Understanding the Data Duplication in VM Cloud Storage

Wei Zhang[⋆†], Hong Tang[†], Hao Jiang[†], Tao Yang[⋆]
[⋆]UC Santa Barbara, {wei, tyang}@cs.ucsb.edu
[†]Alibaba.com, {hongtang, haojiang}@alibaba.com

Abstract

Backup in VM cloud is mainly done by storing snapshots of VM images. Since VM images are huge and backup is frequent, the snapshot storage in VM cloud must deduplicate snapshot data to save cloud resources. Thus it is very important to exploit the duplication pattern of snapshot backup data for designing better cloud storage systems. In this paper we present the data duplication pattern that we observed from Aliyun's publie VM cloud services, and provide some insights for researchers and engineers to design future deduplication strategies for VM cloud backup.

1 Introduction

In a virtualized cloud environment such as ones provided by Amazon EC2 and Aliyun, each instance of a guest operating system runs on a virtual machine, accessing virtual hard disks represented as virtual disk image files in the host operating system. Because these image files are stored as regular files from the external point of view, backing up VM's data is mainly done by taking snapshots of virtual disk images.

Frequent backup of VM snapshots increases the reliability of VM's hosted in a cloud. For example, Aliyun, the largest cloud service provider by Alibaba in China, provides automatic frequent backup of VM images to strengthen the reliability of its service for all users. The cost of frequent backup of VM snapshots is high because of the huge storage demand.

Unlike the legacy backup systems[3] dealing with general file-level backup and deduplication, Backing up VM images is different. Although each VM image is treated as a file logically, its size is very large. On the other hand, a cloud must support parallel backup of a large number of virtual disks everyday. One key requirement we face at Aliyun is that VM snapshot backup should only use a minimal amount of system resources so that most of resources is kept for regular cloud system services or applications. Thus our objective is to exploit the characteristics and duplication pattern of VM snapshot data and pursue a cost-effective deduplication solution.

There are several previous studies on this topic. Jayaram[4] and Jin[5] has investigated the data similarity between VM images using Rabin's fingerprinting[2] algorithm. Silo[7] and Extreme Binning[1] studied the problem of deduplication in a large distributed environment which also helps solving the problem of VM snapshot backup.

In this paper we present the data analysis of Aliyun's VM snapshot data. Our work is different from previous studies at several aspects: First, we are targeting at the problem of snapshot backups, and no previous study has studied VM image with backup data involved. Second, we focus on observing the pattern of data duplication in VM snapshot backups, rather than examining the effect of variable-sized chunking algorithm. Finally, we use real user's VM data rather than hand-made VM images.

2 Experiment Setup

We sampled two data sets from Aliyun's public VM cluster, all VMs are used by real external users running various applications such like database, web server, rendering services or even Hadoop. Each VM has two virtual disks, one is for OS and software installations, and the other one is for storing user data contents.

Data set VOSS composes of the OS disks from 35 VMs in 7 popular OSes: Debian, Ubuntu, Redhat, CentOS, Win2003 32bit, win2003 64 bit and win2008 64 bit. For each OS, 5 VMs are chosen, and every VM come with 10 full snapshots of it OS disk. So there is 350 full snapshot backups in this data set, overall size is about 7 TB. We use VOSS to study the backup duplication characteristics, and OS disk change patterns.

Data set DDS contains the first snapshots of 1323 VMs' data disks from a cluster with 100+ nodes. Since no backup duplication is involved in this data set, this data set helps us to study the duplication pattern of user generated data. The overall size of dataset2 is near 23 TB.

3 Overall Deduplication Effect

Our study start from examining the effect of complete deduplication over both data sets. Each virtual disk image is divided into small variable-sized blocks using the TTTD algorithm[6], under the average 4KB, maximum 16KB and minimum 2KB setting. Complete deduplication is done by caculating the SHA-1 hash of each block and identifying duplicate copies base on compare-by-hash.

Data Set	OS Type	Original Size (GB)	After Complete Deduplication (GB)	Ratio	After 2MB Page Reduction (GB)	Ratio
VOSS	Debian	1034.59	77.10	13.42	218.29	4.74
	Ubuntu	989.32	81.60	12.12	178.38	5.55
	RHEL	1007.28	62.70	16.07	215.33	4.68
	CentOS	973.03	57.34	16.97	522.53	1.86
	Win2003 32Bit	630.37	30.12	20.93	150.31	4.19
	Win2003 64Bit	793.47	44.16	17.97	167.15	4.75
	Win2008 64Bit	1508.97	46.39	32.52	222.54	6.78
	Combined	6937.03	389.65	17.80	1674.53	4.14
DDS		23125.2	10887.2	2.12		

Table 1: Data reduction via complete deduplication and dirty page reduction

Regarding to the VM snapshots storage, one popular technique is to split disk image file into fix-sized pages, and only store the dirty pages since last backup. But unlike complete deduplication considering duplicates across VMs, this technique is limited within each VM's backups. We also tested this data reduction method using 2MB page size over the dataset VOSS.

