Data Analytics and Artificial Intelligence
Vol: 3(6), 2023
REST Publisher; ISBN: 978-81-948459-4-2
Website: http://restpublisher.com/book-series/daai/

Deduplication Approach for Big Data Storage Systems

P. Jegathesh, Dhatchayani K, Nancy S, Nivetha P, Sowmya B
Karpagam College of Engineering, Coimbatore, Tamil Nadu, India.

Abstract:*In the field of IT, cloud computing is often credited with the potential to become the industry standard. Using storage systems, it provides streamlined resource management and system maintenance. Storage has recently become a popular research topic as a fundamental cloud computing technology. The use of innovations in CPUs and software enhancements in hypervisors themselves has effectively addressed the high drain of virtualization. The growing demand for image storage facilities is still a difficult issue. Deduplication within a storage area network system has been used by some systems to reduce storage image consumption. Due to their cost limitations, storage area networks cannot meet the growing need for large-scale storage hosting for cloud computing. In this design, we propose SiLo, a scalable deduplication train system specifically designed for large-scale storage implementation. SiLo stands for "similarity and position." By using deduplication on storage photos, its design enables fast storage provisioning with similarity and location anchor point indicators for data transfer. In addition, SiLo provides a full range of storage functions, such as on-demand costing over a network, caching with original media using dupe-on-read techniques, and fast cloning of storage images. Experiments show that the functions of SILO work well and cause only minor performance degradation*.

Keywords:*Data sharing, Scalability, Data Deduplication, Cloud Computing, Big Data.*

1. INTRODUCTION

Cloud storage is a platform to give large scale data storehouse and service access at a “pay- as- you- go” fashion. Still, a lot of spare data in cloud storage has seriously wasted and enthralled storehouse coffers. Data deduplication is an effective technology to descry and exclude spare data.

Data Deduplication is a approach for removing the duplicate of cloned or repeating data. With the data deduplication implementation, we can improve the storage utilization, which may in turn lower capital expenditure by reducing the overall amount of storage media required to meet storage capacity needs. Data deduplication method needs comparison of data which is also known as ‘byte patterns’ that are unique and contiguous blocks of data. This chunk that is byte patterns are found and stored during a process of analysis and compared to other chunks within existing [already stored data] data. Whenever a match occurs, the unnecessary chunk is replaced with a reference that points to the stored chunk. The same byte pattern may occur in dozens, hundreds, or even thousands of times [the match frequency is depending on the chunk size], the quantity of data that must be stored or transferred on the server can be greatly reduced.

Storage-based data deduplication which reduces the amount of storage space needed for a given set of files. This deduplication approach is most effective in cloud servers where many copies of similar or even identical data that are stored and transferred over the server.

The main objective of this system using hash code for the content of the file, if this code is find into the database then system generates a duplicate file message for the users else file will be divide into three chunks which is stored on the three different location so the load will be divide and automatically load balancing happened.

[1]. SED ensures the semantic security in the random oracle model and has strong anti-attack ability such as the brute-force attack resistance and the collusion attack resistance. Besides, SED can effectively eliminate data redundancies with low computational complexity and communication and storage overhead. [2] after that, only a single copy of the data is uploaded and stored. Thus, the data deduplication technology can reduce the bandwidth requirement of client-side and improve the space utilization efficiency of server-side. Currently, it has widely used in various cloud computing services to improve user experience and save storage space

The classic data deduplication scheme and its variants [2-7], where the framework consists of a key server [KS], a cloud storage provider [CSPs], and users, ensure the security depending on the trusted KS. What is worse, these classic schemes may suffer from the single-point-of-failure and “platform lock-in” issues. If the trusted KS fails, the cloud storage system stops working and data outsourcing protocols cannot be implemented. Recently, a new model of cloud computing, called as Joint Cloud computing system [8], has been designed to solve the above-mentioned issues well. The network architecture of Joint Cloud consists of users and multiple CSPs providing various services.

