Smart Hashing based Queries in the Cloud

Yu Hua
Wuhan National Lab for Optoelectronics
School of Computer Science and Technology
Huazhong University of Science and Technology
Wuhan, China
E-mail: csyhua@hust.edu.cn

***Abstract*—High system dependability and high disaster recovery are important requirements for cloud service providers. Large-scale cloud systems providers often employ remote backups to help satisfy these requirements. However, due to long-distance and limited network bandwidth, backups may take very long time to finish and may become very costly. This problem increases as the size of stored image files in the cloud increases rapidly. To achieve high backup performance, this paper proposes a smart hashing based scheme, called Smarter, to replicate files between long-distance data centers. Efficient queries are important to improve entire system performance. With the aid of efficient queries, the idea behind Smarter is to select the most representative (thus most valuable) files, as base files, for real-time backups, while leaving the remaining non-duplicate ones for offline (or off peak-hour) backups. Smarter leverages a suitable "division of labor" between the metadata and the contents of multimedia images to support cost-effective redundancy detection. By using the smart hashing, Smarter identifies representative files and removes identical files at the source system, and then computes the top-k similar files between the source and destination systems in order to obtain significant bandwidth savings and shortens the backup time. It thus efficiently manages the massive images that account for a large fraction of the data to be transferred during backups. We implemented Smarter and evaluated it in a real cloud backup system. Results demonstrate the superior efficiency.**

I. Introduction

Efficient queries are important to cloud computing that provides plenty of benefits for both businesses and users [1]. As more and more applications are being deployed in the cloud, the dependability and reliability of cloud services are becoming ever more important [2]. For example, Dropbox, one of the largest online storage service providers, undergone a major outage from 3:30 PM on January 10, 2013 to 7:09 AM on January 11, 2013. Similarly, on March 18, 2013, Google users faced slow load times or full-on timeouts while trying to access their Google Drive documents and files. That lasted for about three hours. Again, on January 24, 2014, Google users who use logged-in services like Gmail, Google+, Calendar and Documents found they were unable to access those services for approximately 25 minutes. For about 10 percent of users, the problem persisted for as much as 30 minutes longer. These kinds of service disruptions are not only bad for the end users, but also for the service providers as they cause revenue and user loss.

In order to provide dependability and reliability, many cloud service providers leverage distributed storage strategy instead of centralized strategy to alleviate or avoid the loss of important data and offer fast and efficient data recovery. But in the meantime, the remote location of data centers incurs long backup latency, which is further exacerbated by the limited bandwidth [3], making remote backups for massive data very costly in terms of both network bandwidth and backup time.

The promise of conventional deduplication is its ability to identify redundant data. In this context, two data items are considered redundant only if their bit-streams are completely identical [4]. However, the system-level deduplication solutions are rendered ineffective and inefficient by powerful image processing software whose simple transformation operations will produce data chunks that are often completely different from the original ones [5] (i.e., different fingerprints) but similar or hardly distinguishable from a user's viewpoint.

Existing deduplication [6]–[8] and delta compression [9], [10] leverage exact-matching methodology, which in practice incurs significant performance decrements and consumes substantial system resources. Since the process of identifying near-duplicate images is dependent on image content, access patterns and metadata characteristics, we argue that a suitable "division of labor" between the processing of the data contents and the processing of the metadata information is critical in the design of cost-effective near-duplicate image identification. We propose a methodology of *representative inline backup* with real-time performance, called Smarter. The *representative images* are interpreted as the source files that can be transformed into other near-duplicate images by using geometric "photoshopped" transformations, which may be stored during the backups. By maintaining a representative "master" image with maximum features, the base image is easy for us to recover transformed ones (rooted at the master image) by leveraging geometric parameters, which is a popular and well-recognized approach [5], [11].

It is worth noting that Smarter does not remove or lose any images, hence it guarantees data integrity. Smarter allows representatives to be transmitted with high priority to improve bandwidth efficiency. The delayed near-duplicate images can be transmitted when idle bandwidth is available. Smarter offers a comprehensive affinity-aware system. The affinity is interpreted as semantic correlation derived from multi-dimensional attributes (e.g., metadata and access patterns), as well as representative features of images. The affinity-aware system consists of semantic hashing, data structures and compression schemes, to significantly improve network backup performance.

Smarter functions at both the source end and destination end for a remote backup. At the source end, Smarter selects representative images and delays duplicate and near-duplicate ones. Remote transmission leverages a top-k similar image

978-1-4673-7113-1/15/$31.00 © 2015 IEEE

detection technique to obtain significant bandwidth savings. Smarter transmits the selected representative master images to offer near real-time backup performance, while transmitting the remaining similar or near-duplicate images during the idle times and/or at a low bandwidth.

The novelty of this paper is to propose the methodology of approximate redundancy detection in the application level, compared with conventional exact-matching deduplication in the chunk level. Smarter offers comprehensive system implementations to efficiently leverage the affinity, which obtains better performance than existing multimedia solutions as demonstrated in the performance evaluation. We aim to make the following contributions.

Smart Hashing based Queries. Smarter offers redundancy detection at both the systems and content levels to identify and remove both identical (chunk-level) and geometrically transformed near-duplicate images in an efficient and scalable way. Smarter not only reduces the amount of data to be transmitted, but also decreases the transmission requirements via delta compression. Smarter leverages the metadata information of images to build affinity-aware summary vectors (ASV) that supports fast membership query for near-duplicate images. Smarter is able to identify more near-duplicate images than existing approaches and the amount of data to be replicated is significantly reduced.

Low-cost Image Identification. Smarter alleviates the computation overheads through three schemes. First, Smarter uses the parts-based features of images to identify and aggregate correlated images into affinity-aware groups by using Locality-Sensitive Hashing (LSH) [12] that has a computation complexity of $O(1)$ and low space overhead. Second, Smarter adopts a suitable "division-of-labor" between the metadata and contents of images. Smarter explores the metadata of images to check if they are duplicates via the space-efficient ASV design. Smarter exploits the content-based features only for non-duplicate images, thus significantly decreasing the number of images that are involved in content-based analysis. Third, Smarter slightly relaxes the selection criteria of the base images by using top-k similarity detection in order to reduce the overhead of network transmission. This results in more candidate similar images being identified and significantly reduces overall network bandwidth consumption.

Implementation and Evaluation. We implement the full featured Smarter on a cloud system for evaluation. We examine the performance by using multiple real-world datasets. We compare Smarter with state-of-the-art schemes, including EndRE [13], SiLo [14], Cluster-Based Deduplication (CBD) [15], and Near-Duplicate Detection (NDD) [5]. We evaluate the performance of Smarter in a real remote backup operation between two cities with a distance of over 1200km across a 1Gb/s link. The experimental results demonstrate the efficiency and benefits of Smarter. Moreover, Smarter provides two types of interfaces for users. One is to list all near-duplicate images for users in order to facilitate them to determine the most representative ones. The other is to automatically select from the near-duplicate images with maximum numbers of features, as the most representative images.

