Data and Metadata. 2024; 3:245
doi: 10.56294/dm2024245

ORIGINAL

Big Data De-duplication using modified SHA algorithm in cloud servers for optimal capacity utilization and reduced transmission bandwidth

Big Data Deduplicación utilizando algoritmo SHA modificado en servidores en la nube para una utilización óptima de la capacidad y un ancho de banda de transmisión reducido

Rajendran Bhojan[1], Manikandan Rajagopal[2], Ramesh R[3]

[1]Department of Mathematics and Computer Science, The Papua New Guinea University of Technology.
[2]Lean Operations and Systems, School of Business and Management, CHRIST (Deemed to be University), Bangalore, India.
[3]Department of Computer Science, KPR College of Arts Science and Research, Tamilnadu, India.

Cite as: Bhojan R, Rajagopal M, R R. Big Data De-duplication using modified SHA algorithm in cloud servers for optimal capacity utilization and reduced transmission bandwidth. Data and Metadata 2024;3:245. https://doi.org/10.56294/dm2024245.

Submitted: 30-11-2023 **Revised:** 26-02-2024 **Accepted:** 29-03-2024 **Published:** 30-03-2024

Editor: Adrián Alejandro Vitón Castillo

ABSTRACT

Data de-duplication in cloud storage is crucial for optimizing resource utilization and reducing transmission overhead. By eliminating redundant copies of data, it enhances storage efficiency, lowers costs, and minimizes network bandwidth requirements, thereby improving overall performance and scalability of cloud-based systems. The research investigates the critical intersection of data de-duplication (DD) and privacy concerns within cloud storage services. Distributed Data (DD), a widely employed technique in these services and aims to enhance capacity utilization and reduce transmission bandwidth. However, it poses challenges to information privacy, typically addressed through encoding mechanisms. One significant approach to mitigating this conflict is hierarchical approved de-duplication, which empowers cloud users to conduct privilege-based duplicate checks before data upload. This hierarchical structure allows cloud servers to profile users based on their privileges, enabling more nuanced control over data management. In this research, we introduce the SHA method for de-duplication within cloud servers, supplemented by a secure pre-processing assessment. The proposed method accommodates dynamic privilege modifications, providing flexibility and adaptability to evolving user needs and access levels. Extensive theoretical analysis and simulated investigations validate the efficacy and security of the proposed system. By leveraging the SHA algorithm and incorporating robust pre-processing techniques, our approach not only enhances efficiency in data de-duplication but also addresses crucial privacy concerns inherent in cloud storage environments. This research contributes to advancing the understanding and implementation of efficient and secure data management practices within cloud infrastructures, with implications for a wide range of applications and industries.

Keywords: Preprocessing; De-duplication; SHA; Cloud Servers; Target.

RESUMEN

La deduplicación de datos en el almacenamiento en la nube es crucial para optimizar la utilización de recursos y reducir la sobrecarga de transmisión. Al eliminar copias redundantes de datos, mejora la eficiencia del almacenamiento, reduce los costos y minimiza los requisitos de ancho de banda de la red, mejorando así el rendimiento general y la escalabilidad de los sistemas basados en la nube. La investigación investiga la intersección crítica entre la de duplicación de datos (DD) y las preocupaciones de privacidad dentro de los servicios de almacenamiento en la nube. Datos distribuidos (DD), una técnica ampliamente empleada en estos servicios y cuyo objetivo es mejorar la utilización de la capacidad y reducir el ancho de banda de transmisión. Sin embargo, plantea desafíos a la privacidad de la información, que normalmente se abordan

mediante mecanismos de codificación. Un enfoque importante para mitigar este conflicto es la de duplicación jerárquica aprobada, que permite a los usuarios de la nube realizar comprobaciones de duplicados basadas en privilegios antes de cargar los datos. Esta estructura jerárquica permite a los servidores en la nube crear perfiles de usuarios según sus privilegios, lo que permite un control más matizado sobre la gestión de datos. En esta investigación, presentamos el método SHA para la deduplicación dentro de servidores en la nube, complementado con una evaluación segura de preprocesamiento. El método propuesto se adapta a modificaciones dinámicas de privilegios, proporcionando flexibilidad y adaptabilidad a las necesidades cambiantes de los usuarios y los niveles de acceso. Amplios análisis teóricos e investigaciones simuladas validan la eficacia y seguridad del sistema propuesto. Al aprovechar el algoritmo SHA e incorporar técnicas sólidas de preprocesamiento, nuestro enfoque no solo mejora la eficiencia en la de duplicación de datos sino que también aborda preocupaciones cruciales de privacidad inherentes a los entornos de almacenamiento en la nube. Esta investigación contribuye a avanzar en la comprensión y la implementación de prácticas de gestión de datos eficientes y seguras dentro de las infraestructuras de la nube, con implicaciones para una amplia gama de aplicaciones e industrias.

Palabras clave: Preprocesamiento; Deduplicación; SHA; Servidores Cloud; Target.