Joint Cloud computing has drawn a lot of attention from both academia and assiduity. [9-12] A massive data breach incidents affecting billions of particular data are far common. Thus, the outsourced data are generally asked to be translated to insure data confidentiality in cloud storehouse systems. Still, it's delicate to descry and cancel the duplicated clones in the ciphertext sphere. Because the cipher texts of the same plain text translated by different druggies using traditional encryption algorithms are different. In order to apply translated deduplication, coincident encryption [CE] [13] and its variants [14-17] have been proposed. In these schemes, data are translated using the keys deduced from the data themselves. That is, the secret key is deterministic, and the scheme is vulnerable. Specifically, there are some security vulnerabilities., 1) the label reveals the hash value of plaintext, which is vulnerable to the chosen- plaintext attack; 2) the cipher text dissatisfies the semantic security; 3) the predictable plaintext cannot repel the brute- force attack; 4) druggies have to bear a great computational burden to cover their data against vicious bush hackers, etc.

RELATED WORK

Secure and Efficient data deduplication is one of the main approaches to ensure the security and uniqueness of data in deduplication, which uses Advanced Encryption Standard AES to generate obviously semantic security and anti-brute force attack. SED uses the techniques such as tag generation, encryption, and deduplication. Firstly, it tests 32768 test files with 20% duplicates are selected, in which the size of each and every file is of 32B.However, there is no scalability in distributed data sharing systems and difficult to implement fault tolerance mechanisms when the number of nodes keeps changing. Abadi et al. [11] they proposed a full randomized approach and an encrypted scheme for bounded message based on non- degenerate. Li et al. [15,3] designed secure deduplication with key management based on secret sharing schemes. Followed by Shin et al. [20] proposed a decentralized server-aided encryption for deduplication However it needed several interactions among users and KS, which gave the third parties favourable chance to get the valuable information from the communication [3]. Xia et al. [3] designed a fast and systematic content –defined chunking approach. Apart from that some of the emergingtechnologies like blockchain are used in deduplication system in several applications, which are focused on by the researchers lately.

PROPOSED APPROACH

The proposed system generates a key for the files present in the system with the of Elliptic curve cryptography approach. The system also uses Similarity and locality algorithm which seeks similarity for file content and its location. Then when a duplicate file is uploaded a key gets generated and the keys and location are tested for similarity, if both are found to be similar only one of the files gets saved other file is deduplicated.

Elliptic Curve Cryptography Design:

It's a new form of cryptographic medium to secure the data in both physical and cloud storehouse. ECC is one of the stylish ways grounded on the proposition of elliptical angles. The property of the elliptic wind is used to induce the keys for encryption methodology rather of being ways which use large high figures. ECC uses the elliptic wind equation for crucial generation. In 1985, N. KobiltzandV. Miller proposed the elliptical wind cryptography for changeable data to have secure data. The introductory idea behind ECC is to use an elliptic wind to integrate a separate logarithm problem. The model ECC has the most complex fine problems in public-crucial cryptography. In this case, where calculating power is constantly perfecting, working separate logarithms in a fine sense becomes a fairly simple task. As a result, consider moving the separate logarithm problem result to a further critical and delicate area to apply, similar as elliptic angles.

The most significant of the ECC model uses the lower keys for security. When compared to the other model [1064- bit crucial] ECC takes the 164- bit crucial for the same position of security. This model is a public-crucial cryptosystem for the generation of keys centered on the fine complexity of working the elliptic wind separate logarithm problem to cipher and decipher data. It measures the hops demanded to travel from one point on an elliptic wind to another. Elliptic angles are symmetrical over thex-axis and are double angles. ECC is a cryptographic scheme in which each customer has two keys a public and a private key. Encryption and hand authentication is done with the public key, while decryption and hand generation are done with the private key.

Key Generation:

Any cryptography fashion applied for security, the original and the most significant process is the crucial generation for both public and private. With the help of the receiver's public key, the sender encrypts the communication data, and the receiver decrypts it with its private key. Then we've the law for theellicurv[data] function for all storehouses. The crucial generation procedure is below.

1. Senders public $y[rand_{nopA}$] arbitrary number from $\{1,2, t-1\}$
2. Senders publiccrucial$lS.public_{keyXA} = rand_{nopA} * Gen_{ptQ}$
3. Receivers public $key\left[rand_{nopB}\right]$arbitrary number from {1, 2, t −1}
4. Receiverpublic$keyRpublic_{keyXB} = rand_{nopB} * Gen_{ptQ}$

5. Sender security key,

$$key \ = \ rand_{nopA} \ * R.public_{keyXB}$$

6. Receiver security key,

$key \ = \ rand_{nopB} \ * S.public_{keyXA} \ \ = \ rand_{nopA} \ * Gen_{ptQ} \, \#\# \, sender \ public \ key \ \ =$

