International Journal of Engineering Research and Advanced Technology (IJERAT)

DOI: 10.31695/IJERAT.2022.8.4.2

E-ISSN : 2454-6135

Volume.8, No. 4
April -2022

Techniques of Data Deduplication for Cloud Storage: A Review

Ayad Hasan Adhab[1], & Naseer Ali Hussien[2]

[1] Kufa University, Ph.D.in Computer Science and Mathematics College, Iraq

[2] Wasit University, College of Education for Pure Science, Iraq

ABSTRACT

With the rapid advancement of information technology and network, it's becoming increasingly difficult to keep up, as well as the rapid expansion of data center size, energy consumption as a percentage of IT investment is increasing. As the amount of digital data grows, so does the need for greater storage space, which drives up the cost and performance of backups. Traditional backup solutions don't have any built-in protection against duplicate data being saved up. Duplicate data backups severely lengthen backup times and consume needless resources. Data deduplication is critical for removing redundant data and lowering storage costs. Data deduplication is a new technique of compressing data that helps with storage efficiency while also proving to be a more efficient technique of dealing with duplicate data. Deduplication enables a single data copy to be uploaded to storage and subsequent copies to be provided with a pointer to the original stored copy. This paper consists of extensive literature survey and summarizes numerous storage approaches, concepts, and categories that are used in data reduplication. Also in this paper, the researchers carried out the survey for chunk based data deduplication techniques in detail.

Key Words: *Cloud Computing, Cloud Storage Service, Chunking Algorithm (CA), Data Deduplication, Chunking Method (CM).*

1. INTRODUCTION

With cloud computing's expanding data size; a data compression quantity could suppliers of assistance in lowering the expenses of running huge storage systems and conserving energy [1-4]. Cloud computing refers to everything that involves providing shared services facilities over the Internet. The three kinds of cloud computing (CC) solutions are Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS), the figure 1.1 illustrated types of cloud computing services (CCS) with examples [5]. The term cloud computing (CC) was inspired through the sign of a cloud that is widely utilized to appear the internet in diagrams and flowcharts. You store data in a remote repository instead of on your hard drive or other internal storage device. Your computer and the database (DB) are connected via the internet. Three distinct features of a cloud service can be discovered in modern hosting. It's available on demand, generally by the hour or minute; it's flexible, which means a customer can get as few or as a much service as they want at any one time; and it's completely managed through the supplier. A private cloud or a public cloud can be used. A public cloud authorizes anyone on the Internet to purchasing services. A private cloud is a private data or network center that serves a small number of people [6-8].

Data deduplication (DD) is a storage space-saving technology. Through recognizing excrescent data by comparing data chunks with hash values, saving only one copy, as well as logically pointing to another copies rather than keeping additional copies of the redundant data, logical pointers to other copies can be created. Data volume is reduced by deduplication, allowing network bandwidth and disk space to be lowered, lowering storage scheme costs and energy consumption. Data deduplication (DD) can be used in cloud storage at practically every place where data is stored or delivered. Several cloud service providers support disaster recuperation, and deduplication could be utilized to improve disaster recovery through replication data after it has been deduplicated to reduce replication time and bandwidth costs. Data deduplication can be used in cloud backup and archive storage to decrease network traffic and physical capacity. Furthermore, we must transport a big volume of replication memory image data throughout the live migration procedure. There are three primary migration performance criteria to consider: total amount of data transmitted, total immigration duration, and total service disruption. Service failure would result from a longer immigration time and unavailability. As a result, deduplication can help in migration. Deduplication could help you save space by reducing the amount of efficient data you save, like as virtual machine (VM) images. When utilizing deduplication in essential storage, it's important to evaluate how to balance among storage space savings and performance impact. Deduplication techniques, according to Mandagere et al., reverberate the achievement of deduplicated storage in terms of flexure factor, bandwidth, rebuilding, metadata overhead, and resource utilization [9-12].

IT as a Service (ITaas)			
IaaS	**PaaS**	**SaaS**	**StaaS**
Infrastructure as a Service	Platform as a Service	Software as a Service	Storage as a service
IT Services: ▪ Servers ▪ Network ▪ Storage ▪ Management ▪ Reporting	Application Building Blocks and Standards	Applications	Storage Services: ▪ Primary ▪ Backup ▪ Archive ▪ DR
Examples: BT Telstra T-Systems (ITaas)	Examples: Amazon EC2 Force.com Navitaire	Examples: Yahoo! E-mail Sales Force.com Google apps	Examples: Amazon S3 Nirvanix

Figure 1.1: Types of cloud computing services (CCS) with examples

Data deduplication (DD) is a compression technique. The name deduplication indicated to the overall purpose of a deduplication system: to locate duplicate data and delete repeated instances of that data. The storage needs for systems that deal with huge amounts of data can be dramatically reduced by successfully finding and deleting redundant data. This is especially true for systems that store many files with vast, overlapping sections, such as boilerplate or configuration files that are part of an operating system installation, or for systems that retain several copies of identical files, such as a system that conducts nightly backups. Even if storage capacity isn't an issue, deduplication can be useful for other reasons. For example, when transmitting data from a client to a server, a client that knows the server already has a duplicate of the data it wants to send might avert sending unneeded data and avert network overheads. While removing all duplicate data is fabulous objective, some replication may be tolerated in a system design for a variety of reasons. Allowing some redundant data, for example, may be required in order to terminate crucial activities in an acceptable period of time (identifying all duplicates may be costly!), or to tolerate system failures (if we lose the only copy of a file system superblock, our whole system is hosed). As a result, we will need a metric that indicates the effective "data savings" of a deduplication system. We use the deduplication ratio for this. If B bytes of data are supplied to a storage system S, which unambiguously displays the data using D bytes, then the deduplication ratio of S is determined in equation (1):

$$D_S = \frac{B}{D} \quad \ldots\ldots\ldots (1)$$

A more effective deduplication system will have a higher deduplication ratio (i.e., it stores more data using fewer bytes). However, D_S isn't the only metric that matters. Resiliency, scalability, throughput and Latency are all factors in system design, and the best balance of accomplishment and deduplication effectiveness will be context dependent [13].

