ARTICLE IN PRESS

Future Generation Computer Systems ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

DAC: Improving storage availability with Deduplication-Assisted Cloud-of-Clouds

Suzhen Wu [a], Kuan-Ching Li [c], Bo Mao [b,*], Minghong Liao [b]

[a] Computer Science Department, Xiamen University, Xiamen, 361005, China

[b] Software School, Xiamen University, Xiamen, 361005, China

[c] Department of Computer Science and Information Engineering, Providence University, Taiwan

HIGHLIGHTS

- From the investigations and the preliminary performance evaluations, we show the diversity characteristics of cloud storage systems from the cost and performance aspects.
- A deduplication-assisted cloud storage system is proposed to improve the storage efficiency and network bandwidth.
- Exploiting the data reference characteristics to place data blocks among multiple cloud storage providers.
- The availability of cloud storage system is improved by incorporating both the replication and erasure code schemes.

ARTICLE INFO

Article history:
Received 14 September 2015
Received in revised form
22 January 2016
Accepted 6 February 2016
Available online xxxx

Keywords:
Cloud-of-Clouds
Data deduplication
Availability
Data distribution
Cloud storage

ABSTRACT

With the increasing popularity and rapid development of the cloud storage technology, more and more users are beginning to upload their data to the cloud storage platform. However, solely depending on a particular cloud storage provider has a number of potentially serious problems, such as vendor lock-in, availability and security. To address these problems, we propose a Deduplication-Assisted primary storage system in Cloud-of-Clouds (short for DAC) in this paper. DAC eliminates the redundant data blocks in the cloud computing environment and distributes the data among multiple independent cloud storage providers by exploiting the data reference characteristics. In DAC, the data blocks are stored in multiple cloud storage providers by combing the replication and erasure code schemes. To better utilize the advantages of both replication and erasure code schemes and exploit the reference characteristics in data deduplication, the high referenced data blocks are stored with the replication scheme while the other data blocks are stored with the erasure code scheme. The experiments conducted on our lightweight prototype implementation show that DAC improves the performance and cost efficiency significantly, compared with the existing schemes.

1. Introduction

With the increasing popularity and cost-effectiveness of cloud storage systems, many companies and organizations have migrated or plan to migrate data from their private data centers to the cloud. However, solely depending on a particular cloud storage provider has a number of potentially serious problems. First, it can cause the so-called vendor lock-in problem for the customers [1,2], which results in prohibitively high cost for clients to switch from one provider to another. Second, it can cause service disruptions, which in turn will lead to SLA violation, due to cloud outages, resulting in penalties, monetary or other forms, for the service providers. Examples include a series of high-profile cloud outages in the year of 2013 for cloud providers, such as Amazon, Microsoft and Google [3], from a 5-min failure that costed half a million dollars to a week-long disruption that costed an immeasurable amount of brand damage. From January to March 2014, DropBox has experienced twice service outages [3]. More seriously, Nirvanix filed for Chapter 11 bankruptcy protection on October 1, 2013 [4]. The company gave customers two weeks' notice to retrieve their data. Some users had petabytes of data with single copy stored in Nirvanix. Third, solely depending on a particular cloud storage provider can also result in possible increased service costs and data security issues, such as the data

* Corresponding author. Tel.: +86 5922580033.
E-mail address: maobo@xmu.edu.cn (B. Mao).

http://dx.doi.org/10.1016/j.future.2016.02.001
0167-739X/

Please cite this article in press as: S. Wu, et al., DAC: Improving storage availability with Deduplication-Assisted Cloud-of-Clouds, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.02.001

leakage problem [5]. Therefore, using multiple independent cloud providers, called Cloud-of-Clouds, is an effective way to provide better availability for the cloud storage systems.

In a Cloud-of-Clouds storage system, the data redundancy is introduced to judiciously distribute the data among the clouds. Thus, the redundant data distribution scheme is critically important for the storage availability, performance, cost and space efficiency. Several systems have been proposed for Cloud-of-Clouds. RACS [1] uses the erasure coding to mitigate the vendor lock-in problem encountered by a user when switching the cloud vendors. It transparently stripes the data across multiple cloud storage providers with RAID-like techniques. HAIL [6] provides integrity and availability guarantees for the stored data. It allows a set of servers prove to a client that a stored file is intact and retrievable by the approaches adopted from the cryptographic and distributed-systems communities. NCCloud [7] achieves the cost-effective repair for a permanent single-cloud provider failure to improve the availability of cloud storage services. It is built based on network-coding-based storage schemes called regenerating codes with an emphasis on the storage repair, excluding the failed cloud in repair.

The above three systems are all based on the erasure code or the network code. In contrast, DuraCloud [8] utilizes replication to copy the user content to several different cloud storage providers to provide better availability. Moreover, it ensures that all copies of the user content remain synchronized. However, users should pay more money for the additional storage space and bandwidth required by DuraCloud. DEPSKY [2] improves the availability and confidentiality of the commercial storage cloud services by building a Cloud-of-Clouds on top of a set of storage clouds, combining the Byzantine quorum system protocol, cryptographic secret sharing, replication and the diversity of different cloud providers. Different from these approaches, HyRD [9] integrates both replication and erasure code to the Cloud-of-Clouds. It takes the workload characteristics and the diversity of cloud storage providers, specially the file sizes, into the design of the redundant data distribution strategy so that the advantages of both the replication and erasure code can be exploited while their disadvantages can be hidden. As a result, both the performance and storage efficiency are improved with the availability guarantee. However, the redundant data blocks over the network are not eliminated.