The rest of this paper is organized as follows. Section II discusses the motivations. Section III shows the architecture of Smarter and implementation details. We present experimental setup and evaluation results respectively in Sections IV and V. Section VI discusses the related work. We conclude our paper in Section VII.

II. Background and Motivations

In the context of cloud data centers, it is not cost-effective to execute a backup for a small number of images when considering the available bandwidth and operation complexity. In general, a batch of images can be transmitted to the destination for backups. In fact, Smarter obtains a suitable tradeoff between real-time backup and costs by transmitting representative images.

A. Remote Backup for Data Reliability

Introducing managed data redundancy by means of remote backup is an effective scheme to achieve data reliability for cloud services. Our study of cloud backups draws the following two observations.

(1) With the explosive growth in data volumes, increasing the capacity of data storage and at the same time keeping the data reliable is becoming one of the most critical challenges for cloud service providers. However, the relatively low bandwidth in remote backup environments results in long backup time and makes the backup infeasible especially for dynamic applications with large scale data sets. While conventional storage backup within a typical data center can rely on high-bandwidth local-area networks (LANs), large scale cloud service providers often deploy remote backups on wide-area networks (WANs) with limited bandwidth and over a long distance, for disaster recovery. For example, it takes more than 2 hours to transfer 1 Terabyte of data across a 1Gb/s link on a WAN between two sites in two big cities 1200km apart. Needless to say, efficient and intelligent remote backup policies are in urgent needs to substantially reduce the amount of backup data.

(2) Storing and sharing multimedia files such as images and photos from users is one of the key features for many cloud storage services. This is a trend that is likely to increase steadily given the rapid growth in image contents in the cloud. For example, the image data generated per week in Facebook can easily fill up dozens of Terabyte storage [16]. It is reported that nearly 75% of images have near-duplicates [17]. For these image contents, the conventional bit-level or chunk-level duplicate detection methods often fail to identify transformed copies of the same image as duplicates simply because the original image went through certain digital photometric or geometric transformations, such as adjustments on contrast, saturation, scaling, cropping, framing, etc. This problem can lead to tremendous bandwidth waste when remote backup is performed.

Motivated by these observations, we present Smarter, a novel bandwidth-efficient and affinity-aware scheme. Smarter is an efficient remote image backup system which can substantially alleviate the stress for network and provide high performance long-distance backup for cloud storage. Moreover, if the original set of images is backed up, an incremental backup needs to transmit a batch of new images. Hence, the backup for new images will also incur the non-trivial costs, if Smarter is not used.

Based on and inspired by the observations and analysis, Smarter incorporates the conventional system-level exact-matching deduplication for completely identical images and concentrates on remote backup for near-duplicate images via affinity awareness.

B. Affinity Awareness

Unlike conventional locality, affinity can be implied by or embedded in image attributes that are not necessarily time or space oriented. While in some cases locality may be part of the affinity and simple exploitation of locality may help scale up the performance to some extent, locality alone often fails to faithfully and comprehensively reveal the correlation among images due to limited dimensions and weak semantics.

Affinity in the context of Smarter refers to the semantic correlation derived from multi-dimensional attributes (e.g., metadata and access patterns), together with representative features of images. The affinity-aware design presents a suitable tradeoff between metadata (lightweight but less accurate) and contents (more accurate but expensive) analysis, which depends upon the salient properties of images. In general, the features of images can be efficiently captured to accurately represent their contents (see details in Section III-C2). This offers a special opportunity to identify near-duplicate images in a cost-effective way.

C. Deduplication and Delta Compression

To save bandwidth consumption, previous systems [16], [18], [19] have explored data reduction techniques before transmission to reduce the unnecessary redundancy. The commonly used techniques are *deduplication* in local servers and *delta compression* in network transmission.

Deduplication: The deduplication schemes [6]–[8] split files into multiple chunks. Each chunk is uniquely identified by a hash signature, called a fingerprint. By checking their fingerprints, duplicate chunks can be removed, while avoiding a byte-by-byte comparison and replacing identical data regions with references. However, the existing deduplication cannot meet the performance expectation of remote replication for multimedia data, which take a large percentage of data stored in the cloud.

Delta Compression: The delta compression schemes [9], [10] transmit data in the form of differences between current version and the replicated version. In practice, delta compression suffers from the high overheads in terms of computation and network bandwidth. The reason is that delta compression needs to compute the difference (i.e., the delta) between original and new versions, which generally exist in different locations. Moreover, the deltas from the chunk-level compression, alone fail to identify similar images in an efficient manner. The performance of delta compression can be improved via near deduplication.

III. Design and Implementations

In this section, we present an architectural overview of Smarter to illustrate the main idea and key design principles behind Smarter. The design goal of Smarter is to support bandwidth-efficient remote backups for disaster recovery. In the source end, Smarter selects candidate representative images for transmission via affinity-aware summary vectors and feature-based representation of images. Smarter leverages a top-k similarity set to reduce the bandwidth overhead between the source and destination of the backup via remote image backup. Smarter is orthogonal and complementary to existing non-image chunk-level backup schemes. For non-image data, Smarter can incorporate existing chunk-level fingerprint based schemes [6]–[8], [14].

A. The Source (Local) System

Smarter leverages the affinity of images to support cost-effective image backups. Figure 1 shows the Smarter architecture in the source end system. Smarter can support the operations from users (uploading/downloading and query, etc) and improve system performance (backup, caching, grouping and prefetching, etc). For image backup, there are three main functional components, Affinity-aware Summary Vectors (ASV), an improved Difference of Gaussian (DoG) detector from the original DoG [20] and an improved Principal Components Analysis - Scale-Invariant Feature Transform (PCA-SIFT) from the original PCA-SIFT [11]. ASV captures image affinity by leveraging the image metadata. Different from the original schemes in the multimedia community, the improved DoG and PCA-SIFT of Smarter can efficiently analyze image contents in a lightweight manner by exploiting metadata and feature-based representation of images.

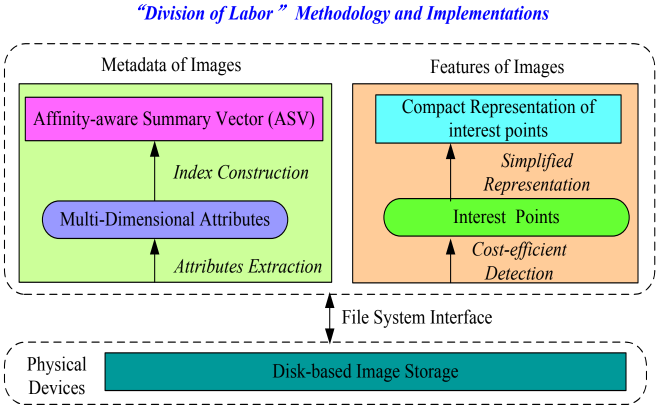

"Division of Labor" Methodology and Implementations
Metadata of Images
Features of Images
Affinity-aware Summary Vector (ASV)
Index Construction
Multi-Dimensional Attributes
Attributes Extraction
Compact Representation of interest points
Simplified Representation
Interest Points
Cost-efficient Detection
File System Interface
Physical Devices
Disk-based Image Storage

Fig. 1. Smarter architecture in the source system, showing a "division of labor" between content and metadata.