Table 1 shows the data reduction of both methods, the reduction ratio is defined as the original size divide by size after reduction. In general, complete deduplication achieves great data reduction, except for DDS data set.

Observation 1: Both methods reduce the data significantly, but there is still a big gap between the effect of dirty page method and what complete deduplication can acheive. We believe complete deduplication can reduces 2x 5x more data than dirty page method for several resons: First, the dirty page method doesn't resolve the duplication across VM backups. Second, the page size is usually much bigger than actual range of modification, so many unchanged data are still backed up.

Observation 2: Locality could be damaged with system upgrade. We notice dirty page method works poorly on CentOS, while complete deduplication works as efficient as on other OSes. We believe this is probably due to the user have upgraded his system heavily during our data sampling period, thus damaged the offset-based locality. In addition, the cloud will only have more and more variations of operating systems with different installation configurations, All these suggest data reduction base on offset may not work across different user snapshot backups.

Observation 3: There is not much duplication on user data if without multiple backups. For DDS data set, because it doesn't contain snapshot backups, the overall reduction ratio is only about 2:1, which suggests we probably should put less effort on the reduction of user generated data. These data are changed slowly and mostly by append, so locality should work well enough.

4 Impact of Locality

Locality is widely used in many deduplication solutions, e.g., Zhu et al. [8] put in memory cache of block indices near the one which result in a disk index look-up and is found. This is base on the observation that duplicates usually come in sequence rather than independently.

We check this fact in VM snapshots by monitoring the modification locations in VOSS. For each snapshot, we compare it to the previous snapshot, in the unit of 2MB fix-sized page, to find out locations of dirty pages since last backup. We end up with a bitmap of dirty pages for every snapshot, except those earliest ones who don't have a previous version to compare against.

In Figure 1, a snapshot's bitmap of dirty pages is represented as a vertical line composed of discontinuous segments. For each vertical line, the solid segments indicate changed regions since last snapshot backup, and the rest represent the unchanged part. All page locations (offset) is normalized with respect to the virtual disk size. Because the earliest snapshots are excluded, every VM has 9 lines and there are 45 lines from each OS. The bitmaps are first grouped by VMs and then by their OS type, borders between OS types are shown as numbers at the xtics.

Observation 4: Each VM do has its own specific interested write regions, as a result, unchanged regions appear to be large and continous. We see almost all VMs have their write regions aligned across 9 snapshots, leaving the unchanged regions aligned as well. So duplicates in the white area do come together during snapshot backup. This observation strongly proves the importance of locality as the primary indication of finding duplication.

Observation 5: Windows tends to write to more disk locations than Linux. Although there are several Linux VMs write to large area of disk locations, this is very likely because those users were more active during our samlping period. On the other hand, all Windows VMs write to almost everywhere of the disk during every backup period. We believe this is due to the design dif-

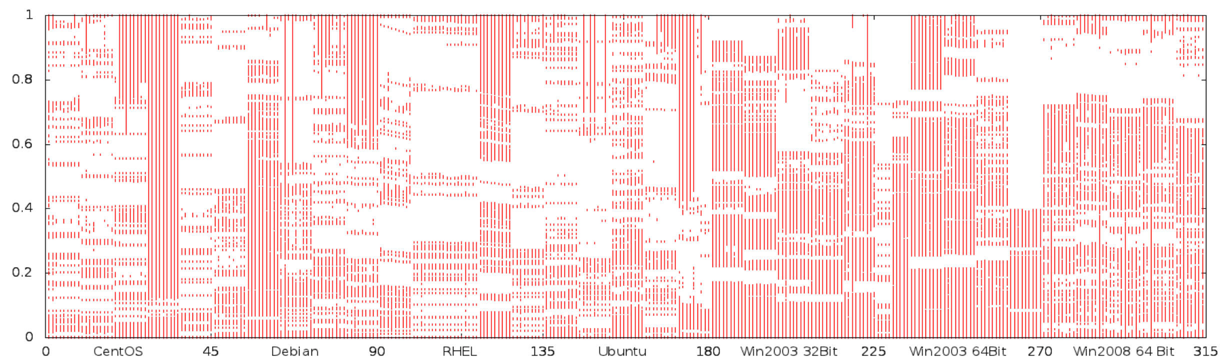

Figure 1: Bitmaps of dirty pages between snapshots

ference between NTFS and ext3, and we also see the upgrade from Win2003 to Win2008 reduces such writes. However, this doesn't mean windows has more dirty pages, it's just more scattered.