On the other hand, previous studies on the workload characteristics have shown that the data redundancy is moderate to high in the cloud storage environments [10–12]. These studies have shown that by applying the data deduplication technology to large-scale data sets, an average space saving of 30%, with up to 90% in VM and 70% in HPC storage systems, can be achieved. The recent studies, such as RACS [1], DuraCloud [8], DepSky [2], NCCloud [7] and HyRD [9], indicate that the replication-based schemes are performance-friendly to the hot data blocks while the erasure-code-based schemes are cost-efficient to the cold data blocks [9,13]. It suggests that a sensible data distribution scheme in the Cloud-of-Clouds should dynamically utilize the replication and erasure codes based on the hotness characteristics of data blocks. To address the important storage availability issue in the Cloud-of-Clouds, we propose a deduplication-assisted data reduction and data distribution approach, called DAC, by exploiting the data redundancy characteristics of applications. DAC utilizes the replication scheme to store the data blocks with high reference count, and utilizes the erasure codes to store the other data blocks on multiple cloud storage providers. By exploiting the data redundancy characteristics and the diversity of cloud providers, both the advantages of erasure codes and replication are exploited and their disadvantages are alleviated. The extensive trace-driven experiments conducted on our lightweight prototype implementation of DAC show that DAC significantly outperforms RACS, DuraCloud and HyRD in the I/O performance measure of average response times. Moreover, our evaluation and analysis results also show that DAC achieves significant cost and space efficiency.

The rest of this paper is organized as follows. The background and motivation are presented in Section 2. We describe the DAC architecture and design in Section 3. The performance evaluation is presented in Section 4. We present the related work in Section 5 and conclude this paper in Section 6.

2. Background and motivation

In this section, we present some important observations drawn from previous and our analysis of the vendor lock-in problem of cloud storage, the diversity characteristics of cloud storage providers, and the data redundancy in primary storage systems to motivate the DAC study.

2.1. The vendor lock-in problem

The services provided by the cloud storage are diverse [1,18]. The cloud storage providers offer different pricing and different performance characteristics, including extra features such as geographic data distribution, access through mountable file systems and specific APIs. Changes in these features, or the emergence of new providers with more powerful and attractive characteristics, might compel some users to switch from one provider to another. However, moving from one provider to another one may be very expensive because the switching cost is proportional to the amount of data that has been stored in the original provider [1]. The more data has been stored in the original provider, the higher switching cost will be paid to the data migration. It puts the users at a disadvantage, that is, when the cloud storage provider that has stored the user's data raises the prices or negotiates a new contract less favorable to the user, the user has no choice but to accept because of the high switching cost, which is called *vendor lock-in problem* [1,2].

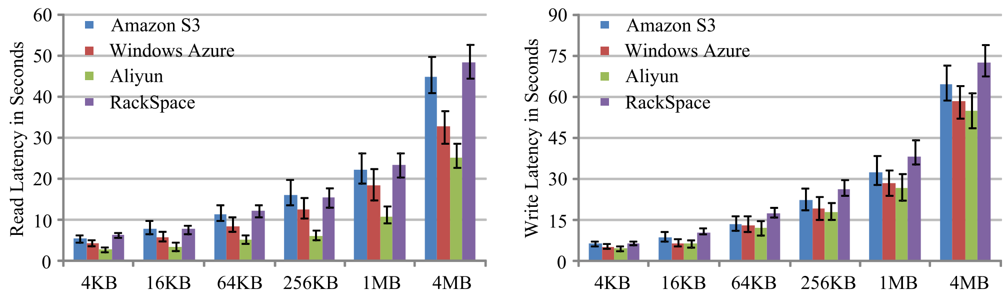
Besides the possible increased prices or pressed unfavorable new contract, the vendor lock-in can also lead to possible data loss or unavailability for users if their cloud storage provider goes out of business or suffers a service outage. Despite of the strict Service-Level Agreements (SLAs) between the cloud provider and the user, the service failures and outage occur and are almost unavoidable [9,19]. The cloud outages in 2013, although infrequent, showed that the service unavailability may last up to several hours and even several days [3]. A study conducted by the ESG (Enterprise Strategy Group) research shown that about 58% of professionals in SMBs (Small and Medium Businesses) can tolerate no more than four hours of downtime before experiencing significant adverse effect [20,21]. More seriously, EMC's Disaster Recovery Survey in 2013 [22] observed that the average cost per hour of downtime is much higher than ever before and 54% of users suffered from lost data or service downtime, which further stresses the importance of the service/data availability in cloud storage systems.

To address the vendor lock-in problem induced by a single individual cloud provider, a Cloud-of-Clouds solution is proposed in the literatures [1,2,7,8]. It redundantly distributes the data across multiple providers by means of the data redundancy schemes, such as replication and erasure codes. As a result, users can maintain their mobility while insuring against the outages of a single individual cloud provider.

Please cite this article in press as: S. Wu, et al., DAC: Improving storage availability with Deduplication-Assisted Cloud-of-Clouds, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.02.001

Table 1
Monthly price plans (in US dollars) for Amazon S3, Windows Azure Storage, Aliyun Open Storage Service and Rackspace Cloud Files, as of September, 10th 2014 in the China region.

Operations and vendors	Amazon S3 [14]	Windows Azure [15]	Aliyun [16]	RackSpace [17]
Storage (per GB/month)	$0.033	$0.157	$0.029	$0.13
Data in (per GB)	Free	Free	Free	Free
Data out to internet (per GB)	$0.201	Free	$0.123	Free
Put, copy, post, and list (per 10k transactions)	$0.047	Free	$0.0016	Free
Get and others (per 10k transactions)	$0.0037	Free	$0.0016	Free



(a) Read latency. (b) Write latency.

Fig. 1. Read/write latency as a function of the file size for single-cloud storage providers.

2.2. Diversity of cloud storage providers

The services provided by different cloud storage providers are diverse [1,18]. The cloud storage providers offer different prices and different performance characteristics. Table 1 shows the monthly price plans for four major providers as of September 10th 2014. For the four cloud providers, we use the prices from the first chargeable usage tier in the China region (*i.e.*, storage usage within 1 TB/month in Amazon S3; the volume of the data transferred out ranges between 1 GB/month and 10 TB/month). From Table 1, we can see that the charged costs of the four cloud storage providers are different in the aspects of storage, data in/out and the metadata operations. Moreover, we also conducted the performance evaluations on these cloud storage providers, as shown in Fig. 1. We can see that there is a huge variance among the performance and the cost of the different cloud providers. It implies an important advantage of the Cloud-of-Clouds that is we can exploit the workload characteristics and the diversity of cloud storage providers to distribute data among multiple cloud storage providers.

