DC Store: A Deduplication -Based Cloud-of-Clouds Storage Services

Spoorthy M Angadi[1], Sanjana Girish[2], Pavithra J C[3], Priyanka[4], Manjula N[5]

[1, 2, 3, 4, 5]Information Science and Engineering, Global Academy Of Technology, Bengaluru, 560098, India

Abstract- The cloud storage systems are becoming more popular and all the organizations are moving towards the direction of cloud for storing the data. Moving all the data in one cloud causes problems like vendor lock-in, increased service costs, and data availability. In this paper, we are introducing the DCStore, a cloud-of-clouds storage services designed for an organization to outsource their data into the clouds. In order to overcome the above mentioned problems we combine three key techniques.First, DCStore will remove all the redundant data at the client-side to save storage using via application-aware chunking method. Second, in order to achieve high availability of data DCStore uses an inner-chunk based erasure coding scheme which distributes unique chunks across multiple clouds. Finally, a Container-based share management strategy is used for optimizing the performance.Load Balancing enables enterprise to manage workload demands or application demands by distributing resources among numerous computers, networks or server. Therefore, our experimental evaluations can improve performance, storage space and can achieve high availability of data.

Index Terms—Cloud Storage Service; Data Deduplication; Erasure Coding; High Availability; Cost Efficiency; Load Balancing.

I. INTRODUCTION

Cloud computing is basically where users who can outsource their computation and stores data/information to the cloud using the Internet. Much of the data stored in cloud like social networks, medical records which are highly sensitive and requires security. Privacy and security are thus very critical issues in cloud computing. Major thing is, the user should verify itself before initiating any transaction, and it must be protect that other users do not know the congruity of the user. The cloud can hold the user for storage with for different purpose, and likewise, the cloud itself liable for the services it provides. The integrity of the user who stores the information is also certify. The cloud is also vulnerable for server colluding attacks and information modification. The opponent can compromise storage servers in server scheme attack, so that server can change data files even though the servers are internally homogenous. Encryption on data is required to provide secure data storage. However, the data is often changed and this dynamic property needs to be taken into considered while designing effective secure storage techniques. Accountability is not responsible for the clouds nor should users not accept any operations requested or performed. It is required to have log of the transactions that are performed. Parallel and distributed systems defines, a number of nodes connected in the network, it can be defined as a collection of processing elements that communicate and collaborate to achieve goal. The node to node communication is done through messages.

As a result, Data De-duplication has received a broad attention in both academia and industry, and gradually it has become a commodity component in data-intensive storage systems and products (especially archival storage systems)[3], [4], [5], [6]. Data De-duplication refers to approaches that use lossless data compression schemes to minimize the duplicated data at inter-file level. It divides each file into a number of non-overlapping chunks and storing only unique chunks on storage devices. However, as a lot of duplications are intended by users and applications for increased reliability or availability, especially in archival storage systems, data should be preserved over long time periods. This requires that the deduplication storage systems provide reliability comparable to other high-available systems. Erasure coding (EC) is a data protection and storage process through which a data object is separated into smaller components/fragments and each of those fragments is encoded with redundant data padding. EC transforms data object fragments into larger fragments and uses the primary data

object identifier to recover each fragment.Erasure coding is also known as forward error correction (FEC). After chunking, the encoder module divides each unique chunk into k same-sized data shares encode them into n-k parity shares through (n, k) erasure coding. Unfortunately, by storing duplicated data chunks just once, de-duped system achieves storage utilization at the cost of error resilience or having potential risk to reduce reliability. As dependencies are introduced between files that share the same chunks, if such a shared chunk is lost, a disproportionately large amount of data becomes inaccessible because of the unavailability of all the files that share this chunk. [8] (If the value of a chunk were measured in the amount of file data (user data) that would be lost on losing a single chunk, then in a traditional data store, each chunk would be equally valuable. While in a de-duplication based store, a chunk with a higher commonality would be more valuable).

II.BACKGROUND AND MOTIVATION

In this session, we present some necessary background, including data deduplication and erasure coding, to motivate our study.