The deduplication technicalities are used on various kinds of data, like as image, video, and text data. Different storage formats and implicit features exist for each of the three categories of data. Deduplication techniques use various approaches to detect and delete duplicate data depending on the type of data. As a result, data type is critical in the development of deduplication techniques. When it comes to reading, discovering, and matching information, the format is crucial. To discover duplication in executable files, it is necessary to match bits on a bit-by-bit basis. Due to the diverse formats of data, the procedures for checking duplicates in image, video, and text have various processes. To achieve high data availability, a large distributed storage system maintains a minimal number of data replicas termed the replication factor. To decrease storage requirements, storage costs, computation, and energy, any duplicate data exceeding the replication factor is eliminated. Deduplication technicalities for large distributed storage schemes have obtained traction in industry and academia as a result of these enormous benefits to industry. However, due to the efficacy and effectiveness of data matching technicalities, these strategies face problems. Academic and industry researchers are collaborating to development efficacy distributed deduplication technicalities [1,14,15,16,17].

DOI : 10.31695/IJERAT.2022.8.4.2

Computer vision (CV), image processing (IP) -rely applications, database systems (DB), cloud computing (CC), face recognition (FR), pattern matching (PM), natural language processing (NLP), big data applications (BDA), and parallel computing (PC) are just a few of the fields where data deduplication can be used in research and other applications. Figure 1.2 depicts the main phases of the data deduplication (DD) technique, where we could see that the data values are reiterated at first, and then the values are determined a "specific value of function" utilizing a convenient algorithm like a hash function (HF) or something identical. The following phase compares all values to the function or standard value, and if the same values are reported, those values are neglected. This could be seen in the final step's result, where the values that were previously reiterating have been negligent and exchanged through a single data value. Figure 1.2 represents a exemplary example of how data deduplication (DD) stages function in any data science or database management system application (DBMS) [18].

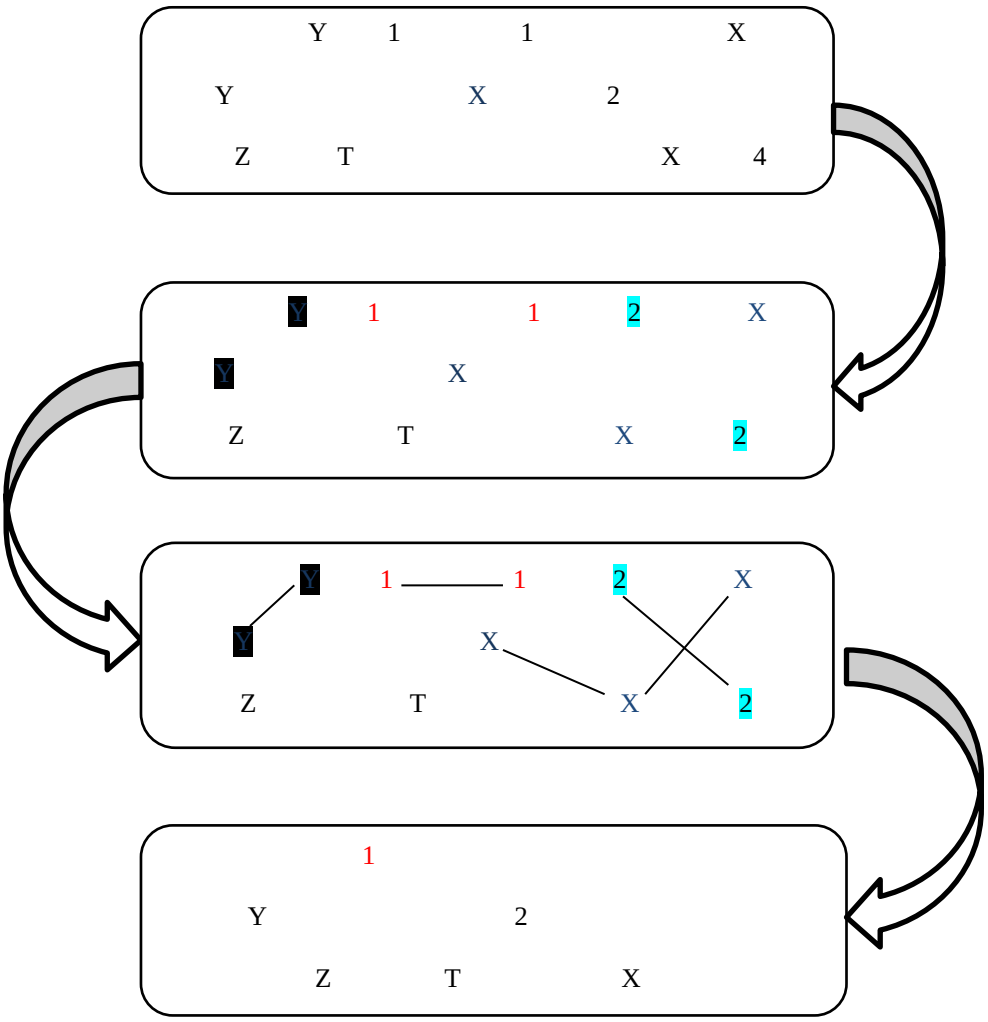

Figure 1.2: The fundamentals of data deduplication

Various kinds of algorithms and methods are employed in data deduplication (DD) strategies and approaches for a variety of applications and data science, but two primary kinds of methods are most frequently utilized in a significant number of applications, namely hash-rely and content-aware methods (CA). Deduplication methods are classified as inline or postprocessing depending on how the data is stored and backed up. The hashing function or algorithm is widely utilized in hash-based methods, and it is also used to identify "chunks" of data. The chunks are small sets of data fragments that depict small related groups and help to ignore the repeated data. The Figure 1.3 depicts a typical schematic of data deduplication using the hashing approach. Different hash function values are allocated to various slices or chunks of data and after comparing a hash value (HV) with all another slices, the updated hash values are returned. This procedure is reiterated until the value convergence of assignment to a state of no change. A hash value (HV) usually requisites a particular number of bits, and when subsequent chunks of data search for and locate chunks with the same hash value; the chunks are viewed as duplicate data and aren't kept in the data deduplication

DOI : 10.31695/IJERAT.2022.8.4.2

(DD) procedure. If the hash value (HV) is unique and not existing among previously recorded values, the hash value is saved, and the matching data chunk is examined and saved in databases (DB). In figure 1.3 we can observe that the next level updates the new slice "K" with a new hash value (HV). The novel hash value is employed as novel indexing value and serves as a crucial indicator component in subsequent data slices or chunks comparisons [18,19].