In general, two common redundant data distribution methods, *i.e.*, replication-based and erasure-code-based schemes, are used in Cloud-of-Clouds to leverage the diversity characteristics of cloud storage providers. Replication provides better performance while erasure codes provide better storage efficiency. However, replication imposes extremely high bandwidth and storage overhead, while erasure codes can provide the robustness and expected high access performance in the Cloud-of-Clouds particularly for large files. It therefore hints at the possibility of a certain combination of the two schemes, which tries to retain their respective advantages and hide their disadvantages to provide the most appropriate redundant data distribution scheme in Cloud-of-Clouds [9].

2.3. Data redundancy in primary storage systems

Previous studies have found that the data redundancy is moderate to high in the cloud storage environments [10–12]. Our analysis on the primary workloads shows that the I/O redundancy is more than 50% on average as illustrated in Fig. 2 on the Web-Vm, Homes and Mail traces that are collected from Florida International University (FIU traces [23]). From Fig. 2, we can also find that I/O

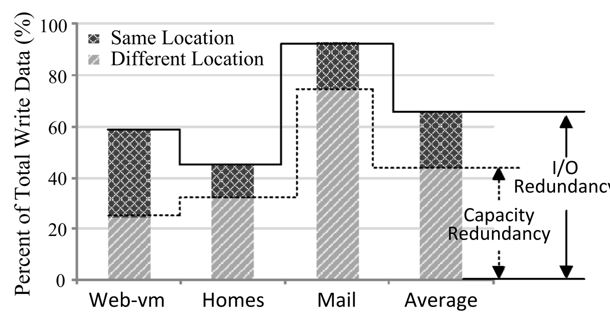


Fig. 2. I/O redundancy vs. capacity redundancy for the three applications (Web-vm, Homes and Mail).

redundancy is noticeably higher than the capacity redundancy. *I/O redundancy* in this context means that data blocks accessed by different write requests on the critical I/O path contain the same content. Differently, the capacity redundancy is analyzed from the static data stored on the storage devices [11,24,25]. For the redundant write data, only the write data addressed to different locations may contribute to the capacity savings. Fig. 2 shows the percentages of the write data that is addressed to the same locations and to the different locations with the same content. The latter indicates the data redundancy targeted by the capacity-oriented deduplication schemes, while the combination of the former and the latter signifies the I/O redundancy. It is clear that the I/O redundancy is noticeably higher than the capacity redundancy, by an average of 21.9% for the three traces, due to the additional repeated accesses to the same locations on the storage devices as a result of the temporal locality of user requests. This new finding implies that the on-line deduplication is much more effective in reducing the I/O traffic than the off-line deduplication [25] for primary storage workloads.

In the data deduplication systems, the effects of the loss of a chunk can be measured by the number of files that are inaccessible as a result of this loss. Accordingly, we measure the importance of a chunk by counting the number of files that depend on the chunk (*i.e.*, the reference count). Fig. 3 shows an example of the reference count in the data deduplication systems. The reference characteristics has high locality in primary storage systems [26–28]. For example, previous study on the 113 VMs from 32 VMware ESX hosts reveals that a high degree of similarity among the VM disk

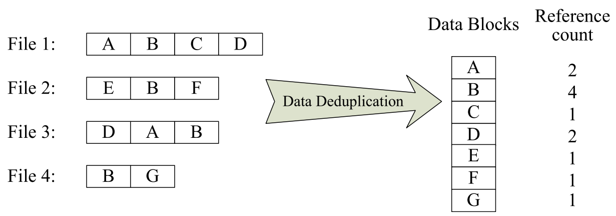


Fig. 3. The data reference characteristics in the data deduplication system.

images results in more than 80% of reduction in the storage space. Moreover, about 6% of the unique data chunks are referenced more than 10 times and some data chunks are referenced over 100,000 times at the 4 kB block size [27]. Our analysis of the Mail workload indicates that the unique data chunks referenced more than 5 times amount to about 4.5% of all data chunks but the accesses on these unique data chunks account for over 42.2% of the total accesses [28]. These unique data chunks with higher reference count are likely to be accessed much more frequently, because they are shared by many files or data blocks. Moreover, the access latency of these unique data chunks with high reference count affects the system performance directly.

The availability of the cloud storage services becomes increasingly important in face of the vendor lock-in problem associated with a single cloud storage provider. Cloud-of-Clouds is a feasible solution to address the problem. However, how to distribute the data blocks across multiple cloud storage providers is non-trivial due to the diversity of different cloud storage providers. Replication-based and erasure-code-based schemes have their own advantages and disadvantages. Moreover, knowing and utilizing the workload characteristics is important for the storage system design. Previous studies have shown that the data redundancy does exist in the primary storage systems in the cloud. They also found that the data redundancy has high access locality that is shown as the reference count characteristics. Thus, the data blocks with high reference count should be stored with a replication-based scheme and other data blocks should be stored with an erasure-code-based scheme for the performance and cost efficiency considerations. These important observations, combined with the urgent need to address the availability problem of cloud storage systems, motivate us to propose DAC.

3. The design of DAC

In this section, we first outline the main design objectives of DAC. Then we present the DAC architecture overview, some design considerations and the prototype implementation.

3.1. The design objectives of DAC

The design of DAC aims to achieve the following three objectives.

- *Improving the availability of the cloud storage systems*—By redundantly distributing the user data in a Cloud-of-Clouds, the vendor lock-in problem is solved. With the data redundancy schemes of replication and erasure codes, the service unavailability problem caused by the outage of a single individual cloud storage provider is avoided.
- *Reducing the access latency of the user request*—By using the data deduplication to reduce the redundant data over the network, the utilization of the network bandwidth has been improved. Moreover, by exploiting the data reference characteristics to replicate data blocks with high reference count, the access latency is reduced significantly.

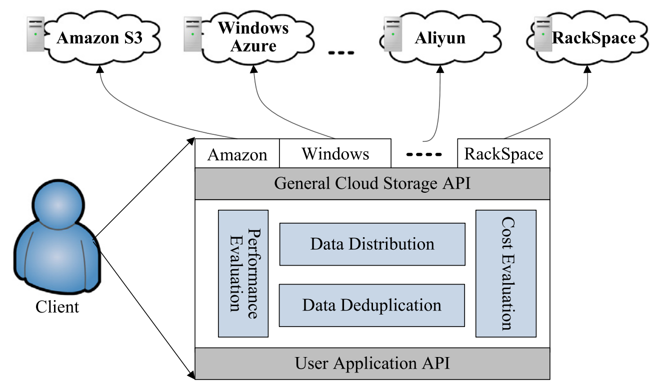


Fig. 4. System architecture of DAC.

