Asian Journal of Information Technology 13 (6): 320-330, 2014
ISSN: 1682-3915

A Survey on Deduplication in Cloud Storage

P. Neelaveni and M. Vijayalakshmi
Anna University, Chennai, India

Abstract: Cloud backup service is the core technology of cloud storage. Cloud backup service is becoming the substantial component of the cloud storage due to emerging trend of user data. Data deduplication techniques are ideal solutions for reducing both bandwidth and storage space requirements for cloud backup services in data centres. The main challenges to be considered in cloud backup services when deduplication is applied are bandwidth, high throughput, computational overhead, deduplication efficiency, read and write efficiency, backup window size and transmission cost. Similarly, in cloud backup services, during virtualization process, deduplication can be done to reduce the storage amount consumed by virtual machine images. When deduplication is performed in VM images, there are issues like high duplicate tracking, space overhead and high computation power that has to be considered. This study investigates the benefits and overhead of various deduplication technique adopted to the cloud backup services.

Key words: Cloud storage, deduplication, optimization, VM images, substantial component

INTRODUCTION

Cloud computing has been defined by NIST (Mell and Grance, 2009) as a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Cloud computing (Mao *et al*., 2013; Ng *et al*., 2011) is a new infrastructure deployment environment that delivers on the promise of supporting on-demand services like computation, software and data access in a flexible manner by scheduling bandwidth, storage and compute resources without required end-user knowledge of physical location and system configuration that delivers the service. The main characteristics of cloud computing are on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service.

Cloud storage is a massive and public accessible storage available for use in the internet. This is termed as Data storage as a Service (DaaS) with respect to services of cloud. Cloud storage refers to scalable and elastic storage capabilities that are delivered as a service using internet technologies with elastic provisioning. The Cloud Computing Model is elastic, meaning that resource allocation can get bigger or smaller depending on demand. Elasticity enables scalability which means that the subscribed services can easily scale up or down.

Cloud backup service is the core technology of cloud storage. Cloud backup stores data located at the client side into the cloud storage service provider through network so as to recover data in time. Cloud backup service has become cost-effective solution that is adopted by many organizations as their alternate data protection strategy.

Although, a cloud backup service has many benefits, several major challenges still exist. One of the main challenges is internet bandwidth since internet bandwidth is significantly lower than a local area network. Therefore, cloud backup and restoration is much slower and costs more than a traditional on-site backup.

The other serious challenge is the (Tan *et al*., 2010) large backup window that represents the time spent on sending specific dataset to backup destination, due to the low network bandwidth between user and service provider constraining the data transmission. For example, it would take >14 days to backup 1TB data to Amazon S3 with the assumed network bandwidth of 800 kb sec^{-1} (Gharaibeh *et al*., 2012). Another challenge stems from the vast storage space and very high data management cost required for the rapidly increasing amount of backed-up data stored at service providers' site.

The another critical performance metric (Tan *et al*., 2013) is Recovery Time Objective (RTO) which specifies the maximum amount of downtime a user is waiting to accept after data disasters. This is a bigger challenge for cloud backup services due to relatively low bandwidth of WAN. Therefore, it is important and critical to adopt network efficient approaches to the cloud backup environment to improve both the data backup and restore performances, if cloud backup as a service is to become practical and cost effective.

Corresponding Author: P. Neelaveni, Anna University, Chennaia, India

Much research (Liguori and Van Hensbergen, 2008) addresses this issue by trying to reduce the amount of data transmitted during the backup process. One important approach is employ data deduplication.

Data deduplication technology identifies duplicate data, eliminate redundancy and reduce the need to transfer or store the data in the overall capacity. Deduplication is an effective technique to optimize the utilization of storage space. Data deduplication can greatly reduce the amount of data, thereby reducing energy consumption and reduce network bandwidth in cloud data centres.

In the deduplication process, duplicate data is determined and only one copy of the data is stored, along with references to the unique copy of data thus redundant data is removed. The most common deduplication technique partitions data into chunks of non-overlapping data blocks (Meyer and Bolosky, 2011). It calculates a fingerprint for each chunk using a cryptographic hash function (e.g., SHA-1) and stores the fingerprint of each chunk in a hash table (chunk index). Each chunk stored on the storage system has a unique fingerprint in the chunk index. To determine whether a chunk is already stored on the system or not, the fingerprint of the incoming data item is first looked up in the chunk index and if there is a match, the system only stores a reference to the existing data. Otherwise the incoming chunk is considered unique and is stored on the system and its fingerprint inserted in the chunk index.

DEDUPLICATION STRATEGIES

Data deduplication strategies are basically classified into three types (Harnik *et al*., 2010) based on data units, location where deduplication can be performed and based on disk placement as illustrated in Fig. 1.

Data unit based methods: Data deduplication strategies are basically classified into two types based on data units as illustrated in Fig. 1. They are: file-level deduplication and block (chunk) level deduplication. Only one copy of the file is stored in file level deduplication. If the two files are producing same hash value then they are identical.

In block-level deduplication, each file is fragmented into blocks and one copy of each block is stored. Each block may be of fixed-sized (static) or variable-sized chunks. In fixed size chunks, size of each block is same. In case of variable, size of each chunk is different.