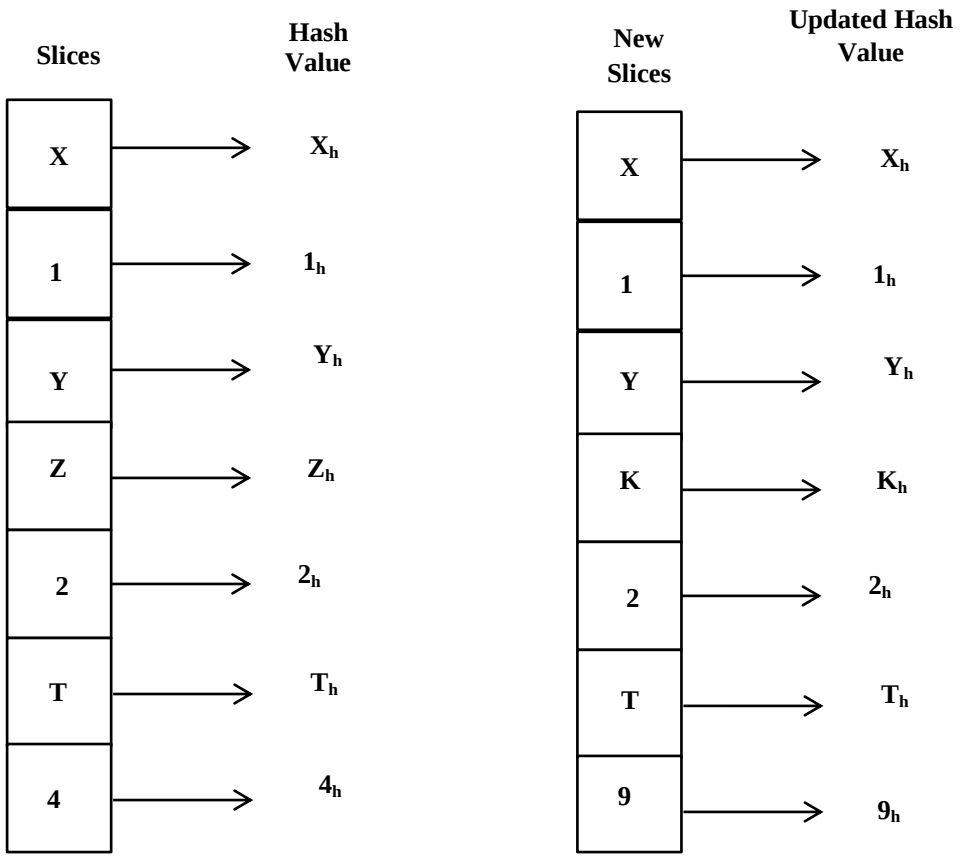

Figure 1.3: Hash values are assigned.

Data patterns are explored in the content-aware method, and when the same patterns are reported, the patterns are obviously as though it were reiterate data. In this technique, a file name mechanism is used, which means that different data patterns are given appropriate file names, and new patterns are given new file names. When a match of files is identified in this method, a bitwise comparison is performed, and similarity is scored rely on this comparison; this implies that identical file names aren't the only criterion in the procedure of discovering duplicate data; the bitwise matching result is also important. Due the number of files might be in the millions, database knowledge becomes critical in this technique. As a result, memory indexing is required to achieve efficient file management and updating, as well as to avoid duplicate data. The steps in this deduplication technique are self-explanatory and may be seen in Figure 1.4 [18, 20].

These techniques are based on the amount of time it takes to deduplicate data. The "inline" method is used when the process is performed as the data is being saved, while the "postprocessing" method is used after the data initial processed in the deduplication operation has been stored. The performance of the inline method is independent of the type of method utilized for data deduplication, whether it is hash-rely or content-aware (CA), but the performance of the postprocessing method is dependent on the kind of method utilized for data deduplication (DD) [20].

