DOI: http://dx.doi.org/10.26483/ijarcs.v8i8.4689

ISSN No. 0976-5697

Volume 8, No. 8, September-October 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A STUDY ON DATA DEDUPLICATION IN CLOUD COMPUTING

S. Supriya MCA, M.Phil
Research Scholar
Department of Computer Science
Kongunadu Arts and Science College
Coimbatore, Tamil Nadu, India

Dr. S. Mythili MCA, M.Phil, Ph.D
Associate Professor and Head
Department of Information Technology
Kongunadu Arts and Science College
Coimbatore, Tamil Nadu, India

Abstract—Cloud computing provides scalable, low-cost and location-independent services over the internet. The services provided ranges from simple backup services to cloud storage infrastructures. The fast growth of data volumes has greatly increased the demand for techniques for saving disk space and network bandwidth. Cloud storage services like Dropbox, Mozy, Google Drive choose a deduplication technique where the cloud server stores only a single copy of redundant data and creates links to the copy instead of storing actual copies. The security of users data become a new challenge. Hence the users encrypt the data before outsourcing to the cloud. Conventional encryption techniques are incompatible with deduplication while convergent encryption resolves this problem effectively. Various research papers have been studied from the literature, as a result, this paper attempts to survey data deduplication techniques in cloud storage along with concepts, categories and methods used in data deduplication.

Keywords—Data Deduplication; Encryption; Hashing; Chunking; Hybrid Cloud.

I. INTRODUCTION

There is a significant increase in the amount of data generated each day and in 2020 it is expected 44 zettabytes of data will be produced [1]. The storage and management of these large volumes of data is becoming the most challenging job today. By re-arranging various resources over the internet cloud computing offers a new way of service provision. Among the services provided cloud storage service is the most important and popular one. Making the data management scalable in cloud computing deduplication technique has attracted more and more attention recently [2]. When the same data is being outsourced to the cloud storage by multiple users deduplication is most effective. Data deduplication is a one of the data compression techniques. This technique keeps only one physical copy and eliminates multiple data copies with the same content and links other redundant data to that copy. Many cloud storage services employ a deduplication technique reducing resource consumption thus saving disk space and network bandwidth.

Cloud users upload confidential and their personal data to the data center of the cloud service provider. It is judgmatic to assume that cloud service providers cannot be fully trusted by cloud users as the users sensitive data is susceptible to both insider and outsider attacks. Even though data deduplication promises lot of benefits, security and privacy becomes a serious issue due to the rapid development in data mining and analysis techniques. So as a god practice the user need to encrypt the data to be stored on cloud in order to ensure data security and user privacy. Deduplication have proved high cost savings , i.e., it reduces up to 68 percent in standard file systems [3] and 90-95 percent of storage needs in case of backup applications [4].

II. DATA DEDUPLICATION PROCESS

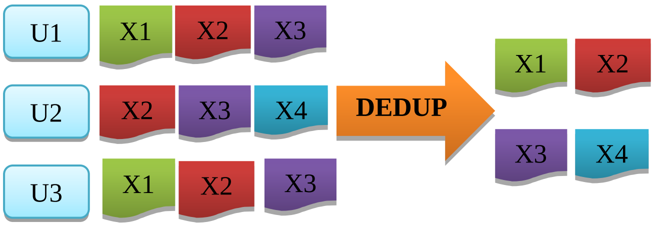

Figure 1. Data Deduplication Process

Data deduplication, also called as Intelligent Compression is the means of reducing the amount of data that needs to be stored. The process of data deduplication works by eliminating the repeated data and storing only the first unique instance of any data. If the user tries to store the same data again only a pointer is created to the originally stored data rather than storing the redundant data. For each file or chunk (in block level) a unique hash number is created using the hash algorithms such as MD5 or SHA1. The created hash number is compared with existing hash numbers in the index. If it exists then the data is not stored else the new hash number and data is stored. Sometimes the hash algorithm may produce the same hash number for different chunks of data which is termed as hash collision. Avoiding hash collision becomes a necessity to prevent data loss. Figure 1 explains the deduplication process involving three users. The users upload their files to the storage server. The files X1, X2 and X3 are repeated and hence deduplicated during the process. Deduplication not only saves the storage space and network bandwidth but also speeds up remote backup and disaster recovery process.

III. CLASSIFICATION OF DATA DEDUPLICATION

Data deduplication process can be classified based on Data unit, location and Disk placement which are explained below.