- *Improving the cost efficiency*—Since DAC uses the data deduplication to eliminate the redundant data blocks, the overall storage efficiency is improved. Moreover, by reducing the data blocks transferred over the network, the access cost is also reduced in the cloud storage systems.

3.2. DAC architecture overview

Fig. 4 shows a system architecture overview of our proposed DAC in a Cloud-of-Clouds. Since more cloud storage services are provided by commercial cloud providers, the providers are not allowed to execute users' codes on the cloud storage side. As shown in Fig. 4, DAC resides on the client side and interacts with the cloud storages via their standard interfaces without any modifications. Thus, DAC can be easily applied to any cloud storage providers to use their cloud storage services.

DAC has four main functional modules: Data Deduplication, Data Distribution, Performance Evaluation and Cost Evaluation. The *Data Deduplication* module is responsible for dividing the incoming data into multiple data blocks and calculating their hash values (SHA1 or MD5) to eliminate the redundant data blocks. Moreover, the reference values of the data blocks are also updated. Based on the reference values of the data blocks, the *Data Distribution* module decides which redundancy scheme should be used for the incoming data, and distributes the data blocks to the corresponding cloud storage providers. The *Performance Evaluation* and the *Cost Evaluation* modules are responsible for evaluating the cloud storage services from the perspectives of performance and cost. The performance characteristics are mainly described in terms of the access latency while the cost characteristics of the cloud storage providers are summarized in Table 1 in Section 2. These evaluation results will enable the Data Distribution module to select the appropriate cloud storage providers.

3.3. Data deduplication

The data deduplication process is intended to be transparent to the upper layer users and applications. By definition, the data deduplication in the primary storage systems is designed for optimal performance rather than possible low cost in the secondary storage systems [12]. The design criteria for the deduplication-based primary storage systems is to improve the performance, thus leading to the careful system design considerations to alleviate any operations that can negatively impact the performance.

The data deduplication in DAC includes four steps: data splitting, hash computing, index querying and index updating, as shown in Fig. 5. DAC uses the fixed-sized chunking algorithm to split the data to reduce the computing overhead. Though the variable-sized chunking can exploit much more data redundancy

Please cite this article in press as: S. Wu, et al., DAC: Improving storage availability with Deduplication-Assisted Cloud-of-Clouds, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.02.001

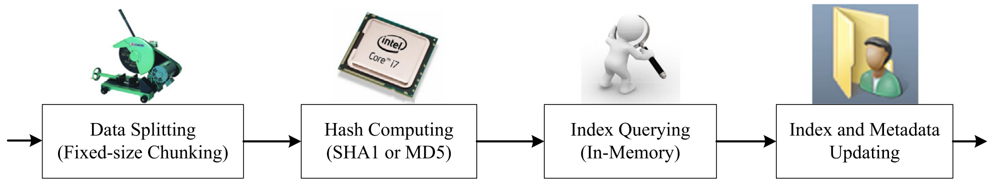


Fig. 5. The workflow of the data deduplication in DAC.

than the fixed-sized chunking, the extra computing overhead of the former is also more significant than the latter. The previous studies also suggest that fixed-size chunking method is suitable to provide a better trade-off between performance and data redundancy in primary storage systems [10,28,26,12]. DAC also uses the SHA1-based or MD5-based hash computing algorithms that are already embedded as an independent module in the Operating Systems because it locates in the client side. Moreover, since the whole data is small for each Client, the hash index is stored in the memory for fast querying and updating. In order to prevent data loss caused by the power failure, the metadata and newly updated index are stored on persisted storage devices, such as HDDs and SSDs. During the index updating process, the reference count of the data chunk is also updated. Overall, DAC takes the performance as the prior design principle in the data deduplication process.

3.4. Data distribution

Our previous study has shown that the hybrid data placement is better than the pure replication-based or erasure-code-based schemes [9]. Thus, the key idea of DAC is exploiting the reference characteristics to choose either the replication-based scheme or erasure-code-based scheme to distribute the data among multiple cloud storage providers. In deduplication-based storage systems, the data blocks with high reference values are critical to the system performance. The reason is that these data blocks are referenced by many files, which indicates much higher probability to be accessed frequently. DAC uses the replication-based scheme to store these data blocks to exploit the access frequency, as shown in Fig. 6. For the other data blocks that occupy a disproportionally large storage capacity, DAC uses the erasure-code-based scheme to store them. Initially, all the data blocks are written with the erasure-code-based scheme. Along with the written data increased, the reference count values of the data blocks are also changed. Upon the reference count of a data block reaches the preset reference threshold, the data block will be replicated. However, how to determine the reference threshold is nontrivial as it sensitively depends on the applications. We have conducted the sensitivity experiments to investigate the reference threshold, as shown in Section 4. The reference threshold is determined and set when the system is build.

Data distribution method determines the recovery process. DAC utilizes hybrid data distribution methods. Thus, the recovery workflow in DAC should follow the data layout schemes. However, an outage of the cloud storage service is different from a disk failure in a disk array [29,30]. The former results in a period of time during which the cloud storage service is unavailable. The period may be hours and up to days. However, most outages will return to the normal state eventually. Thus, the recovery in case of the service outage in DAC includes two phrases: (1) the reconstruction on-demand during the unavailable period and (2) the consistency update upon the service's return to the normal state.

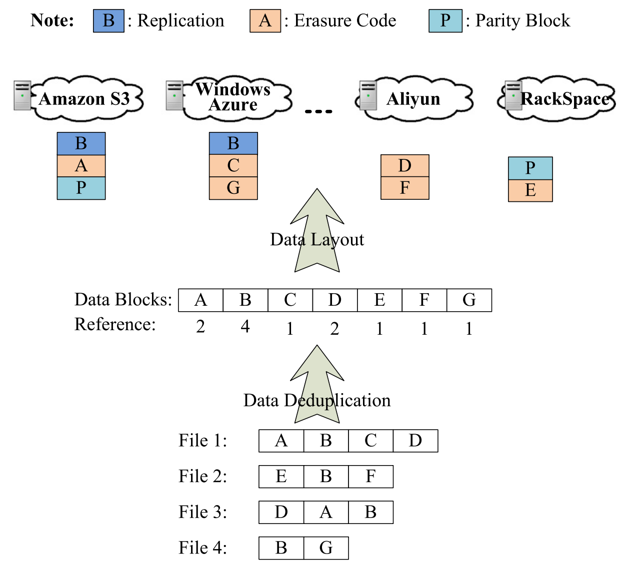


Fig. 6. Data distribution based on the reference characteristics.