DOI : 10.31695/IJERAT.2022.8.4.2

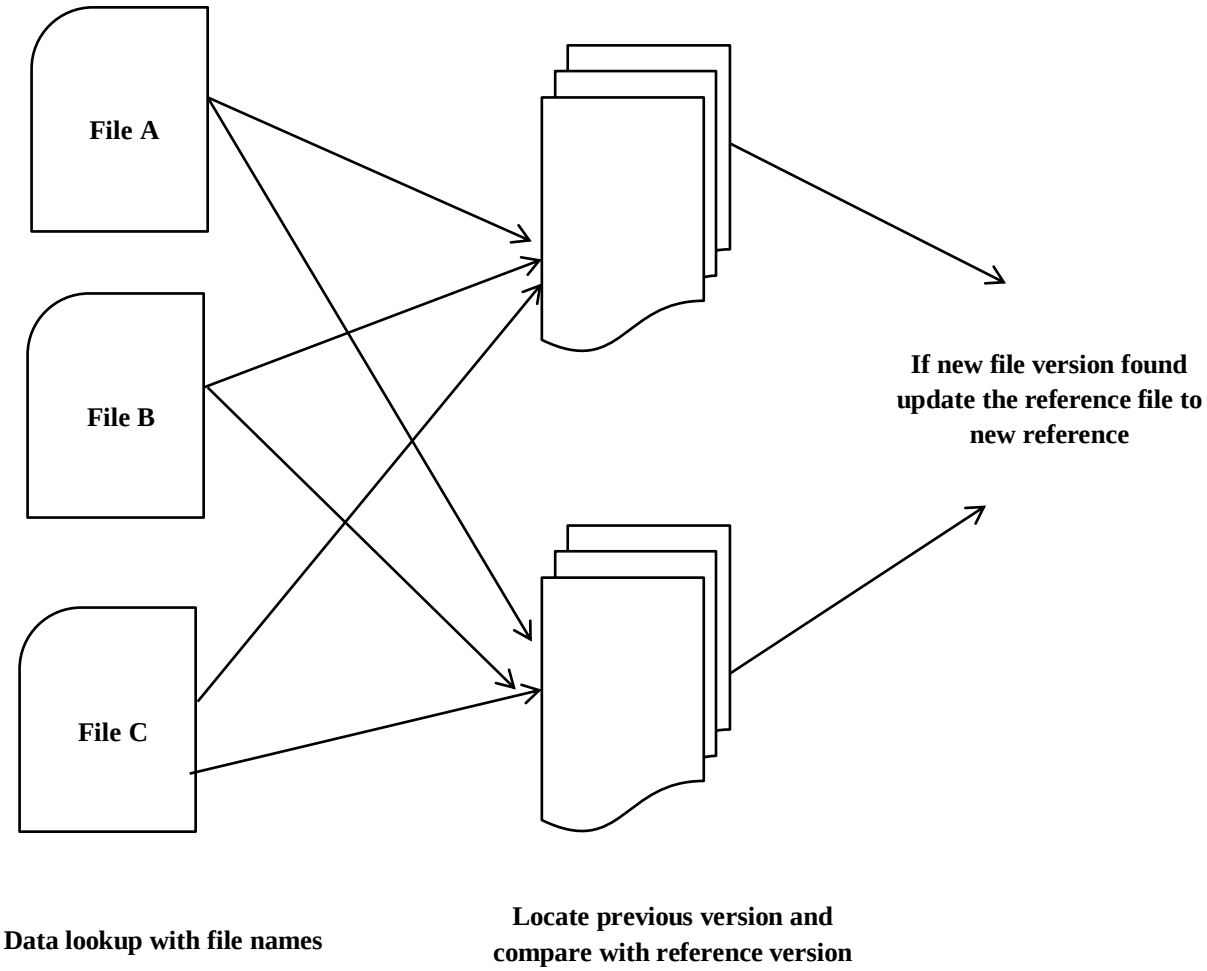

Figure 1.4: File name concept of deduplication

2. LITERATURE SURVEY

Various studies relating to the data deduplication have been investigated recently. A few of such studies are discussed in this section.

2.1 Data Deduplication

N. Kumar and S. Jain [21] in this paper, we suggest Differential Evolution (DE) rely on technique which is optimized Two Thresholds Two Divisors (TTTD-P) Content Defined Chunking (CDC) to decrease the number of computing process utilizing single dynamic optimal parameter divisor D with optimal threshold value exploiting multi-process nature of TTTD. The TTTD technique provides an additional backup divisor D′ that has a higher probability of discovering cut points to reduce chunk size variance; nevertheless, introducing a second divisor reduces chunking throughput. To this purpose, Asymmetric Extremum (AE) considerably enhances chunking throughput through overcoming Rabin and TTTD boundary shift problems by using local extreme values in a variable-sized asymmetric window, while maintaining a near-identical deduplication ratio (DR). On Hadoop Distributed File System, we suggest DE-rely on TTTD-P optimized chunking to maximize chunking throughput while increasing DR; The use of a scalable bucket indexing strategy minimizes the time it takes to find and declare duplicated hash values (HV). chunks about 16 times greater than Rabin CDC, 5 times greater than AE CDC, and 1.6 times greater than FAST CDC (HDFS).

W. Leesakul et al., [22] in this research, suggested a dynamic data deduplication (DD) strategy for cloud storage, in arrange to strike a balance among changing storage efficacy and criteria for fault tolerance, as well as to increase cloud storage performance. We adjust the amount of copies of files in real time to match the changing degree of QoS. Show results of the experiments reveal

DOI : 10.31695/IJERAT.2022.8.4.2

that our suggested scheme works effectively and can deal with scalability issues. We also intend to keep track of how users' file demands change. We also intend to assess the system's availability and performance.

Y. Fan et al., [23] in this research article, every cloud user is given a set of privileges in our system, and deduplication can only be done if and only if the cloud users have the appropriate privileges. Furthermore, our system improves the capacity of like cryptosystems to resist selected plaintext and selection ciphertext attacks by augmenting convergent encryption with users' privileges and relying on TEE to provide secure key management. Our system is secure sufficient to facilitate data deduplication (DD) as well as protecting the privacy of sensitive data, according to a security analysis. Moreover, we create a prototype of our system and analyze its performance. Experiments reveal that our system overhead is practical in real-world scenarios.

M. Oh et al., [24] in this paper, suggest a novel deduplication technique that is extremely compatible and scalable with the exhausted storage currently in use. Our deduplication technique, in particular, uses a double hashing algorithm (HA) that takes advantage of hashes employed through the implicit scale-out storage, which overcomes the limitations of present fingerprint hashing (FH). Furthermore, our approach combines file system and deduplication meta-information into a single object, and it manages the deduplication ratio online through initial aware of post-processing-related scheme demands. On open source scale-out storage, we implemented the proposed deduplication approach. When executing a variety of standard storage workloads, the experimental findings illustrate that our solution could save greater than 90% of total storage space while providing the same or similar performance as traditional scale-out storage.

P. Anitha et al., [25] in this article, suggested method uses an access control mechanism on data belonging to privileged authorities to safeguard the data deduplication (DD) process. The data that will be outsourced will be encrypted files. The secure authorities are given access control mechanisms to do data deduplication (DD) on the data that was outsourced. Encryption techniques are used in the Access Control Mechanism. It employs convergent randomized encryption and a reliable distribution of owning party keys to allow the cloud service provider to manage outsourced data access even when control shifts on a regular basis. This prohibits data from being leakage not only to individuals who have had their rights to the data revoked, however also to a degree to a trustworthy but dubious cloud storage server. In addition, the suggested technique safeguards data integrity against attacks rely on label discrepancies. As a precaution, the suggested technique has been changed to improve security. The suggested scheme is virtually as effective as the current ones, with only a minor raise in computational expenses, according to the efficiency study.

R. Kirubakaran et al., [26] in this article, present a cloud-based technique for achieving deduplication of a huge amount of data available. The approach includes data deduplication before uploading to cloud storage as well as data reverse deduplication when obtaining the required data. Any of the algorithms indicated in the literature survey can be used to implement the suggested technique. The model is more effective and accurate than existing deduplication systems because of the type of algorithm utilized.

M. V. Maruti et al., [27] in this paper, proposed data deduplication technique, the main goal of this technique is to delete reiterate data from the cloud. It can also aid in the reduction of bandwidth and storage space usage. This suggested technique is to delete reiterate data; however each user has their own unique token and has been allocated various privileges based on the duplication check. The hybrid cloud architecture is used to achieve cloud deduplication. The proposed technique is more secure and uses fewer cloud resources. It was also demonstrated that, when compared to the standard Deduplication technique, the proposed system had a low overhead in duplicate removal. On this work, both content level and file level deduplication of file data is examined in the cloud.