$rand_{nopB} \ * Gen_{ptQ} \, \#\# \, receiver \ public \ key \ \ key \ =$

$rand_{nopA} \ * R.public_{keyXB} \, \#\# \, sender \ security \ key \ \ key \ =$

$rand_{nopB} \ * S.public_{keyXA} \, \#\# \, sender \ security \ key$

Encryption Code:

From this offer, the stoner data is divided into three corridors. The third part of 85 of stoner information is translated with the ECC system. The plain textbook istransferred to the cloud, before saving the data it can be translated into points and it can be saved as translated points [F1, F2]. The translated points are stored in cloud storehouse. The first stoner data is saved in the separate warehouses and during the third part of data, it calls theellpcurve function. Now sender, shoot communication e to the end receiver. e has any pointEllpcurve on elliptic wind, Now the sender choose an arbitrary number, r from$\{ \ 1, 2 \ t \ - 1\}$Cipher textbook will be written from points$[F1, F2]$ then, $F1 \ = \ r \ * Gen_{ptQ} \, and \ \ F2 \ = \ Ellp_{curve \, [\, r \, * Gen_{ptQ} \,]}$

Decryption Code:

Once the stoner stores the data in translated points$[F1, F2]$ during the reclamation of the data, the translated points are deciphered as plain text that can be stoner readable.

On calculating the receiver side, addition of $F1$ with its $private \ key \ [rand_{noPb} \ * \ F1]$ also subtracts from the result of former step

– Original communication
= $F2 - [rand_{noPb} \ * \ F1]$
= $F2$ – $[pB \ * \ F1]$

SILO ALGORITHM:

Files in the backup stream are first chunked, fingerprinted, and packed into segments by grouping strongly correlated small files and segmenting large files in the File Agent. For an input segment Snew.

Step 1: Check to see if Snew is in the SHT Table. If it hits in SHT Table, SiLo checks if the blockBbk containing Snew’s similar segment is in the cache. If it is not in the cache, SiLo will loadBbk from the disk to the Read Cache according to the referenced block ID of Snew’s similar segment, where a block is replaced in the FIFO order if the cache is full.

Step 2: The duplicate chunks in Snew are detected and eliminated by checking the fingerprint sets of Snew with LHT Table [fingerprints index] of Bbk in the cache.

Step 3: If Snew misses in SHT Table, it is then checked against recently accessed blocks in the read cache for potentially similar segment [i.e., locality-enhanced similarity detection].

Step 4: Then SiLo will construct input segments into blocks to retain access locality of the input backup stream. For an input block Bnew, SiLo does follow.

Step 5: The representative fingerprint of Bnew will be examined to determine the stored backup nodes of data block Bnew.

Step 6: SiLo checks if the Write Buffer is full. If the Write Buffer is full, a block there is replaced in the FIFO order by Bnew and then written to the disk

After the process of deduplication indexing, SiLo will record the chunk-to-file mapping information as the reference for each file, which is managed by the Job Metadata of the Backup Server. For the read operation, SiLo will read the referenced metadata of each target file in the Job Metadata that allows the corresponding data chunks to be read from the data blocks in the Storage Server. These data chunks will then be used to reconstruct the target files in the File Daemon according to the index mapping relationship between files and deduplicated data chunks.

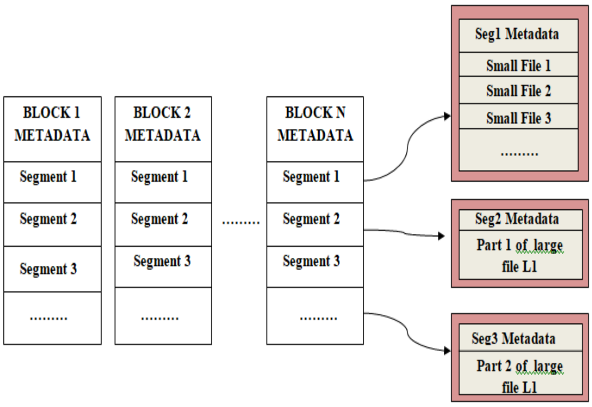

FIGURE 1. Design of SiLo Algorithm

Now we further dissect the relationship between similarity and position with respect to backup aqueducts. As mentioned before, knob position can be exploited to store and prefetch groups of conterminous gobbets that are likely to be penetrated together with a high probability in the backup sluice, while lines' similarity may be booby-trapped so that the similarity characteristics rather of the whole sets of fingerprints of lines, are listed to minimize the indicator size in the memory. The exploitation of position maximizes the RAM application to ameliorate the outturn but can beget RAM overflows and frequent accesses to on- fragment indicator when datasets warrant or are weak in position project