A.Datadeduplication: Data deduplication is an efficient data reduction technology to reduce network bandwidth and storage overhead by eliminating duplicate data. As shown in Figure1, a typical data type ded uplication system splits the input data stream into multiple chunks which will be identified with a unique hash identifier. The system then removes the duplicate data chunks by the identifiers and then only the original copy gets stored or transferred this is done to achieve the goal of saving storage space or network bandwidth.

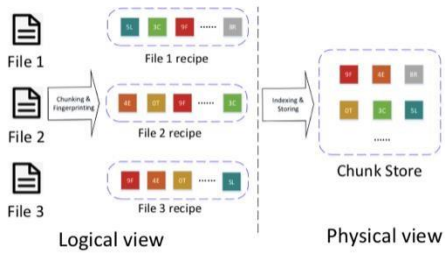

Fig1: Overview of deduplication process

There are two approaches that is used to split the input file. The first one is called static chunking it splits the data into chunks of a fixed size. The main advantage of this technique lies in the fact that it offers a very high throughput and is easy to implement. We use Dropbox [13], this approach has a major drawback: when the bytes are added at the beginning of a file, all the chunk boundaries are shifted [14], i.e., all the chunks after addition will have a different hash sums and will become obsolete for that file. The second approach is called as content-defined chunking (CDC) and eliminates the problems caused due to the first approach. It was proposed by Muthitacharoen et al. for the Low Bandwidth File System (LBFS) [15]. With content-defined chunking, the system moves a sliding window across the data and hashes the windowed data in each step, using Rabins fingerprinting method. A new chunk boundary is found if the hash value satisfies some pre-defined condition.The chunking methods discussed above do not make use of the file characteristics of the underlying data. If the chunking method understands the data stream of the file, the deduplication method can provide the best redundancy detection ratio and throughput because this method could set boundaries more natural than other algorithm methods [16].

B.Erasure Coding: In a Cloud-of-Clouds storage services, erasure coding allows client to mask temporary outages and get higher data availability by adding chunk redundancies across multiple providers. As shown in the Figure 2, a (k, r) erasure coding encodes k data blocks and generates r parity blocks are sufficient to decode the original k data blocks. In many storage systems [11], [17]-[19], erasure coding is applied across fixed-size objects: k objects are encoded to generate r additional objects. Read requests to an object are served from the original object unless it is missing. If the object is missing, reconstruction using parities incurs huge bandwidth overheads [20]. Besides, deduplicated storage systems need pack varied-size chunks into fixed-size objects, which may cause zero-byte padding thereby making the space utilization low [21].

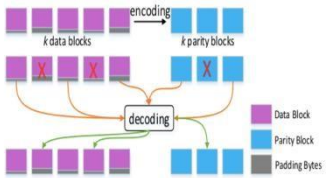

Fig2: Overview of Erasure Coding

In contrast, DCStore uses a inner-chunk based erasure coding scheme, which divides individual chunks into k fixed-sized data

block and creates r additional data blocks like Per-file RAID [22]. Read requests to an chunk are served by reading any k of the (k+r) splits and decoding them to recover the desired chunk. This approach provides two benefits which help customers save unnecessary expenses. First, the space overhead consumed by the zero-byte padding is reduced. Then, decoding the chunk using the parities upon a cloud failure does not incur any additional bandwidth overhead. It is also worth noting that the computational overhead of erasure coding is negligible compared to the overhead of reading, writing or sending data in the WAN.

III. THE DESIGN OF DC-STORE

Here in this part,we are introducing the DCStore architecture overview and design details.