Instead of using conventional random hash functions (e.g., SHA-1 and MD5), ASV employs locality sensitive hashing (LSH) [12], [21] to efficiently capture image affinity that can accurately identify correlated images and support detection of near-duplicate images. In essence, ASV strikes a suitable tradeoff between Bloom filter [22], which is space efficient but only for exact-match queries, and conventional LSH [12], which is good for similarity queries but suffers from space inefficiency. The in-memory ASV, as a space-efficient data structure, is designed to detect similarity-based membership in the image backup process, as detailed next (Section III-C1). By checking the ASV array that consists of the ASVs of other servers in a cloud storage system, we can find the servers that contain near-duplicate images.

To support the feature-based content analysis of images, which is an important integral part of near-duplicate detection, improved DoG and PCA-SIFT are employed to respectively detect and represent interest points. In the computer vision field, an interest point refers to the point that is stable under local and global perturbations in the image domain. Typical perturbations include deformations (e.g., affine transformations, scale changes, rotations and translations) and illumination/brightness variations. The interest points represent the corresponding images in a space-efficient manner. By capturing the interest points using DoG and PCA-SIFT, Smarter identifies near-duplicate images as detailed in Section III-C2.

The improved and combined use of the three functional modules of ASV, DoG and PCA-SIFT enables Smarter to substantially reduce the need for on-disk index lookups during backup and decrease the complexity of identifying near-duplicate images in three steps. First, ASV fast determines if an image is non-duplicate from the system's viewpoint, thus avoiding the unnecessary and costly lookups in the disk-resident hash tables for images that are non-duplicate. Second, DoG is used to detect the interest points in the non-duplicate images determined by ASV, instead of exploring their full contents. Finally, the detected interest points are represented by PCA-SIFT in a compact way to obtain substantial space savings.

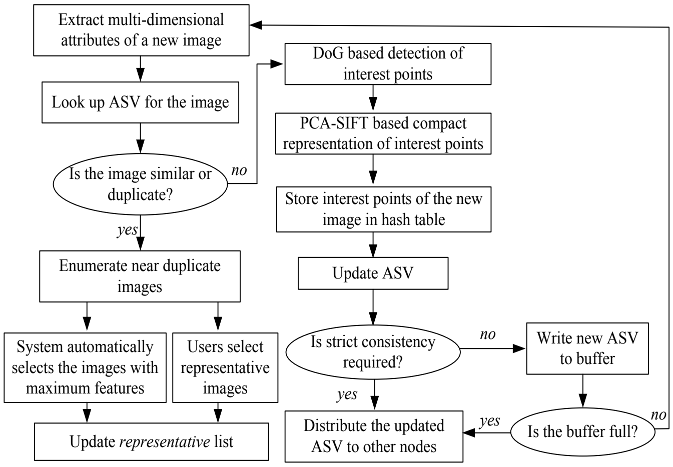

Extract multi-dimensional attributes of a new image
Look up ASV for the image
Is the image similar or duplicate?
no
yes
Enumerate near duplicate images
System automatically selects the images with maximum features
Users select representative images
Update *representative* list
DoG based detection of interest points
PCA-SIFT based compact representation of interest points
Store interest points of the new image in hash table
Update ASV
Is strict consistency required?
no
yes
Write new ASV to buffer
Is the buffer full?
yes
no
Distribute the updated ASV to other nodes

Fig. 2. The workflow of identifying representative images.

Figure 2 shows the workflow of locally identifying the representative images. Note that Smarter offers optional interfaces to select the representative images that can be replicated in a near real-time manner. In other words, a user is given the option of choosing a select subset of images from the Smarter-determined near-duplicate images for online transmission. Otherwise, if the user defers the decision to Smarter, the latter will select a representative image that contains the maximum number of features from its identified near-duplicate images for transmission.

For an arriving image in the backup image stream, Smarter first extracts its multi-dimensional attributes and looks up the image in ASV. If it is determined to be similar or duplicate to some existing images (see Section III-C1), Smarter enumerates top-d near-duplicate images of this image, where d is a constant pre-defined by users, and presents them to the user who then *optionally* determines if any of the selected images may become a representative image to update the representative image list. Otherwise, if the user opts not to choose, Smarter selects one with the maximum number of features as the representative image to add to the representative image list. The images selected by users, while subjective, offer a means (interface) and flexibility for a user to double-check if Smarter's selection is accurate/acceptable and define what he/she wants/allows to be deduplicated. Otherwise, the image is considered to be non-duplicate, which then goes into the feature-based representation, where DoG detects and PCA-SIFT represents the image's interest points that are stored in disk-resident hash tables. To support the scalability of the source system that generally consists of multiple nodes, the new ASV, updated by the new image, is distributed to other nodes in the source system appropriately, based on the consistency requirement and buffer status.

B. The Destination (Remote) System

The destination system in Smarter is used to backup the image dataset from the source system over a long distance for the purpose of disaster recovery. To avoid transmitting the entire image dataset for backup, which would result in an unacceptably long backup window and consuming extremely high WAN bandwidth, only the images from the representative image list identified by Smarter (or confirmed by the user) are considered for online transmission. The workflow of remote backup consists of procedures in both the source and destination servers. For the source, ideally it needs to send the images as fast and as little bandwidth as possible. For the destination, it needs to divide semantically correlated images into groups based on image metadata and content analysis to query the membership of similar images. The membership query of similar images helps determine if an image in the source needs to be transmitted. The meaning of efficient recovery in the Smarter context is to fast retrieve from the backup system the images that have been lost in the source due to a disaster.

Smarter substantially cuts down the cost of remote backups of image datasets following a three-pronged approach. First, Smarter only transmits a tiny fraction of all images, namely, the representative images, in a near real-time manner. For the representative images selected for transmission, Smarter actually transmits unique ones that are not already in the destination server by means of the application-level similarity detection techniques. Second, for non-duplicate images, the interest-points based identification technique is used to support correlation-aware grouping and queries via LSH in the destination system. As a result, similar images can be found within one or a limited number of correlation-aware groups. Third, Smarter uses a top-k-base-feature set technique, to identify more similar images in the image data transmission. In addition, for non-image data, Smarter adopts the conventional chunk-level fingerprint based deduplication approaches [15], [19], [23].

To reduce the overhead of network bandwidth and improve long-distance data transmission efficiency, we propose a feature-based top-k similarity scheme. This scheme is designed from an application's point of view, which allows Smarter to identify more similar images to significantly reduce the number of images transmitted. Note that since the interest-point based local features of each image have been obtained when locally selecting the representative images, Smarter can directly use these extracted features without re-computing.

