International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

Available at http://inpressco.com/category/ijcet

Research Article

Blockchain based framework for data storage management with deduplication in cloud computing

Miss.Kirti Mahesh Deshpande and Prof.Aparna A. Junnarkar

Department of Computer Engineering, P.E.S.Modern College of Engineering, Pune

Received 10 Nov 2020, Accepted 10 Dec 2020, Available online 01 Feb 2021, **Special Issue-8 (Feb 2021)**

Abstract

Data reduction has become increasingly important in storage systems due to the explosive growth of digital data in the world that has ushered in the decentralized storage approach. One of the main challenges facing large-scale data reduction is how to maximally detect and eliminate redundancy at very low overheads. To guarantee the security of cloud users, data is always outsourced in encrypted format. In this paper, a deduplication resemblance detection and elimination scheme that effectively exploits existing duplicate information for highly efficient resemblance detection in data deduplication based storage systems. Conventional deduplication conspires always focus on specific application situations, where deduplication is handled by data owners or servers in the cloud. They can not adaptably fulfill demands that are unique from data owners based on the level of data sensitivity. In this paper, the plan all the while gives the executives of the deduplication plan and access control over cloud service providers (CSPs).

Keywords: *Blockchain,Cloud Computing, Data Deduplication, Access Control, Storage Management.*

Introduction

Cloud storage system has been mostly adopted, it does not meet some important emerging needs, such as the ability to verify cloud file integrity from cloud clients and the detection of duplicate files on servers in the cloud. We present the two problems below. These servers in the cloud can free customers from the heavy burden of storage management and maintenance[3][4]. The biggest difference between cloud storage and traditional internal storage is that data is transferred over the Internet and stored in an uncertain domain, which is not under the control of customers, which inevitably raises further concerns about data integrity. These concerns stem from the fact that cloud storage is affected by security threats both inside and outside the cloud, and servers in the uncontrolled cloud can passively hide some episodes of client data loss to maintain reputation[7]. The most serious thing is that to save money and space, servers in the cloud can even exclude an active and deliberate data file that we only have access to and that belong to a common client. Given the large size of outsourced data files and the limited capacity of client resources, the first problem is widespread so that the client can perform integrity checks effectively, even without a local copy of the data file. Cloud computing is computing in which large groups of remote servers are networked to enable centralized data storage and online access to services or IT resources[1].

With cloud computing, large groups of resources can be connected via a private or public network. In the public cloud, services (ie applications and storage space) are available for general use on the Internet. A private cloud is a virtualized data center that operates within a firewall. Cloud computing provides computing and storage resources on the Internet. The increasing amount of data is stored in the cloud, and users with specific privileges share it, which defines special rights to access stored data[8]. Managing the exponential growth of a growing volume of data has become a critical challenge[1]. According to the IDC Cloud report, companies in India are gradually moving from the legacy of the premise to different forms of cloud. Because the process is gradual, it started during the migration of some application workloads into the cloud. To perform scalable management of data stored in cloud computing, deduplication has been a wellknown technique that has become more popular recently[6][9]. Deduplication is a specialized data compression technique that reduces storage space and charges bandwidth in cloud storage. In deduplication, there is only one instance of data on the server and the redundant data is replaced with a pointer to the copy of the unique data. Deduplication can occur at the file or block level from the user's point of view, security and privacy issues arise, as data is susceptible to internal and external attacks. We must correctly implement the mechanisms of confidentiality, integrity verification and access control of both attacks.

Deduplication does not work with traditional cryptography. The user encrypts their files with their own individual encryption key, another cryptographic text can also appear for identical files. Therefore, traditional cryptography is incompatible with data duplication Converged cryptography is a widely used technique for combining storage savings with deduplication to ensure confidentiality. In converged encryption, data copy is encrypted with a key derived from the data hash. This converging key is used to encrypt and decrypt a copy of the data. After key generation and data encryption, users keep keys and send encrypted text to the cloud. Because cryptography is deterministic, copies of identical data will generate the same convergent key and the same ciphertext. This allows the cloud to duplicate encrypted texts. Cryptographic texts can only be decrypted by the owners of the corresponding data with their converging keys. Differential authorization duplication control is an authorized duplication elimination technique in which each user is granted a set of privileges during system initialization. This privilege set specifies which types of users can perform duplicate checks and access files[1].