A. Architecture Overview

It is the modern approach to the implementation of DCStore. We represent the Client and Server architecture in below figure. Organization : Here client is responsible for data chunking, encoding/decoding of chunks, finally the file recipe collection.Cloud Proxy server is managing data transferring between client and corresponding cloud storage in

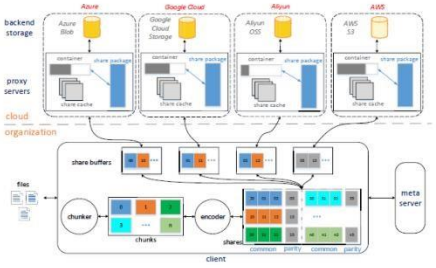

Fig3: System Architecture of DCStore

B.Application-Aware Chunking: This is implemented in DCStore client for betterment of the trade-off between deduplication ratio and deduplication throughput. This model divides files intotwo categories: static and dynamic files. We have advantage of editing in dynamic files whereas it is not possible in static files. Here static files split into fixed size chunks with ideal chunk size and split the dynamic files to variable size by using CDC based on Rabin fingerprinting. In there are two distinctive process, 1.Hashing in which fingerprints of data are generated. 2. Hash judgement in which fingerprints are compared against a given value to identify chunk - cut points.

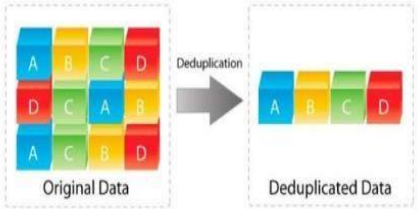

Fig4:DCStore data deduplication

C. Inner-Chunk Based Erasure Coding: In a storage system data gets stored on large number of commodity disks, it has high possibilities of disk failures which just cannot be considered as exceptions, but as rule. Hence storage system should be able to store redundant data so when disk fails it can be it can be accessed from other disks. When the redundancy has high degree its cost increases. In cloud storage concurrent cloud outages are rare so we fix n – k = 1. As chunk size are in KB when the size is increased it effects the process of concurrent share.Erasure coding tolerates number of disk failures with a greater efficiency. In this coding method

Maximum Distance Separable codes for example ReedSolomon codes will achieve optimal storage efficiency.

1.After chunking, the encoder module divides each unique chunk into k sized data shares encode them into n-k parity shares through(n,k) erasure coding.

2. These n shares gets sorted on different cloud service providers so high availability and resist against possible failures of particular providers can be achieved.

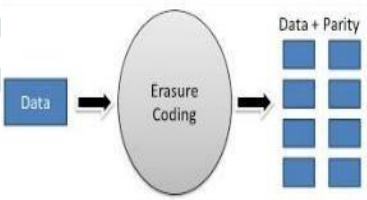

Fig5: Overview of Erasure Coding

D.Container – Based Share Management: Container in cloud computing is basically an approach to operating system virtualization which are packages that relay on virtual isolation to deploy and applications that access a shared operating system kernel without the use of virtual machines. Deduplication technology splits large files into small chunks which is stored into large storage units at client side to avoid high input/output overhead.Cloud services provide only limited support to operations, it may find network traffic overhead for client to access a file.

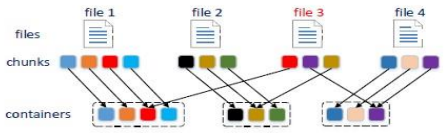

Fig6:Network Traffic Overhead We employ co-locating virtual machines at each cloud as a proxy server. Proxy servers are used to by DCStore client to upload a file by batches shares in client buffer and uploads the buffer when it is full.The Container module writes to cloud storage backend through internal network when container is filled with fixed size like for example 10MB. It increases the performace of data restoration.The container usage in online services provides cloud computing information security, availability and elasticity. Additional security is incorporated by data re-encryption. After authentication user can view desired information by receiving a key for decryption.We use a least recently used cache for the most recently accessed shares to optimize further performances.Clouds are sorted and accessed on access monetary cost in parallel. Proxy servers retrieves corresponding containers and only returns required shares to client.Since there is no billing for high-speed network between co-locating proxy server and cloud backend storage service. Our approach has limited overhead in expense and performance.