The similarity- grounded approaches minimize the RAM operation at the cost of potentially missing large quantities of spare data which is dependent on the similarity degree of the backup sluice. We've examined the similarity degree and the duplicate-elimination measure of our similarity-only deduplication approach on four datasets The four datasets represent one- backups, incremental backups, Linux- performances and full- backups independently

Silo Workflow:

Lines in the backup sluice are first chunked, fingerprinted and packed into parts by grouping explosively identified small lines and segmenting large lines in the train Agent.

Each recently generated member Snew is checked against SHT Table for similarity discovery. If the new member hits in SHT Table, SiLo checks if the blockBbk containing Snew's analogous member is in the, Bbk is read from If it isn't in the cache.the fragment to the read cache, where a block is replaced in the FIFO order if the cache is full. If Snew misses in SHT Table, it's also checked against lately penetrated blocks in the read cache for potentially analogous member in one of the cached blocks [Bbk].

The indistinguishable gobbets in Snew are excluded by checking LHT Table of Bbk in the read cache. Also, the gobbets in the neighbouring parts of Snew in the backup sluice are filtered by the locality enhanced similarity discovery [i.e., these gobbets are checked against LHT Table of Bbk for possible duplication].

After knob filtering and constructing a new and non-duplicate block Bnew from the backup sluice, Silo checks if the write buffer is full. If the write buffer is full, a block there's replaced in the FIFO order by Bnew and also written to the fragment.

As demonstrated in Section 4, SiLo is able to minimize both the time and space overheads of indexing fingerprints while maintaining a duplicate elimination performance comparable to exact deduplication methods such as Chunk Stash.

IMPLEMENTATION

In deduplication frame, propose system apply block position deduplication system and named as similarity and position grounded deduplication frame that's a scalable and short outflow near-exact deduplication system, to master the forenamed failings of being schemes. The main idea of T3S is to consider both similarity and position in the backup sluice coincidently.

Specifically, expose and use further similarity through grouping explosively identified small lines into a division and segmenting large lines, and influence position in the backup sluice by grouping closest parts into blocks to confine analogous and indistinguishable data missed by the probabilistic similarity discovery.

By keeping the resembling indicator and conserving spatial position of help aqueducts in RAM [i.e., hash table and position cache], T3S is suitable to remove huge quantities of spare data, dramatically reduce the figures of accesses to on- fragment indicator, and mainly increase the RAM application. Thisapproach divides a large train into numerous little parts to more expose similarity among large lines while adding the effectiveness of the deduplication channel.

The Detailed Construction:

Private Cloud storage can be built from the unused resources to store the data that belongs to an organization. Many organizations have set up private Clouds as they result in 19 better utilizations of resources. Since private Cloud storage have limited amount of hardware resources, they need to be utilized optimally so has to accommodate maximum data. Deduplication is an effective technique to optimize the utilization of storage space. The work in this paper focuses on de-duplication. Two methods adopted for deduplication, namely, chunk level and file level, are studied in following modules.

The virtualization is being used to provide ever-increasing number of servers on virtual machines [storage], reducing the number of physical machines required while preserving isolation between machine instances. This approach better utilizes server resources, allowing many different operating system instances to run on a small number of servers, saving both hardware acquisition costs and operational costs such as energy, management, and cooling. Individual storage instances can be separately managed, allowing them to serve a wide variety of purposes and preserving the level of control that many users want. In this module, clients store data into data servers for future usages. Then data servers stored data in Meta servers.

Deduplication is a technology that can be used to reduce the amount of storage required for a set of files by identifying duplicate “chunks” of data in a set of files and storing only one copy of each chunk. Subsequent requests to store a chunk that already exists in the chunk store are done by simply recording the identity of the chunk in the file’s block list; by not storing the chunk a second time, the system stores less data, thus reducing cost. In this module, we implement fingerprint scheme to identifying chunksdiffer, both fixed-size and 20 variable-size chunking use cryptographically secure content hashes such as ECC to identify chunks, thus allowing the system to quickly discover that newly-generated chunks already have stored instances.