K. Vijayalakshmi and V. Jayalakshmi [28] in this research article, suggest data duplication in clouds, which is managed using the deduplication technique. Although some deduplication techniques are used to prevent data redundancy, they are inefficient. The major goal of this research is to gain enough knowledge and a decent concept of deduplication techniques through reviewing existent ways, and this work may aid future research in establishing effective cloud storage management (CSM) solutions for researchers and practitioners.

X. Xu et al., [29] in this article, focus on non-center cloud storage data deduplication and present a new two-side data deduplication (DD) mechanism. Duplicate data is first recognized and stopped from being uploaded to the cloud data center on the client side, and then new duplicate data is detection and erased on the cloud server side. The Chord algorithm (CA) is optimized using a compressed finger table and search optimization of the finger table, and it is then utilized to build a logical network that makes searching for fingerprints in storage nodes considerably faster. Show the results illustrate that the enhanced Chord algorithm outperforms the original in terms of the number of fingerprint values and the cost of searching time, and the suggested

DOI : 10.31695/IJERAT.2022.8.4.2

two-side data deduplication (DD) technique outperforms the traditional data deduplication (DD) mechanism in terms of deduplication rate.

X. Xu and Q. Tu [30] in this work suggest a deduplication scheme architecture for cloud storage environments (CSE), as well as the procedure for averting duplication on the client side at the file and chunk levels. DelayDedupe, a delayed target deduplication strategy rely on chunk-level deduplication and chunk access frequency, is suggested to decrease response time in storage nodes (Snodes). When used in conjunction with replica arrangement, this technique evaluates whether fresh multiplied chunks for data update are hot and, if they aren't, eliminates the hot duplicated chucks. The findings of the experiment show that the DelayDedupe method may successfully minimize response time while also balancing the storage demand on Snodes.

2.2 Chunking Algorithm for Data Deduplication

M. Ellappan and S. Abirami [31] in this paper, suggest a novel chunking algorithm (CA) called Dynamic Prime Chunking (DPC). DPC's major purpose is to modify the window size during the prime value dynamically rely on the maximum and minimal chunk size. According to the results of the paper, DPC in the deduplication scheme gives good throughput while avoiding large chunk variance. The multimedia and operating system datasets were used for implementation and experimental evaluation. Existing algorithms such as AE, MAXP, TTTD, and Rabin have been compared to DPC. The performance indicators we looked at were throughput, chunk count, Bytes Saved per Second (BSPS), chunking time, processing time and Deduplication removal Ratio (DER). According to the study of the data, BSPS and throughput have both improved. To begin, DPC boosts throughput performance through greater than 21% when compared to AE. Second, BSPS improves performance by up to 11% over the previous AE method. In comparison to existing Content Defined Chunking (CDC) algorithms, our algorithm reduces total processing time and achievement higher deduplication effectiveness due to the aforementioned rationale.

H. A. Jasim and A. A. Fahad [32] in this work uses a novel fingerprint function (FF), a multi-level hashing and matching mechanism, and a novel indexing technicality to hold metadata to progress the TTTD chunking algorithm. These novel technicality include four hashing algorithms to handle the collision issue, as well as adding a novel chunk stipulation to the TTTD chunking criterion to improve the amount of small chunks and hence the Deduplication Ratio. This modification enhances the TTTD algorithm's Deduplication Ratio while also lowering the algorithm's system resource requirements. In terms of deduplication ratio, metadata size, and execution time, the suggested technique is put to the test.

W. Xia et al., [33] in this article, suggest FastCDC, a Fast and effective CDC approach, which constructs and enhances on the latest Gear-based on CDC technique, one of the fastest CDC techniques to our knowledge. FastCDC's main idea is to integrate five key mechanics: gear-rely on rapid rolling hash, improving and simplifying Gear hash (GH) verdict, skipping sub-minimal chunk cut-points, normalizing the chunk-size distribution in a small specific region to address the issue of reduction deduplication ratio caused by cut-point skipping. FastCDC is around 10 times quicker than the best open-source Rabin-based on CDC, and about 3 times greater than the state-of-the-art Gear- and AE-rely on CDC, while obtaining almost the same deduplication ratio as the standard Rabin-rely solution, according to our evaluation results.

W. Xia et al., [34] in this research article, suggest FastCDC, a Fast and effective Content Defined Chunking method, for data deduplication-rely on storage schemes. FastCDC's main idea is to integrate five key mechanics: gear-rely on rapid rolling hash, improving and simplifying Gear hash (GH) verdict, skipping sub-minimal chunk cut-points, normalizing the chunk-size distribution in a small specific region to address the issue of reduction deduplication ratio caused by cut-point skipping, and, last but not least, rolling two bytes every time to speed CDC. FastCDC is 3-12X faster than state-of-the-art CDC approaches when employing a combination of the five strategies, while achieving approximately the same or even greater deduplication ratio than the standard Rabin-rely CDC. Furthermore, our research into the deduplication throughput of FastCDC-rely Destor found that FastCDC aids in achieving 1.2-3.0X higher throughput than Destor using state-of-the-art chunkers.

Y. Zhang et al., [35] in this work, suggested a novel CDC algorithm indicated the Asymmetric Extremum (AE) algorithm. The major idea behind AE is relies the observance that in dealing with the boundaries-shift issue, the maximum value in an asymmetric local domain is improbable to be exchanged through a novel extreme value, which motivates AE's utilize of asymmetric (instead of symmetric, as in MAXP) local domain to distinguish cut-points and attain high chunking throughput while minimizing chunk size variance. According the result, AE addresses the issues of low chunking throughput in MAXP and Rabin, as well as excessive chunk-size volatility in Rabin, at the same time. AE enhances the throughput speed of state-of-the-art CDC algorithms by 3x while achieving equivalent or greater deduplication efficacy, according to experimental results rely on four real-world datasets.