3.5. Data consistency

The data consistency in DAC means that (1) the write data must be reliably stored in the cloud storage providers, (2) the index data and metadata must be reliably stored in the persisted storage, and (3) the user read requests must fetch the integrated data.

First, the write data must be reliably stored in the cloud storage providers. Since DAC uses either replication or erasure codes to distribute the data blocks, each write request will produce multiple write operations which involve multiple cloud storage providers. DAC makes sure that all the data blocks and the corresponding parity blocks are completely written in the cloud storage providers when performing a write request. Until all the write operations to the cloud storage providers are completed, the write request is completed. Otherwise, the uncompleted write operations will be re-performed.

Second, to prevent the loss of the index data and metadata in the event of a power supply failure or a system crash, DAC stores them in the persisted storage devices, such as HDDs and SSDs. Since the size of the hash index is generally small, it will not incur significant hardware cost. Moreover, in order to improve the deduplication performance, DAC stores the whole hash index in the memory for fast querying. Only the newly updated hash index will be flushed to the persisted storage devices periodically.

Third, since the whole data for a read request may be stored on multiple cloud storage providers or different locations within an individual cloud storage provider, each read request is first checked in the metadata to determine whether it should be serviced by the former or the latter to keep the fetched data always integrated. After all the data blocks are fetched, the requested data will be reconstructed and returned to the upper layer.

Table 2
The characteristics of the three traces.

Traces	Write ratio	I/Os	Average request size
Mail	78.5%	328,145	40.8 kB
Homes	80.5%	64,819	13.1 kB
Web-vm	69.8%	154,105	14.8 kB

4. Performance evaluation

In this section, we first describe the prototype implementation of our proposed DAC scheme and the experimental setup. Then we evaluate the performance of DAC through extensive trace-driven experiments.

4.1. Prototype implementation

The DAC scheme is embedded in our previous HyRD prototype [9] as an independent module on the client side. In the experiments, the fix-size chunking method is used in DAC scheme. To interact with multiple cloud storage providers, we have implemented a middleware of general cloud storage API, short for GCS-API. The GCS-API middleware hides the complexity of the cloud storage providers at the system level. Moreover, with such middleware, it is easy to add new cloud storage providers to the DAC system.

Currently, each cloud storage service is modeled as a passive storage functional entity that supports five functions: List (lists the files of a container in the cloud), Get (reads a file), Create (creates a container), Put (writes or modifies a file in a container) and Remove (deletes a file). By passive storage functional entity, we mean that no operations other than what is needed to support the aforementioned five functions are executed. To easily use the various cloud storage services, DAC uses the REpresentational State Transfer APIs (short for RESTful APIs) to perform the operations. The RESTful APIs are application program interfaces (APIs) that use the HTTP requests to perform the above five functions and explicitly take the advantage of the HTTP methodologies that are defined by the RFC 2616 protocol. Besides the above five functions, the evaluation module in DAC will directly interact with the individual cloud storage providers to evaluate the corresponding values.

4.2. Experimental setup

Our experiments are conducted in a desktop PC (*i.e.*, the client) with an Intel i5-3470 3.2 GHz quad-core processor and 4 GB of DRAM, and with 1 Gigabit Ethernet connected to the China Education and Research Network [31]. Currently, our evaluations use the following four cloud storage providers in their default configurations: Amazon [14], Windows Azure [15], Aliyun [16] and RackSpace [17]. Table 1 shows the monthly price plans for the four providers as of Sep. 10th 2014. For all the cloud providers, we use the prices from the first chargeable usage tier in the China region.

In the experiments, we used three FIU traces which contain the hash values for trace-driven evaluations [23]. The three traces were collected from three production systems, a virtual machine running two web-servers (web-vm), a file server (homes) and an email server (mail). They cover a duration of three weeks. We used the 15th day of the three traces for our evaluations and the characteristics of the three traces are summarized in Table 2. Moreover, we compare DAC with three other Cloud-of-Clouds schemes: RACS [1], DuraCloud [8] and HyRD [9].

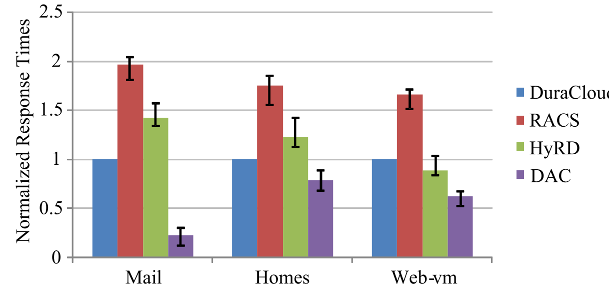


Fig. 7. The normalized average response times.

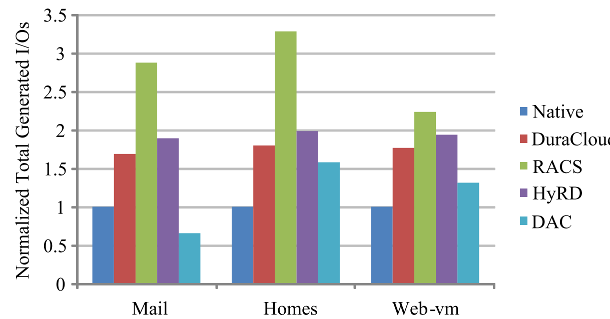


Fig. 8. The normalized total generated I/O requests.

4.3. Performance results and analysis

Fig. 7 shows the normalized average response times for difference Cloud-of-Clouds schemes driven by the three traces. We can see that DAC has the lowest average response times than the other three schemes. Compared with DuraCloud, DAC reduces the response times by 77.4%, 22.1%, and 37.8% for the Mail, Homes, and Web-vm traces, respectively. Moreover, we can see that the performance of the replication-based data layout is better than the erasure-code-based scheme and the hybrid schemes, such as RACS and HyRD. The reason is that all the three traces are write-intensive workloads and most requests in them are small. These small write requests will incur significant write amplification problem in the RACS and HyRD schemes, thus increasing the response times. For the Web-vm trace that has large request sizes, we can see that the HyRD scheme outperforms the DuraCloud and RACS schemes since it integrates the advantages of both replication and erasure codes.