In this module, we first broke storage disk images into chunks, and then analysed different sets of chunks to determine both the amount of deduplication possible and the source of chunk similarity. We use the term disk image to denote the logical abstraction containing all of the data in storage, while image files refer to the actual files that make up a disk image. A disk image is always associated with a single storage; a monolithic disk image consists of a single image file, and a spanning disk image has one or more image files, each limited to a particular size. Files are stored in data server with block id, and this can be monitored by Data servers. Data servers are mapped by using Meta servers.

In this module, we can analyse data sharing components and Meta server in SILO responsible for managing all data servers. It contains SHT and LHT table for indexing each files details for improving search mechanisms. A dedicated background daemon thread will immediately send a heartbeat message to the problematic data server and determines if it is alive. This mechanism ensures that failures are detected and handled at an early stage. The stateless routing algorithm can be implemented since it could detect duplicate data servers even if no one is communicating.

TABLE 1. Comparison Of Deduplication with Previous Schemes

Parameters	Existing system	Proposed system
Data Size (TB)	700	1.16
Algorithm Efficiency	low compared to the ECC algorithm	higher than the previous schemes
indexing runtime (MS)	438.191	380.314
scalability	no scalability	scalability in distributed data sharing systems

We have made the approach with the help of   modules such as:

- Cloud resource allocation
- Deduplication scheme
- File system analysis
- Data sharing components
- Evaluation criteria

Cloud Resource Allocation:

Virtualization is deploying more and more servers on virtual storage machines, reducing the number of physical machines needed while maintaining isolation between machine instances. With this approach, can be better utilised because many different operating system instances can run on a small number of servers, reducing not only hardware acquisition costs but also software costs. Individual instances of STOR instances can be managed independently, allowing them to be used for different purposes and maintaining the level of control desired by many users. In this module, the client stores the data on the data server for later use. The data server then stores the data in the meta server.

Deduplication Scheme:

Deduplication is a technology used to reduce the storage space required for a set of files by identifying duplicate "chunks" of data in a set of files and storing only one copy of each chunk. Subsequent requests to store a chunk that already exists in chunk memory are handled by simply recording the identity of the chunk in the file's block list; by not storing the chunk a second time, the system stores less data, reducing costs. In this module, we implement a fingerprint scheme to identify chunks. Both fixed-size chunking and variable-size chunking use cryptographically secure content hashes such as ECC to identify chunks, allowing the system to quickly determine that newly created chunks already have stored instances.

File System Analysis:

In this module, we first decomposed storage disk images into chunks and then analyzed different sets of chunks to determine both the extent of possible deduplication and the source of chunk similarity. We use the term disk image to refer to the logical abstraction that contains all the data in storage, while image files refer to the actual files that make up a disk image. A disk image is always associated with a single storage; a monolithic disk image consists of a single image file, and a spanned disk image has one or more image files, each limited to a specific size. Files are stored in data servers with block IDs and can be monitored by data servers. Data servers are mapped using meta servers.

Data sharing components:

In this module, we can analyze the data sharing components and the meta server in SILO, which is responsible for managing all data servers. It contains SHT and a LHT table to index each file details to improve the search mechanisms. A special daemon thread in the background immediately sends a heartbeat message to the problematic data server and determines if it is still active. This mechanism ensures that failures are detected and handled early. The stateless routing algorithm can be implemented as it can detect duplicate data servers even if no one is communicating with them.

EVALUATION CRITERIA:

Deduplication is an efficient approach to reduce storage requirements in environments with a large number of storage disk images. As we have shown, deduplication of storage disk images can save 80% or more of the disk space required to store the operating system and application environment; we have studied the impact of many factors on the effectiveness of deduplication. We have shown that data locality has little impact on the deduplication rate. However, factors such as the base operating system or even the Linux distribution can have a large impact on deduplication effectiveness. Therefore, we recommend that hosting centres suggest "preferred" OS distributions to their users to ensuremaximum space savings. If this preference is followed, subsequent user activities will have little impact on deduplication effectiveness.