When the source system needs to transmit an image for backup, it first sends the interest-point based features to the destination system. The destination system uses LSH to hash similar images into the correlated groups. Within the narrowed search scope, Smarter can easily and fast determine the membership of the image to be transmitted. A hit means that the destination system already has images that are highly similar to the image targeted for transfer. Smarter further checks if the existing image in the destination system has richer content/metadata information (i.e., more representative) than the one to be transmitted. If so, the latter needs not to be transmitted. Otherwise, this image, having richer information and being more representative (valuable), is transmitted to guarantee the recovery quality. In the meantime, the similarity search in the correlation-aware groups will yield the top-k images with the most matching interest points to the queried interest points.

Near-duplicate images generally contain identical features. In the source system where the image dataset has not been deduplicated for near-duplicates, a feature corresponds to several images. The features of the identified top-k most similar images are sent to the source system to identify more similar images, further exploiting the content locality in cloud storage [14], [19]. For example, besides the mentioned geometric transformation, there generally exists a large fraction of images that are semantically similar in content, e.g., taking pictures of the same scene from slightly different angles. It is very difficult to accurately identify such semantically similar images using the computer identification tools alone, due to the lack of accurate semantic identification techniques. Nevertheless, we believe intuitively that these pictures may share similar or identical features. The top-k similarity set has a large probability of covering these semantically similar images in content. Our evaluation results, presented in Section V-A1, confirm our intuition.

C. Implementation Details

1) Affinity-aware Summary Vectors: ASV maintains the near-duplicate memberships of stored data to determine whether a newly arriving image is near-duplicate. Given an image q, ASV needs to determine whether it is approximate (i.e., near-duplicate) to any image in a dataset S by examining the distance measured by l_s norm where the metric can be Hamming or Euclidean distance [21]. To detect near-duplicate images, affinity-aware summary vectors store and maintain the metadata of images. In ASV, we use the metadata of images, such as time stamps, file size, location and color histograms in images' Exif (Exchangeable image file format) headers, to quickly identify the images that are not duplicate. Exif is a standard that describes the formats for images, camera model, shot parameter and image properties [24]. In general, if two photos are near duplicates, their Exif metadata should be similar to each other with a high probability.

To rapidly and accurately identify similar images, we leverage LSH that is well-suited for processing high-dimensional images [21] and maps similar items into the same hash buckets with a high probability [12]. The LSH function families have a locality-aware property.

Definition 1: LSH function family, i.e., $H = \{h : S \rightarrow U\}$, is called (R, cR, P_1, P_2)-sensitive for distance function $||*||$ if for any $p, q \in S$

- If $||p,q|| \leq R$ then $Pr_H[h(p) = h(q)] \geq P_1$,
- If $||p,q|| > cR$ then $Pr_H[h(p) = h(q)] \leq P_2$.

To detect the similarity, we choose $c > 1$ and $P_1 > P_2$. In practice, we need to increase the gap between P_1 and P_2 via several hash functions. Each hash function $h_{a,b} : R^d \rightarrow Z$ is able to map a d-dimensional vector v onto a set of integers. The hash function in H is defined as $h_{a,b}(v) = \lfloor \frac{a \cdot v + b}{\omega} \rfloor$, in which a is a d-dimensional random vector with chosen entries following an s-stable distribution and b is a real number chosen uniformly from the range $[0, \omega)$. In the context of LSH, ω is a constant.

In the implementation of ASV, an ASV is a bit-vector with m bits, where each bit is initialized to 0. There are totally L LSH functions, h_i $(1 \leq i \leq L)$, of which each h_i hashes an item into one of the m bits based on the hash computation. Hence, it is possible for a bit to be set to 1 more than once, with only the first setting taking effect. All items belonging to a dataset S can be inserted into the m-bit space that serves as a summary vector of dataset S to support approximate (near-duplicate) queries.

When an approximate query request for item q arrives, we execute the same operations as the item insertion by hashing $h_i(q)$ $(1 \leq i \leq L)$ to L bit positions in the m-bit vector. If all the corresponding L bit positions in the vector are "1", we determine that item q is an approximate (i.e., near-duplicate) member of the dataset S in the metric R, i.e., $\exists p \in S$, $||p,q|| \leq R$. On the other hand, if any one of the corresponding bit positions in the vector has a "0", item q is quickly determined to be a non-duplicate (i.e., neither duplicate nor near-duplicate). It must be noted that, although ASV clearly shares some features with Bloom filter [22], they are significantly different in that ASV uses locality sensitive hashing to identify high-dimensional near duplicate images while Bloom filter uses random hashing for exact-matching queries.

To alleviate the false positives and false negatives, typical efforts, such as Multi-probe LSH [25], Locality-Sensitive Bloom Filter [26] and locality sensitive B-tree [27], have been proposed. Their idea is to leverage extra probing results to guarantee query accuracy. Unlike these approaches that incur substantial temporal or spatial overheads, Smarter leverages a simple but efficient approach that combines both the ASV results and users' feedback. The rationale behind this approach comes from two key observations. First, ASV itself can offer high accuracy (more than 90%) of identifying similar images as shown in the performance evaluation (Section V-A1). This means that ASV can support efficient grouping of similar images to facilitate users' choices. Second, the notion of similar images is essentially derived from a combination of quantitative analysis (computation results) and qualitative judgment (user choice). Therefore, Smarter can effectively compensate for the inaccuracy caused by the false answers (positive or negative) by asking for users' feedbacks through a user-friendly interface to determine the final results, i.e., a double-confirmation.

2) Feature-based Images: To perform reliable and accurate matching between different views of an object or scene that characterize near-duplicate images, we extract distinctive invariant features from the images. The features that are invariant to image scale and rotation are shown to provide robust matching across a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination. These features are highly distinctive in the sense that a single feature can be correctly matched with a high probability against a large dataset of images with many features. Feature-based management can detect and represent similar images to support correlation-aware grouping and similarity search.

Metadata per image are used to comprehensively represent the features of images, which unfortunately far exceeds the size of images themselves if their global features are to be explored. The global features of images generally require expensive representation and retrieval. To address this problem, we use interest points, an effective common form of local descriptors [20], that are widely employed in real-world applications such as object recognition and image retrieval. The reason is that interest points are robust to photometric changes and geometric variations and can be computed efficiently, so that several thousand points can be extracted from an image with a near real-time performance.

Using interest points entails two important operations, i.e., detection and representation. First, we localize interest points in position and scale. Interest points are placed at local peaks in a scale-space search, and filtered to preserve those that are likely to remain stable over transformations. Second, we need to build a description of the interest points to facilitate fast retrieval. The description is distinctive (i.e., differentiating one interest point from another), concise and invariant over transformations that may be caused by changes in camera pose and lighting.

3) Delta Compression and Recovery: Compression techniques are widely used in computer networks and storage systems to increase the efficiency of data transmission and reduce space requirements on the end systems. Delta encoding greatly reduces data redundancy. Collections of unique deltas are substantially more space-efficient than their non-encoded equivalents. There are two inputs, i.e., target file to be compressed and a reference source file. Encoding deltas needs to compress the difference between the target and source file as a delta. On the other hand, decoding delta takes the delta and source files as inputs to generate an exact copy of the target.