DOI : 10.31695/IJERAT.2022.8.4.2

Z. Xu and W. Zhang [36] in this paper, use Content Defined Chunking (CDC), which plays an essential role in basic storage and backup systems. Rabin CDC's chunking speed and deduplication ratio are lower due to time-consumption byte-by-byte calculations on data streams, which is central processing unit (CPU) expensive and relies on sliding window. QuickCDC improves CDC chunking speed, deduplication ratio, and throughput by combining three methods. Initially, QuickCDC can move instantly to the chunk boundaries of duplicate chunks that arise frequently. The mapping of the duplicate chunk's first n bytes and last m bytes to chunk length must be registered. The first n bytes and last m bytes of the current chunk are checked to see if they are in the mapping table when chunking is performed. QuickCDC can skip relevant chunk lengths (CL) if they are in the mapping table. Second, QuickCDC can skip the minimal chunk length for unique chunks. Finally, QuickCDC may dynamically alter mask bits length such that chunk length (CL) is permanently more than the minimal chunk length and is distributed in a limited particular location. When the current chunk length (CL) is less than the expected chunk length (CL), we should use longer mask bits, and when the current chunk length (CL) is more than the expected chunk length (CL), we should utilize shorter mask bits. Experiments show that QuickCDC's chunking speed is 11.4x that of RapidCDC, and the associated deduplication ratio is somewhat increased, with a maximum deduplication ratio improvement of 222.3% and a throughput improvement of 111.4%.

P. Minishapriya and S. Maheswari [37] in this research article, use throughput, provide performance analyses of Rabin-rely on chunking and Rapid Asymmetric Maximum (RAM) chunking. The procedure considers an inhabitance dataset case study research in which the state and county take into account the parameter for chunking and deduplication analyses.

S. Luo and M. Hou [38] in this paper, suggest a new chunk coalescing algorithm (CCA), this refers to the minimal and maximum amount of subchunks which should be coalesced to form superchunks (SC). Experiments demonstrate that our algorithm eliminates the expenses of the chunk coalescing (CC) procedure and speeds up the entire data deduplication procedure.

H. Wu et al., [39] in this article, suggests a sampling-rely on chunking algorithm and improve SmartChunker, a tool to predict the appropriate chunking configuration for deduplication schemes. SmartChunker's effectiveness and efficacy have been demonstrated in real-world datasets.

P. Krishnaprasad and B. A. Narayamparambil [40] in this paper, suggested a novel Dual Side Fixed Size Chunking (DSFSC) algorithm to achieve a rising deduplication ratio for comparison to conventional FSC. This approach can successfully be utilized for audio or video files to produce a best Deduplication ratio without requiring computationally exorbitant variable size chunking or content determined chunking. This data compression will save network bandwidth while also allowing more data to be stored on a given quantity of cloud or disk storage. Storage management and energy expenses will be reduced if storage requests are reduced.

3. COMPARISON OF SCHEMES

Table 1: Description of the comparison between previous systems for data deduplication

Ref	Authors	Year	Algorithm/method/ Techniques	Performance
[21]	N. Kumar and S. Jain	2019	Differential Evolution (DE), Two Thresholds Two Divisors (TTTD-P) algorithm, Two Thresholds, Two Divisors with Switch (TTTD-S) Algorithm, Two Thresholds, Two Divisors with Optimal Parameter (TTTD-P) Algorithm, Rabin Algorithm	Hash values (chunks about 16 times greater than Rabin CDC, 5 times greater than AE CDC, and 1.6 times greater than FAST CDC (HDFS).)
[22]	W. Leesakul et al.,	2014	Dynamic Data Deduplication	experiments reveal that our proposed system works effectively
[23]	Y. Fan et al.,	2019	deduplication system that includes the processes of duplicate checking, proofs of ownership and convergent encryption. (Preliminaries, Adversarial model)	implement the security analysis and also performance evaluation is effective and feasible in practice
[24]	M. Oh et al.,	2018	a novel deduplication technique (a global deduplication design for current shared-nothing scale-out storage system)	experimental findings illustrate that our solution could save greater than 90% of total storage space

DOI : 10.31695/IJERAT.2022.8.4.2

[25]	P. Anitha et al.,	2021	secure rising scalable data deduplication architecture (utilizing authorized accessing control mechanisms in the outsourced data), encryption is utilizing RSA	The system is virtually as successful as the existing ones (minor increase in computational overhead)
[26]	R. Kirubakaran et al	2015	a cloud-rely technique for deduplication of huge data	The model is more efficient and accurate compared to that of the existent deduplication techniques.
[27]	M. V. Maruti et al.	2015	novel duplication check technique that configuration the token for the private file and check content level deduplication	the system achieve is 98 %
[28]	K. Vijayalakshmi and V. Jayalakshmi	2021	data duplication (DD) in clouds	the system achieves efficient knowledge and a good idea concerning deduplication techniques
[29]	X. Xu et al	2016	focus on non-center cloud storage data deduplication, a new two-side data deduplication (DD) technique, Chord algorithm	two-side data deduplication (DD) technique outperforms the traditional data deduplication technique in terms of deduplication rate
[30]	X. Xu and Q. Tu	2015	deduplication scheme architecture for cloud storage environments (CSE)	DelayDedupe method may successfully minimize response time while also balancing the storage demand on Snodes.

Table 2: Description the comparison between previous systems for chunking algorithm

Ref	Authors	Year	Chunking Method	Performance
[31]	M. Ellappan and S. Abirami	2021	Dynamic Prime Chunking (DPC), Existing algorithms such as AE, MAXP, TTTD, and Rabin have been compared to DPC	DPC's durable performance over the another existent algorithms in terms of BSPS and the efficacy of the backup storage scheme
[32]	H. A. Jasim and A. A. Fahad	2018	a novel fingerprint function (FF), a multi-level hashing and matching mechanism, TTTD chunking algorithm	good deduplication ratio and rapid execution time, efficacy of the suggest algorithm was evaluated utilizing two relatively datasets
[33]	W. Xia et al.,	2016	FastCDC, a Fast and effective CDC approach	FastCDC is around 10 times quicker than the best open-source Rabin-based on CDC, and about 3 times greater than the state-of-the-art Gear- and AE-rely on CDC
[34]	W. Xia et al.,	2020	FastCDC, a Fast and effective Content Defined Chunking (CDC) method	Achieving 1.2-3.0X higher throughput than Destor using state-of-the-art chunkers.
[35]	Y. Zhang et al.,	2015	a novel CDC algorithm indicated the Asymmetric Extremum (AE) algorithm	Illustrates the superior and robust performance of AE in terms of deduplication efficacy and chunking throughput over the state of-the-art Rabin and MAXP chunking algorithms.
[36]	Z. Xu and W. Zhang	2021	Content Defined Chunking (CDC)	Show that QuickCDC's chunking speed is 11.4x that of RapidCDC, and the associated deduplication ratio is somewhat increased, with a maximum deduplication ratio improvement of 222.3% and a throughput improvement of 111.4%.
[37]	P. Minishapriya and S. Maheswari	2018	analyses of Rabin-rely on chunking and Rapid Asymmetric Maximum (RAM) chunking	illustrate that RAM offers minimize computational expenses compared to the Rabin algorithm (RA)
[38]	S. Luo and M. Hou	2013	a new chunk coalescing algorithm (CCA)	demonstrate that our algorithm eliminates the expenses of the chunk coalescing procedure and enhance the efficacy of hash-comparison
[39]	H. Wu et al.,	2018	a sampling-based on chunking algorithm and	illustrate that a sampling-based chunking