INTRODUCTION

The absence of dedicated assistance for data-escalated logical work processes, data administration is the misplaced ability restricts greater acceptance of clouds for logical processing.[1] Currently, work process information handling in the cloud is attained through the use of both the MapReduce programming model, and an application-specific overlays, which direct the output of one project to the contribution of an alternative in a pipeline mold. Superior storage frameworks that enable virtual machines (VMs) to utilize shared information simultaneously are necessary for such applications.[2] However, the current reference business clouds only provide high-dormancy REST (HTTP) interfaces for accessing the storage. Also, conditions could occur by which the implementations are required to modify the method information is accomplished to conform to the definite access technique.[3] The necessity of efficient storage for workloads involving a lot of data. Using such open cloud question stores in conjunction with a more conventional parallel record structure for the application would be the first method of data oversight. In any event, because of the aforementioned data access protocols, compute hubs and storage hubs are separated in the current cloud topologies, and communication among the two is quite inactive.

Furthermore, as these facilities mainly target storage, they only facilitate data sharing as an indication, meaning they do not facilitate transfers among the optional VMs deprived of a middleman to store the data. Along with the expense of renting the VMs, clients must pay for the storage and transfer of data in and out of these archives.[4] Recently, cloud providers: Azure Drives or Amazon EBS offered the option to link cloud storage in the form of the virtual volumes to the registration hubs. Not only is this option susceptible to identically elevated delays by the usual storage access, but it also offers flexibility and sharing restrictions because only one VM can mount such a volume at a time. In order to prevent data misuse when storage and transmitting work process data, other to cloud storage is to provide an equivalent record architecture on the process hubs. Distributed storage solutions, like Gfarm[5], activate in the host operating system of the physical hub with the purpose of storing the data in the machine's surrounding storage rings. They were delivered in a register cloud called Eucalyptus.

Data De-Duplication is a unique form of data firmness that requires entire data owners who upload identical data to share a single copy of the duplicate data, hence removing the duplicate copies from the storage.

The cloud storage server verifies whether or not uploaded data has been placed once users send it. The data will really be written in the storage if it hasn't been stored; if it has, the cloud storage server will merely store a pole pointing to the initial data copy rather than the complete set. As a result, it can prevent repeatedly storing the same data. In general, there are two main methods for DD:

a) Target-DD.
b) Source-DD.

DD is the procedure of eliminating redundant data, which lowers data redundancies within (intra file) and between (inter file) files. Identify common data segments and store them only once, both within and between files. DD reduces the usefulness of a given quantity of storage, which can result in cost savings. The various DD methods and data sets all have somewhat varying levels of the DD performance. While DD can result in significant space savings, it is a data-intensive process that has greater resource overheads on the existing storage systems.

One compression technique called DD is primarily used to remove duplicate files or data by maintaining

a single copy in the storage system to minimise space consumption. Additionally, it makes searches faster and more efficient in terms of outcomes. Sometimes, it is referred to as storage capacity optimisation, finds the files that are duplicated from the data repository or from the storage systems and specifically employs the "reference pointer" to identify the chunks that are not necessary. Block level DD and file level DD are two possible methods of DD. Keeping the original physical copy of the data after DD allows you to eliminate unnecessary data without keeping multiple duplicate copies of the same file or data with comparable content. [6] There are numerous cloud storage services available, including Memopal, Mozy, and the Dropboxthat use de-duplication techniques to protect customer data. Concerns regarding privacy and security are brought up by outsourcing. De-duplication is engineered to take these aspects into account, optimising data storage capacity and network bandwidth while aligning with the latest convergent key management features. Proof of protocols are used to secure data from unauthorised users. Cloud load balancing is a feature that cloud providers like Amazon Web Services (AWS), Microsoft Azure, and Google to facilitate easy task distribution. Elastic Load Balancing (ELB) technology is one example of AWS offers to distribute traffic between EC2 instances. The majority of applications supported by AWS with ELBs implemented as a crucial architectural element.

Objective of the study

I. Design and implementation a modified SHA algorithm within cloud servers for efficient and secure data de-duplication, addressing the conflict between capacity optimization and information privacy.
II. Proposing a hierarchical structure enabling privilege-based duplicate checks by cloud users before data upload, aiming to reconcile de-duplication requirements with privacy concerns.
III. To enable the adaptation of user privileges within the de-duplication process, facilitating dynamic changes in access levels and enhancing user control over data management.

Literature survey

For effective, adaptive P2P streaming of scalable video, error-resilient, Murat Tekalp et al.[7] suggest two changes to the Torrent protocol: variable sized chunking and adaptive P2P. The suggested changes produce better P2P video streaming outcomes by the quantity of decoded frames and, consequently, better user experience. The suggested changes to BitTorrent for video streaming produce better outcomes by the quantity of decoded frames (greater experience quality) and the chunks that are shared among the (P2P task) leechers. The amount of decodable frames has greatly augmented, as evidenced by the variable-sized chunk testing, boosting the PSNR and QoE. The suggested adaptive windowing enables improved scalability against an aggregating amount of leechers, as demonstrated by the varying size chunking test. As a result, the suggested changes would end in higher-quality video being received by peers and less bandwidth costs for content creators (seeders).