Location based methods: Depending on the location where redundant data is eliminated, deduplication can be categorized into two basic approaches (Harnik *et al*., 2010). In the target-based approach, deduplication is performed in the Destination Storage System. The client is not having knowledge about the deduplication strategies. This method have the advantage of increasing storage utilization but does not save bandwidth.

In source based deduplication, elimination of duplicate data is performed close to where data is created, rather than where data is stored as in the case of target deduplication. The source deduplication approach works on the client machine before it is transmitted specifically, the client software communicates with the backup server (by sending hash signatures) to check for the existence of files or blocks. Duplicates are replaced by pointers and the actual duplicate data is never sent over the network.

Further, Source Deduplication Method (Zhu *et al*., 2008) is classified based on different deduplication granularities as source local chunk level deduplication; source global chunk level deduplication (Harnik *et al*., 2010; Meyer and Bolosky, 2011). In the local chunk level, the redundant data chunks are removed before sending them to the remote backup destination within the

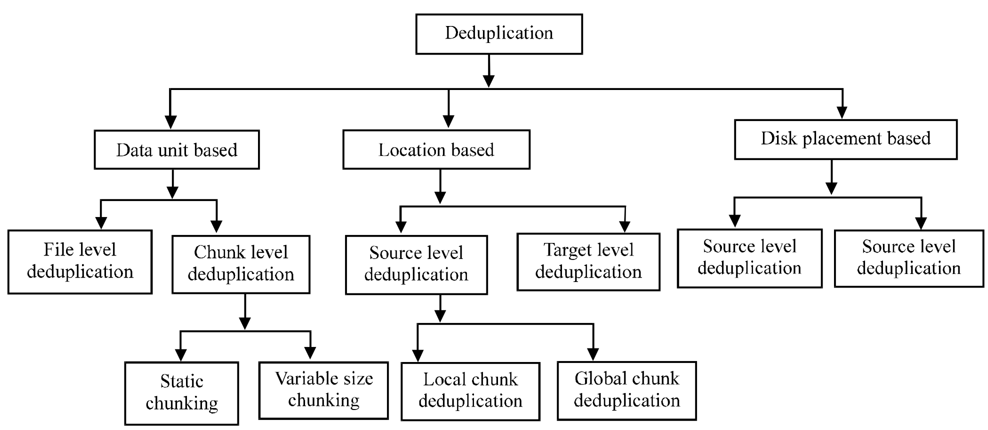

Fig. 1: Various deduplication strategies

same client. In the global chunk level, the duplicate chunks are removed globally across different clients. Due to the out of memory fingerprint accesses to massive backed-up data (Zhu *et al*., 2008; Bhagwat *et al*., 2009; Lillibridge *et al*., 2009) chunk level deduplication has an inherent latency and throughput problem that significantly decreases the backup performance. In this approach, the overhead is alleviated since searching the duplicate chunks is restricted to the same client. This reduced overhead, however, comes at the cost of severely limited compression ratio which increases the backup window due to the increased data transmission cost.

According to Tan *et al*. (2010), in source global chunk level deduplication, the overhead of massive disk accesses strangles the deduplication process which increases the backup window size.

The deduplication can also be performed at source file level (Fu *et al*., 2011). This method gives higher efficiency and low computational overhead than chunk level.

Disk placement based deduplication: Based on how data is placed in disks (Mao *et al*., 2013), data deduplication methods can be classified into backward-reference deduplication and forward-reference deduplication. In the backward-reference, the recent redundant data chunks are associated with pointers that point backward to the older identical data chunks. In the forward-reference, the recent redundant data chunks are maintained in their entirety and all the old identical data chunks are associated with pointers that point forward to the recent data chunks. The forward-reference approach provides the fastest read performance on the recent backups. However, it introduces much more fragmentation for the past data chunks and induce more index update and metadata update operations which degrades the system performance. Therefore, most existing data deduplication methods are based on the backward-reference deduplication, especially in the primary storage platforms such as VM servers.

EXISTING DEDUPLICATION TECHNIQUES

Recently, deduplication in distributed systems has motivated lots of researchers due to the demand of scalability and efficiency. Here, researchers reviewed the recently done works in the literature for source level deduplication and the different approaches used for it.

Application based source deduplication scheme: To achieve high data transfer rates, cloud clients require significant processing to deduplicate data in cloud backup services, resulting in index size and performance challenges.

Application Aware Deduplication (AA-Dedupe) (Fu *et al*., 2011) technique reduces the computational overhead by implementing an intelligent data chunking scheme and the adaptive use of hash functions based on application awareness and to alleviate the on-disk index lookup bottleneck by separating the entire index into small independent and application specific indices in an application-aware index structure. In the architecture of AA-Dedupe tiny files are first filtered out by file size and backup data streams are broken into chunks by an intelligent chunker using an application aware chunking strategy. Data chunks from the same type of files are then deduplicated in the application-aware deduplicator by looking up their hash values in an application-aware index that is stored in the local disk. If a match is found, the metadata for the file containing that chunk is updated to point to the location of the existing chunk. If there is no match, the new chunk is stored based on the container management in the cloud, the metadata for the associated file is updated to point to it and a new entry is added into the application-aware index to index the new chunk.