A. *Data unit based deduplicaton*

Data deduplication can operate at the file-level or block-level. In file-level deduplication two files are compared with their unique hash values. If the values are same then the files are assumed to have similar contents and thus only one copy is saved and the pointers are created for other copies. In block-level deduplication chunks are formed by splitting up the file contents. The chunks formed may be of fixed length or of variable length. As the name implies fixed size chunking divides the file into same sized chunks. It is faster among other chunking algorithms but it suffers from "Boundary Shift" problem when there is any modification in the data. Variable length blocks achieve good data deduplication throughput.

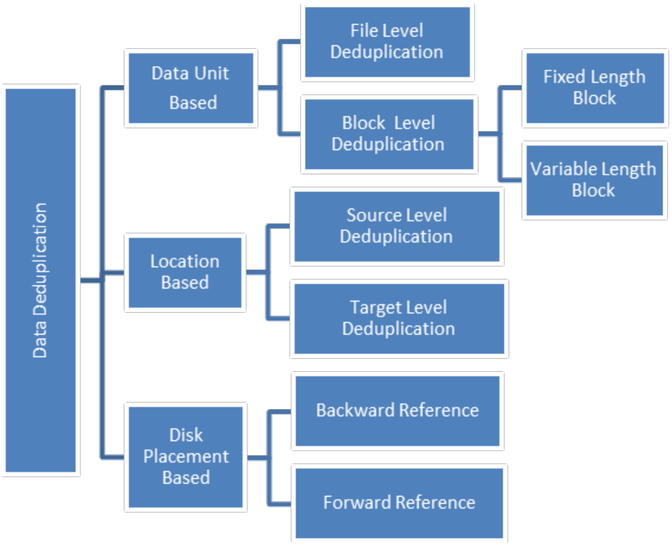

Figure 2. Classification of Data Deduplication

B. *Location based deduplication*

Location based deduplication is further divided into two types source based and target based deduplication. Source based deduplication is performed at the client side. Before the data is being transmitted to the storage server the deduplication process is carried out. This results in saving network bandwidth as well as storage space. Target based deduplication is done at the server side and the client is unaware of the process. There is no overhead on the client. Target based deduplication saves storage space but fails to save network bandwidth.

C. *Disk Placement based Deduplication*

Disk placement deduplicationt is based on how the data is stored in disk. Two techniques are used namely forward reference and backward reference. Forward reference maintains new data chunks while creating pointers to old data chunks. In Backward reference past data chunks are fragmented highly.

IV. PERFORMANCE EVALUATOR OF A DATA DEDUPLICATION SYSTEM

The performance of the any deduplication system is measured by two important computation. Dedupe ratio and Throughput.

1. Dedupe ratio=size of actual data / size of data after deduplication.
2. Throughput= Megabytes of data deuplication/ second.

V. METHODS USED IN DATA DEDUPLICATION

The following are the secure primitives used in deduplication.

A. *Symmetric Encryption*

Symmetric Encryption utilizes a common secret key *k* for both encryption and decryption. Symmetric encryption can be defined by three primary functions.

- $\mathrm{KeyGen}_{SE}\ (1^{\lambda}) \rightarrow k$ -is the key generation algorithm that generates k using security parameter 1^{λ} .
- $\mathrm{Enc}_{SE}\ (k\ ,\ M) \rightarrow C$ -is the symmetric encryption algorithm that takes the secret key k and message M as input and outputs the ciphertext C.
- $\mathrm{Dec}_{SE}\ (C,\ k\ ) \rightarrow M$ -is the symmetric decryption algorithm that takes the secret key k and cipher text C as input and outputs the message M.

B. *Convergent Encryption*

Convergent Encryption ensures data secrecy in deduplication. For each message M the user derives a convergent key and encrypts the message with that convergent key. In addition a tag is also derived for message M which is used to detect duplicates. If two messages are same then the tags are also the same. Convergent encryption can be defined by four primary functions.

- $\mathrm{KeyGen}_{CE}\ (M\ ) \rightarrow K$ -is the key generation algorithm that generates the key K and maps the message M to convergent key K .
- $\mathrm{Enc}_{CE}\ (K,\ M) \rightarrow C$ -is the encryption algorithm that takes the key K and message M as input and outputs the ciphertext C.
- $\mathrm{Dec}_{CE}\ (C,\ K) \rightarrow M$ -is the decryption algorithm that takes the key K and cipher text C as input and outputs the message M.
- $\mathrm{TagGen}(M) \rightarrow T(M)$ -is the tag generating algorithm that maps the tag T with message M.

Convergent encryption encrypts/decrypts with a convergent key that is obtained by computing the cryptographic hash value of the content of the message. Identical data from different users generate the same cipher text which makes deduplication feasible along with data confidentiality.

