International Journal of Aquatic Science
ISSN: 2008-8019
Vol 12, Issue 03, 2021

Data Deduplication Techniques: A Comparative Analysis

*Priya J[1], *Vinothini C[2], Dinesh P S[3], Reshmi T S[4]

[1,4]*Department of Information Technology, Bannari Amman Institute of Technology, Tamilnadu*
[2]*Department of Computer Science and Engineering, Dr.N.G.P. Institute of Technology, Tamilnadu*
[3]*Department of Computer Science and Engineering, Bannari Amman Institute of Technology, Tamilnadu*

[1]*priyaajothimani@gmail.com,* [2]*vinuchidambaram@gmail.com,*
[3]*dineshps@bitsathy.ac.in,*[4]*reshmits@bitsathy.ac.in*

Abstract. Cloud Storage Service affords users with plentiful storage space and provides users accessibility in synchronizing files across platform and devices from anyplace. In today's world, emerging technologies are relying on a large number of data, hence data management and storage becomes a serious problem. Therefore data deduplication significantly reduces the storage overhead particularly in cloud environment. Deduplication mechanism applied on server-side and client-side. In this paper, we discussed about various types and mechanisms of data deduplication, importance of client-side deduplication and additionally comparative analysis of the few among the different existing client-side deduplication schemes are done.

Keywords: Data De-duplication, techniques, Comparative analysis, Cloud Storage

1. INTRODUCTİON

Expelling of repeating and copy data, known as deduplication, is generally used to spare extra room and network bandwidth. Data Deduplication, commonly known as "Dedup", is a division that can help in decreasing the influence of redundant data on volume costs. Repetition disposal or deduplication over network packets requires huge registering assets to discover fundamental units of rehashed substance, called pieces, by checking each byte in each packet [1]**.** With the quickly expanding measures of data delivered around the world, networked.
What's more, multi-client stockpiling frameworks are getting extremely famous [2]. Data Deduplication improves in removing redundancies without bargaining data loyalty or trustworthiness. Data Deduplication enables volume overseers, to decrease expenses that are related to copied data. Large datasets frequently have a lot of duplication, which expands the expenses of putting away the data.
However, the inside storage limit of every client PC doesn't bolster a lot to this cause [3] Any client can store his boundless data in a circulated storage area which is named as a Cloud. Cloud storage alludes to stockpile the data online in cloud environment. It offers you to store and get to data effectively on the cloud's footing. Data can be put away in advanced, physical

and legitimate pools with numerous ranges of servers. Cloud storage improves the application's execution to minimize expenses [3].

Data deduplication is fundamental for the modern endeavors to limit the hidden costs related to support up their data utilizing Public Cloud administrations. Wasteful data storage on its own can get costly, and such issues are compounded in the Public Cloud when you factor in making numerous duplicates of single datasets for filing or different purposes.

To defeat the previously mentioned issues in cloud storage, deduplication gets basic here. Deduplication can be accomplished on the client-side and as well' the server side. Client-side deduplication scheme saves the transfer bandwidth expenditures, however, require portion of registering limit and expends so much time. Server side deduplication spares the time at the client end yet causes part of bandwidth costs. Numerous client-side as well' server-side deduplication mechanisms are there in existing. This paper analyzes hardly few client-side deduplication algorithms in a nutshell. Remainder of the paper comes here; Section 2 gives about the basics of Deduplication, its types and mechanisms. Comparative analysis given in Section 3, whereas Section 4 briefs about the paper as summary and followed by references lists the contents referred.

2. DE DUPLİCATİON

2.1 Overview

Data deduplication, or "dedupe” is a compression technique that expects to expel copy data from a dataset. Deduplication opens up a great deal of capacity, especially when it is performed over huge volumes of data.

a. Cloud and Deduplication

Cloud Computing has risen as of late as an amazing possibility to serve a wide scope of big business IT capacities. The Public Cloud, wherein merchants give a scope of the process, stockpiling, and framework administrations available by means of an Internet association, is a financially savvy, versatile alternative for aids up and chronicling data [4][5]. Cloud stockpiling is an assistance model where data is kept up, oversaw and back up for clients over a system. Not quite the same as conventional nearby gadget stockpiling, as a rule, cloud stockpiling is an on-request self-administration which can be effortlessly gotten to by means of standard Internet APIs and corresponding protocols [6].