They experimentally showed that deduplication efficiency is measured by a metric called "bytes saved per second", deduplication efficiency in AA-Dedupe is 2 times that of backup PC, 5 times that of SAM and 7 times that of Avamarson average. Backup window size of AA-Dedupe is redudced 10-32% than other methods. Cloud cost of AA-Dedupe is lower than those of other schemes by about 12-29% for backup datasets. AA-Dedupe incurs only one fourth of the power consumption of Avamar and one third of that of SAM by adaptively using weaker hash functions in deduplication.

A semantic attributes based source de-duplication: The Deduplication System produces increased latency and reduced throughput, high data transmission costs which results in a large backup window. Researchers developed novel technique called SAM, a Semantic-Aware Multi tiered source de-duplication framework that first combines the global file-level deduplication and local chunk level deduplication. They also considered file level semantic attributes like file locality, file time stamps, file size and file type which are used to find redundant data. Unmodified data removal, removing the unmodified files and data chunks that are kept intact after backups from data transmission for cloud restore operations. Among these features, hybrid deduplication and semantic-aware elimination work in synchronized manner to remove the redundant data from data transmission to reduce backup times and storage costs while unmodified data removal aims to reduce the restore times in some restore scenarios.

According to their system architecture there are 3 subsystem as follows: file agent which is used to collect, send, restore data sets, master server globally maintains and schedules all backup files and it also uses catalogue database to keep track of files stored in the storage server It is the repository of backup data. It consists of chunk store responsible for storing backup data chunks of different clients. Whenever, files to be backup, SAM filters out completely unchanged files then it performs global file level deduplication by removing duplicate files across different client in the master server based on the file locality and file size. Finally, it performs local chunk deduplication across similar files with in the same client.

The performance evaluation proved that they achieved high ratio of deduplication efficiency, reduced throughput and shortens the backup time by an average of 38.7% during backup operations. During some restore operations when the local unmodified data is available, SAFE significantly reduces the restore time by a reduction ratio of up to 9.7:1 after removing up to 91.8% of this unmodified data. The main drawback of this hybrid approach is client overhead since SAM uses more CPU power and storage space at the client and restore performance not able to calculate since they are not considered real time data sets.

GPU based source data deduplication: Deduplication needs high computing potential since it has to compute the fingerprint of large data using a Hash algorithm. For example Intel core i7 has 4 processing cores whereas Nvidia GeForce GTX480 has 480 cores. Thus, GPU has much more computing power than CPU for massively computing tasks. Hence, (Suttisirikul and Uthayopas, 2012) GPU can be utilized to perform deduplication on large volume of cloud backup data. The GPU can be placed at the cloud provider or cloud consumer side. In this system, the source level (Cloud provider) deduplication process is performed.

The executable units called threads in GPU are grouped into blocks. Each block is then executed in parallel on each core. The deduplication process is parallelized by partioning a file into fixed size chunks. Then, multiple chunks are passed to threads in GPU system. Each thread calculates fingerprint of each chunk using SHA256 algorithm since SHA256 Method produces very lower hash collision than SHA Methods.

They also constructed mathematical model to describe the speed up obtained using GPU. The experimental results shows that GPU speed up the fingerprint generation process up to 53 times better than CPU for large data file. They achieved higher throughput but fingerprint calculation is slower for each chunk.

Hadoop based deduplication: Existing approaches which require locality do not have any exact deduplication mechanism and those approaches using fingerprints as well as an index cannot solve the problems of RAM usage very well.

The deduplication system developed based on Hadoop (Sun *et al*., 2011) composed of two main elements, a front-end deduplication application and Hadoop Distributed File System (HDFS) used with a Hadoop database. Hadoop database is used to build up a fast indexing system. Two types of files, i.e, source files and link files are saved in HDFS. In the DeDu System, each source file is read by its primary value and every link file records one hash value for each source file and the logical path to the source file. HBase records all the hash values for each file, the number of links and the logical path to the source file.

During deduplication process, users select the files or folders which are going to be uploaded and saved. The hash value of file has been computed using MD5 and SHA-1 and its value is passed to HBase. HBase table stores all hash values. It compares new hash value with the existing values. If it does not exist, a new hash value will be recorded in the table and then HDFS request the clients to upload the files and record the logical path. If it does exist, HDFS will check the number of links and if the number is not zero, the counter will be incremented by one. In this case, HDFS will tell the clients that the file has been saved. if the number is zero, HDFS will ask the client to upload the file.

The experimental results shows that the deduplication efficiency is achieved three times better than existing system. They also proved that their system achieves static, dynamic load balancing. It also improves reading and writing efficiency.

Hash cluster based deduplication: The data deduplication system identifies the redundant data by computing hash values of data chunks and stores in a centralized server. Due to centralized server, the throughput is reduced, concurrency problem is raised and slow seek time of hard disk degrades the response of hash look up. To overcome the above issues, scalable hybrid hash cluster (Xu *et al*., 2011) is developed. In a cluster of machines, each hybrid node composed of RAM and Solid State Drives (SSD). The fast random access inherent in SSD gives fast response to hash lookup services.