The reason that DAC reduces the response times is twofold: (1) The data deduplication can reduce the redundant I/O requests from the source node. Fig. 8 shows the normalized total I/O requests that are generated in the Cloud-of-Clouds. The Native system indicates that its total I/O requests are the trace itself contains. From Fig. 8, we can see that the DAC scheme only slightly increases the extra I/O requests, or even fewer I/O requests in the Mail trace. The reason is that all these three traces have moderate to high data redundancy. Especially, the mail trace contains a lot of redundant I/O requests. In contrast, the replication-based and erasure-code-based schemes will incur the write amplification problem, thus significantly increasing the total I/O requests. Obviously, reducing the I/O requests over the network will directly reduce the average response times. (2) DAC exploits the reference characteristics to distribute the hot data blocks with the replication-based scheme, which further reduces the response times. The DAC scheme is much more simple and effective than the HyRD scheme.

In order to evaluate the sensitivity of the reference values to the system performance, we also conduct experiments with different reference values, as shown in Fig. 9. We can see that, the lower reference values, the lower response times. The reason is

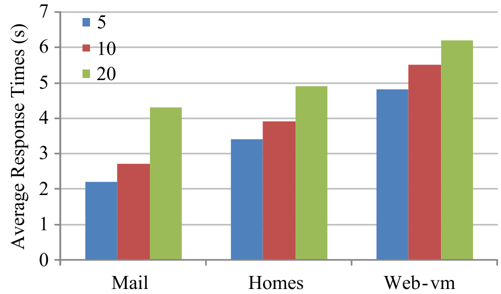


Fig. 9. The sensitivity study of the different reference values.

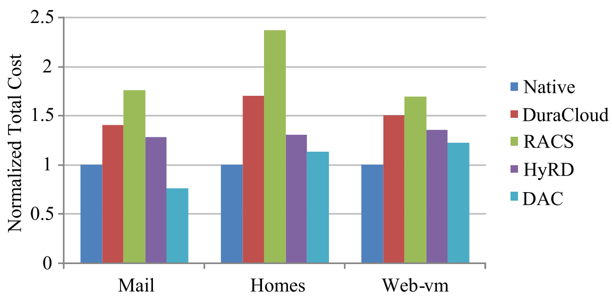


Fig. 10. The normalized total cost for the trace replay.

that with the lower reference value, much more data blocks will be stored with the replication-based scheme, which will reduce the response times. However, with a lower reference value, the storage efficiency will be degraded, thus increasing the storage cost. The reference count characteristics of the three traces are also investigated in the previous studies [23,28]. In the DAC design, the reference threshold value is configurable by the administrators to make a better design tradeoff between the performance and cost efficiency.

Fig. 10 shows the normalized total cost for the different Cloud-of-Clouds schemes based on the Cloud prices that are shown in Table 1 driven by the three traces for one day. The Native system shows that the total cost for a single cloud storage provider (*i.e.*, Aliyun). We can see that the total cost of the single cloud storage provider is the lowest. The reason is that for a single cloud storage provider, none extra I/O requests and redundant data are generated. However, the data availability and security of a single cloud storage provider is much lower than the Cloud-of-Clouds schemes. In contrast, the DAC scheme achieves the lowest total cost among all the Cloud-of-Clouds schemes since it reduces significant I/O requests and storage capacity. We also see that the RACS scheme has the highest total cost due to the increased total generated I/O requests and data redundancy that are shown in Fig. 8.

5. Related work

As the cloud storage becomes popular and cost efficient, more and more organizations and individual users have moved or will move their data to the cloud. Besides the performance and security, the availability of the cloud storage service is becoming increasingly more important for the users. The notion of the Cloud-of-Clouds is an effective approach to addressing the availability issue that is caused by the service outages of a single cloud storage provider.

Several systems are proposed for the Cloud-of-Clouds. RACS [1] uses the erasure codes to mitigate the vendor lock-in problem encountered by a user when switching the cloud vendors. It transparently stripes the data across multiple cloud storage providers with the RAID-like algorithm used by disks and file systems. HAIL [6] provides the integrity and availability guarantees for the stored data. It allows a set of servers to prove to a client that a stored file is intact and retrievable by the approaches adopted from the cryptographic and distributed-system communities. D2DRR (DeDuplication-aware Deficit Round Robin) [33] is proposed to provide flexible reconfiguration and fast deduplication for Avionics Full DupleX (AFDX) networks. Moreover, it offers the salient features of simplicity and ease of use by leveraging conventional DRR with the functionality improvements upon deduplication. NCCloud [7] achieves the cost-effective repair for a permanent single-cloud provider failure to improve the availability of the cloud storage services. It is built on top of the network-coding-based storage schemes called regenerating codes with an emphasis on the storage repair, excluding the failed cloud in repair.

The above systems are all based on the erasure codes or network codes. In contrast, DuraCloud [8] utilizes the replication to copy the user content to multiple different cloud storage providers to provide better availability. Moreover, it ensures that all copies of the user content remain synchronized. However, users will pay more money for the additional storage space and bandwidth required by DuraCloud. DEPSKY [2] improves the availability and confidentiality of the commercial storage cloud services by building a Cloud-of-Clouds on top of a set of storage clouds, combining the Byzantine quorum system protocols, cryptographic secret sharing, replication and the diversity provided by the use of several cloud providers. All the data stored in a single cloud storage provider will be exposed once the provider is maliciously attacked by the hackers. If the user data can be striped and stored in multiple cloud storage providers, the data integrity is destroyed from the viewpoint of a single cloud storage provider. Thus, the Cloud-of-Cloud diversity ensures confidentiality and integrity of outsourced data against outsider attacks, as long as a tolerable number of clouds are uncompromised.