C. *Proof of Ownership*

The notion of proof of ownership (Pow) permits the user to prove the ownership of data copy M to the storage provider. Pow is implemented as an interactive algorithm by the user and the storage server. The storage server derives φ(M) for the data copy M. The user sends φ' to the storage server to prove the ownership. If $\varphi'=\phi(M)$ then the user is accepted as the data owner of the data copy M by the storage server.

D. *Identification Protocol*

The identification protocol has two phases Proof and Verify. In the proof phase the user can prove his identity to the verifier by presenting the recognizable proof. In the verify phase the verifier checks the identification proof submitted by the user and outputs the accept or reject message according to the proof submitted.

VI. DEDUPLICATION STORAGE SYSTEMS

An effective deduplication system is defined by the support it provides in terms of three correlating competing goals.

1. Deduplication efficiency: This is the primary compression goal which refers how efficiently the system detects the duplicate data units. Storage cost is reduced by good deduplication efficiency.
2. Scalability: It is the ability of the system to support enormous amount of raw storage with stable performance. A good scalability helps in reducing the overall cost by reducing the total number of nodes where each node can handle more data.
3. Throughput: Throughput refers to the data transfer rate in and out of the system. High throughput results in fast backups.

A deduplication system shares data among files by default which is antithetical to a traditional backup system. So there arises a need for reliable reference management which keeps track of segment usage and claim back the freed space. A few deduplication storage systems are discussed below where each one is preferred for different storage purposes.

Venti is block level network storage system, intended for archival data. As the blocks are addressed by the fingerprint (a unique hash produced by a collision resistant hash function) of their contents the modification on a block cannot be done without changing its address. This property implements write-once policy which distinguishes Venti from other storage systems. Although Venti is considered to be the building block of many storage applications it cannot efficiently deal with large amount of data and suffers from scalability.

HYDRAstor is a scalable, secondary storage solution. The front end is the traditional file interface where the back end is a grid of storage nodes. The implementation of variable-sized block, inline and hash-verified global duplicate elimination on storage nodes makes the system highly scalable.

Extreme Binning is a scalable and parallel deduplication system for chunk based file backup. It uses file similarity rather than locality thus increasing the throughput of the system. The arrangement of similar data files into bins makes deduplication easier by removing duplicated chunks from each bin. Extreme Binning is very powerful making data management tasks robust with low overhead.

MAD2 is an accurate deduplication network backup service which works on both file level and block level. The techniques which help the system in accelerating the deduplication process are organizing fingerprints into Hash Bucket Matrix, Bloom Filter Array to quickly identify incoming non duplicate object, dual cache and Load Balance technique.

Duplicate Data Elimination (DDE) highlights are address-by-block, only operate as a background process, block-level content hashing (160 bit SHA1), lazy update and copy-on-write that guarantees consistency between data and data hash. DDE can continuously improve storage efficiency as the data set grows.

VII. LITERATURE SURVEY

The following papers are surveyed in the following section along with its merits and demerits.

Wang et al, [5] proposed a system where the data owner has the encryption keys and access certificate. Any user who needs to access the data first sends a request to the data owner who in turn provides access certificate along with the encryption keys to the end user. Now the end user sends the access certificate to the data storage provider. After verifying the access certificate the data storage provider will send the encrypted data blocks to the end user. Although this system has low storage overhead, each time the data owner is being disturbed.

Roxana et al, [6] proposed a new system named FADE in which a new protocol called Vanish is introduced which includes data privacy and self-destruction of data. Vanish encrypts users data with the help of encryption key where the user is also unaware of the encryption key. After a particular time, the data is not accessible to the user nor to the data owner. Vanish protocol has self-deleting property and is applicable only for sensitive data. As the system promises assured deletion based on time, even the legitimate users and the data owner are unable to access the data after the time has expired.

The above discussed systems fall under traditional encryption and do not support deduplication.

Douceur et al, [7] introduced convergent encryption technique that ensures data privacy in deduplication. This scheme is safe for unpredictable messages and not suitable if the target message is drawn from the finite space. If an attacker can generate the convergent key of every single message and compute the corresponding cipher text, and if one computed cipher text is equal to target cipher text then the target message is inferred.

Bellare et al, [8] proposed a new encryption scheme Message Locked Encryption (MLE) where the key K for encryption and decryption is derived from the message itself. The encryption algorithm generates the cipher text C using key K mapped to message M. Ciphertext C is mapped to the tag T which is used to check for duplicates in the server. Keys used here are of fixed length and does not result in much storage overhead. But the system is susceptible to brute-force attacks.