The client in the cloud sends a fingerprint to hash node. If the hash value already exists in main memory then it is informed to the client. Otherwise node tries to locate this fingerprint in the hash table on SSD. If this fingerprint exists on SSD then node loads it into RAM and replies to the client. Else new entry is created in the hash table on SSD and client has to retransmit to store the data in cloud.

Researchers experimental setup shows that finger prints of real work loads are issued to hash cluster for different numbers of cluster nodes. The network bandwidth, inline deduplication throughput and load balancing is achieved.

Casualty based deduplication: In cloud backup services, due to the relatively low bandwidth of network, the backup time and restore time is increased. Existing solutions that employ the deduplication technology for cloud backup services only focus on removing redundant data from transmission during backup operations to reduce the backup time. The restore time affects the overall quality of service of the cloud backup services. Researchers proposed a (Tan *et al*., 2011) Causality-based deduplication performance booster for both cloud backup and restore operations, called CABdedupe. It captures the causal relationship among chronological versions of datasets to remove the unmodified data from transmission during not only backup operations but also restore operations. CABdedupe is a middleware that is orthogonal to and can be integrated into any existing backup system. Their extensive experiments where CABdedupe is intergrated into two existing backup systems and real world datasets shows that both the backup time and restore time are significantly reduced, with a reduction ratio of upto 69:1.

SSD based deduplication: The data deduplication technology has been demonstrated to be very effective in shortening the backup window and saving the network bandwidth and storage space in cloud backup, archiving and primary storage systems such as VM platforms. However, the delay and power consumption of the restore operations from a deduplicated storage can be significantly higher than those without deduplication. The main reason lies in the fact that a file or block is split into multiple small data chunks that are often located in non-sequential locations on HDDs after deduplication which can cause a subsequent read operation to invoke many HDD I/O requests involving multiple disk seeks.

To address this problem, researchers proposed SAR (Mao *et al*., 2013), an SSD Assisted Restore scheme, that effectively exploits the high random-read performance and low power consumption properties of SSDs. This system implies unique data sharing characteristic of Deduplication-Based Storage System by storing in SSDs having the unique data chunks with high reference count, small size and non-sequential characteristics. In this way, many critical random-read requests to HDDs are replaced by read requests to SSDs thus significantly improving the system performance and energy efficiency. The extensive trace-driven and VM restore evaluations on the prototype implementation of SAR show that SAR outperforms the traditional deduplication-based schemes significantly, in terms of both restore performance and energy efficiency.

Deduplication in disk: Cloud-based backup and archival services (Guo and Efstathopoulos, 2011) use large tape libraries as a cost-effective element in their online storage hierarchy today. These services influences deduplication to reduce the disk storage capacity required by their customer data sets but they usually re-duplicate the data when moving it from disk to tape. Deduplication does not add significant I/O overhead when performed on disk storage pools. However, when deduplicated data is naively placed on tape storage, the high degree of data fragmentation caused by deduplication combined with the high seek and mount time leads to high retrieval time. This negatively impacts the Recovery Time Objectives (RTO) that the service provider has to meet as a part of the Service Level Agreement (SLA).

This research proposes (Gharaibeh *et al*., 2012) CloudDT, an extension to cloud backup and archival services to efficiently support deduplication on tape pools. This system introduces a class of solutions based on graph-modeling of similarity between data items that enables efficient placement on tapes and presents the design and initial evaluation of algorithms that alleviate tape mount time overhead and reduce on-tape data fragmentation.

Using 4.5 TB of real-world workloads, initial evaluation shows that the algorithms retain at least 95% of the deduplication storage efficiency and offer upto 40% faster restore performance compared to the case of restoring non-deduplicated data. Therefore, the proposed techniques allow the backup service provider to increase tape resource utilization using deduplication while also improving the restore time performance for the end user.

Signature based deduplication: The existing deduplication systems used in the cloud storage server spends too much time on examining duplicate blocks. Researchers proposed a scheme that utilizes the capacity of cloud storage server more properly and also improve the speed of data deduplication. Furthermore, a signature is computed for every uploaded file to ensure the integrity of file. Lin and Chien (2012) developed a scheme which is derived from Zhang's digital signature method with fault tolerance which is based on RSA cryptosystem.

This system consists of two phases: file translation phase. When the user uploads a file to cloud storage, it is translated into to block of matrix and computes feature values for each row and column. The signature of file is calculated by using user's private key. In data deduplication phase when the same user uploads another file, server implements the same procedure as in file translation phase and compare the signature of a file.

The fixed-sized block-level deduplication is used in the system. The computation cost $o(n^2)$ indicates the complexity of translating a file into a matrix. They calculated frequency of block verification. $o(n)$ represents the complexity of verifying the hash value between uploaded file blocks and cloud storage blocks to check whether the same blocks exist in the cloud storage or not. Their experimental setup shows that less time on block verification and computation cost is also less than Venti's Archival Storage System.

A lightweight deduplication mechanism: The traditional data deduplication techniques consumes extra computing resource for compare and delete data. The proposed (Wang *et al*., 2013) Lightweight Deduplication Mechanism (LDMCS) reduces the number of backups which provides more storage capacity and also reduces the computing resource to compare data processing.