The rationale of delta compression comes from the fact that both sender and receiver contain a reference file that is similar to the transmitted file. Hence, we only need to transmit the difference (or delta) between the two files, which requires a significantly smaller number of bits. Formally, we have two files f_{new}, f_{old}, and a client and a server connected by a communication link. The client has a copy of f_{new} and the server has a copy of f_{old}. The design goal is to compute a file f_{δ} of minimum size, such that the server can reconstruct f_{new} from f_{old} and f_{δ}. f_{δ} is called as a delta of f_{new} and f_{old}.

Users may choose to recover some images at some points after their deletion. Although the design goal of Smarter is to remove near duplicate images, the recovery function is useful in improving system usability and allows users to flexibly and easily use the Smarter scheme. In order to support the recovery functionality, a simple but naive solution is to maintain the deleted images as backups, which consumes substantial system resources (computation and space overheads).

In order to offer a cost-effective and efficient recovery solution, we propose a delta-based decompression scheme that computes the difference among multiple similar images against a base one. The base image can be artificially selected by users or automatically by Smarter that leverages well-recognized clustering algorithms, such as *k-means*. Smarter only needs to maintain the base image (not deduplicated) and the deltas from other deduplicated images. The delta-based design can significantly reduce the demand for system resources, while supporting the recovery functionality. In order to facilitate the delta decoding in the delta recovery, Smarter maintains all images that were ever base ones, even if they are selected to be removed by some other users. When users want to recover images, Smarter executes the delta decoding operations by computing the base images and the deltas. The delta-based recovery incurs acceptable computation and space overheads.

IV. Experimental Setup

A. Evaluation Platform and Datasets

We have implemented a Smarter prototype between two data centers separated by a distance of more than 1200km and connected by a 1Gb/s network link. Each center consists of 128 servers and each server has a 16-core CPU, a 32GB RAM, a 500GB 7200RPM hard disk and Gigabit network interface card. The Smarter prototype implementation required approximately 6000 lines of C code in Linux. To comprehensively evaluate Smarter's performance, we use 4 image datasets, i.e., a *real* collection from cloud systems, and three typical image sets.

(1) *real* Cloud System. The cloud system offers storage services for more than many users, including faculty, staff, graduate students (Ph.D and Master) and undergraduate students. Each user is allowed to use 50GB capacity and the total allocated user storage capacity is more than 1.5PB. The backup period is 1 month. Specifically, in terms of file numbers, image, web and audio files dominate. Correspondingly, the image files (including gif, jpeg, bmp, etc.) consume the largest storage space (i.e., 35.2% capacity). Among these image files, the percentage of near duplicates, which come from geometric transformation via image processing software, is 15.7%. When considering 26.3% exact-matching duplicates, more than 42% (near) duplicates can be identified.

(2) Three Typical Applications: TRECVID [28], Online Art Gallery (OAG) [29] and MM270K [30]. TRECVID is an annual retrieval evaluation benchmark, which consists of low and high-quality parts. The low-quality part contains about 165 hours (17.8M frames, 127 GB) of MPEG-1 news footage. A total of 146,588 keyframes are in the formats of .jpeg, .bmp and .gif. Each keyframe is normally of low quality and at a resolution of 352×240 pixels. For the high-quality part, 8,000 Internet archive videos (50 GB, 200 hours) in MPEG-4/H.264 with durations between 10 seconds and 3.5 minutes were used as our test data. This set has more than 84,100 duplicate images. Moreover, OAG consists of 6,261 images of fine art downloaded from an online art gallery [29], which has

more than 30,000 duplicate images. Furthermore, we randomly chose 100,000 photos from MM270K [30] image set. This set has more than 50% duplicate images.

B. Metrics and Parameters

To examine the semantic correlation in Smarter's near-duplicate backup, we use similarity degree (SD) to measure the similarity between the compared image pair X,Y as $SD(X,Y) = (\frac{|S_X \cap S_Y|}{|S_X|} + \frac{|S_X \cap S_Y|}{|S_Y|})/2$, where $|S_X \cap S_Y|$ is equal to the number of similar interest points [5]. $|S_X|, |S_Y|$ represent the number of interest points in images X and Y, respectively. The similarity degree demonstrates the correlation of (near) duplicate images from the views of both system level and application level, which traditionally are evaluated respectively in the network systems and multimedia fields. SD ranges from 0 to 1, where the value 0 means that the images are totally different and the value 1 means that they are identical (duplicate).

By executing many sampling operations and incorporating users' feedback from the evaluated datasets, we study the value of the similarity degree. We find that it is proper and suitable for the similarity degree to be set at 0.75 for the four image datasets. We observe that about 80% near-duplicate images have their similarity degree located in the range of $[0.75, 1]$. To evaluate the image similarity detection performance, we classify the (near) duplicates into 2 levels, i.e., *format level and transformation level*. Specifically, the format-level (near) duplicates include the images that have the same contents but different formats. The transformation-level (near) duplicates include the images that are derived from the same source images under different transformation policies.

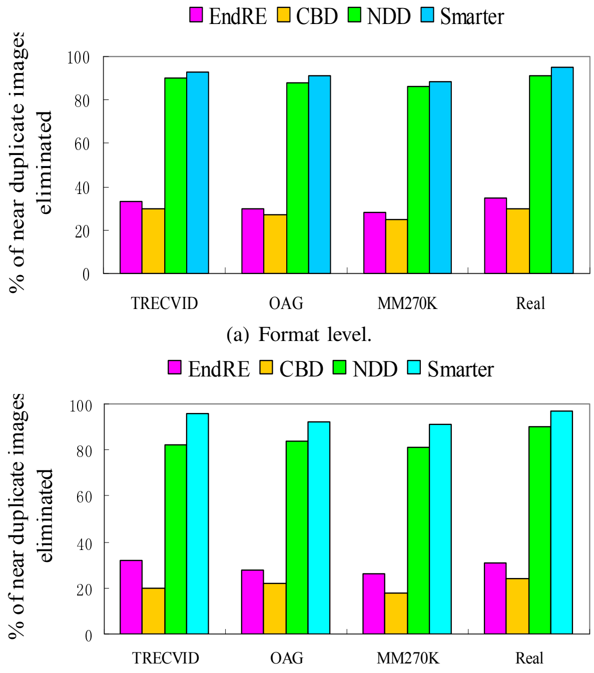
To evaluate image similarity detection, we compare Smarter with state-of-the-art schemes in both the network systems field, EndRE [13] and Cluster-Based Deduplication (CBD) [15], and the multimedia field, Near-Duplicate Detection (NDD) [5]. Since there are no open source codes of EndRE, CBD, and NDD, we choose to re-implement them. Specifically, we have implemented EndRE's redundancy elimination, which includes an adaptive algorithm (i.e., SampleByte) and an optimized data structure. We also implement CBD [15], including fingerprint cache, containers and super-chunk based data routing in the deduplication clusters. The implementation of NDD follows its processing steps and computation models [5]. Moreover, for remote backups, we compare Smarter with Stream-Informed Delta Compression (SIDC) [19] that is a feature of backup replication in EMC backup recovery systems. SIDC is re-implemented, including Bloom filter, fingerprint index, and containers, to load the stored sketches into a stream-informed cache.