For P2P live streaming, Jin Li et al.[8] suggest a chunk-driven overlay with DHT support that aims for greater scalability, improved accessibility, and reduced latency. A video source selection algorithm, 2-layer hierarchical DHT-infrastructure, and chunk sharing algorithm, comprise the three primary parts of the architecture. The DHT-based hierarchical infrastructure has a great scalability. High availability is ensured by the chunk sharing algorithm, which offers services for chunk index discovery as well as collection. The technique used for provider selection allows for complete system bandwidth utilization. Consequently, the overlay has the ability to stream videos in high quality. Additionally, they suggest a decentralized provider selection algorithm that is simplified and centralized. When it comes to handling churn, DCO outperforms Tree-systems in terms of latency and bandwidth consumption. More significantly, by dynamically matching chunk requesters and suppliers, it may flexibly utilize all available system bandwidth. According to the experimental findings, DCO enhances the scalability, availability, latency, and overhead of mesh-systems (pull and push) as well as tree-systems. The test outcomes further validate the significance of choosing chunk providers with adequate bandwidth for chunk distribution and offering reasons to nodes to act as coordinators in the DHT-architecture.

Extreme Binning is a new technique for scalable and parallel deduplication that was established by Kave Eshghil et al.[9] It is particularly well-suitable for workloads that comprise distinct files with low locality. With such a workload, current methods that depend on locality to guarantee a fair throughput perform badly. In order to reduce the disk bottleneck issue, Extreme Binning uses file comparison rather than position to use a single disc access for each file's chunk in search as opposed for each chunk. In comparison to a flat index method, it divides the chunk index as two tiers, ending in a small RAM footprint that enables the method to sustain throughput for a bigger data set. It is simple and straightforward to partition the data chunks and the two-tier chunk index. There is no data or index exchange amongst nodes in a dispersed method by several backup nodes. A stateless routing mechanism is used to assign files to a distinct node for storage, and deduplication; this indicates the backup nodes content are not need to be known at the time of allocation. The distribution of one file per backup node allows for maximum parallelization. Since there are no dependences among the bins or among pieces tied to various bins, redistributing indices and chunks is a clean process, and backup nodes can be augmented to increase throughput. Data management operations including integrity checks, data restore,

https://doi.org/10.56294/dm2024245

and the trash collection requests are made efficient by the autonomy of backup nodes. The improvements in RAM applications and scalability more than make up for the minor loss of de-duplication.

The SHA family was expanded by Shih-Pei Chien et al.[10] which admits messages of any length as input and produces a message digest of the necessary length. The eight working variables, the for-loop operation, the first hash values, constants, Boolean expressions and functions, the padding and parsing, and the message program are all modified in this generalised version of SHA-mn. Additionally, the ith intermediary hash values are computed.

In[11] , the authors have addressed the LHV problem, which was absent from the initial SHA standard, has been resolved. Due to security concerns, SHA-mn is standardized using the SHA family design guidelines. The structure of SHA was significantly enhanced, even though many people may disagree on the birthday paradox technique for determining complexity because the full SHA-1 collision was discovered in 2005. Determining effective methods for SHA-256 collision detection is still a major area of study for several scientists.

The SHA processor architecture described by Sang-Hyun Lee et al.[12] implements 3 hash algorithms: SHA-512/256, SHA-512/224, and SHA-512. Based on hash algorithms, the SHA processor produces digests with 512, 224, and 256 bits, which are the three different lengths. Because it was built on a 32-bit data path and employed SHA-512 to construct the initial hash values of SHA512/224 and SHA-512/256, the application is region effective. The FPGA implementation validated the HDL-designed SHA processor. Operating at up to 185 MHz clock frequency, the SHA processor created by 0,18μm CMOS cell library takes up 27,368 GEs (Gate Equivalents). IoT security applications can make advantage of the SHA processor. Key words: security for IoT devices, hash, integrity and SHA.

Proposed work

The proposed preprocessing Approach

The proposed methodology is shown in figure1. Suggesting the NPFD (Novel preprocessing framework for de- duplication) architecture as a solution to these problems with cloud data management. It is a PaaS-level cloud storage system that leverages virtual discs and is concurrency-optimized. It associates the local discs of these virtual machines into a comprehensively shared data store for an application that consists of an excessive number of VMs. Applications thus exchange input files and save intermediate data or output files directly on the local disc of the virtual machine instance. The findings shown in this chapter show that this strategy improves the throughput over remote cloud storage by more than two times. Moreover, by creating an Azure prototype that employs the suggested storage strategy as the data management back-end and implements this computation paradigm, the advantages of the NPFD method were verified in the context of MapReduce.