E. Load Balancing: Cloud load balancing is a type of load balancing that is performed in cloud computing .Cloud load balancing is a process of distributing work loads across multiple computing resources.Cloud load balancing reduces costs associated with document management systems and maximizes availability of resources.Load balancing refers to efficiently distributing incoming network traffic across a group of backend servers also knows as server pool.Functions: Distributes client requests or networks, loads efficiently across multiple networks. Types of load balancers:1.Elastic Load Balance supports Application load balancing,network load balancing,classic load balancing.2.Amazon ECS services can use either type of load balancer.3 Application load balancer are used to route HTTP/HTTPS traffic. A load balancer serves as the single point of contacts for clients .The load balancer distributes incoming application traffic across multiple targets ,such as EC2 instances in multiple availability zones . It acts as reverse proxy and distributes network or application traffic across number of servers, and are used to increase capacity and reliability of applications Benefits of using load balancing:

1.Improves responsiveness

2.Availability of applications

3.Distributes traffic

Load Balancing Techniques/Methods:

1.Round Robin :Asimple method load balancing servers.

2.Weighted Round Robin:Builds of simple round robin load balancing method 3.Source IP Hash

4.URL Hash

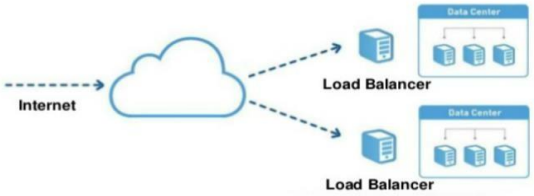

Fig7: Load Balancer

IV.EVALUATION

We evaluate the performance and the costs advantages of DCStore by both running real-world dataset and simulating on trace dataset.

A. Experimantal Setup

Client : Our experimental client runs in a virtual machine on Aliyun, which has a dual-core 2.50GHz Intel Xeon Platinum 8163 processors, 8GB RAM and a $0GB HDD. This virtual machine's bandwidth is limited to 100Mbps to simulate network bandwidth between organization to clouds. The machine runs Ubuntu 16.04.

Cloud-Side Proxy Servers: The proxy server in the cloudside have the similar configuration as the client machine, which has a dual-core processor, 8GB RAM and 400GB HDD.

Cloud Storage Services: In our experiments, we choose Amazon S3, Aliyun OSS, Microsoft Azure Blob and Google Cloud Storage as the backend storage services. **Datasets**: We use both real-world benchmark files and open source trace datasets to evaluate DCStore.

1) Random Generated Benchmark Files*:* We generate eleven sets of files with different duplication ratio from *0%* to *100%*. ***2) FSL Trace Dataset:***We use the FSL dataset [24] to simulate DCStore's cost-saving effect. This dataset is published by the File systems and Storage Lab at Stony Brook

University.Due to the large dataset size, we use the *Fslhomes*dataset in 2014, containing daily snapshots of 17 students' home directories from a shared network file system. The dataset is represented in 48-bit chunk fingerprints and corresponding chunk sizes obtained from content-defined chunking. B. Benefits

In this section, we evaluate both the performance and cost benefits of DC Store.

1)Performance Improvement: *Put* and *Get* are the two main operations in DCStore, among which *Put* is the onethat is relevant to deduplication while *Get* is not. This isbecause the corresponding content must be fetched from the clouds whether the redundant content is removed or not. As for a redundant Level *r%*, we call the first-uploaded file the uniquefile and the second-uploaded file the dup-file. The green horizontal line in the figure represents the upper bound of the network bandwidth, which is about 100Mbps. The red line represents the upload speed of unique files, which can achieve about 70Mbps, *70%* of the bandwidth. As the duplication ratio goes higher, the uploading of the dup-file becomes faster. When the duplication ratio comes to *30%* or higher, the speed can become faster than the native bandwidth. When the duplication ratio is *100%*, which means uploading two files with identical content, the speed can achieve nearly 800Mbps. Since no new content needs to be delivered to the cloud, the transfer speed at the *100%* point can represent the speed of the client-side chunking and encoding process.