V. Results and Analysis

We present the experimental results from the perspectives of local similarity detection and remote backups.

A. The Source (Local) System

1) Detection Accuracy: The image similarity detection helps identify representative images by examining the identical and geometrically transformed images that are derived from the same source images. Figure 3 shows the percentage of near-duplicate images detected at the format level and the transformation level, respectively. At the format level, as shown in Figure 3(a), since EndRE and CBD carry out the exact-matching identification for completely identical images, they detect on average 22.6% of all images as duplicate images. In general, they fail to detect (near) duplicate images that are stored in different file formats. NDD, which uses content-based feature approach, can identify more (near) duplicate (i.e., 87.5%) images than the system-level identification schemes. Smarter identifies the most (near) duplicates (i.e., 92.7%). The reason is that Smarter leverages efficient feature-based detection technique to detect similar (near duplicate) images and execute semantic-aware grouping to facilitate accurate identification.



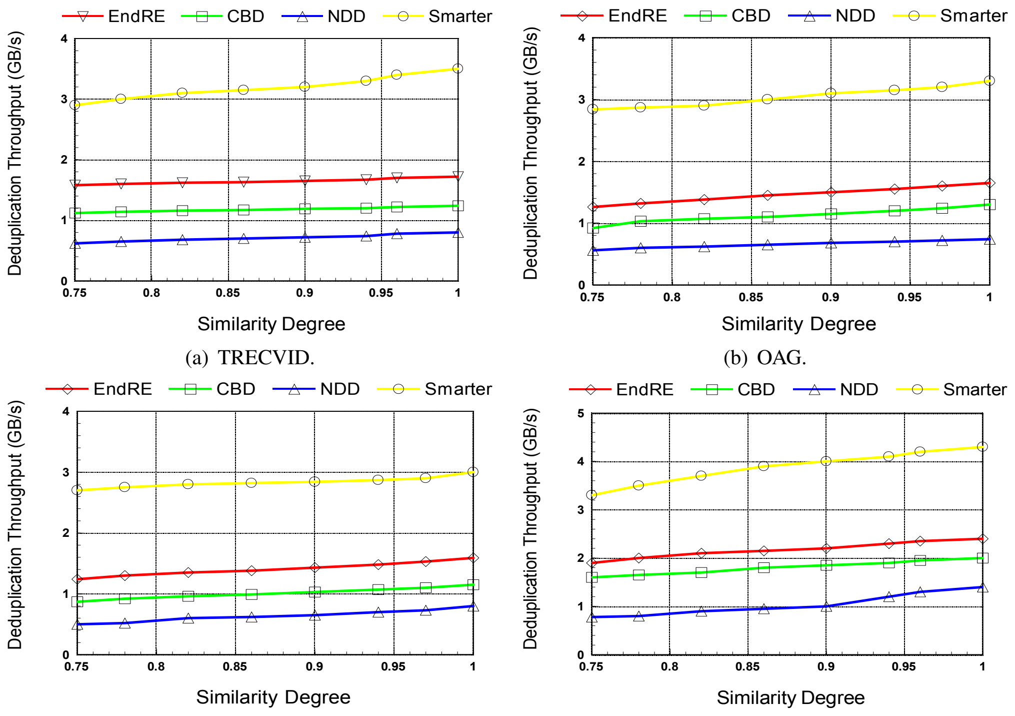
(a) Format level.

(b) Transformation level.

Fig. 3. Percentage of similar (near-duplicate) images identified at the format and transformation levels.

At the transformation level, as shown in Figure 3(b), the percentages of images identified are 21.6% and 18.4% for EndRE and CBD, which are much lower than 86.5% and 92.8% for NDD and Smarter. The detected near duplicates by the system-level schemes come from the images generated from transformations that happen to preserve the partitioned chunks and fingerprints, such as downsampling and scaling. They fail to detect the near duplicates generated from the majority of the transformation policies that tend to alter data chunks and fingerprints. We also observe that the detected near duplicates by the system-level schemes are mostly completely identical images or cropping results that account for a small fraction of original image datasets.

2) Identification Throughput: Figure 4 shows the results in terms of identification throughput. Smarter achieves an average throughput of about 3.25GB/s, higher than the 1.85GB/s, 1.22GB/s and 0.81GB/s throughputs respectively by EndRE, CBD and NDD. The substantial throughput advantage of Smarter can be attributed to the space-efficient vector rep-



(a) TRECVID.

(b) OAG.

(c) MM270K.

(d) Real.

Fig. 4. The throughputs in deduplicating image datasets.

resentation and content-based feature analysis as described in Section III-A. Furthermore, due to the high computation complexity in NDD, its throughput is relatively low. Although EndRE optimizes data structures to reduce memory overhead, its fingerprint hash table consumes substantial space that is much larger than the main memory size, leading to frequent and random accesses to the uncached fingerprints in hard disks and decreasing the throughput.

3) False Rates of Space-efficient Structures: We examine the false probabilities of the space-efficient, compared with the widely used Bloom filters. Figure 5 shows the evaluation results in the four datasets. The false probabilities consist of two parts, i.e., false positives and false negatives. First, for false positives, we observe that ASV and Bloom filters obtain comparable probabilities, i.e., around 0.01%. ASV uses probabilistic grouping, while Bloom filters incur hash collisions. Moreover, for false negatives, ASV achieves much better performance than Bloom filters. The main reason is the exact-matching feature in Bloom filters, which fail to identify near-duplicate images.

4) Time Overhead: We examine the time overhead in processing near-duplicate image identification, with evaluation results shown in Figure 6. Since NDD uses the expensive scheme for content-based feature extraction, it incurs the highest time overhead, i.e., 1252.7s, 469.5s, 752s and 2472.5s, respectively for the TRECVID, OAG, MM270K and *real* image datasets. EndRE leverages the sample-based fingerprinting algorithm to accelerate the identification. Smarter incurs the lowest time overhead for two reasons. One is the proper management and analysis in the content and metadata of images. The other is the usage of the LSH scheme to implement the fast and accurate detection of near duplicates of images.

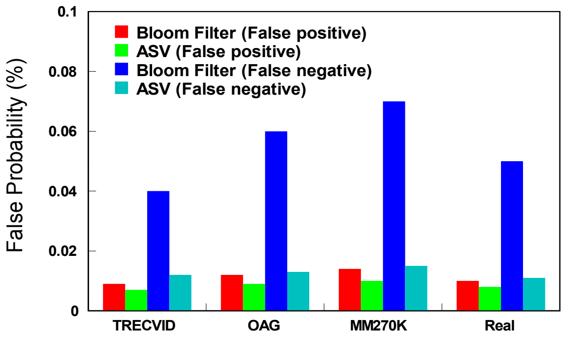


Fig. 5. False probabilities in space-efficient structures.