Bellare et al, [9] proposed new system which resists brute-force attacks called DupLESS. DupLESS introduces an additional key server which encrypts the data rather than storage server. Users authenticate on their own to the key server without leaking any information about their data. DupLESS ensures high security as long as the key server remains safe that is inaccessible to attackers. DupLESS can transform the predictable message into an unpredictable one ensuring data security.

Haveli et al, [10] introduced proof-of-ownership (PoWs) which overcomes client-side exploitation. The exploitation of client-side deduplication is explained as follows. When the attacker knows the hash signature of others file and convince the server that he owns the file he can gain access to the file. This problem has been overcomed by the notion of proof-of-ownerships where the client has to prove to the server that he owns the file. Solutions are based on Merkle trees. The system incurs only a small overhead compared to naïve client-side deduplication but has high I/O requirements at the client side, leaving heavy computational toll on the client. Sorniotti et al, [11] gave the improved version of PoW where the I/O and computational costs do not depend on the input file size and thus reducing the computational toll on the client.

Bugiel et al, [12] proposed a twin cloud architecture consisting of trusted cloud and commodity cloud. Security critical operations are performed by the trusted cloud which is responsible for encrypting data. Queries related to outsourced data are performed by the commodity cloud. This approach claims protection against various security issues including leakage of data, computation manipulations, etc.

Jin Li et al, [13] proposed hybrid cloud approach which consists of a private and a public cloud supporting differential authorization duplicate check. The private cloud acts as a proxy allowing the data owners to securely perform duplicate check with differential privileges. The data storage is done in the public cloud while the data operation is managed in the private cloud. Convergent encryption technique is used here allowing the cloud to perform deduplication on the cipher texts along with proof-of-ownership to prevent unauthorized access.

Zheng et al, [14] proposed a novel based scheme to manage encrypted data storage with deduplication based on data ownership challenge and proxy Re-Encryption(PRE). The scheme uses ECC for verification of data ownership, proxy re-encryption algorithm for deduplication, SHA1 for hash function and AES for encryption/decryption of a file. The scheme solves the situation where the data holder is not available. There is a trusted third party called Authenticated party (AP) which is responsible for providing the re-encrypted key. The scheme supports data updataion, data deletion, data owner management and access control on encrypted data, saves storage space and also resists offline brute force attacks caused by convergent encryption. The disadvantage is that the same key is used for encryption/decryption which is not highly secure.

Wang et al, [15] propose a scheme for deduplication with secure access control using attribute based encryption [ABE]. The system uses SHA1 for hash function, CP-ABE for deduplication, AES for symmetric encryption and RSA for Public Key Cryptography. The scheme supports data updation, deletion and data owner management with low operational and implementation cost as the third party is not involved for key generation. A demerit of this system is that it takes more time for key generation.

Junbeom et al, [16] proposed a deduplication scheme for encrypted data that has dynamic ownership management capability that uses randomized convergent encryption. It uses MD5 for key and token generation and AES with electronic code book algorithm for encrypting and decrypting a file. The system is secure against the chosen-plaintext attack, collusion attack and poison attack. It supports forward and backward secrecy of outsourced data. The disadvantage of the scheme is when a data loss attack occurs the system cannot recover the original data as all duplicates are removed from the cloud service provider.

Jan et al, [17] proposed a system that deals with a new concept called data popularity. When the file becomes popular the secure cipher text of the file is downgraded to convergent cipher text of file that supports deduplication. The scheme uses Identity Provider and the Index Repository Service [IRS] (the two trusted authority), AES-128-CTR for symmetric encryption and SHA 256 for hashing function. The advantage of this system is that it has low cost and resists user collision attack. The disadvantage is that it cannot prevent attacks based on knowledge of hash of the plaintext file.

Rodel et al, [18] proposed a system ensuring deduplication with the help of key server deployed at CSP that is used to issue Data Encryption Key to the users. The system uses AES 128 bit algorithm for symmetric encryption and MD5/SHA algorithms for hash functions. Homomorphic XOR operation is included to enhance the security. Although the system resists man in the middle attack as file encryption is done at client side it is vulnerable to passive attack such as unauthorized reading of the file and traffic analysis etc.

Hui cui et al, [19] proposed attribute based storage system which employs ciphertext-policy attribute based encryption for deduplication of encrypted data in cloud. The system uses a method to modify the ciphertext over one access policy into ciphertext of the same plain text but under another access policy without revealing the underlying plaintext. A new approach is proposed based on zero-knowledge proof of knowledge and a commitment scheme to achieve data consistency in the system. The merits of the system is that it can share data with other users by specifying an access policy rather than sharing the decryption key and it achieves the standard notion of semantic security.