In the proposed system architecture, several storage nodes are grouped to form cluster. Each cluster consists of two types of storage node called name node and a data node. The name node is responsible for performing deduplication. Transfer Agent System (TAS) uploads the data when the requests of user are received. The data is partitioned into n chunks and translated into unique identifier by SHA-1 and is delivered to cluster by TAS for comparison of data. Then, LDMCS check the status of data existence by using bloom filter. If this data is not exist, comparison of data is not performed and searches link structure to find the same data. Using bloom filter to check the status of data existence and compare the same data position, reduces the computing resource of data deduplication compare process.

Deduplication in VM images: Virtualization is becoming widely deployed in servers to efficiently provide many logically separate execution environments while reducing the need for physical servers in cloud backup services. While this approach saves physical CPU resources, it still consumes large amounts of storage because each Virtual Machine (VM) instance requires its own multi-gigabyte disk image. Deduplication can be employed to reduce the storage amount occupied by VM images. In this study, researchers are reviewing benefits and overhead of deduplication techniques adopted in VM images.

Live deduplication: Standard applications like web servers and file storage can be hosted in open-source cloud should provide users with Virtual Machines (VMs). Cloud must provide variety of versions of virtual machines for different configurations (e.g., 32 bit/64 bit Hardware, File Systems, Operating Systems, etc.) to make deployment flexible. A serious challenge is to scale up the storage of a large number of VM images, each of which is a file that could be of gigabytes. When the storage capacity of VM images are increased during hosting, it requires higher operating costs. To improve the storage efficiency of VM images, deduplication techniques can be used.

Recent studies (Jin and Miller, 2009; Liguori and Van Hensbergen, 2008) showed that the VM images of different versions of the same Linux distribution generally have a high proportion of identical data blocks (e.g., about 30% of overlap in adjacent Fedora distributions). Therefore, deduplication can increase enhance the storage utilization of VM images. But when deduplication for VM image storage in a cloud is performed, several deployment issues are also addressed like performance of VM operations, i.e., deduplication degrades the performance of existing VM operations such as VM start-up and support of general file system operations are not allowed.

To handle VM images more effectively, a deduplication solution should allow general file system operations such as data modification and deletion. However, current deduplication techniques are mainly designed for backup systems which require data be immutable and impose a write once policy (Quinlan and Dorward, 2002) to prevent data from being modified or deleted.

To overcome this problem, researchers developed a system called a (Ng *et al*., 2011) LiveDFS (Deduplicate File System) for VM image storage in an open-source cloud. This system provides the file system design layout in Linux and allows general I/O operations such as read, write, modify and delete while enabling inline deduplication. To support inline deduplication, LiveDFS exploits spatial locality to reduce the disk access overhead for looking up fingerprints that are stored on disk. LiveDFS is implemented as a Linux kernel driver module that can be deployed without the need of modifying the kernel source. LiveDFS is integrated into a cloud platform based on OpenStack and evaluates the deployment. The experimental setup shows that LiveDFS saves at least 40% of storage space for different distributions of VM images while its performance overhead in read/write throughput is minimal overall.

Scalable deduplication in VM images: The growing number of VMs being deployed leads to increased burden on the underlying storage systems. To ensure that advanced VM features like migration and high availability could work fluently, VM images need to be accessible from more than one host machine. This leads to the common practice of storing VM images on shared network storage such as Network-Attached Storage (NAS) and Storage Area Network (SAN). The problem of such an approach is that network storage systems usually cost more and they have high demand on network IO performance. Moreover, the critical need to store thousands of VM images would be an extremely challenging problem for network storage systems because of the significant scale of storage consumption.

Studies have shown that the storage consumption issue brought by a large number of VM images could be addressed by deduplication techniques (Lin and Chien, 2012; Zhang *et al*., 2013). SAN systems operated in a decentralized fashion such that deduplication is done at the host machines running VM and unique data blocks are then stored on the SAN cluster. However, SANs are very expensive and thus difficult to satisfy the ever-growing need of VM image storage in the future.

Researchers proposed system called Liquid which is a distributed file system particularly designed to simultaneously address the above problems faced in large scale VM deployment. Its client side breaks VM images into small data blocks, references them by their fingerprints and uses deduplication techniques to avoid storing redundant data blocks. The deduplicated data blocks are then saved to a group of data servers and the set of fingerprints is saved to a meta server. When a VM image is to be accessed, a Liquid client downloads its set of fingerprints from the meta server, fetches data blocks from data servers and peer clients in a P2P fashion and exports an integrated VM image layout to hypervisors. Liquid's P2P data block transfer protocol reduces requests directly issued to data servers and guarantees high scalability of the whole system. The experimental environment proved that data block size in the range 256 kb 1Mb achieves moderate IO performance during deduplication.

Deduplication during VM migration: Live migration of a Virtual Machine (VM) refers to the transfer of a running VM over the network from one physical machine to another. Some of the key metrics to measure the performance of VM migration are as follows: total migration time is the time from the start of migration at the source to its completion at the target. Downtime is the duration for which a VM's execution is suspended during migration. Network traffic overhead is the additional network traffic due to VM migration. Application degradation is the adverse performance impact of VM migration on applications running anywhere in the cluster.