Different from these approaches, the HyRD [9] scheme integrates both the replication and erasure codes into the Cloud-of-Clouds. It takes the workload characteristics and the diversity of the cloud storage providers, specially the file sizes, into the design of the redundant data distribution strategy so that the advantages of both the replication and erasure codes are exploited while their disadvantages are alleviated. As a result, the performance, storage efficiency and availability are improved. However, the transferred data blocks over the network have not reduced, especially for the redundant data blocks that are repeated transferred. In contrast, our proposed DAC integrates the data deduplication technique into the Cloud-of-Clouds to reduce the number of data blocks and distribute the data blocks with both replication and erasure codes by exploiting the data reference characteristics.

Table 3 summarizes the state-of-the-art data distribution schemes in the Cloud-of-Clouds. In general, replication provides better performance while erasure codes provide better storage efficiency. However, the former imposes extremely high bandwidth and storage overhead while the latter does not provide the robustness and expects high access performance in the Cloud-of-Clouds particularly for large files. It therefore hints at the possibility of a certain combination of the two, which tries to retain their respective advantages and hide their disadvantages to provide the most appropriate redundant data distribution scheme in the Cloud-of-Clouds. Moreover, none of the existing Cloud-of-Clouds schemes incorporates the data deduplication technique to reduce the redundant I/O traffic and improve the overall system availability.

6. Conclusion

Depending on a single cloud storage provider has the inherent vendor lock-in problem that can potentially cost the user deeply. In this paper we propose a Deduplication-Assisted primary storage

Table 3
Comparisons between DAC and the state-of-the-art schemes.

Scheme	Distribution	Deduplication	Performance	Cost
RACS [1]	Erasure codes	No	Low for small updates	Moderate
DuraCloud [8,32]	Replication	No	Low for requests with large size	High
DepSky [2]	Replication	No	Low for requests with large size	High
NCCloud [7]	Network codes	No	Low for small updates	Moderate
HyRD [9]	Replication and erasure code	No	Moderate	Moderate
DAC	Replication and erasure code	Yes	High	Low

system in the Cloud-of-Clouds, short for DAC. DAC eliminates the redundant data blocks in the cloud computing environments and distributes the data among multiple independent cloud storage providers by exploiting the data reference characteristics. In DAC, the data blocks are stored in multiple cloud storage providers by combing the replication and erasure code schemes. To better utilize the advantages of both the replication and erasure code schemes and exploit the reference characteristics in the data deduplication process, the high referenced data blocks are stored with the replication scheme and the other data blocks are stored with the erasure code scheme. The experiments conducted on our lightweight prototype implementation of DAC show that DAC improves the performance and cost efficiency significantly compared with the existing schemes.

DAC is an ongoing research project and we are currently exploring several directions for the future research. First, we will incorporate the reliability analysis into the DAC design by providing multiple replicas (more than two) for the data blocks with high reference count. Second, we will incorporate the data encryption into the DAC design to further improve the security of cloud storage systems. Since deduplication exploits the identical content and encryption attempts to make all contents appear random, the same contents encrypted with two different keys denotes to different ciphertexts [34,35]. It is important to provide secured data deduplication in cloud storage systems.

Acknowledgments

We thank the SyLab in FIU for providing us the I/O traces. We are also grateful to all the members of ASTL group for their continuous support and discussion. This work is supported by the National Natural Science Foundation of China under Grant No. 61100033, No. 61472336 and No. 61402385, National Key Technology R&D Program Foundation of China (No. 2015BAH16F02), Fundamental Research Funds for the Central Universities (No. 20720140515).