In order to see the impact of locality further, we investigate deeper into the dirty pages. We split those fix-sized pages into variable-sized 4KB blocks, for every block in the dirty pages, we look into the corresponding page (base on offset) of previous snapshot to see if a duplicate can be found.

OS Type	After Reduction Within Dirty Page (GB)	Ratio
Debian	161.11	6.42
Ubuntu	145.02	6.82
RHEL	185.34	5.43
CentOS	479.78	2.03
Win2003 32Bit	80.14	7.87
Win2003 64Bit	95.07	8.35
Win2008 64Bit	172.22	9.76
Combined	1318.68	5.26

Table 2: Data reduction via 4KB block deduplication within dirty pages

Observarion 6: Data reduction ratio is generally improved, but there is still lots of improvment space compare to complete deduplication. We see significant improvment for all OS types, which definitely deserves adding an additional layer to dirty-page based data reduction. In addition, the cost of looking one page's block hash index shall be very small compare to full hash index lookup.

However, this is probably the best of what we can get from locality. The rest of duplication mainly lies between VMs, e.g., same or slightly different versions of OS distributions, similar software installations, etc. Such duplications can not be easily eliminated and worth further research efforts.

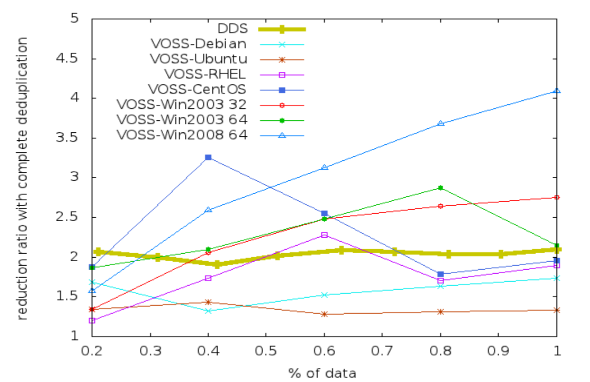

Figure 2: Reduction ratio of complete deduplication at different scale

5 Duplication V.S. Scale

Dealing with large scale VM cloud, one of the most important questions is whether data duplication will grow with the system scale. One hypothesis we have is that when the total number of block increases, the chance that it being duplicated with another is going to be higher. Thus we shall get better data reduction with larger scale of data. Past study[5] has suggested that virtual disks with operating systems installed should share large amount of software related data, which indirectly supports this argument. However, some other data study[4] showed the opposite.

In this experiment we test the complete deduplication under the scenario that no snapshot backup is involved. To simulate the scale factor, we start with partial data sets and scale them up by adding disk images. Complete deduplication is performed at each stage to examine the reduction ratio. Our DDS data set already contains no snapshot backup data. For VOSS, we simply pick the earliest snapshot for each VM.

Observation 7: Reduction ratio by complete deduplication does not appear to increase. In general we see almost no increment of reduction ratio along with the data scale, especially for user generated data, shown as DDS in Figure 2, its reduction ratio is always near 2:1. For OS disks, some Windows distributions appear to have incremental reduction, which is suspected to because the size of Windows system is much bigger than Linux, thus it has much more redundant data that widely exist in all VMs. We suspect the operating system and software related data will occupy a significant percentage of data reduction regardless of number of VMs, will be studied in the next section.

6 Heavily Duplicated Data

Knowing that locality has ruled the data reduction within individual VM's snapshot backups, in this section we investigate the dominant factor of data duplication between VMs.

Many deduplication study has mentioned that zero-filled blocks were widely exist, so we count the zero-filled blocks using the variable 4KB scheme to examine their weights in VM storage. Table 3 shows the total size of zero-filled blocks divide by data set size.