Gang migration (Deshpande *et al*., 2013), i.e., the simultaneous live migration of multiple VMs that run on multiple physical machines in a cluster. Gang migration affects the performance at the network edges where the migration traffic competes with the bandwidth requirements of applications within the VMs. Since, gang migration can transfer hundreds of gigabytes of data over the network, it can overload the core links and switches of the datacenter network. Reducing the network traffic overhead can also indirectly reduce the total time for migrating multiple VMs and the application degradation depending upon how the traffic reduction is achieved. Researchers specifically focuses on reducing the network traffic overhead due to gang migration.

VMs within a cluster often have similar memory content, given that they may execute the same operating system, libraries and applications. Hence, a significant number of their memory pages may be identical (Quinlan and Dorward, 2002). The network overhead of gang migration can be reduced using deduplication. Deshpande *et al*. (2013) presented an approach called Gang Migration using Global (cluster-wide) Deduplication (GMGD).

During normal execution, a duplicate tracking mechanism keeps track of identical pages across different VMs in the cluster. During gang migration, a distributed coordination mechanism suppresses the retransmission of identical pages over the core links. Specifically, only one copy of each identical page is transferred to a target rack (i.e., the rack where a recipient physical machine for a VM resides). Thereupon, the machines within each target rack coordinate the exchange of necessary pages. In contrast to GMGD, Gang Migration using Local Deduplication (GMLD) suppresses the retransmission of identical pages from among VMs within a single host. A prototype is implemented using GMGD on the KVM platform. then GMGD is evaluated on a 30 node cluster testbed having three switches, 10GigE core links and 1Gbp edge links. GMGD is compared against two techniques the QEMU/KVM's default live migration technique.

Imaged based deduplication: In cloud computing, IAAS (Infrastructure As A Service) platform provides services to the user through the virtual machine. Some recent studies (Jin and Miller, 2009; Liguori and Van Hensbergen, 2008) have shown that >80% of the data blocks in virtual machine storage are duplicated. Therefore, the application of deduplication technology is helpful for IAAS cloud computing platforms to manage data more efficiently. The management of virtual machine images not only consumes a huge amount of storage space but also gives large pressure on network transmission. By using deduplication technology in openstack (Openstack is an open source project developed by NASA and Rackspace which can be used to build private and public cloud) an image management system IM-dedup (Zhang *et al*., 2013) is designed. This system uses Static Chunking (SC) to divide image file into blocks of data, avoid duplication data blocks transmission on network by using fingerprint pre transmission technology and reduce storage space by deploying kernel mode file system with deduplication in the image storage server.

The components of openstack are used to manage virtual machine images and network connections. In order to implement the deduplication and reduce the local storage space of the images, the liveDFS (Xu *et al*., 2011) Data Deduplication Method is used into the image data stored in the systems.

The openstack is deployed on the Ubuntu Linux kernel which uses EXT3 file system to manage files and devices. The block file consist of 128 byte inode which gives information about file properties and description information of a file. When a new file comes, system will first chunk it into blocks. The calculated fingerprint is transferred into fingerprint filter. The fingerprint filter is the index structure in memory. It aims to accelerate the searching speed of the fingerprint on disk.

If the fingerprint is not in the filter, it indicates that the block is not duplicated. Then, the system will write the block into disk and then modify the inode. If the fingerprint is in the filter, it will start another process into disk to search the fingerprint. If fingerprint is not in the disk, it indicates that the block is a new one. It should be stored into disk and then some modification should be done to the inode. If fingerprint is in the disk which means the block is a duplicated one, the block data should be dropped. The experimental results showed that the system not only reduced 80% usage of the virtual machine image storage but also saved at least 30% of transmission time.

Multilevel selective deduplication in VM: In a virtualized cloud environment such as one provided by Amazon EC and Alibaba Aliyun (Liguori and Van Hensbergen, 2008), 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. 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 (Vrable *et al*., 2009). Frequent backup of VM snapshots increases the reliability of VM's hosted in a cloud.

The cost of frequent backup of VM snapshots is high because of the huge storage demand. Using a backup service with full deduplication support (Zhu *et al*., 2008; Harnik *et al*., 2010) 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.

Thus, researchers developed a system 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.

Based on the VM 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.

The distributed multilevel deduplication is system is developed by researchers to conduct segment-level and block-level inner VM deduplication within VM images. Then, cross-VM deduplication is performed by excluding a small number of popular common data blocks from being backed up. 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.

Their experimental setup showed that level-1 elimination can reduce the data size to about 23% of original data, namely it delivers close to 77% reduction. Level-2 elimination is applied to data that could pass level-1, it reduces the size further to about 18.5% of original size, namely it delivers additional 4.5% reduction. Level-3 elimination together with level 1 and 2 reduces the size further to 8% of original size, namely it delivers additional 10.5% reduction.