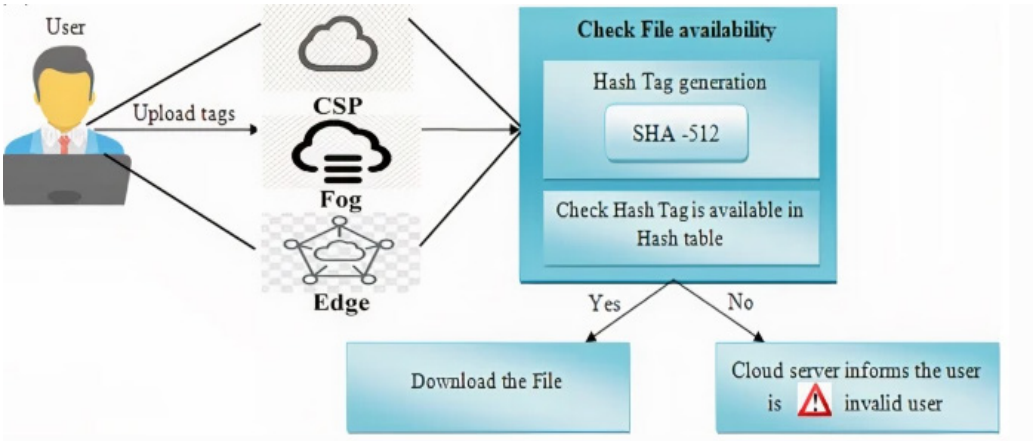

Figure 1. Proposed methodology

Novel preprocessing framework for deduplication(NPFD): Virtual Disc Associating for a Communication-Efficient Storage NPFD technique for analyzing virtual discs in VMs is presented in this part of the presentation. The architecture addresses each of the fundamental requirements of data-serious apps point-by-point by providing simultaneous access to low-inertia data storage. We start from the assumption that many cloud organizations do not abuse the discs secretly linked to the VMs, which have storage limits of several GBs that are available at no further cost.

As a result, we suggest pooling some of the virtual discs' total storage space into a shared basic pool can be attained in a dispersed way. The purpose of this pool is to hold application-level data. To optimize the heap and provide flexibility, information is stored in a striped fashion, meaning that it is divided into chunks that are uniformly distributed over the storage device's neighboring discs. Every lump is duplicated across other nearby

https://doi.org/10.56294/dm2024245

discs with the ultimate purpose of enduring setbacks. This method greatly improves read and write throughput under simultaneous conditions by distributing the global input/output effort evenly across the adjacent discs. Additionally, this approach reduces latencies by enabling data space and has the capacity to be highly versatile because an increasing VMs counts calls for a larger storage structure.

De-duplication in SHA Algorithm

High availability, scalability, security, fault tolerance, and cost-effective services are all possible with cloud storage. De-duplication is a well-accepted technique that has gained increasing traction recently for its ability to build scalable data organization in cloud-based settings. The method known as "DD" stores only one copy of the data and provides a link to it instead of storing the actual copy of the data. Block level or file level deduplication is accomplished using this method. Eliminating duplicate data blocks at the block level and the file level results in de-duplication.

Security and privacy are the main issues with DD since users' sensitive information is susceptible to attacks from within as well as outside. Conventional encryption is incompatible using data de-duplication techniques when it comes to maintaining data security. Utilising their unique key, each user encrypts their data during the traditional encryption process. As a result, different users' matching copies of the data will produce different cypher texts, making data deduplication impossible. In the proposed approach, we are addressing that problem by employing a technique that offers data security and scalability. Although the DD technique has several benefits, customers still worry about security and confidentiality because their sensitive information is susceptible to threats that are internal as well as external.

Every file is constrained to a hash function in this manner. Duplicate file detection will be aided by the hash function's continuous encoding; such as message direct or secure hash algorithm-1 encoded hash value. Analysts calculated the efficacy of the technique in the same data sets. Two pieces of data will be regarded as having duplicate content if their hash values are the same. The method was substituted at the bottom rating score with predefined block size and Chunking techniques. This method of updating data condenses throughput by using the SHA-1 algorithm for a large number of data sources, making it inefficient for saving a significant amount of storage space. However, because this method is quick and requires little math, it is inexpensive and highly effective.

Improved SHA De-Duplication Algorithm – Pseudo code

1. *Initialize cloud server with modified SHA algorithm for data de-duplication.*
2. *Establish hierarchical structure for privilege-based duplicate checks.*
3. *Define data upload process for cloud users:*
   DO
   Receive data upload request with associated user privilege and
   If duplicates found:
   - Proceed with data upload if privilege-based duplicate check is approved.
   Else
   Proceed with data upload.
4. *Implement dynamic privilege modification mechanism:*
   a. Allow users to modify their privileges dynamically.
   b. Update hierarchical structure accordingly.
5. *Develop pre-processing assessment for data security:*
   a. Conduct pre-processing assessment on incoming data.
   b. Ensure data meets security criteria before de-duplication.
6. *Integrate SHA algorithm for de-duplication:*
   a. Apply modified SHA algorithm to identify duplicate data chunks.
   b. Eliminate redundant copies while maintaining data integrity.
   Calculate capacity utilization improvement.
   b. Evaluate reduction in transmission bandwidth.
   While File upload = YES
10. *Iterate and refine methodology based on evaluation results.*

RESULTS

The results of the pre-processing steps are depicted in table 1. The result indicate that the time taken for Input- Output of the proposed method is better than the P1 and P2 methods discussed in section that followed.

https://doi.org/10.56294/dm2024245

Table 1. Comparison Table of I/O Time per Job			
P1	**P2**	**P3**	**Proposed**
0,581	0,6	0,475	0,299
0,699	0,71	0,5	0,35
0,726	0,789	0,654	0,391
0,78	0,812	0,699	0,423
0,81	0,856	0,765	0,501

The metrics of the suggested approaches are low in comparison to the proposed method. Values for the suggested method range from 0,299 to 0,501 as it is seen in figure 2.

