Multi-level Selective Deduplication for VM Snapshots in Cloud Storage

Wei Zhang*, Hong Tang†, Hao Jiang†, Tao Yang*, Xiaogang Li†, Yue Zeng†
*Dept. of Computer Science, UC Santa Barbara. Email: {wei, tyang}@cs.ucsb.edu
†Alibaba Inc. Email: {hongtang, haojiang, xiaogang, rachael}@alibaba-inc.com

***Abstract*—In a virtualized cloud computing environment, frequent snapshot backup of virtual disks improves hosting reliability but storage demand of such operations is huge. While dirtybit-based technique can identify unmodified data between versions, full deduplication with fingerprint comparison can remove more redundant content at the cost of computing resources. This paper presents a multi-level selective deduplication scheme which integrates inner-VM and cross-VM duplicate elimination under a stringent resource requirement. This scheme uses popular common data to facilitate fingerprint comparison while reducing the cost and it strikes a balance between local and global deduplication to increase parallelism and improve reliability. Experimental results show the proposed scheme can achieve high deduplication ratio while using a small amount of cloud resources.**

***Index Terms*—Cloud storage backup, Virtual machine snapshots, Distributed data deduplication**

I. Introduction

In a virtualized cloud environment such as ones provided by Amazon EC2[**?**] and Alibaba 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.

A snapshot preserves the data of a VM's file system at a specific point in time. VM snapshots can be backed up incrementally by comparing blocks from one version to another and only the blocks that have changed from the previous version of snapshot will be saved [**?**], [**?**].

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. Using a backup service with full deduplication support [**?**], [**?**] can identify content duplicates among snapshots to remove redundant storage content, but the weakness is that it either adds the extra cost significantly or competes computing resource with the existing cloud services. In addition, data dependence created by duplicate relationship among snapshots adds the complexity in fault tolerance management, especially when VMs can migrate around in the cloud.

Unlike the previous work dealing with general file-level backup and deduplication, our problem is focused on virtual disk image backup. Although we treat each virtual disk as a file logically, its size is very large. On the other hand, we need to support parallel backup of a large number of virtual disks in a cloud every day. One key requirement we face at Alibaba 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 of VM snapshot data and pursue a cost-effective deduplication solution. Another goal is to decentralize VM snapshot backup and localize deduplication as much as possible, which brings the benefits for increased parallelism and fault isolation.

By observations on the VM snapshot data from production cloud, we found snapshot data duplication can be easily classified into two categories: *inner-VM* and *cross-VM*. Inner-VM duplication exists between VM's snapshots, because the majority of data are unchanged during each backup period. On the other hand, Cross-VM duplication is mainly due to widely-used software and libraries such as Linux and MySQL. As the result, different VMs tend to backup large amount of highly similar data.

With these in mind, we have developed a distributed multi-level solution to conduct segment-level and block-level inner-VM deduplication to localize the deduplication effort when possible. It then makes cross-VM deduplication by excluding a small number of popular common data blocks from being backed up. Our study shows that common data blocks occupy significant amount of storage space while they only take a small amount of resources to deduplicate. Separating deduplication into multi levels effectively accomplish the major space saving goal compare the global complete deduplication scheme, at the same time it makes the backup of different VMs to be independent for better fault tolerance.

The rest of the paper is arranged as follows. Section **??** discusses on some background and related work. Section **??** discusses the requirements and design options. Section **??** presents our snapshot backup architecture with multi-level selective deduplication Section **??** presents our evaluation results on the effectiveness of multi-level deduplication for snapshot backup. Section **??** concludes this paper.

II. Background and Related Work

In a VM cloud, several operations are provided for creating and managing snapshots and snapshot trees, such as creating snapshots, reverting to any snapshot, and removing snapshots.

For VM snapshot backup, file-level semantics are normally not provided. Snapshot operations are taken place at the virtual device driver level, which means no fine-grained file system metadata can be used to determine the changed data. Only raw access information at disk block level are provided.

VM snapshots can be backed up incrementally by identifying file blocks that have changed from the previous version of the snapshot [**?**], [**?**], [**?**]. The main weakness is that it does not reveal content redundancy among data blocks from different snapshots or different VMs.