2) **Storage Cost Saving:**To evaluate the cost-saving benefits, we run a simulation based on the FSL trace mentioned in

Section IV-A2. We combine all the users' trace data together,which is about 1.41TB in 168 days totally. From day #1 to day #168, we simulate the upload process for the modified files according to four different strategies, which are used bySingle Cloud, DuraCloud, RACS, and DCStore, respectively.The first three strategies simply upload the whole file when its modify-timestamp changes while DCStore only uploads thechanged chunks. We choose Google Cloud as the Single Cloud instance, which is the cheapest among the four cloud vendors. DuraCloud simply stores two copies of a single file. RACS do a (3,1) erasure-encoding for all the files without chunk-level deduplication.. The vertical axis is the accumulated cost. DuraCloud stores the most data so it costs the most, about 4970USD on the last day. RACS also provides high availability but still costs too much, about 3765USD on the last day. Storing all the data in the cheapest cloud costs lower than the former two, about 2975USD on the last day. But it may suffer from cloud outages. The blue line at the bottom is DCStore'scost. DCStore removes redundant data via chunk-level deduplication and provide high availability by performing erasure-encoding on chunks. The cost of DCStore is the lowest among these four strategies, about only 277 USD on the last day.

CONCLUSION:

Deduplication-based on Cloud-of-Clouds storage service practically addresses the reliability of cloud storage service. DCStore not only achieves fault tolerance of storage, but also achieves cost savings via client-side data deduplication, innerchunk encoding, and container-based share management strategy.Our lightweight prototype implementation of DCStore shows that DCStore improves the performanceand cost efficiency significantly compared with existing Cloud-ofClouds storage system. Load balancing helps in improving the system by shifting of workload among the processors.It also minimizes the the power consumption and maximize the user satisfaction forth service being offered. The goal of load balancing is to increase client satisfaction and maximize resource utilization and increase the performance of the cloud system thereby reducing the energy consumed and the carbon emission rate.

V.ACKNOWLEDGMENT

This work is supported by the National Key R&D Program of China (2016YFB1000105), the Science Fund for Creative Research Groups of China under Grant, No. 61421091

REFERENCES

[1] M. Greer, "Survivability and information assurance in the cloud," in Dependable Systems and Networks Workshops (DSN-W), 2010 International Conference on. IEEE, 2010, pp. 194–195.

[2] "Serious cloud outages," https://www.bvoip.com/blog/ the-10-biggest-cloud-outages-of-2018-so-far, 2018. [3] M. Naldi and L. Mastroeni, "Cloud

storage pricing: a comparison of current practices," in Proceedings of the 2013 international workshop on Hot topics in cloud services. ACM, 2013, pp. 27–34.

[4]"Duracloudproject,"http://www.duracloud.org/, 2019.

[5] H. Abu-Libdeh, L. Princehouse, and H. Weatherspoon, "RACS: a case for cloud storage diversity," in Proceedings of the 1st ACM symposium on Cloud computing. ACM, 2010, pp. 229–240.

[6] Y. Hu, H. C. Chen, P. P. Lee, and Y. Tang, "NCCloud: applying network coding for the storage repair in a cloud-ofclouds." in FAST, 2012, p. 21.

[7] W. Xia, H. Jiang, D. Feng, F. Douglis, P. Shilane, Y. Hua, M. Fu, Y. Zhang, and Y. Zhou, "A comprehensive study of the past, present, and future of data deduplication," Proceedings of the IEEE, vol. 104, no. 9, pp. 1681–1710, 2016.

[8] D. Bhagwat, K. Pollack, D. D. Long, T. Schwarz, E. L. Miller, and J.- F. Paris, "Providing high reliability in a minimum redundancy archival storage system," in null. IEEE, 2006, pp. 413–421.

[9] X. Li, M. Lillibridge, and M. Uysal, "Reliability analysis of deduplicated and erasure-coded storage," ACM SIGMETRICS Performance Evaluation Review, vol. 38, no. 3, pp. 4–9, 2011.

[10] S. Wu, K.-C. Li, B. Mao, and M. Liao, "DAC: improving storage availability with deduplication-assisted cloud-ofclouds," Future Generation Computer Systems, vol. 74, pp. 190–198, 2017.