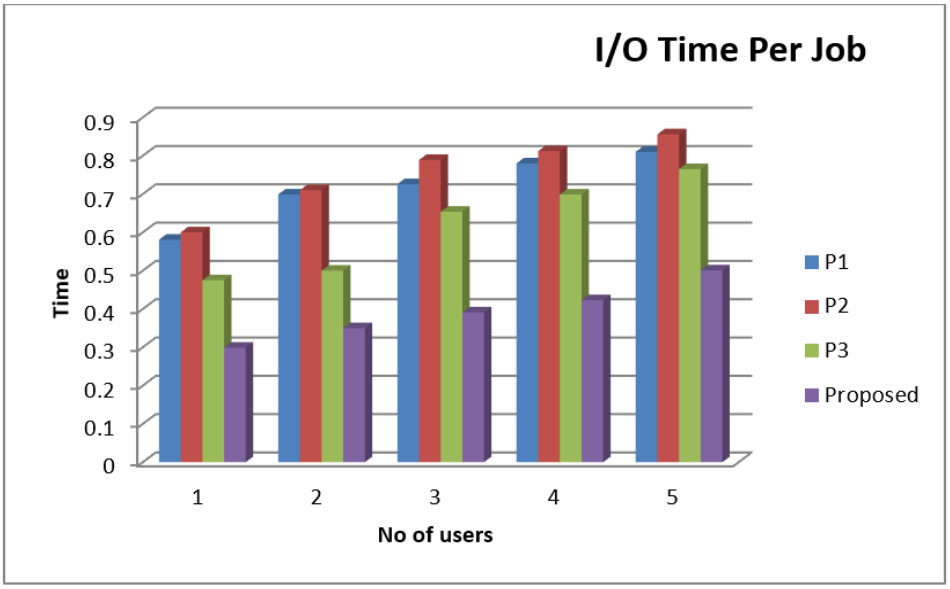

Figure 2. Comparison Chart of I/O Time per Job

The suggested approach and the current techniques' differing figures are displayed in the I/O time per job comparison table. Both the sequence level and the number of records on the X and Y axes. The values of the suggested approach are less than those of the present technique when compared with it. Values for the suggested technique range from 0,299 to 0,501.

Table 2. Comparison Table of Computing Time			
P1	**P2**	**P3**	**Proposed**
0,388	0,499	0,555	0,167
0,459	0,49	0,541	0,201
0,501	0,569	0,615	0,28
0,575	0,599	0,629	0,31
0,59	0,62	0,65	0,388

Table 2 outlines one suggested method and 3 previous methods (P1, P2, and P3). The values of the suggested approaches are low in comparison to the current approaches. Values for the suggested approach range from 0,167 to 0,388 as shown in figure 3.

The computation time comparison chart displays the disparities in values between the suggested approach and the previous methods is shown in table 3. Record count on the x-axis and series levels on the y-axis. The values of the suggested approach are lower than those of the previous method when compared with it. Values for the suggested approach range from 0,167 to 0,38 as shown in figure and table 3.

https://doi.org/10.56294/dm2024245

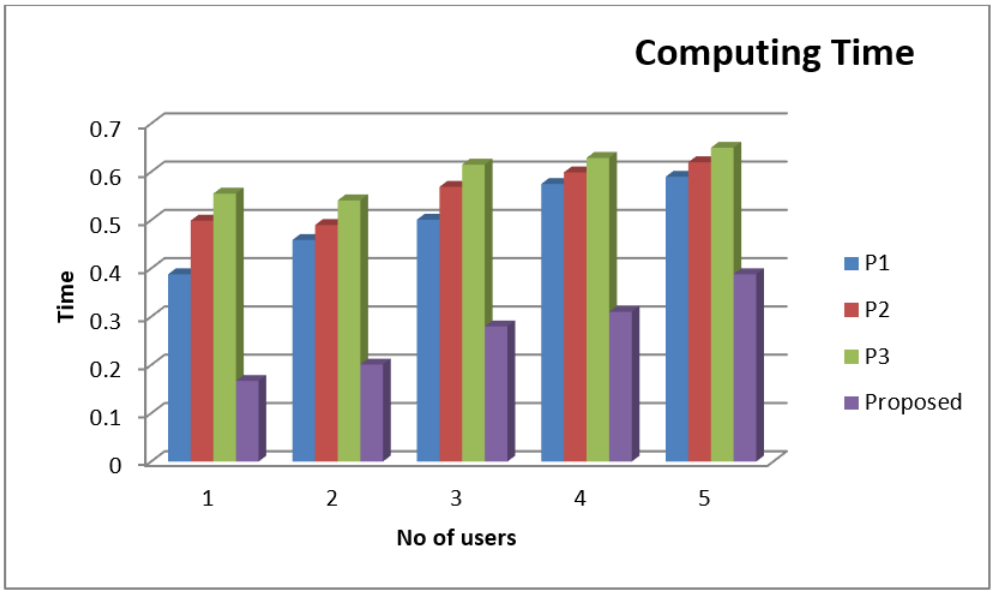

Figure 3. Comparison Chart of Computing Time

Table 3. Comparison Table of Read/Write Processing Throughput

P1	P2	P3	Proposed
0,145	0,423	0,286	0,5
0,193	0,457	0,345	0,543
0,222	0,494	0,377	0,588
0,276	0,52	0,434	0,621
0,301	0,552	0,471	0,666

A comparison table outlining the read/write execution throughput of 3 previous methods (P1, P2, and P3) and one planned approach is provided. The suggested approaches have higher values than the previous ones. Values for the suggested approach range from 0,5 to 0,666.