VIII. CONCLUSION

Data deduplication is an emerging trend and secure deduplication is one of the most important concerns for users. The paper focuses on basics of deduplication including the process of deduplication, classification, methods used in deduplication and few data deduplication systems. Various techniques available for secure data deduplication are also discussed along with their advantages and disadvantages. In future secure data deduplication systems should be build with standards providing high performance ratio and throughput along with user privacy.

IX. REFERENCES

[1] J. Gantz and D. Reinsel, “The digital universe in 2020: Big data, bigger digital shadows, biggest growth in the far east,” IDC iView: IDC Analyze the Future, 2012.

[2] S. Quinlan and S. Dorward, “Venti: A new approach to archival storage,” Proc. 1st USENIX conf. File Storage Technol., Jan. 2002, pp.7.

[3] D. T. Meyer and W. J Bolosky, “A study of practical deduplication,” ACM Trans. Storage, vol. 7, no. 4, pp. 1-20, 2012, doi:10.1145/2078861.2078864.

[4] Opendedup. (2016). [online]. Available: http://opendedup.org/

[5] W. Wang, Z.Li, R. Owens and B. Bhargava, “Secure and Efficient Access to Outsourced Data,” Proc. ACM CCSW, Nov. 2009, pp. 55-66.

[6] R. Geambasu, T. Kohno, A. Levy, “Vanish: Increasing Data Privacy with Self-Destructing Data,” Proc. USENIX Security Symp., Aug. 2009, pp. 316-299.

[7] J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon and M. Theimer, “Reclaiming space from duplicate files in a serverless distributed file system,” ICDCS, pp 617-624, 2002.

[8] M. Bellare, S. Keelveedhi and T. Ristenpart, “Message-Locked Encryption and Secure Deduplication,” EUROCRYPT, pp 296-312, 2013.

[9] M. Bellare, S. Keelveedhi and T. Ristenpart, “Dupless: Serveraided Encryption for Deduplicated Storage,” USENIX Security Symposium, 2013.

[10] S. Haveli, D. Harnik, B. Pinkas and A. Shulman-Peleg, “Proofs of ownership in remote storage systems,” Proc. 18th ACM Conference on Computer and Communications Security,” pp. 491-500, 2011.

[11] R. D. Pietro and A. Sorniotti, ”Boosting efficiency in proof of ownership for deduplication,” ACM Symposium on Information, Computer and Communications Security, pp. 81-82, 2012.

[12] S. Bugiel, S. Nurnberger, A. Sadeghi and T. Schneider, “Twin clouds: An architecture for secure cloud computing,” Proc. Workshop Cryptography Security Clouds, pp.32-44, 2011.

[13] Li, Jin, Yan Kit Li, Xiaofeng Chen, Patrick PC Lee and Wenjing Lou, “A Hybrid Cloud Approach for Secure

Authorized Deduplication, "IEEE Transactions on Parallel and Distributed Systems, Vol. 26, No. 5, May 2015, pp. 1206-1216.

[14] Zheng Yan, Wenxiu Ding, Xixun Yu, Haiqi Zhu and Robert H. Deng, "Deduplication on Encrypted Big Data in Cloud," IEEE Transactions on Big Data, Vol.2, No.2, April_June 2016, pp. 138-150.

[15] Zheng Yan, Mingjun Wang, Yuxiang Li and Athanasios V. Vasilakos, "Encrypted Data Management with Deduplication in Cloud Computing," IEEE Cloud Computing, March-April 2016, pp. 29-35.

[16] Junbeom Hur, Dongyoung Koo, Youngjoo Shin and Kyungtae Kang, "Secure Data Deduplication with Dynamic Ownership Management in Cloud Storage," IEEE Transactions on Knowledge and Data Engineering, Vol. 28, No. 11, November 2016, pp. 3113-3125.

[17] Jan Stanek and Lukas Kencl, "Enhanced Secure Thresholded Data Deduplication Scheme for Cloud Storage, "IEEE Transactions on Dependale and Secure Computing.

[18] Miguel, Rodel and Khin Mi Mi Aung, "HEDup: Secure Deduplication with Homomorphic Encryption," IEEE International Conference on Networking, Architecture and Storage (NAS)," pp. 215-223, 2015.

[19] Hui Cui, Robert H. Deng, YingJiu Li and Guowei Wu, "Attribute-Based Storage Supporting Secure Deduplication of Encrypted Data in Cloud," IEEE Tranactions on Big Data, 2016.