[11] C. Liu, Y. Gu, L. Sun, B. Yan, and D. Wang, "RADMAD: High reliability provision for large-scale deduplication archival storage systems," in Proceedings of the 23rd international conference on Supercomputing. ACM, 2009, pp. 370–379.

[12] J. Y. Chung, C. Joe-Wong, S. Ha, J. W.-K. Hong, and M. Chiang, "CYRUS: Towards client-defined cloud storage," in Proceedings of the Tenth European Conference on Computer Systems. ACM, 2015, p. 17.

[13] I. Drago, M. Mellia, M. M Munafo, A. Sperotto, R. Sadre, and A. Pras, "Inside Dropbox: understanding personal cloud storage services," in Proceedings of the 2012 Internet Measurement Conference. ACM, 2012, pp. 481–494.

[14] C. Policroniades and I. Pratt, "Alternatives for detecting redundancy in storage systems data." in USENIX Annual Technical Conference, General Track, 2004, pp. 73–86.

[15] A. Muthitacharoen, B. Chen, and D. Mazieres, "A lowbandwidth network file system," in ACM SIGOPS Operating Systems Review, vol. 35, no. 5. ACM, 2001, pp. 174–187.

[16] H. Abu-Libdeh, L. Princehouse, and H. Weatherspoon, "RACS: A Case for Cloud Storage Diversity," Proc. ACM First ACM Symp. Cloud Computing (SoCC '10), 2010.

[17] R. Ahlswede, N. Cai, S.-Y.R. Li, and R.W. Yeung, "Network Information Flow," IEEE Trans. Information Theory, vol. 46, no. 4,pp. 1204-1216, July 2000.

[18] Amazon Web Services, "AWS Case Study: Backupify," http://aws.amazon. com/solutions/case-studies/backupify/, 2013.

[19] Amazon Web Services, "Case Studies," https://aws.amazon.com/solutions/case-studies/#backup, 2013.

[20] Amazon Web Services, "Amazon Glacier," http://aws.amazon.com/glacier/, 2013.

[21] Amazon Web Services, "Amazon S3," http://aws.amazon.com/s3, 2013.

[22] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A.Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M.Zaharia, "A View of Cloud Computing," Comm. the ACM, vol. 53,no. 4, pp. 50-58, 2010.

[23] Asigra, "Case Studies," http://www.asigra.com/product/casestudies/,2013.

[24] Amazon Web Services, "Amazon S3 Availability Event: July 20,2008," http://status.aws.amazon.com/s3-20080720.html, July 2008.

[25] A. Bessani, M. Correia, B. Quaresma, F. Andre´, and P. Sousa, "DEPSKY: Dependable and Secure Storage in a Cloud-of-Clouds,"Proc. ACM European Conf. Computer Systems (EuroSys '11), 2011.

[26] K.D. Bowers, A. Juels, and A. Oprea, "HAIL: A High-

Availability and Integrity Layer for Cloud Storage," Proc. 16th ACM Conf. Computer and Comm. Security (CCS '09), 2009.

[27] H. Blodget, "Amazon's Cloud Crash Disaster Permanently Destroyed Many Customers' Data," http://www.businessinsider.com/amazon-lost-data-2011-4/, Apr. 2011.

[28] B. Calder et al., "Windows Azure Storage: A Highly Available Cloud Storage Service with Strong Consistency," Proc. 23rd ACM Symp. Operating Systems Principles (SOSP '11), 2011.

[29] B. Chen, R. Curtmola, G. Ateniese, and R. Burns, "Remote Data Checking for Network Coding-Based Distributed Storage Systems,"MProc. ACM Workshop Cloud Computing Security Workshop (CCSW '10), 2010.

[30] H.C.H. Chen and P.P.C. Lee, "Enabling Data Integrity Protection in Regenerating-Coding-Based Cloud Storage," Proc. IEEE 31st Int'l Symp. Reliable Distributed Systems (SRDS '12), 2012.