Literature Survey

”G. Wallace, F. Douglis, H. Qian, P. Shilane, S. Smaldone, M. Chamness, and W. Hsu has developed Characteristics of backup workloads in production systems The author presents a complete characterization of backup workloads by analyzing statistics and content metadata collected from a large set of EMC Data Domain backup systems in production use. This analysis is complete (it covers the statistics of over 10,000 systems) and in depth (it uses detailed traces of the metadata of different production systems that store almost 700TB of backup data). We compared these systems with a detailed study of Microsoft’s primary storage systems and demonstrated that back-up storage differs significantly from the primary storage workload in terms of data quantities and capacity requirements, as well as the amount of data storage capacity. redundancy within the data. These properties offer unique challenges and opportunities when designing a disk-based file system for backup workloads[1].A. El-Shimi, R. Kalach, A. Kumar, A. Ottean, J. Li, and S. Sengupta have developed Primary data deduplication-large scale study and system designThe author presents a largescale study of primary data deduplication and uses the results to guide the design of a new primary data deduplication system implemented in the Windows Server 2012 operating system. The file data were analyzed by 15 servers of globally distributed files that host data for over 2000 users in a large multinational company. The results are used to achieve a fragmentation and compression approach that maximizes deduplication savings by minimizing the metadata generated and producing a uniform distribution of the portion size. Deduplication processing resizing with data size is achieved by a frugal hash index of RAM and data partitioning, so that memory, CPU and disk search resources remain available to meet the main workload of the IO service [2]. P. Kulkarni, F. Douglis, J. D. LaVoie, and J. M. Tracey, “Redundancy elimination within large collections of files”. Propose a new storage reduction scheme that reduces data size with comparable efficiency to the most expensive techniques, but at a cost comparable to the fastest but least effective. The scheme, called REBL (Block Level Redundancy Elimination), exploits the advantages of compression, deletion of duplicate blocks and delta encoding to eliminate a wide spectrum of redundant data in a scalable and efficient way. REBL generally encodes more compactly than compression (up to a factor of 14) and a combination of compression and suppression of duplicates (up to a factor of 6.7). REBL is also coded similarly to a technique based on delta encoding, which significantly reduces the overall space in a case. In addition, REBL uses super fingerprint, a technique that reduces the data needed to identify similar blocks by drastically reducing the computational requirements of the matching blocks: it converts the comparisons of O (n2) into searches of hash tables. As a result, the use of super fingerprints to avoid enumerating the corresponding data objects decreases the calculation in the REBL resemblance phase of a couple of orders of magnitude [3].Shweta D. Pochhi, Prof. Pradnya V. Kasture have represents“Encrypted Data Storage with De-duplication Approach on Twin Cloud. The data and the private cloud where the token generation will be generated for each file. Before uploading the data or file to the public cloud, the client will send the file to the private cloud for token generation, which is unique to each file. Private clouds generate a hash and token and send the token to the client. The token and hashes are kept in the private cloud itself, so that whenever the next token generation file arrives, the private clone can refer to the same token. Once the client gets the token for a given file, the public cloud looks for the token similar if it exists or not. If the public cloud token exists, it will return a pointer to the existing file, otherwise it will send a message to load a file. A system that achieves confidentiality and allows block-level deduplication at the same time. Before uploading the data or file to the public cloud, the client will send the file to the private cloud for token generation, which is unique to each file. The private cloud generates a hash and token and sends them to the client. The token and the hash are kept in the private cloud itself so that whenever the next token generation file arrives, the private clone can refer to the same token [4].Jin Li, Yan Kit Li, Xiaofeng Chen, Patrick P. C. Lee, Wenjing Lou have developed A Hybrid Cloud Approach for Secure Authorized De-duplication[9]. In the proposed system, we are getting data deduplication by providing data evidence from the data owner. This test is used when the file is uploaded. Each file uploaded to the cloud is also limited by a set of privileges to specify the type of users who can perform duplicate verification and