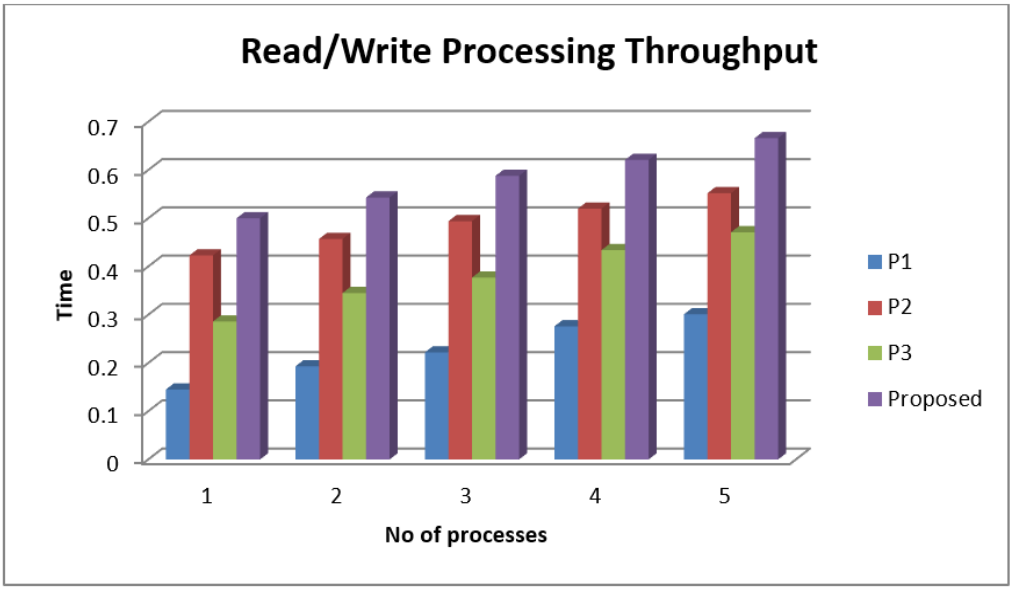

Figure 4. Comparison Chart of Read/Write Processing Throughput

The various values of the suggested approach and the current methods are displayed in the Read/Write Processing Throughput Comparison Chart. Record count on the x-axis and series level on the y-axis. The values of the suggested approach are greater than those of the previous technique when compared with it. Values for

https://doi.org/10.56294/dm2024245

the suggested approach range from 0,5 to 0,666 as shown in figure 4.

SHA Algorithm Results

Cross Dataset Sharing

Three methods Data Routing Technique, Multilayer Metadata, and Byte Index Chunking technique as well as the suggested technique are compared in the Cross Dataset Sharing Comparison as shown in table 4. The values of the suggested method are higher as compared to the previous method when compared to it.

Table 4. Comparison table of Cross Dataset Sharing

Byte Index Chunking Method	Data Routing Technique	Multilayer Metadata	Proposed
19,76	16,99	0,7	21,3
28,91	20,01	1,89	32,65
34,79	24,05	2,5	39,01
40,05	28,65	3,2	46,66
44,89	33,89	4,2	50,55

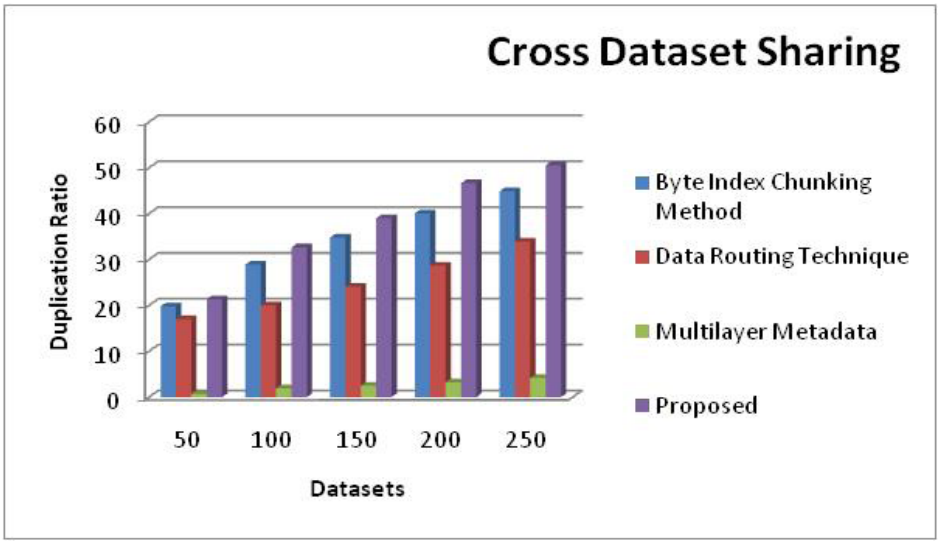

Figure 5. Comparison chart of Cross Dataset Sharing

The Cross Dataset Sharing comparison graph displays the disparities between the suggested and previous methods' values. Duplication Proportion on the y axis and datasets in the x axis. The values of the current approaches are quite low in comparison to the suggested techniques. Values between 21,3 and 50,55 are suggested as shown in figure 5.