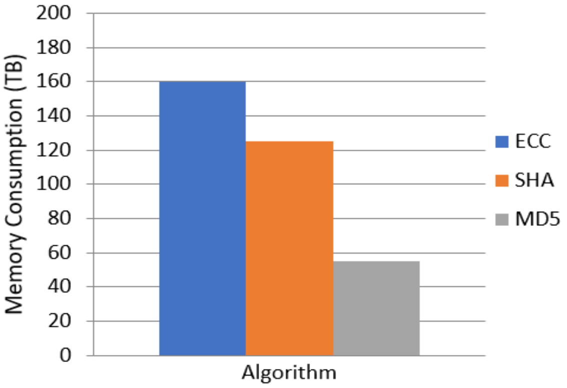

FIGURE 2. Record of Memory Consumption by Various Algorithm

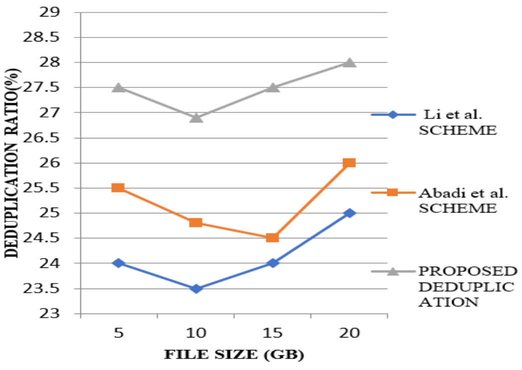

FIGURE 3. Record of Compression Ratio Graph

We expand on the post processing deduplication aspects of our system, including data chunking, chunk indexing, and data partitioning and reconciliation. To provide context for this discussion, we provide here an overview of our overall primary data deduplication system, as illustrated

In the architecture diagram we can find that the user is given a user. Then the user id is proceeded to SHA algorithm. The next step in the architecture design is the process of giving a digital signature. Then the registered user can upload file into the cloud system. After which the uploaded files get through the checking process, where the id, file name and content is checked. 15 Then the duplicated files are found.

After the checking process is completed, it is sent to backup server and through meta server the file reaches the data centre. It is applied for LHT construction and SHT construction then unduplicated files are stored in the system. In deduplication framework, propose system implement block level deduplication system and named as similarity and locality-based deduplication [SILO] framework that is a scalable and short overhead near-exact deduplication system, to defeat the aforementioned shortcomings of existing schemes.

The main idea of SiLo is to consider both similarity and locality in the backup stream concurrently. Specifically, expose and utilize more similarity through grouping strongly correlated small files into a division and segmenting large files, and leverage locality in the backup stream by grouping closest segments into blocks to confine similar and duplicate data missed by the probabilistic similarity detection. By keeping the parallel index and preserving spatial locality of help streams in RAM [i.e., hash table and locality cache], SiLo is able to remove huge amounts of redundant data, dramatically reduce the numbers of accesses to on-disk index, and substantially increase the RAM utilization. This approach divides a large file into many little segments to better expose similarity among large files while increasing the efficiency of the deduplication pipeline.

CONCLUSION

In cloud server much information are stored again and again by user. So, the user who operates need further spaces to store another data. This will lessen the memory space of the cloud for the users. To get the better of this problem uses the deduplication approach. Data deduplication approach is a method for sinking the amount of storage space for an organization that wants to save its data and information.

In several associations, the storage systems have many duplicate copies of many sections of data. In case, the similar file or content that might be keep in several dissimilar places by different users, two or many files that aren't the same may still include much of the similar content. Deduplication approach remove these extra and duplicate copies by saving just one version of the data and exchange the other copies with pointers that lead reverse to the distinctive copy.

So, we proposed this Block-level deduplication approach that frees up more spaces in the storage servers and exacting categorization recognized as variable block or variable length deduplication approach has become very popular. [21] In cloud using the SHT and LHT tables the users easily search the content and retrieves the searched content from the cloud server. And implemented heartbeat protocol which is to recover the data from the corrupted cloud server. Experimental metrics are proved that our proposed deduplication approach provide improved results in deduplication process.

REFERENCES

[1]. D. Meyer And W. Bolosky,"A study of practical deduplication," in Proceedings of the 9th USENIX Conference on File and Storage Technologies, 2011.

[2]. B. Debnath, S. Sengupta, And J. Li, "Chunkstash: speeding up inline storage deduplication using flashmemory," in Proceedings of the 2010 USENIX conference on USENIX annual technical conference. USENIX Association, 2010.

[3]. W. Dong, F. Douglis, K. Li, H. Patterson, S. Reddy, And P. Shilane,"Tradeoffs in scalable data routing for deduplication clusters," in Proceedings of the 9th USENIX conference on File and storage technologies. USENIX Association, 2011.