access the files. New duplication constructs compatible with authorized duplicate verification in the cloud hybrid architecture where the private cloud server generates duplicate file verification keys. The proposed system includes a data owner test, so it will help implement better security issues in cloud computing [5].M. Lillibridge, K. Eshghi, and D. Bhagwat represents the improvement in recovery speed for backup systems that use block-based online deduplication. The slow recovery due to the fragmentation of the parts is a serious problem faced by data deduplication systems in one piece: the recovery speeds for the most recent backup can eliminate orders of magnitude during the life cycle of a system. We have studied three techniques: increase the size of the cache, limit the containers and use a direct assembly area to solve this problem. Limiting the container is a time-consuming task and reduces fragmentation of fragments at the cost of losing part of the deduplication, while using a direct assembly area is a new technique of recovery and caching in the recovery process which exploits the perfect knowledge of the future access to the fragments available during the restoration of a backup to reduce the amount of RAM needed for a certain level of caching in the recovery phase [6].D. Meister, J. Kaiser, and A. Brinkmann represented caching of data deduplication locations. The author proposes a new approach, called Block Locality Cache (BLC), which captures the previous backup execution significantly better than existing approaches and always uses up-to-date information about the location and is therefore less prone to aging. We evaluated the approach using a simulation based on the detection of multiple sets of real backup data. The simulation compares the Block Locality Cache with the approach of Zhu et al. and provides a detailed analysis of the behavior and the IO pattern. In addition, a prototype implementation is used to validate the simulation[7].D. T. Meyer and W. J. Bolosky has represents A study of practical Deduplication. We collect data from the file system content of 857 desktop computers in Microsoft for a period of 4 weeks. We analyze the data to determine the relative efficiency of data deduplication, especially considering the elimination of complete file redundancy against blocks. We have found that full file deduplication reaches about three quarters of the space savings of more aggressive block deduplication for live file system storage and 87 of backup image savings. We also investigated file fragmentation and found that it does not prevail, and we have updated previous studies on file system metadata, and we have found that file size distribution continues to affect very large unstructured files[8].V. Tarasov, A. Mudrankit, W. Buik, P. Shilane, G. Kuenning, and E. Zadok having represents generating realistic datasets for the deduplication analysis. The author has developed a generic model of file system changes based on properties measured in terabytes of real and different storage systems. Our model connects to a generic framework to emulate changes in the file system. Based on observations from specific environments, the model can generate an initial file system followed by continuous changes that emulate the distribution of duplicates and file sizes, realistic changes to existing files and file system growth [9].P. Shilane, M. Huang, G. Wallace, and W. Hsu discovered the optimized WAN replication of backup data sets using delta compression reported by the stream. Offsite data replication is critical for disaster recovery reasons, but the current tape transfer approach is cumbersome and error prone. Replication in a wide area network (WAN) is a promising alternative, but fast network connections are expensive or impractical in many remote locations, so better compression is needed to make WAN replication very practical. We present a new technique for replicating backup data sets through a WAN that not only removes duplicate file regions (deduplication) but also compresses similar file regions with delta compression, which is available as a feature of EMC Data Domain systems [10]."

Proposed Methodology

In this paper, Author propose another methodology in the test of data ownership and cryptography to deal with the capacity of encoded information with deduplication. We will probably take care of the issue of deduplication in the circumstance where the data owner is not available or it is difficult to get involved. Meanwhile, the data size does not affect the performance of data deduplication in our schema. We are motivated to save space in the cloud and to protection of information of data owners by proposing a scheme to manage the storage of encrypted data with deduplication. We test safety and evaluate the performance of the proposed scheme through analysis and simulation. The results show its efficiency, effectiveness and applicability.

A. Architecture

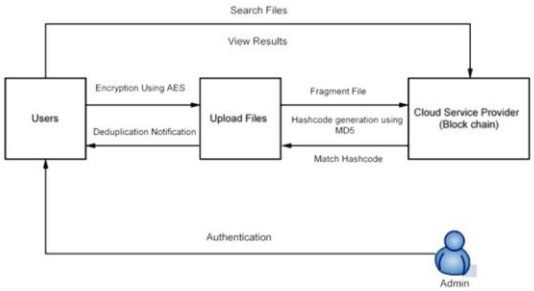

Fig. 1. System Architecture

Cloud Service Provider: The CSP allows the owner of data for data storage services. You can not trust completely. This is why the content of the stored data is curious. It must be done honestly in the conservation of data for profit.

Data Holder: The data owner can upload and save his data and files in the CSP. In this system, it is possible

that the number of data holders may store their files in raw cryptographic data in the CSP. The owner of the data that produces or creates the file considers the file as the owner of the data. The owner of the data is in normal form that the highest priority of the owner.
Authorized Party: An authorized party where data owners trust completely. Data holders to verify data ownership and manage data deduplication. It does not converge with the CSP. In this case, CSP does not need to know the user data in its memory.

B. Algorithms

1. AES Algorithm for Encryption and Decryption. AES (advanced encryption standard).It is symmetric algorithm. It used to convert plain text into cipher text .The need for coming with this algo is weakness in DES. The 56 bit key of des is no longer safe against attacks based on exhaustive key searches and 64-bit block also consider as weak..