The space saving in a storage pool from data deduplication depends on the dataset and the file type that client use to work with. Datasets that have high redundant data can see progression strides of up to 95%. The below given Fig.1 features distinctive deduplication space savings for different file content types:

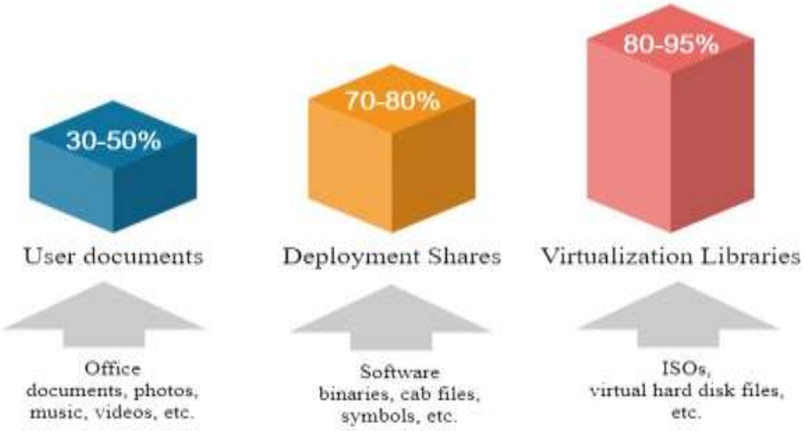

Fig.1. Space Savings in Storage after Deduplication

2.2 Types of Deduplication

Deduplication techniques are classified as Source deduplication, Target deduplication and Global deduplication. Further Target deduplication has been classified into Post-Process deduplication and Inline deduplication. The classification chart is given below in Fig.2

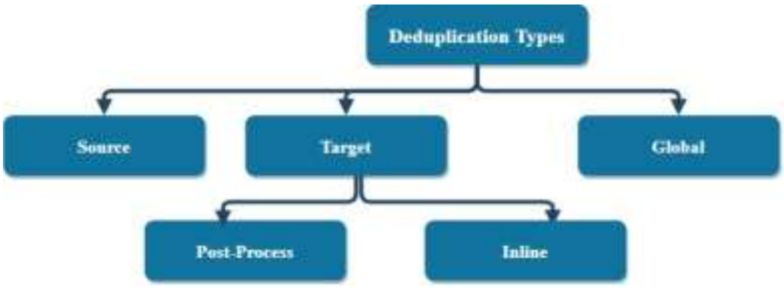

Fig.2. Types of Deduplication

a. Source Deduplication

Source deduplication is termed as expulsion of duplications from the data set before transmission to the backup target. It utilizes a customer programming for contrasting new data blocks on the essential stockpiling device with the recently upheld up data blocks.

b. Target Deduplication

Target deduplication expels all the excess information in the standby apparatus frequently on the virtual tape library. It decreases the storage limit required for standby information. Target deduplication sub-classified as:

- Inline deduplication: Inline deduplication is removal of duplications from dataset previously stored or transformed[19]. Inline deduplication diminishes the measure of excess data in an application and the limit required for the backup disk targets[20]. Here, the deduplication control devise is placed before the storage pool, it's feasible to eject repeated data reasonably during transformation or even before [7]. However, it additionally fundamentally decreases the essential measure of physical memory on the objective framework contrasted with a post-processing deduplication.
- Post-Process Deduplication: It composes the backup data into the disk cache before it begins the "dedupe" process. It is for the most part utilized in backup applications, virtual

tape libraries and such, where a decrease of backup time is required. In the event that the deduplication motor is incorporated in the cluster, a non-controlled adaptation of the backup data first saves money on the storage gadget and afterward deduplicates. This is known as post-process deduplication. This sort of deduplication has the benefit of being a moderately quick method of moving data to the storage goal. Be that as it may, the data isn't promptly accessible after the exchange as the backup process should initially be finished before redundancies can be eliminated [3]. Data on the hard drive is tended to multiple times before it is made accessible for a replication or recovery. This requires unmistakably more physical memory than with inline deduplication. Conversely, in any case, post-process deduplication empowers a progressively effective data decrease dependent on factor data blocks.

c. Global Deduplication

Global data deduplication is a technique for forestalling repetitive data when back up data to numerous deduplication gadgets. It expels all the conceivable reinforcement data redundancies over various frameworks.

2.3 Techniques

a. File-level deduplication

It lookouts for numerous duplicates of a similar file, stores the main duplicate data, and afterward references to the primary file. Just one duplicate gets put away on the disk/tape file. Let's assume a business organization employs 100 workers, sharing a similar file states "document.txt" which is 15MB in memory size. Each representative makes similar changes and spares the specific comparative 100 duplicates of file on server end. So assessed capacity requires to spare a text file on the server side is 1.5GB. Here, the fact of the matter is in the event that all the files are indistinguishable, at that point for what reason to transfer all the files to server. Just spares a solitary duplicate on the server and put a pointer in a client's envelope that focuses to a solitary duplicate on the server. So that is the means by which Information Deduplication procedure used to spare the TB's of capacity.