Data Set	Weight of Zero-filled Blocks
VOSS	8.1%
DDS	21.3%

Table 3: Weight of zero-filled blocks in our data sets

Observation 8: User generated data are much less compact than operating system data. We see 21.3% reduction from DDS by removing zero-filled blocks. Consider that full deduplication can only reduce about 50% on DDS, this indicates we could achieve great reduction by simply avoid storing zero-filled blocks in storage system.

Data on OS disks seem much more compact than user generated contents. However, their duplication may come in from another way. Consider 99% of our VM users run either Linux or Windows, and most of them only uses a small selection of software (e.g., MySQL, Apache), such operating system and software related data are very likely to be duplicated across VMs.

We examine this by using the public VM images in Aliyun's cloud environment as a reference. Every VM is generated from one of the public VM images by copying its data from base image, so in order to see whether these data are changed after VM usage, we first divide the public images into 4KB variable-sized blocks, then for each type of OS, we check its 50 VM snapshot backups in VOSS, to see if this data block still exist in all 50 backups. This is a very strict criteria, a data block will only be marked as "unchanged" if it appear in all 50 snapshots.

Observation 9: OS and software related data rarely change. This strongly indicates we should avoid storing such data in any VM's snapshot backups, if we know they can be obtained from somewhere else. In addition, knowing these data are widely used by VMs shall help us developing fast content delivery scheme for VM migration and snapshot restoration.

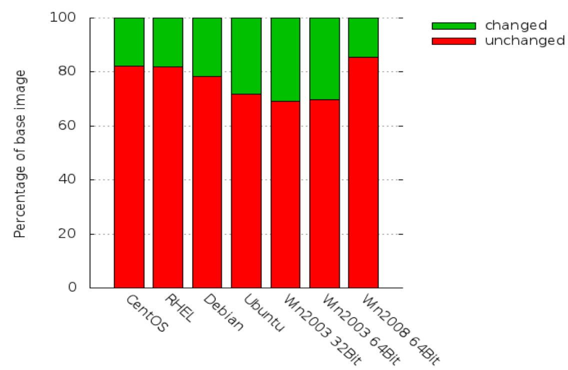

Figure 3:

7 Conclusion

In this paper, a empirical study using Aliyun's real user VM data traces is conducted. We present a number of observations and insights on the data duplication in VM snapshot backups. We describe in detail the patterns of VM access, VM snapshot duplication, locality reduction, and the dintinct difference between OS and user generated data. We hope this will facilitate future designs of cloud storage for VMs.

References

[1] D. Bhagwat, K. Eshghi, D. D. E. Long, and M. Lillibridge. Extreme Binning: Scalable, parallel deduplication for chunk-based file backup. In *Modeling, Analysis Simulation of Computer and Telecommunication Systems, 2009. MASCOTS '09. IEEE International Symposium on*, pages 1–9, 2009.

[2] A. Z. Broder. Identifying and Filtering Near-Duplicate Documents. In *COM '00: Proceedings of the 11th Annual Symposium on Combinatorial Pattern Matching*, pages 1–10, London, UK, 2000. Springer-Verlag.

[3] K. Eshghi, M. Lillibridge, L. Wilcock, G. Belrose, and R. Hawkes. Jumbo store: providing efficient incremental upload and versioning for a utility rendering service. In *FAST '07: Proceedings of the 5th USENIX conference on File and Storage Technologies*, pages 123–138, Berkeley, CA, USA, 2007. USENIX Association.

[4] K. R. Jayaram, C. Peng, Z. Zhang, M. Kim, H. Chen, and H. Lei. An empirical analysis of similarity in virtual machine images. In *Proceedings of the Middleware 2011 Industry Track Workshop on - Middleware '11*, pages 1–6, New York, New York, USA, Dec. 2011. ACM Press.

[5] K. Jin and E. L. Miller. The effectiveness of deduplication on virtual machine disk images. In *Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference on - SYSTOR '09*, page 1, New York, New York, USA, May 2009. ACM Press.

[6] E. Kave and T. H. Khuern. A Framework for Analyzing and Improving Content-Based Chunking Algorithms. Technical Report HPL-2005-30R1, HP Laboratory, Oct. 2005.

[7] W. Xia, H. Jiang, D. Feng, and Y. Hua. SiLo: a similarity-locality based near-exact deduplication scheme with low RAM overhead and high throughput. pages 26–28, June 2011.

[8] B. Zhu, K. Li, and H. Patterson. Avoiding the disk bottleneck in the data domain deduplication file system. In *FAST'08: Proceedings of the 6th USENIX Conference on File and Storage Technologies*, pages 1–14, Berkeley, CA, USA, 2008. USENIX Association.