Input:
128 bit /192 bit/256 bit input (0, 1) Secret key (128 bit) +plain text (128 bit).
Process:
10/12/14-rounds for-128 bit /192 bit/256 bit input
Xor state block (i/p)
Final round:10,12,14
Each round consists: sub byte, shift byte, mix columns, add round key.
Output:
cipher text(128 bit)

2. FRAGMENTATION ALGORITHM

Input: File
Output: Chunks
Step1: If file is to be split go to step 2 else merge the fragments of the file and go to step
Step2: Input source path, destination path
Step3: Size = size of source file
Step4: Fs = Fragment Size
Step5: NoF = number of fragments
Step6: Fs = Size/Nof
Step7: We get fragments with merge option
Step8: End

3. MD5 (Message-Digest Algorithm)

The MD5 message digest algorithm is a widely used cryptography hash function that produces a 128-bit (16byte) hash value, typically expressed as text in 32-digit hexadecimal numbers. MD5 has been used in a wide variety of cryptographic applications and is also commonly used to verify data integrity. Steps:

1. A message digest algorithm is a hash function that acceptsa sequence of bits of any length and produces a sequence of bits of a small fixed length.
2. The output of a message digest is considered as a digitalsignature of the input data.
3. MD5 is a message digest algorithm that produces 128 bitsof data.
4. Use the constants derived from the trigonometric sinefunction.
5. Go through the original message in blocks of 512 bits

C. Hardware and Software Requirements

Hardware Requirements:

1. Processor - Pentium –III
2. RAM - 2 GB(min)
3. Hard Disk - 20 GB
4. Key Board - Standard Windows Keyboard
5. Mouse - Two or Three Button Mouse
6. Monitor - SVGA

Software Requirements:

1. Operating System - Windows
2. Application Server - Apache Tomcat
3. Coding Language - Java 1.8
4. Scripts - JavaScript.
5. Server side Script - Java Server Pages.
6. Database - My SQL 5.0
7. IDE - Eclipse

D. Mathematical Model

KeyGenCE (M): K is the key generation algorithm that maps a data copy M to a convergent key K;
EncryptCE(K, M): C is the symmetric encryption algorithm that takes both the convergent key K and the data copy M as inputs and then outputs a ciphertext C;
DecryptCE(K,C): M is the decryption algorithm that takes both the ciphertext C and the convergent key K as inputs and then outputs the original data copy M
TagGenCE(M): T(M) is the tag generation algorithm that maps the original data copy M and outputs a tag T(M). We allow TagGenCE to generate a tag from the corresponding ciphertext, by using T(M)=TagGenCE(C), where C=EncryptCE(K,M).

Result and Discussion

A. Proposed Results

Proposed System tested the time spent to encryption and decryption a file with different sizes by applying AES with 2 different key sizes, namely 128 bits and 256 bits and observe from graph that encrypting or decrypting a file of 10 to 20 megabytes (MB) with 128-bit AES takes about 100 seconds. It is a reasonable and practical choice to apply symmetric encryption for data protection.

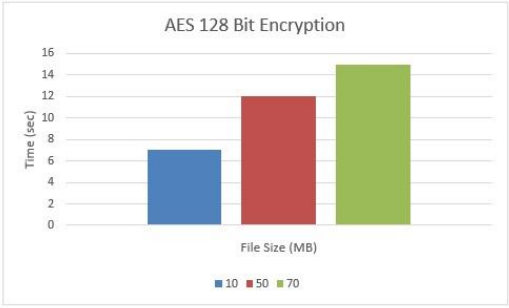

Fig. 2. Graph 1

Table 1:AES 128 bit Comparative Result

Parameter	AES 128 Bits	AES 128 Bits	AES 128 Bits
Time(sec)	7	12	15
File Size(MB)	10	50	70

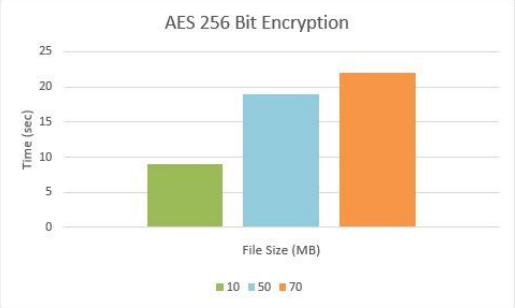

Fig. 3. Graph 2

Table 2: AES 256 bit Comparative Result

Parameter	AES 256 Bits	AES 256 Bits	AES 256 Bits
Time(sec)	9	19	22
File Size(MB)	10	50	70

B. Algorithm Comparison

In this subsection, our System evaluates the performance of the proposed scheme by several experiments. System runs these experiments on a window machine with an Intel Pentium 2.30GHz processor and 8GB memory. All these experiments use Java programming language with the various encryption algorithms such as AES, RSA. In our experiments, System first Install required Software.