[4]. E. Kruus, C. Ungureanu, And C. Dubnicki,"Bimodal content defined chunking for backup streams," in Proceedings of the 8th USENIX conference on File and storage technologies.USENIX Association, 2010.

[5]. G.Wallace, F. Douglis, H. Qian, P. Shilane, S. Smaldone, M. Chamness, And W. Hsu, "Characteristics of backup workloads in production systems," in Proceedings of the Tenth USENIX Conference on File and Storage Technologies, 2012.

[6]. Broder, "On the resemblance and containment of documents," in Compression and Complexity of Sequences 1997.

[7]. D. Bhagwat, K. Eshghi, And P. Mehra,"Content-based document routing and index partitioning for scalable similarity-based searches in a large corpus," in Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2007, pp. 105–112.

[8]. Y. Tan, H. Jiang, D. Feng, L. Tian, Z. Yan, And G. Zhou, "Sam: A Semantic-Aware Multi-Tiered Source De-duplication Framework for Cloud Backup," in IEEE 39th International Conference on Parallel Processing. IEEE, 2010, pp. 614–623.

[9]. M. Lillibridge, K. Eshghi, D. Bhagwat, V. Deolalikar, G. Trezise, And P. Camble,"Sparse indexing: large scale, inline de-duplication using sampling and locality," in Proceedings of the 7th conference on File and storage technologies, 2009, pp. 111–123.

[10]. D. Bhagwat, K. Eshghi, D. Long, And M. Lillibridge,“Extreme binning: Scalable, parallel de-duplication for chunk-based file backup,” in IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems. IEEE, 2009, pp. 1–9.

[11]. Xu, Jia, AndJianyingZhou."Leakage-resilient proofs of ownership in cloud storage, revisited." International Conference on Applied Cryptography and Network Security. Springer International Publishing, 2014.

[12]. M. Bellare, S. Keelveedhi, And T. Ristenpart. Dupless:Server aided encryption for deduplicated storage. In USENIX Security Symposium, 2013

[13]. Halevi, Shai, Et Al."Proofs of ownership in remote storage systems."Proceedings of the 18th ACMconference on Computer and communications security. ACM, 2011

[14]. Armknecht, Frederik, Et Al."Transparent data deduplication in the cloud."Proceedingsof the 22nd ACM SIGSAC Conference on Computer and Communications Security. ACM, 2015.

[15]. Bellare, Mihir, SriramKeelveedhi, And Thomas Ristenpart. “Message-locked encryption and secure deduplication.” In Advances in Cryptology EUROCRYPT 2013, pp. 296-312. Springer Berlin Heidelberg, 2013.

[16]. S. Zhang, H. Xian, Z. Li, And L. Wang, “Secdedup:Secure encrypted data deduplication with dynamic ownership updating,” IEEE Access, vol. 8, pp. 186 323–186 334, 2020.

[17]. S. Srisakthi And G. A. Ansari, “Pcsp: A protected cloud storage provider employing light weight techniques,” Information Security Journal A Global Perspective, no. 4, pp. 1–12, 2021.

[18]. F. M. Awaysheh, M. N. Aladwan, S. Alawadi, J. C. Cabaleiro, And T. F. Pena,“Security by design for big data frameworks over cloud computing,” IEEE Transactions on Engineering Management, vol. PP, no. 99, 2021.

[19]. N. Almrezeq, M. Humayun, A. El-Aziz, And N. Z. Jhanjhi, “An enhanced approach to improve the security and performance for deduplication,” Turkish Journal of Computer and Mathematics Education [TURCOMAT], vol. 12, no. 6, pp. 2866–2882, 2021.

[20]. J. Li, X. Chen, M. Li, J. Li, P. Lee, And W. Lou.Secure deduplication with efficient and reliable convergent key management. In IEEE Transactions on Parallel and Distributed Systems 2013.

[21]. U. Habiba, R. Masood, M. A. Shibli, And M. A. Niazi,“Cloud identity management security issues & solutions: a taxonomy,” Complex Adaptive Systems Modeling, vol. 2, no. 1, pp. 1-37, 2014.

[22]. W. K. Ng, Y. Wen, And H. Zhu,“Private data deduplication protocols in cloud storage,” in Proc. 27th Annu. ACM Symp. Appl. Comput., 2012, pp. 441–446.