DOI : 10.31695/IJERAT.2022.8.4.2

			improve SmartChunker	algorithm and enhance SmartChunker application-specified chunk configurations
[40]	P. Krishnaprasad and B. A. Narayamparambil	2013	a novel Dual Side Fixed Size Chunking (DSFSC) algorithm	illustrates that the algorithm can completely utilized for video and audio deduplication, that is having the higher potential for the contiguous modulation. DSFSC can be efficiently built on DeDu to set up a cloud with fixed size chunking deduplication.

4. CONCLUSION

In this paper, we presented various techniques used to analyze many data deduplication in cloud computing and chunking algorithm for data deduplication during the period (2013-2021). Cloud storage researchers have been drawn to data deduplication. Existing works usually adopt the chunking algorithm for data deduplication. Those schemes of data deduplication are thoroughly investigated and analyzed in order to improve the elimination data deduplication for cloud computing efficiency by using various several algorithms such as chunking algorithm for data deduplication and to ensure improve performance. According to the findings of this work, each of these techniques are useful for data deduplication in cloud computing. Each plan is unique in its approach, which might be convenient for a diversity of purposes. Recently, novel technology of data deduplication is being developed on a daily basis, and recently, suggested chunking algorithm for data deduplication have also improve performance the cloud computing. Each technology its own set of advantages and disadvantages and therefore novel technologies are beginning developed in the future.

ACKNOWLEDGMENT

The authors would like to thank University of Kufa (https://uokufa.edu.iq/, Baghdad- Iraq) and University of Wasit (https://uowasit.edu.iq/university-of-wasit//, Baghdad- Iraq) for its support in this work.

REFERENCES

[1] R. Kaur, I. Chana, and J. Bhattacharya, "Data deduplication techniques for efficient cloud storage management: a systematic review," The Journal of Supercomputing, vol. 74, no. 5, 2018, pp. 2035-2085, doi.org/10.1007/s11227-017-2210-8.

[2] M. Gu, X. Li, and Y. Cao, "Optical storage arrays: a perspective for future big data storage," Light: Science & Applications, vol. 3, no. 5, 2014, pp. e177-e177, doi:10.1038/lsa.

[3] H. S. Chyad, R. A. Mustafa, and D. N. George, "Cloud resources modelling using smart cloud management," Bulletin of Electrical Engineering and Informatics, vol. 11, no. 2, April 2022, pp. 1134~1142, ISSN: 2302-9285, doi: 10.11591/eei.v11i2.3286.

[4] H. S. Chyad, R. A. Mustafa, and K.T. Saleh, "Subject Review: Cloud Computing using RSA Algorithm", International Journal of Engineering Research and Advanced Technology (IJERAT), Volume.7, No. 7, July - 2021, ISSN: 2454-6135, doi: 10.31695/IJERAT.2021.3731.

[5] J. Wu, L. Ping, X. Ge, Y. Wang, and J. Fu, "Cloud storage as the infrastructure of cloud computing," in 2010 International conference on intelligent computing and cognitive informatics, 2010, pp. 380-383, DOI 10.1109/ICICCI.2010.119, IEEE.

[6] H. S. Chyad, R. A. Mustafa, and K. T. Saleh, "Study and Implementation of Resource Allocation Algorithms in Cloud Computing," International Journal of Engineering & Technology, vol. 7, no. 4.28, pp. 591-594, 2018, doi: 10.14419/ijet.v7i4.28.25394.

[7] A. A. Maryoosh, R. S. Mohammed, and R. A. Mustafa, "Subject Review: Cloud Computing Security Based on Cryptography," International Journal of Engineering Research and Advanced Technology-IJERAT (ISSN: 2454-6135), vol. 5, no. 9, pp. 20-23, 2019, doi: 10.31695/IJERAT.2019.3569.

[8] P. A. Kumar, E. Pugazhendhi, and K. V. Lakshmi, "Cloud Data Storage Optimization by Using Novel De-Duplication Technique," in 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT), 2022, pp. 436-442, doi: 10.1109/ICSSIT53264.2022.9716508, IEEE.

DOI : 10.31695/IJERAT.2022.8.4.2

[9] V. Javaraiah, "Backup for cloud and disaster recovery for consumers and SMBs," in 2011 Fifth IEEE International Conference on Advanced Telecommunication Systems and Networks (ANTS), 2011, pp. 1-3: IEEE.

[10] 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 2010 39th International Conference on Parallel Processing, 2010, pp. 614-623, doi: 10.1109/ICPP.2010.69, IEEE.

[11] S. K. Bose, S. Brock, R. Skeoch, N. Shaikh, and S. Rao, "Optimizing live migration of virtual machines across wide area networks using integrated replication and scheduling," in 2011 IEEE International Systems Conference, 2011, pp. 97-102: IEEE.

[12] S. K. Bose, S. Brock, R. Skeoch, and S. Rao, "CloudSpider: Combining replication with scheduling for optimizing live migration of virtual machines across wide area networks," in 2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2011, pp. 13-22, doi: 10.1109/CCGrid.2011.16, IEEE.

[13] W. Jannen, "Deduplication: Concepts and Techniques," 2020.

[14] J. Paulo and J. Pereira, "A survey and classification of storage deduplication systems," ACM Computing Surveys (CSUR), vol. 47, no. 1, 2014, pp. 1-30, doi: http://dx.doi.org/10.1145/2611778.