Full File Duplicates

Three methods **namely**, the **Multilayer** Metadata, Data Routing Technique, Byte Index Chunking technique as well as the proposed technique are compared in the Full File Duplicates Comparison Table 5. The values of the suggested approach are higher than those of the previous method when compared to it. The values of the suggested method range from 10,26 to 30,06.

Table 5. Comparison table of Full File Duplicates

Byte Index Chunking Method	Data Routing Technique	Multilayer Metadata	Proposed
0,77	6,99	5,76	10,26
2,45	9,01	8,91	16,66
2,99	12,05	4,79	20,01
3,56	13,65	10,05	25,69
4,02	19,89	15,89	30,06

https://doi.org/10.56294/dm2024245

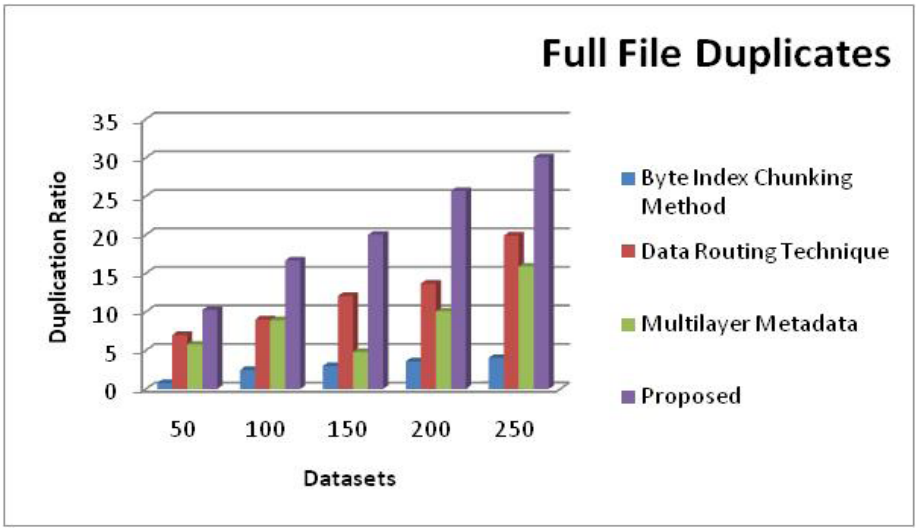

Figure 6. Comparison table of Full File Duplicates

The Full File Duplicates Comparison Chart displays the disparities between the suggested and previous methods' values. Duplication Proportion on the y axis and datasets in the x axis. The values of the current approaches are quite low in comparison to the suggested techniques. Values between 10,26 and 30,06 are considered as shown in figure 6.

Zero Chunk Removal

The suggested approach and the 3 currently utilizing methods Multilayer Metadata, Data Routing Technique, Byte Index Chunking approach are compared in the Zero Chunk Removal Comparison table 6. The values of the proposed approach are higher than those of the previous technique when compared to it. The values of the proposed approach range from 12,02 to 30,06

Table 6. Comparison table of Zero Chunk Removal

Byte Index Chunking Method	Data Routing Technique	Multilayer Metadata	Proposed
4,77	2,76	8,99	12,02
6,45	5,91	10,87	15,55
12,99	9,79	13,26	19,65
18,56	12,05	18,77	24,65
24,02	16,89	25,1	30,06

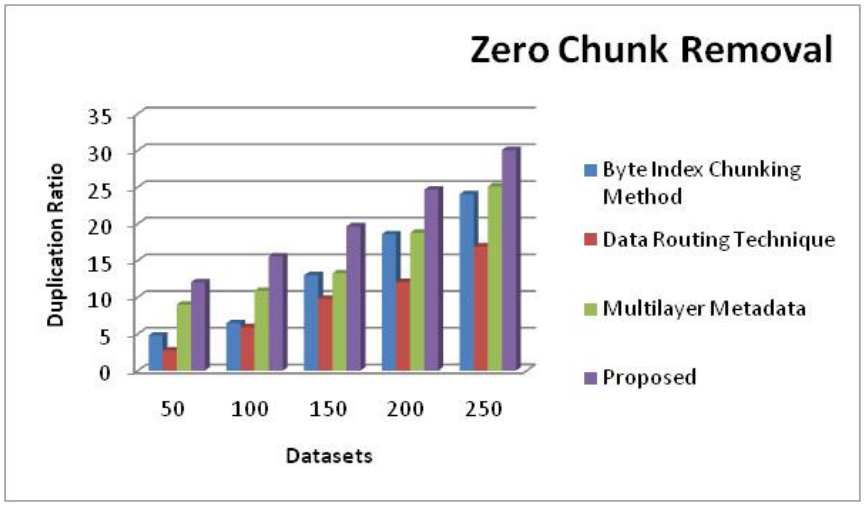

Figure 7. Comparison chart of zero Chunk Removal

https://doi.org/10.56294/dm2024245

As shown in figure 7, The Zero Chunk Removal comparison table displays the disparities between the suggested and previous methods' values. Duplication Ratio on the y axis and datasets in the x axis. The values of the current approaches are quite low in comparison to the suggested techniques. Values in proposal range from 12,02 to 30,06.