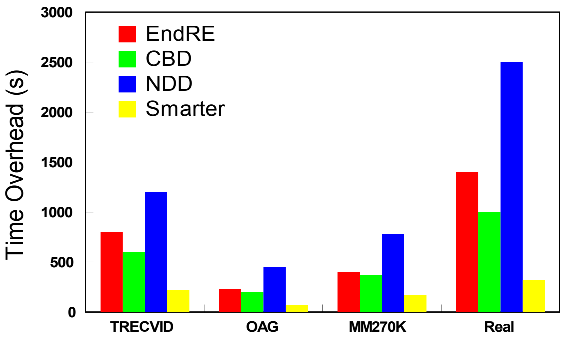


Fig. 6. Time overhead in hashing-based identification.

B. The Quality of Remote Backups

We focus on the experimental results from real remote backup operations by examining the chunk-level and similarity-level detection, effective throughput and similarity computation overhead.

1) Chunk-level and Similarity-level Detection: Smarter aims to reduce the number and size of transmitted images between source and destination systems in remote backup via similarity detection to prevent duplicate and near-duplicate images from being transmitted over the low-bandwidth WAN link. The similarity examines the near-duplicate images with identical features. We compare the detection accuracy of chunk-level exact-matching deduplication, Stream-Informed Delta Compression (SIDC) [19] and Smarter. Figure 7 shows the results. The lowest segment on each vertical bar represents the percentage of all identified images (thus compressed) that are attributed to chunk-level detection. The middle and upper segments on each bar indicate the fractions of the compression attributed to SIDC and Smarter. Smarter contributes by far the most to the reduction of data transmitted over the WAN link in the remote backup.

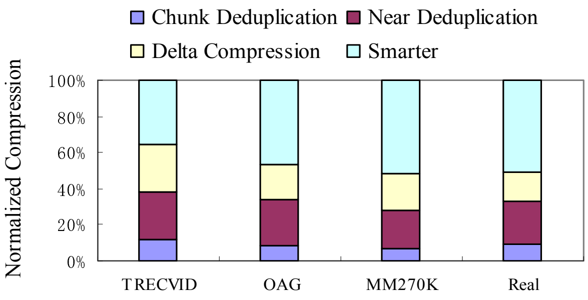


Fig. 7. Chunk-level and similarity-level detection.

2) Effective Throughput: We perform numerous backup experiments to measure effective network throughput between two remote cities. Figure 8 shows a representative backup result for the *real* dataset. WAN link has a bandwidth of 1Gbps and the backup throughput is measured every 2 seconds. We observe that compared with NDD, the average effective throughputs in Smarter and SIDC are 152.82Gb/s and 32.65Gb/s respectively, much higher than the NDD throughput of 0.78Gb/s. The main reason is that SIDC leverages the chunk-level delta compression that physically reduces the amounts of transmitted data. However, the chunk-level SIDC only relies on the closest fingerprint to determine the similarity. The too strict design, i.e., top-1 nearest chunk, often fails to identify similar chunks to be compressed.

Smarter obtains significant improvements via application-level and top-k similarity detection. The former allows Smarter to execute content-based detection, while the latter gives an opportunity to find more similar images within the correlation-aware groups. The throughput improvements come from local near-deduplication and delta compression in transmission. For non-images, we deduplicate identical chunks to obtain bandwidth savings and improve effective throughput.

VI. Related Work

To capture traffic redundancy, CoRE [31] uses two-layer redundancy detection for chunking and fingerprinting, while supporting cooperative operations between the sender and the receiver. CARE [32] supports content-aware redundancy elimination and incorporates image similarity detection algorithms in the forwarding path to handle the large generation of redundant content. Information-bound referencing [33] supports multimedia dissemination and access via fingerprinting algorithms. By using a SampleByte scheme to deliver compression gains and adapt CPU usage, EndRE [13] offers redundancy elimination as an end system service. Moreover, in order to improve video accessibility and availability, Information-Bound References [3] are used to bind the video content references to the underlying information, thus being effective at reducing redundant transfers.

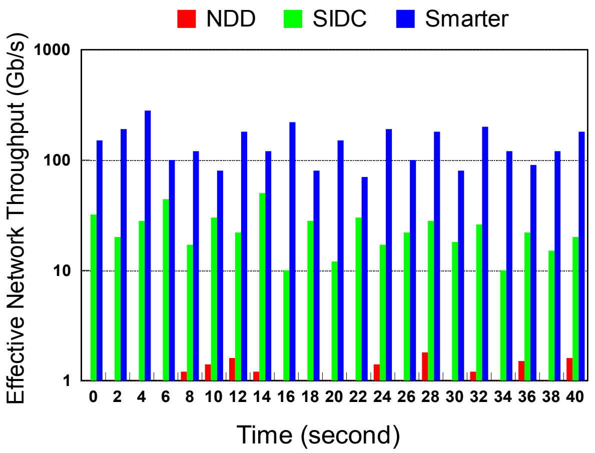


Fig. 8. Effective throughput in remote backups.

Efficient backup of data in remote sites is important to disaster recovery. Stream-Informed Delta Compression (SIDC) [19] adds stream-informed delta compression to the existing deduplication systems to eliminate the need for persistent indices. A scalable, geo-replicated storage system was proposed in [34] to offer a rich data model and provide stronger semantics. Inside Dropbox [35] reveals that Dropbox service is highly impacted by the distance between the clients and the data-centers, which can be addressed by chunk bundling.

By exploiting the skewness in the communication patterns, a tradeoff between improving fault tolerance and reducing bandwidth usage is obtained in [36]. Fault tolerant DCell [37] alleviates the single point of failure and obtains near shortest-path routing by using a distributed fault-tolerant routing protocol. Paxos-based replicated state machine [38] implements a storage service and can tolerate arbitrary machine restarts, disk failures and some Byzantine faults. WheelFS [39] allows multi-site applications to share data and gain fault tolerance. NetStitcher [40] uses a network of storage nodes to aggregate un-utilized bandwidth, and leverages a store-and-forward algorithm to schedule data transfers.

Unlike existing work, such as EndRE [13], FAST [41], Cluster-Based Deduplication (CBD) [15], FastVR [42] and SiLo [14], Smarter improves network transmission efficiency that is important in disaster discovery. Our scheme leverages a suitable division-of-labor between the metadata and data, rather than only exploiting data contents in existing work. Smarter implements both system-level and application level image identification and obtains semantics-aware grouping to improve system efficiency and scalability.

VII. Conclusion

In order to improve the dependability and reliability of cloud services, we propose Smarter, a bandwidth-efficient remote backup system for large-scale cloud applications. Smarter explores and exploits the semantic correlation that exists in both the metadata and data contents. By leveraging a suitable "division of labor" between the metadata and feature-based contents of images, Smarter narrows the feature-based analysis to focus only on non-duplicate images, and significantly reduces the overheads of processing images. To evaluate Smarter, we implement Smarter and conduct extensive experiments with several large-scale datasets. Results demonstrate Smarter significantly outperforms existing schemes.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China (NSFC) under Grant 61173043 and National Basic Research 973 Program of China under Grant 2011CB302301.