Table 1: Comparison of performance measures of various deduplication techniques

Techniques	SHHC	Semantic aware	GPU based	CAB dupe	AA-dupe	De-du
Bandwidth	-	-	-	-	-	-
Computational overhead	-	34.6% GCDS 26.2% LCDS	Acceleration ratio-53 times > CPU	- -	- -	- -
Throughput	1 order of magnitude	-	-	-	-	-
Data transfer rate	-	-	-	-	-	-
Reading efficiency	-	I/O reduction 40% through small file filtering	-	-	-	295 items amounting to 3.3 GB at a cost of 356 sec
Writing efficiency	-		-	-	-	295 items uploaded amounting to 3.3 GB at a cost of 401 sec
Static load balancing	Workload is balanced in each node 25%	-	-	-	-	4 data nodes stores average of 6.8 GB
Dynamic load balancing	-	-	-	-	-	8 GB data is stored on each node after new node is inserted
Backup window size	-	1.49% < deduplication time than GCDS	-	30% reduced	Reduced 10-32%	-
Deduplication efficiency	-	1.35%	-	18% > existing system	5 times > SAM	38% > perfect deduplication

GCDS: Global Chunk-level Deduplication Scheme; LCDS: Local Chunk-level Deduplication Scheme

Data deduplication in live VM migration: One of the key characteristics of virtualization, live Virtual Machine (VM) migration provides great benefits for load balancing, power management, fault tolerance and other system maintenance issues in modern clusters and data centers. Although, pre-copy is a widespread used Migration algorithm, it does transfer a lot of duplicated memory image data from source to destination which results in longer migration time and downtime. This system proposes a novel VM migration approach, named Migration with Data Deduplication (MDD) which introduces data deduplication into migration. MDD utilizes the self-similarity of run-time memory image, uses hash based fingerprints to find identical and similar memory pages and employs Run Length Encode (RLE) to eliminate redundant memory data during migration. Experiment demonstrates that compared with Xen's Default Pre-Copy Migration algorithm, MDD can reduce 56.60% of total data transferred during migration, 34.93% of total migration time and 26.16% of downtime on average. Table 1 compares the performance measures of various deduplication techniques.

RESEARCH CHALLENGES

In source level deduplication, elimination of duplicate data is performed close to where data is created. The source deduplication approach works on the client machine before it is transmitted. The client system communicates with the backup server (by sending hash signatures) to check for existence of files or blocks. Duplicates are replaced by pointers and the actual duplicate data is never sent over the network. This strategy saves bandwidth in turn decreases transmission cost, computational overhead and backup window size and also increases deduplication efficiency. Due to these benefits source level deduplication can be used in the cloud backup services.

The serious research challenges can be considered while doing source level deduplication are reading and writing efficiency. It can be increased based on the number of nodes. Time to calculate hash values can be reduced invariant of the file size which increases deduplication efficiency. Deduplication throughput can be increased and backup window size can be reduced.

When deduplication is performed in VM images high performance can be achieved. During live migration of VM the basic issues like high duplicate tracking, space overhead and computation power can be considered.

CONCLUSION

A number of researches have investigated the performance and efficiency of the various deduplication techniques. The experience suggests that research challenges like a deduplication system for cloud backup services should have the following features: the emerging hardware components such as flash-based Solid State Drives (SSD), General-Purpose computing on Graphics Processing Units (GPGPU) and multi-core CPU can be considered to perform deduplication process efficiently. Cloud backup service architecture can be organized regarding the data similarity for the deduplication efficiency. Inline deduplication and post-processing deduplication can bring significant improvement in terms of deduplication efficiency. As a future research, researchers plan to develop a framework which can optimize the performance measures like network bandwidth, high throughput, computational overhead, deduplication efficiency, read and write efficiency, backup window size and transmission cost for cloud backup services during deduplication process. As a direction of future research, researchers also plan to develop a system to achieve high performance when deduplication is performed in VM images and during live migration of VM by considering the basic issues like high duplicate tracking, space overhead and computation power.

REFERENCES

Bhagwat, D., K. Eshghi, D. Long and M. Lillibriage, 2009. Extreme binning: Scalable, parallel deduplication for chunk-based file backup. Proceedings of the IEEE International Symposium Modeling, Analysis and Simulation of Computer and Telecommunication Systems, September 21-23, 2009, London, UK., pp: 1-9.

Deshpande, U., B. Schlinker, E. Adler and K. Gopalan, 2013. Gang migration of virtual machines using cluster-wide deduplication. Proceedings of the 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, May 13-16, 2013, Delft, Netherlands, pp: 394-400.

Fu, Y., H. Jiang, N. Xiao, L. Tian and F. Liu, 2011. AA-Dedupe: An application-aware source deduplication approach for cloud backup services in the personal computing environment. Proceedings of the IEEE International Conference on Cluster Computing, September 26-30, 2011, Austin, TX., pp: 112-120.

Gharaibeh, A., C. Constantinescu, M. Lu, A. Sharma and R.R. Routray *et al*., 2012. CloudDT: Efficient tape resource management using deduplication in cloud backup and archival services. Proceedings of the 8th International Conference and Workshop on Systems Virtualization Management, October 22-26, 2012, Las Vegas, USA., pp: 169-173.

Guo, F. and P. Efstathopoulos, 2011. Building a high-performance deduplication system. Proceedings of the USENIX Annual Technical Conference, June 15-17, 2011, Portland, OR., USA., pp: 25.