[15] B. Mao, H. Jiang, S. Wu, Y. Fu, and L. Tian, "Read-performance optimization for deduplication-based storage systems in the cloud," ACM Transactions on Storage (TOS), vol. 10, no. 2, pp. 1-22, doi:http://dx.doi.org/10.1145/2512348, 2014.

[16] R. Di Pietro and A. Sorniotti, "Proof of ownership for deduplication systems: a secure, scalable, and efficient solution," Computer Communications, vol. 82, 2016, pp. 71-82, http://dx.doi.org/10.1016/j.comcom.2016.01.011.

[17] J. Wang and X. Chen, "Efficient and secure storage for outsourced data: A survey," Data Science and Engineering, vol. 1, no. 3, 2016, pp. 178-188, doi: 10.1007/s41019-016-0018-9.

[18] T. T. Thwel and G. Sinha, Data Deduplication Approaches: Concepts, Strategies, and Challenges. Academic Press, 2020.

[19] A. Gracia, S. González, V. Robles, and E. Menasalvas, "A methodology to compare dimensionality reduction algorithms in terms of loss of quality," Information Sciences, vol. 270, pp. 1-27, 2014.

[20] L. Xu, "Online Deduplication for Distributed Databases," Carnegie Mellon University, 2016.

[21] N. Kumar and S. Jain, "Efficient data deduplication for big data storage systems," in Progress in Advanced Computing and Intelligent Engineering: Springer , 2019, pp. 351-371, https://doi.org/10.1007/978-981-13-0224-4_32.

[22] W. Leesakul, P. Townend, and J. Xu, "Dynamic data deduplication in cloud storage," in 2014 IEEE 8th International Symposium on Service Oriented System Engineering, 2014, pp. 320-325 , doi: 10.1109/SOSE.2014.46, IEEE.

[23] Y. Fan, X. Lin, W. Liang, G. Tan, and P. Nanda, "A secure privacy preserving deduplication scheme for cloud computing," Future Generation Computer Systems, vol. 101, pp. 127-135, 2019.

[24] M. Oh et al., "Design of global data deduplication for a scale-out distributed storage system," in 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), 2018, pp. 1063-1073, doi:10.1109/ICDCS.2018.00106IEEE.

[25] P. Anitha, R. Dhanushram, D. H. Sudhan, and T. Indhresh, "Security Aware High Scalable paradigm for Data Deduplication in Big Data cloud computing Environments," in Journal of Physics: Conference Series, 2021, vol. 1916, no. 1, p. 012097: IOP Publishing, doi:10.1088/1742-6596/1916/1/012097.

[26] R. Kirubakaran, C. M. Prathibhan, and C. Karthika, "A cloud based model for deduplication of large data," in 2015 IEEE International Conference on Engineering and Technology (ICETECH), 2015, pp. 1-4: IEEE.

[27] M. V. Maruti and M. K. Nighot, "Authorized data Deduplication using hybrid cloud technique," in 2015 International Conference on Energy Systems and Applications, 2015, pp. 695-699: IEEE.

[28] K. Vijayalakshmi and V. Jayalakshmi, "Analysis on data deduplication techniques of storage of big data in cloud," in 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021, pp. 976-983: IEEE, doi: 10.1109/ICCMC51019.2021.9418445.

DOI : 10.31695/IJERAT.2022.8.4.2

[29] X. Xu, N. Hu, and Q. Tu, "Two-side data deduplication mechanism for non-center cloud storage systems," in 2016 IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB), 2016, pp. 1-4: IEEE.

[30] X. Xu and Q. Tu, "Data deduplication mechanism for cloud storage systems," in 2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, 2015, pp. 286-294: IEEE, DOI 10.1109/CyberC.2015.71.

[31] M. Ellappan and S. Abirami, "Dynamic Prime Chunking Algorithm for Data Deduplication in Cloud Storage," KSII Transactions on Internet and Information Systems (TIIS), vol. 15, no. 4, pp. 1342-1359, 2021, http://doi.org/10.3837/tiis.2021.04.009.

[32] H. A. Jasim and A. A. Fahad, "New techniques to enhance data deduplication using content based-TTTD chunking algorithm," International Journal of Advanced Computer Science and Applications, vol. 9, no. 5, p. 116, 2018.

[33] W. Xia et al., "{FastCDC}: A Fast and Efficient {Content-Defined} Chunking Approach for Data Deduplication," in 2016 USENIX Annual Technical Conference (USENIX ATC 16), 2016, pp. 101-114.

[34] W. Xia et al., "The design of fast content-defined chunking for data deduplication based storage systems," IEEE Transactions on Parallel and Distributed Systems, vol. 31, no. 9, pp. 2017-2031, 2020, doi: no. 10.1109/TPDS.2020.2984632.

[35] Y. Zhang et al., "AE: An asymmetric extremum content defined chunking algorithm for fast and bandwidth-efficient data deduplication," in 2015 IEEE Conference on Computer Communications (INFOCOM), 2015, pp. 1337-1345: IEEE.

[36] Z. Xu and W. Zhang, "QuickCDC: A Quick Content Defined Chunking Algorithm Based on Jumping and Dynamically Adjusting Mask Bits," in 2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), 2021, pp. 288-299: IEEE, DOI 10.1109/ISPA-BDCloud-SocialCom-SustainCom52081.2021.00049.

[37] P. Minishapriya and S. Maheswari, "Performance Analysis of Cloud Storage Using Chunking Algorithm," in 2018 Fourth International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), 2018, pp. 1-5: IEEE.

[38] S. Luo and M. Hou, "A novel chunk coalescing algorithm for data deduplication in cloud storage," in 2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), 2013, pp. 1-5: IEEE.

[39] H. Wu, C. Wang, K. Lu, Y. Fu, and L. Zhu, "One size does not fit all: The case for chunking configuration in backup deduplication," in 2018 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), 2018, pp. 213-222: IEEE, DOI 10.1109/CCGRID.2018.00036.

[40] P. Krishnaprasad and B. A. Narayamparambil, "A proposal for improving data deduplication with dual side fixed size chunking algorithm," in 2013 Third International Conference on Advances in Computing and Communications, 2013, pp. 13-16: IEEE, DOI 10.1109/ICACC.2013.10.

* C author email: ayadwasit1978@gmail.com

DOI : 10.31695/IJERAT.2022.8.4.2