Data deduplication techniques can eliminate redundancy globally among different files from different users. Backup systems have been developed to use content hash (finger prints) to identify duplicate content [**?**], [**?**]. Today's commercial data backup systems (e.g. from EMC and NetApp) use a variable-size chunking algorithm to detect duplicates in file data [**?**], [**?**]. As data grows to be big, fingerprint lookup in such schemes becomes too slow to be scalable. Several techniques have been proposed to speedup searching of duplicate content. For example, Zhu et al. [**?**] tackle it by using an in-memory Bloom filter and prefetch groups of chunk IDs that are likely to be accessed together with high probability. It takes significant memory resource for filtering and caching. NG et al. [**?**] use a related filtering technique for integrating deduplication in Linux file system and the memory consumed is up to 2GB for a single machine. That is still too big in our context discussed below.

Duplicate search approximation [**?**], [**?**], [**?**] has been proposed to package similar content in one location, and duplicate lookup only searches for chunks within files which have a similar file-level or segment-level content fingerprints. That leads to a smaller amount of memory usage for storing meta data in signature lookup with a tradeoff of the reduced recall ratio.

III. Requirements and Design Options

We discuss the characteristics and main requirements for VM snapshot backup in a cloud environment. which are different from a traditional data backup.

1) *Cost consciousness.* There are tens of thousands of VMs running on a large-scale cluster. The amount of data is so huge such that backup cost must be controlled carefully. On the other hand, the computing resources allocated for snapshot service is very limited because VM performance has higher priority. At Aliyun, it is required that while CPU and disk usage should be small or modest during backup time, the memory footprint of snapshot service should not exceed 500MB at each node.
2) *Fast backup speed.* Often a cloud has a few hours of light workload each day (e.g. midnight), which creates an small window for automatic backup. Thus it is desirable that backup for all nodes can be conducted in parallel and any centralized or cross-machine communication for deduplication should not become a bottleneck.
3) *Fault tolerance.* The addition of data deduplication should not decrease the degree of fault tolerance. It's not desirable that small scale of data failure affects the backup of many VMs.

There are multiple choices in designing a backup architecture for VM snapshots. We discuss the following design options with a consideration on their strengths and weakness.

1) *An external and dedicated backup storage system.* In this architecture setting, a separate backup storage system using the standard backup and deduplication techniques can be deployed [**?**], [**?**], [**?**]. This system is attached to the cloud network and every machine can periodically transfer snapshot data to the attached backup system. A key weakness of this approach is communication bottleneck between a large number of machines in a cloud to this centralized service. Another weakness is that the cost of allocating separate resource for dedicated backup can be expensive. Since most of backup data is not used eventually, CPU and memory resource in such a backup cluster may not be fully utilized.
2) *A decentralized and co-hosted backup system with full deduplication.* In this option, the backup system runs on an existing set of cluster machines. The disadvantage is that even such a backup service may only use a fraction of the existing disk storage, fingerprint-based search does require a significant amount of memory for fingerprint lookup of searching duplicates. This competes memory resource with the existing VMs.
   Even approximation [**?**], [**?**] can be used to reduce memory requirement, one key weakness the hasn't been addressed by previous solutions is that global content sharing affects fault isolation. Because a content chunk is compared with a content signature collected from other users, this artificially creates data dependency among different VM users. In large scale cloud, node failures happen at daily basis, the loss of a shared block can affect many users whose snapshots share this data block. Without any control of such data sharing, we can only increase replication for global dataset to enhance the availability, but this incurs significantly more cost.

With these considerations in mind, we propose a decentralized backup architecture with multi-level and selective deduplication. This service is hosted in the existing set of machines and resource usage is controlled with a minimal impact to the existing applications. The deduplication process is first conducted among snapshots within each VM and then is conducted across VMs. Given the concern that searching duplicates across VMs is a global feature which can affect parallel performance and complicate failure management, we only eliminate the duplication of a small but popular data set while still maintaining a cost-effective deduplication ratio. For this purpose, we exploit the data characteristics of snapshots and collect most popular data. Data sharing across VMs is limited within this small data set such that adding replicas for it could enhance fault tolerance.