Harnik, D., B. Pinkas and A. Shulman-Peleg, 2010. Side channels in cloud services: Deduplication in cloud storage. IEEE Security Privacy, 8: 40-47.

Jin, K. and E.L. Miller, 2009. The effectiveness of deduplication on virtual machine disk images. Proceedings of SYSTOR: The Israeli Experimental Systems Conference, May 2009, Haifa, Israel, pp: 1-12.

Liguori, A. and E. Van Hensbergen, 2008. Experiences with content addressable storage and virtual disks. Proceedings of the 1st Conference on I/O Virtualization, December 10-11, 2008, USENIX Association Berkeley, CA., USA., pp: 5-5.

Lillibridge, M., K. Eshghi, D. Bhagwat, V. Deolalikar, G. Trezise and P. Camble, 2009. Sparse indexing: Large scale, inline deduplication using sampling and locality. Proceedings of the 7th USENIX Conference on File and Storage Technologies, February 24-27, 2009, San Francisco, CA., USA., pp: 111-123.

Lin, I.C. and P.C. Chien, 2012. Data deduplication scheme for cloud storage. Int. J. Comput. Consumer Control, 1: 26-31.

Mao, B., H. Jiang, S. Wu, Y. Fu and L. Tian, 2013. AR: SSD assisted restore optimization for deduplication-based storage systems. Proceedings of the IEEE 7th International Conference on Networking, Architecture and Storage, June 28-30, 2012, Xiamen, Fujian, pp: 328-337.

Mell, P. and T. Grance, 2009. The NIST definition of cloud computing. Version 15, National Institute of Standards and Technology, October 7, 2009. http://www.nist.gov/itl/cloud/upload/cloud-def-v15.pdf.

Meyer, D.T. and W.J. Bolosky, 2011. A study of practical deduplication. Proceedings of the 9th Conference on File and Storage Technologies, February 15-17, 2011, San Jose, CA, USA., pp: 1-1.

Ng, C.H., M. Ma, T.Y. Wong, P.P.C. Lee and J.C.S. Lui, 2011. Live deduplication storage of virtual machine images in an open-source cloud. Proceedings of the ACM/IFIP/USENIX 12th International Middleware Conference, December 12-16, 2011, Lisboa, Portugal, pp: 81-100.

Quinlan, S. and S. Dorward, 2002. Venti: A new approach to archival storage. Proceedings of the 1st USENIX Conference on File and Storage Technologies, January 28-30, 2002, Monterey, CA., USA., pp: 89-102.

Sun, Z., J. Shen and J. Yong, 2011. DeDu: Building a deduplication storage system over cloud computing. Proceedings of the 15th International Conference on Computer Supported Cooperative Work in Design, June 8-10, 2011, Lausanne, pp: 348-355.

Suttisirikul, K. and P. Uthayopas, 2012. Accelerating the cloud backup using GPU based data deduplication. Proceedings of the 18th International Conference on Parallel and Distributed Systems, December 17-19, 2012, Singapore, pp: 766-769.

Tan, Y., H. Jiang, D. Feng, L. Tian and Z. Yan, 2011. CABdedupe: A causality-based deduplication performance booster for cloud backup services. Proceedings of the International Symposium on Parallel and Distributed Processing, May 16-20, 2011, Anchorage, AK., USA., pp: 1266-1277.

Tan, Y., H. Jiang, D. Feng, L. Tian, Z. Yan and G. Zhou, 2010. SAM: A semantic-aware multi-tiered source de-duplication framework for cloud backup. Proceedings of the 39th International Conference on Parallel Processing, September 13-16, 2010, San Diego, CA., USA., pp: 614-623.

Tan, Y., H. Jiang, E.H.M. Sha, Z. Yan and D. Feng, 2013. SAFE: A source deduplication framework for efficient cloud backup services. J. Signal Process. Syst., 72: 209-228.

Vrable, M., S. Savage and G.M. Voelker, 2009. Cumulus: Filesystem backup to the cloud. ACM Trans. Storage, Vol. 5. 10.1145/1629080.1629084.

Wang, S.C., K.Q. Yan, S.S. Wang and B.W. Chen, 2013. LDMCS: A lightweight deduplication mechanism under cloud storage. Proceedings of the International Conference on Business and Information, July 7-9, 2013, Bali, Indonesia, pp: E32-E40.

Xu, L., J. Hu, S. Mkandawire and H. Jiang, 2011. SHHC: A scalable hybrid hash cluster for cloud backup services in data centers. Proceedings of the 31st International Conference on Distributed Computing Systems Workshops, June 20-24, 2011, Minneapolis, MN., USA., pp: 61-65.

Zhang, J., S. Han, J. Wan, B. Zhu, L. Zhou, Y. Ren and W. Zhang, 2013. IM-Dedup: An image management system based on deduplication applied in DWSNs. Int. J. Distrib. Sensor Networks, 10.1155/2013/625070.

Zhu, B., K. Li and H. Patterson, 2008. Avoiding the disk bottleneck in the data domain deduplication file system. Proceedings of the 6th USENIX Conference on File and Storage Technologies, February 26-29, 2008, San Jose, CA., USA., pp: 269-282.