CONCLUSION

This study has tackled the pivotal challenge of harmonizing data de-duplication (DD) with privacy considerations within cloud storage systems. Recognizing the inherent conflict between optimizing capacity and safeguarding information privacy, we devised a method that harnesses a modified SHA algorithm within cloud servers. The proposed methodology incorporates hierarchical approved de-duplication, empowering users to conduct privilege-based duplicate checks before data upload, thereby alleviating privacy concerns. Through rigorous theoretical analysis and simulated investigations, we rigorously evaluated the efficiency and security of our proposed system. Our findings underscored notable enhancements in capacity utilization and reductions in transmission bandwidth, affirming the effectiveness of our approach in real-world scenarios.This research not only provides a concrete solution to the pressing issue of DD in cloud storage but also offers valuable insights into the intricate balance between data optimization and privacy preservation. By offering a robust methodology backed by empirical validation, our study contributes significantly to advancing the discourse on secure and efficient data management practices in cloud environments. In future research, further refinement of the hierarchical structure and exploration of dynamic privilege modifications could enhance the system's adaptability and privacy preservation capabilities. Additionally, extending empirical studies to real-world cloud environments would offer valuable validation and insights into practical implementations.

REFERENCES

1. Gurler CG, Savas SS, Tekalp AM. Variable chunk size and adaptive scheduling window for P2P streaming of scalable video. In: 2012 19th IEEE International Conference on Image Processing; IEEE; 2012.

2. Shen H, Li J. A DHT-Aided Chunk-Driven Overlay for Scalable and Efficient Peer-to-Peer Live Streaming. IEEE Transactions on Parallel and Distributed Systems. 2013 Nov;24(11):22 Oct 2012.

3. Bhagwat D, Eshghi K, Long DDE, Lillibridge M. Extreme Binning: Scalable, Parallel Deduplication for Chunk-based File Backup. In: 2009 IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems; IEEE; 2009.

4. Lin CH, Lee CY, Yeh YS, Chien HS, Chien SP. Generalized Secure Hash Algorithm: SHA-X. In: 2011 IEEE EUROCON - International Conference on Computer as a Tool; IEEE; 2011.

5. Lee SH, Shin KW. An Efficient Implementation of SHA processor Including Three Hash Algorithms (SHA-512, SHA-512/224, SHA-512/256). In: 2018 International Conference on Electronics, Information, and Communication (ICEIC); IEEE; 2018.

6. Mosquera ASB, Román-Mireles A, Rodríguez-Álvarez AM, Esmeraldas E del CO, Nieves-Lizárraga DO, Velarde-Osuna DV, et al. Gamification and development of social skills in education. AG Salud 2024;2:58-58. https://doi.org/10.62486/agsalud202458.

7. Ahmad I, Das AS. Analysis and Detection Of Errors In Implementation Of SHA-512 Algorithms On FPGAs. The Computer Journal. 2007 Nov;50(6).

8. Kunhu A, Al-Ahmad H, Taher F. Medical Images Protection and Authentication using hybrid DWT-DCT and SHA256-MD5 Hash Functions. In: 2017 24th IEEE International Conference on Electronics, Circuits and Systems (ICECS); IEEE; 2017.

9. Aziz MVG, Wijaya R, Prihatmanto AS, Henriyan D. HASH MD5 Function Implementation at 8-bit Microcontroller. In: 2013 Joint International Conference on Rural Information & Communication Technology and Electric-Vehicle Technology (rICT & ICeV-T); IEEE; 2013.

10. Solano AVC, Arboleda LDC, García CCC, Dominguez CDC. Benefits of artificial intelligence in companies. AG Managment 2023;1:17-17. https://doi.org/10.62486/agma202317.

11. Sediyono E, Santoso KI, Suhartono. Secure Login by Using One-time Password Authentication Based on

MD5 Hash Encrypted SMS. In: 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI); IEEE; 2013.

12. Wua H, Liua X, Tang W. A Fast GPU-based Implementation for MD5 Hash Reverse. In: 2011 IEEE International Conference on Anti-Counterfeiting, Security and Identification; IEEE; 2011.

13. Gonzalez-Argote J, Castillo-González W. Update on the use of gamified educational resources in the development of cognitive skills. AG Salud 2024;2:41-41. https://doi.org/10.62486/agsalud202441.

14. Kim WB, Lee IY, Ryou JC. Improving dynamic ownership scheme for data Deduplication. In: 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT); IEEE; 2017.

15. Bhalerao A, Pawar A. A Survey: On Data Deduplication for Efficiently Utilizing Cloud Storage for Big Data Backups. In: 2017 International Conference on Trends in Electronics and Informatics (ICEI); IEEE; 2017.

FINANCING

The authors did not receive financing for the development of this research.

CONFLICT OF INTEREST

None.

AUTHORSHIP CONTRIBUTION

Conceptualization: Manikandan Rajagopal.
Research: Rajendran Bhojan and Ramesh.
Drafting - original draft: Manikandan Rajagopal, Rajendran Bhojan & Ramesh R.
Writing - proofreading and editing: Ramesh R & Manikandan Rajagopal.