References

[1] P. Shankaranarayanan, A. Sivakumar, S. Rao, and M. Tawarmalani, "Performance sensitive replication in geo-distributed cloud datastores," *Proc. IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2014.

[2] Y. Hua, B. Xiao, and X. Liu, "NEST: Locality-aware approximate query service for cloud computing," in *Proceedings of IEEE INFOCOM*, pp. 1303–1311, IEEE, 2013.

[3] A. Anand, A. Balachandran, A. Akella, V. Sekar, and S. Seshan, "Enhancing video accessibility and availability using information-bound references," *Proc. ACM CoNEXT*, 2013.

[4] Y. Hua and X. Liu, "Scheduling heterogeneous flows with delay-aware deduplication for avionics applications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, pp. 1790–1802, 2012.

[5] Y. Ke, R. Sukthankar, and L. Huston, "Efficient near-duplicate detection and sub-image retrieval," in *ACM Multimedia*, pp. 869–876, 2004.

[6] B. Zhu, K. Li, and H. Patterson, "Avoiding the disk bottleneck in the data domain deduplication file system," *Proc. FAST*, 2008.

[7] K. Srinivasan, T. Bisson, G. Goodson, and K. Voruganti, "iDedup: Latency-aware, Inline Data Deduplication for Primary Storage," *Proc. FAST*, 2012.

[8] M. Lillibridge, K. Eshghi, D. Bhagwat, V. Deolalikar, G. Trezise, and P. Camble, "Sparse indexing: large scale, inline deduplication using sampling and locality," *Proc. FAST*, 2009.

[9] D. Trendafilov, N. Memon, and T. Suel, "zdelta: a simple delta compression tool," *Technical Report, CIS Department, Polytechnic University*, 2002.

[10] J. MacDonald, "File system support for delta compression," *Master's thesis, Department of Electrical Engineering and Computer Science, University of California at Berkeley*, 2000.

[11] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," *Prc. CVPR*, 2004.

[12] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," *Proc. STOC*, 1998.

[13] B. Aggarwal, A. Akella, A. Anand, A. Balachandran, P. Chitnis, C. Muthukrishnan, R. Ramjee, and G. Varghese, "EndRE: an end-system redundancy elimination service for enterprises," *Proc. NSDI*, 2010.

[14] 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," *Proc. USENIX ATC*, 2011.

[15] W. Dong, F. Douglis, K. Li, H. Patterson, S. Reddy, and P. Shilane, "Tradeoffs in scalable data routing for deduplication clusters," *Proc. FAST*, 2011.

[16] D. Beaver, S. Kumar, H. C. Li, J. Sobel, and P. Vajgel, "Finding a needle in haystack facebook's photo storage," *Proc. OSDI*, 2010.

[17] J. Gantz and D. Reinsel, "The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East," *International Data Corporation (IDC) iView*, December 2012.

[18] A. Muthitacharoen, B. Chen, and D. Mazieres, "A low-bandwidth network file system," *Proc. SOSP*, 2001.

[19] P. Shilane, M. Huang, G. Wallace, and W. Hsu, "WAN Optimized Replication of Backup Datasets Using Stream-Informed Delta Compression," *Proc. FAST*, 2012.

[20] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[21] A. Andoni and P. Indyk, "Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions," *Communications of the ACM*, vol. 51, no. 1, pp. 117–122, 2008.

[22] B. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[23] B. Debnath, S. Sengupta, and J. Li, "ChunkStash: speeding up inline storage deduplication using flash memory," *Proc. USENIX ATC*, 2010.

[24] M. Huiskes and M. Lew, "The MIR flickr retrieval evaluation," *Proc. ACM Conference on Multimedia Information Retrieval*, 2008.

[25] Q. Lv, W. Josephson, Z. Wang, M. Charikar, and K. Li, "Multi-probe LSH: Efficient indexing for high-dimensional similarity search," in *VLDB*, pp. 950–961, 2007.

[26] Y. Hua, B. Xiao, B. Veeravalli, and D. Feng, "Locality-Sensitive Bloom Filter for Approximate Membership Query," *IEEE Transactions on Computers*, vol. 61, no. 6, pp. 817–830, 2012.

[27] Y. Tao, K. Yi, C. Sheng, and P. Kalnis, "Quality and efficiency in high dimensional nearest neighbor search," in *SIGMOD*, pp. 563–576, 2009.

[28] TRECVID *http://trecvid.nist.gov/*.

[29] CGFA - A Virtual Art Museum *http://cgfa.sunsite.dk/*.

[30] Media Graphics International *270,000 Multimedia Graphics Pack*, 1998.

[31] L. Yu, K. Sapra, H. Shen, and L. Ye, "Cooperative end-to-end traffic redundancy elimination for reducing cloud bandwidth cost," *Proc. ICNP*, 2012.

[32] U. Weinsberg, Q. Li, N. Taft, A. Balachandran, V. Sekar, G. Iannaccone, and S. Seshan, "CARE: content aware redundancy elimination for challenged networks," *Proc. ACM HotNets*, 2012.

[33] A. Anand, A. Akella, V. Sekar, and S. Seshan, "A case for information-bound referencing," *Proc. ACM HotNets*, 2010.

[34] W. Lloyd, M. J. Freedman, M. Kaminsky, and D. G. Andersen, "Stronger semantics for low-latency geo-replicated storage," *Proc NSDI*, 2013.

[35] I. Drago, M. Mellia, M M Munafo, A. Sperotto, R. Sadre, and A. Pras, "Inside dropbox: understanding personal cloud storage services," *Proc. ACM IMC*, 2012.

[36] P. Bodík, I. Menache, M. Chowdhury, P. Mani, D. Maltz, and I. Stoica, "Surviving failures in bandwidth-constrained datacenters," *Proc. SIGCOMM*, 2012.

[37] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, "Dcell: a scalable and fault-tolerant network structure for data centers," *Proc. SIGCOMM*, 2008.

[38] W. Bolosky, D. Bradshaw, R. Haagens, N. Kusters, and P. Li, "Paxos replicated state machines as the basis of a high-performance data store," *Proc. USENIX NSDI*, 2011.

[39] J. Stribling, Y. Sovran, I. Zhang, X. Pretzer, J. Li, M. Kaashoek, and R. Morris, "Flexible, wide-area storage for distributed systems with WheelFS," *Proc. USENIX NSDI*, 2009.

[40] N. Laoutaris, M. Sirivianos, X. Yang, and P. Rodriguez, "Inter-datacenter bulk transfers with netstitcher," *Proc. SIGCOMM*, 2011.

[41] Y. Hua, H. Jiang, and D. Feng, "FAST: Near Real-time Searchable Data Analytics for the Cloud," *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pp. 754–765, November 2014.

[42] Z. Nie, Y. Hua, D. Feng, Q. Li, and Y. Sun, "Efficient Storage Support for Real-time Near-duplicate Video Retrieval," *Proceedings of the 14th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP)*, pp. 312–324, 2014.