efficiency.
4) Compression is another option to save the disk space.
5) Deduplication overhead consists of hash computation and large amount of hash entry reads.

For each item of challenges, we suggest the blueprints thoroughly based on these lessons.

First, the deduplication engine should be constructed not to be bottleneck of the cloud storage system. For the deduplication, we need the deduplication engine which computes the key hash value from a written data block and search the identical hash entry from hash index table. Due to enormous size of cloud storage system, the deduplication will bring large amount of hash operations and hash index entries which is the direct overhead described in Lesson 4. To resolve the bottleneck, our suggested deduplication engine equips the emerging hardware components such as flash-based solid state drives(SSD) and general purpose computing on graphics processing units (GPGPU). GPGPU can compute the hash operations in parallel so that the system can be freed from the hash operation bottleneck. While disk reads due to the excessive hash index entries makes a latency of the write requests higher, SSD which has better latency than that of a hard disk can mitigate the latency problem.

Second, cloud storage architecture should be organized in more smart way. In the fundamental view, the deduplication efficiency gets higher when the similar data resides in the same storage. We verified this from Lesson 1. However, data amount of the VM image repository is the minor part of entire cloud storage space. To save the disk space significantly, the user data repository should be deduplicated efficiently. The best solution to enhance the deduplication efficiency is increasing the number of filesystems in a deduplication set as described in Lesson 3 while it also brings more performance overhead. Monitoring the deduplication efficiency and overhead, a cloud

provider can decide the appropriate number of filesystems which fits into the cloud provider's SLA and charging policy. Further, we can save the disk space more using the compression related with Lesson 5. If a data block is not pointed by multiple users and the block is not accessed recently, it is wasting the disk space. In other word, it is better to compress the block to save the space.

In the last moment, policies of the deduplication will decide the cost and efficiency of the deduplication. The cloud provider should choose when the deduplication will be done. Current deduplication techniques are classified into inline deduplication [5] and post-processing deduplication [11]. While the former is done in front of the storage before writing the data into the disk space, the latter write the data once and reproduce the deduplicated image for the written data. In the viewpoint of the cloud providers, inline deduplication is more appropriate for the cloud storage system. Since write requests to the storage with inline deduplication should come by the deduplication engine before to be written, inline deduplication has more latency than post-processing deduplication. Instead, inline processing can earn space efficiency and reduced network transmission to the cloud storage system. Moreover, if the data block to be written finds its identical block in the cloud storage system, it can avoid a disk write. When the cloud provider uses SSD as a basic disk unit for the cloud storage system, the benefit from the deduplication hit of inline deduplication will be doubled since the write performance of SSD is much below the read performance and the number of write operations is limited for endurance of SSD.

## IV. Related Work

A number of works have investigated the performance and efficiency of the deduplication. Among them, the most recent research work of Meyer et al. [12] investigated the deduplication effects in practical. They profiled about 857 filesystems in use. While it provides the practical results of many deduplication issues, its work is bounded to the investigation for the traditional deduplication system. We extended the viewpoint of investigation for the cloud storage system, and suggested the future cloud storage system based on the results of the investigation on the cloud storage system.

## V. Conclusion

In an attempt to adopt the deduplication technologies into the cloud storage system, we have thoroughly analyzed the dataset from diverse VMs and multiple users' data. Consequently, we have reported on our experience on deriving the blueprint for deduplication system for cloud storage system. Our experience suggests that a deduplication system for cloud storage should have the following features:

1) The emerging hardware components such as flash-based solid state drives (SSD), general-purpose computing on graphics processing units (GPGPU), and multi-core CPU offer a new possibility to realize a high-end and cost-efficient de-duplication system without performance bottleneck;
2) Cloud storage architecture should be organized regarding the data similarity for the deduplication efficiency;
3) Cognitively applying deduplication techniques (inline deduplication and post-processing deduplication) brings significant improvement in terms of deduplication efficiency.

We hope that these lessons will help to light the way to realize deduplication system for cloud computing environment. The initial discussions in this paper are the first step in our attempts to realize the deduplication system for cloud storage system. As the next step of this study, we plan to expand on some key lessons learned from our experiences and continue to work on going forward to realize a deduplication system for cloud storage.

## References


[1] Google.com, "Google docs: Online documents, spreadsheets, presentations," Google Co. Ltd., http://docs.google.com, April 2011. [Online]. Available: http://docs.google.com

[2] DropBox, "Dropbox: Online backup, file sync, and data sharing," DropBox Co. Ltd., http://www.dropbox.com, April 2011. [Online]. Available: http://www.dropbox.com

[3] Amazon.com, "Amazon simple storage service s3," Amazon Co. Ltd., http://aws.amazon.com/s3/, April 2011. [Online]. Available: http://aws.amazon.com/s3/

[4] B. Zhu, K. Li, and H. Patterson, "Avoiding the disk bottleneck in the data domain deduplication file system," in *Proceedings of the 6th USENIX Conference on File and Storage Technologies*, ser. FAST'08. Berkeley, CA, USA: USENIX Association, 2008, pp. 18:1–18:14. [Online]. Available: http://portal.acm.org/citation.cfm?id=1364813.1364831

[5] M. Lillibridge, K. Eshghi, D. Bhagwat, V. Deolalikar, G. Trezise, and P. Camble, "Sparse indexing: large scale, inline deduplication using sampling and locality," in *Proccedings of the 7th conference on File and storage technologies*. Berkeley, CA, USA: USENIX Association, 2009, pp. 111–123. [Online]. Available: http://portal.acm.org/citation.cfm?id=1525908.1525917

[6] A. T. Clements, I. Ahmad, M. Vilayannur, and J. Li, "Decentralized deduplication in san cluster file systems," in *Proceedings of the 2009 conference on USENIX Annual technical conference*, ser. USENIX'09. Berkeley, CA, USA: USENIX Association, 2009, pp. 8–8. [Online]. Available: http://portal.acm.org/citation.cfm?id=1855807.1855815

[7] S. Quinlan and S. Dorward, "Venti: A new approach to archival data storage," in *Proceedings of the 1st USENIX Conference on File and Storage Technologies*, ser. FAST '02. Berkeley, CA, USA: USENIX Association, 2002. [Online]. Available: http://portal.acm.org/citation.cfm?id=1083323.1083333

[8] M. O. Rabin, "Fingerprinting by random polynomials," in *Harvard University Technical Report*, 1981.

[9] O. Rodeh and A. Teperman, "zfs-a scalable distributed file system using object disks," in *Proceedings of the 20th IEEE Goddard Conference on Mass Storage Systems and Technologies*, 2003, p. p.207.

[10] R.Coker, "Bonnie++," September 2001. [Online]. Available: http://www.coker.com.au/bonnie++

[11] T. Yang, D. Feng, Z. Niu, K. Zhou, and Y. Wan, "Debar: a scalable high-performance deduplication storage system for backup and archiving," in *IEEE International Symposium on Parallel and Distributed Processing*, 2010.

[12] D. T. Meyer and W. J. Bolosky, "A study of practical deduplication," in *Proceedings of the 9th USENIX conference on File and stroage technologies*, ser. FAST'11. Berkeley, CA, USA: USENIX Association, 2011, pp. 1–1. [Online]. Available: http://portal.acm.org/citation.cfm?id=1960475.1960476