References

[1] H. Abu-Libdeh, L. Princehouse, H. Weatherspoon, RACS: a case for cloud storage diversity, in: Proceedings of the ACM Symposium on Cloud Computing, SOCC'10, 2010, pp. 229–240.
[2] A. Bessani, M. Correia, B. Quaresma, F. André, P. Sousa, DepSky: Dependable and secure storage in a cloud-of-clouds, in: Proceedings of the 6th European Conference on Computer Systems, EuroSys'11, 2011, pp. 31–46.
[3] The Worst Cloud Outages, 2014. http://www.cio.com/category/cloud-computing/.
[4] Nirvanix Files for Chapter 11 Bankruptcy, 2013. http://blogs.wsj.com/venturecapital/2013/10/01/nirvanix-files-for-chapter-11-bankruptcy/.
[5] Tim Cook Says Apple to Add Security Alerts for iCloud Users, the Wall Street Journal, September 2014, 2014. http://online.wsj.com/articles/tim-cook-says-apple-to-add-security-alerts-for-icloud-users-1409880977.
[6] K.D. Bowers, A. Juels, A. Oprea, HAIL: A high-availability and integrity layer for cloud storage, in: Proceedings of the 16th ACM Conference on Computer and Communications Security, CCS'09, 2009, pp. 187–198.
[7] Y. Hu, H. Chen, P. Lee, Y. Tang, NCCloud: Applying network coding for the storage repair in a cloud-of-clouds, in: Proceedings of the 10th USENIX Conference on File and Storage Technologies, FAST'12, 2012, p. 21.
[8] DuraCloud Project, 2015. http://www.duracloud.org/.
[9] B. Mao, S. Wu, H. Jiang, Improving storage availability in cloud-of-clouds with hybrid redundant data distribution, in: Proceedings of the 29th IEEE International Parallel & Distributed Processing Symposium, IPDPS'15, 2015, pp. 633–642.
[10] C. Ng, M. Ma, T. Wong, P.P.C. Lee, J.C.S. Lui, Live deduplication storage of virtual machine images in an open-source cloud, in: Proceedings of ACM/IFIP/USENIX 12th International Middleware Conference, Middleware'11, 2011, pp. 80–99.
[11] D. Meister, J. Kaiser, A. Brinkmann, T. Cortes, M. Kuhn, J. Kunkel, A study on data deduplication in HPC storage systems, in: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, SC'12, 2012, pp. 1–11.
[12] B. Mao, H. Jiang, S. Wu, L. Tian, POD: Performance oriented I/O deduplication for primary storage systems in the cloud, in: Proceedings of the 28th IEEE International Parallel & Distributed Processing Symposium, IPDPS'14, 2014, pp. 767–776.
[13] A. Thomson, D.J. Abadi, CalvinFS: Consistent WAN replication and scalable metadata management for distributed file systems, in: Proceedings of the 13th USENIX Conference on File and Storage Technologies, FAST'15, 2015, pp. 1–14.
[14] Amazon S3. http://aws.amazon.com/s3/.
[15] Windows Azure. http://www.windowsazure.cn/zh-cn/.
[16] Aliyun Open Storage Service, 2015. http://www.aliyun.com/.
[17] RackSpace. http://www.rackspace.com/cn/.
[18] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, M. Zaharia, Above the clouds: A Berkeley view of cloud computing. Technical Report No. USB/EECS-2009-28, EECS Department, University of California, Berkeley.
[19] F. Jrad, J. Tao, I. Brandic, A. Streit, SLA enactment for large-scale healthcare workflows on multi-cloud, Future Gener. Comput. Syst. 43–44 (2015) 135–148.
[20] J. Gahm, J. Mcknight, Medium-size Business Server & Storage Priorities, Enterprise Strategy Group.
[21] S. Wu, H. Jiang, D. Feng, L. Tian, B. Mao, Improving availability of RAID-structured storage systems by workload outsourcing, IEEE Trans. Comput. 60 (1) (2011) 64–79.
[22] EMC Presents Disaster Recovery Survey 2013, 2013. http://emea.emc.com/microsites/2011/emc-brs-survey/index.htm.
[23] R. Koller, R. Rangaswami, I/O deduplication: Utilizing content similarity to improve I/O performance, in: Proceedings of the 8th USENIX Conference on File and Storage Technologies, FAST'10, 2010, pp. 211–224.
[24] D.T. Meyer, W.J. Bolosky, A study of practical deduplication, in: Proceedings of the 9th USENIX Conference on File and Storage Technologies, FAST'11, 2011, pp. 1–14.
[25] A. El-Shimi, R. Kalach, A. Kumar, A. Oltean, J. Li, S. Sengupta, Primary data deduplication—large scale study and system design, in: Proceedings of the 2012 USENIX Annual Technical Conference, USENIX ATC'12, 2012, pp. 285–296.
[26] K. Srinivasan, T. Bisson, G. Goodson, K. Voruganti, iDedup: Latency-aware, inline data deduplication for primary storage, in: Proceedings of the 10th USENIX Conference on File and Storage Technologies, FAST'12, 2012, pp. 299–312.
[27] A.T. Clements, I. Ahmad, M. Vilayannur, J. Li, Decentralized deduplication in SAN cluster file systems, in: Proceedings of the 2009 USENIX Annual Technical Conference, USENIX ATC'09, 2009, pp. 101–114.
[28] B. Mao, H. Jiang, S. Wu, Y. Fu, L. Tian, Read performance optimization for deduplication-based storage systems in the cloud, ACM Trans. Storage 10 (2) (2014) 1–22.
[29] S. Wu, H. Jiang, B. Mao, IDO: Intelligent data outsourcing with improved RAID reconstruction performance in large-scale data centers, in: Proceedings of the 26th USENIX Large Installation System Administration, LISA'12, 2012, pp. 17–32.
[30] S. Wu, H. Jiang, B. Mao, Proactive data migration for improved storage availability in large-scale data centers, IEEE Trans. Comput. 64 (9) (2015) 2637–2651.
[31] China Education and Research Network, 2015. http://www.edu.cn/english_1369/index.shtml.
[32] E. Allen, C.M. Morris, Library of congress and duracloud launch pilot program using cloud technologies to test perpetual access to digital content, in: Library of Congress, News Release, 2009. http://www.loc.gov/today/pr/2009/09-140.html.
[33] Y. Hua, X. Liu, Scheduling heterogeneous flows with delay-aware deduplication for avionics applications, IEEE Trans. Parallel Distrib. Syst. 23 (9) (2012) 1790–1802.
[34] Y. Yu, L. Niu, G. Yang, Y. Mu, W. Susilo, On the security of auditing mechanisms for secure cloud storage, Future Gener. Comput. Syst. 30 (2014) 127–132.
[35] M. Li, C. Qin, P.P.C. Lee, CDStore: Toward reliable, secure, and cost-efficient cloud storage via convergent dispersal, in: Proceedings of the 2015 USENIX Annual Technical Conference, USENIX ATC'15, 2015, pp. 111–124.

Please cite this article in press as: S. Wu, et al., DAC: Improving storage availability with Deduplication-Assisted Cloud-of-Clouds, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.02.001

Suzhen Wu received the B.E. and Ph.D. degrees in Computer Science and Technology and Computer Architecture from Huazhong University of Science and Technology, in 2005 and 2010 respectively. She is an associate professor at Computer Science Department of Xiamen University since August 2014. Her research interests include computer architecture and storage system. She has over 30 publications in journal and international conferences including IEEE-TC, IEEE-TPDS, ACM-TOS, USENIX FAST, USENIX LISA, IPDPS, ICCD, MASCOTS, and ICPADS. Dr. Wu is a Member of IEEE and ACM.

Kuan-Ching Li is a Professor of Department of Computer Science and Information Engineering (CSIE) of Providence University in Taiwan. He is also a guest professor of "Minjiang Scholars" in Xiamen University. He has been involved actively in conferences and workshops as a program/general/steering conference chairman positions, numerous conferences and workshops as a program committee member, and has organized numerous conferences related to high-performance computing and computational science and engineering. He is a Fellow of IET society, a senior member of the IEEE and a member of TACC.

Bo Mao received the B.E. degree in Computer Science and Technology in 2005 from Northeast University; and the Ph.D. degree in Computer Architecture in 2010 from the Huazhong University of Science and Technology. His research interests include storage system, Cloud computing and Big Data. He is an associate professor at the Software School of Xiamen University. He has over 30 publications in international journals and conferences including IEEE-TC, IEEE-TPDS, ACM-TOS, USENIX FAST, IPDPS, USENIX LISA, ICCD, MASCOTS, Cluster, and ICPADS. Dr. Mao is a Member of IEEE, ACM and USENIX.

Minghong Liao is a Professor of Software School of Xiamen University. He received his Ph.D. degree in Computer Science in 1993 from Harbin Institute of Technology of China. His research interests include Intelligent Network, Pervasive Computing and Computer Graphics.

Please cite this article in press as: S. Wu, et al., DAC: Improving storage availability with Deduplication-Assisted Cloud-of-Clouds, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.02.001