Table 3: AES vs Blowfish algorithm comparative result

SR No	File size	Time in ms(RSA)	Time in ms(AES)
1	30kb	30	28
2	50kb	35	31
3	100kb	60	58
4	1mb	100	93
5	3mb	250	245

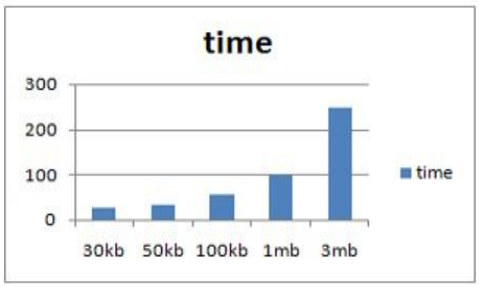

Fig. 4. Show File Size and Time to Upload

Here, In Table 3 entity Analysis of person such as Admin Cloud which performances different role. In Our System Data Owner Upload the Data in Dynamic Group which are show to Admin Side after show the Data "Click to Encryption" when User Request to Data access of Owner than System first Check Fault Tolerances Value and Then access To User

Conclusion

Data deduplication is important and significant in the practice of data storage in the cloud, in particular for the management of big data filing. In this paper, we proposed a heterogeneous data storage management scheme, which offers flexible data deduplication in the cloud and access control. Our schema can be adapted to different scenarios and application requests and offers cost-effective management of big data storage across multiple CSPs. Data deduplication and access control can be achieved with different security requirements. Security analysis, comparison with existing work and implementation-based performance evaluation have shown that our scheme is safe, advanced and efficient.

References

[1]. D. Meister, J. Kaiser, and A. Brinkmann, "Block locality caching for data deduplication," in Proc. 6th Int. Syst. Storage Conf., 2013, pp. 1–12.

[2]. M. Lillibridge, K. Eshghi, and D. Bhagwat, "Improving restore speed for backup systems that use inline chunk-based deduplication," in Proc. 11th USENIX Conf. File Storage Technol, Feb. 2013, pp. 183–197.

[3]. V. Tarasov, A. Mudrankit, W. Buik, P. Shilane, G. Kuenning, and E. Zadok, "Generating realistic datasets for deduplication analysis," in Proc. USENIX Conf. Annu. Tech. Conf., Jun. 2012, pp. 261–272.

[4]. D. T. Meyer and W. J. Bolosky, "A study of practical deduplication," ACM Trans. Storage, vol. 7, no. 4, p. 14, 2012.

[5]. G. Wallace, F. Douglis, H. Qian, P. Shilane, S. Smaldone, M. Chamness, and W. Hsu, "Characteristics of backup workloads in production systems," in Proc. 10th USENIX Conf. File Storage Technol., Feb.2012,pp.33–48.

[6]. El-Shimi, R. Kalach, A. Kumar, A. Ottean, J. Li, and S. Sengupta, "Primary data deduplication-large scale study and system design," in Proc. Conf. USENIX Annu. Tech. Conf., Jun. 2012, pp.285–296.

[7]. P. Shilane, M. Huang, G. Wallace, and W. Hsu, "WAN optimized replication of backup datasets using stream-informed delta compression," in Proc. 10th USENIX Conf. File Storage Technol.,Feb.2012,pp.49–64.

[8]. P. Kulkarni, F. Douglis, J. D. LaVoie, and J. M. Tracey, "Redundancy elimination within large collections of files," in Proc.USENIXAnnu.Tech.Conf. Jun.2012, pp.59–72.

[9]. Jin Li, Yan Kit Li, Xiaofeng Chen, Patrick P. C. Lee, Wenjing Lou "A Hybrid Cloud Approach for Secure Authorized De-duplication" IEEE Transactions on Parallel and Distributed Systems: PP Year 2014.

[10]. Shweta D. Pochhi, Prof. Pradnya V. Kasture "Encrypted Data Storage with De-duplication Approach on Twin Cloud " International Journal of Innovative Research in Computer and Communication Engineering