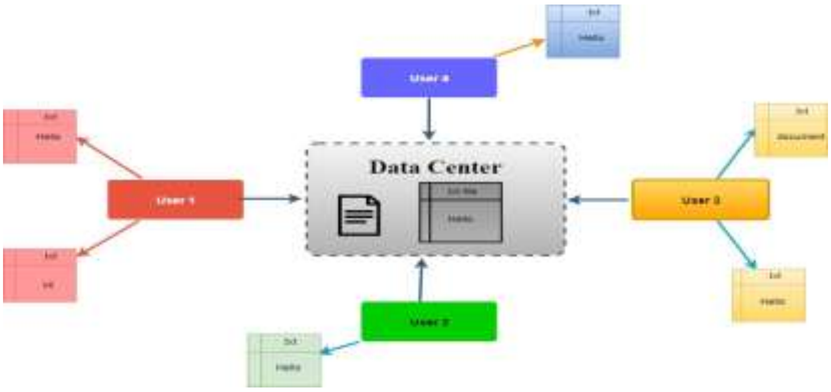

Fig.3. File-level Deduplication

b. Block-level Deduplication

“Variable block-level deduplication” is also termed as block-level deduplication, presences at the data block to look-after whether another duplicate of this block previously occurs or not. If so, the successive duplicates are not stockpiled on any disk or tape, but a link/pointer is created to point to the original copy.

Consider, we have three clients each taking 3 data blocks. Orange, Yellow and blue block are common with in two clients, hence they backed up in a Data center. Therefore, the point to be considered is the memory space required for a data block is very smaller.

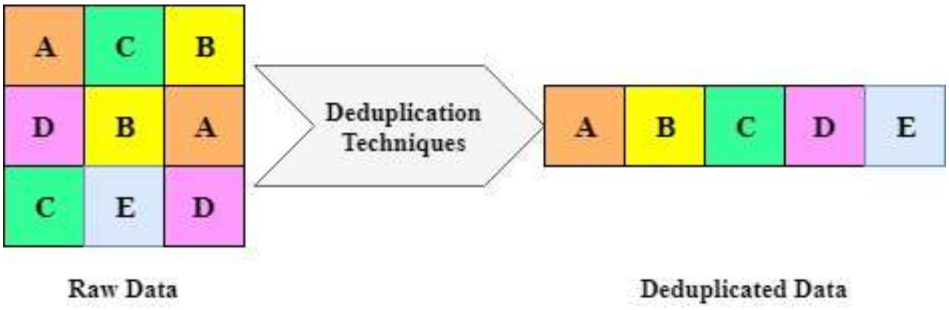

Fig.4. Block-level Deduplication

Block level deduplication is efficient all time than file-level deduplication for the reason that the file-level deduplication get dumps entire file in data storage.

c. File-level vs. block-level deduplication

File-level or Block-level is used in data deduplication. File-level deduplication eliminates duplicate files more, yet it's not an effective method for deduplication.

File-level deduplication reflects about a file to be either endorsed or chronicled with duplicates the data that are as of now put away. This is get completed by examine its merits beside a list. The outcome is an occasion of a file is spared, and results in duplicates get replaced with a remnant that emphasis to the first file.

Block-level deduplication searches inside a data file and spares one of a kind emphasis of every block. Every block is breaks into small chunks with the equivalent fixed length. Every piece of information is handled utilizing a hash calculation, like SHA.

This procedure produces a one of a kind number for each piece, which is then put away in a list. In the event that a file is refreshed, just the changed information is spared, regardless of whether just a couple of bytes of the archive or introduction have changed. The progressions don't establish a completely new file. This conduct makes block deduplication unquestionably progressively productive. In any case, block deduplication takes all the more handling force and uses a lot bigger file to follow the individual pieces.

2.4 Deduplication Mechanism

Data Deduplication can be implemented on Server-side as well' the Client-side. A number of Server-side as well' the Client-side deduplication mechanisms are there in existing. As said above, the deduplication can be done either at the file level or block level. Either type of deduplication mechanism aids to eradicate the redundant of data. Altogether, data deduplication techniques are suitable with conventional encryption mechanisms [3].

A data deduplication technique used on the client end to remove repetitive data in backup and documentation handling procedure before the data is moved to the server comes under the client-side deduplication. Client-side deduplication is of a three-step process. Initially, a client creates extents, and then the server and the client acts together to identify duplicate data and extents and at the last client sends original or non-duplicated data to the server. Whereas, server-side deduplication has two-step process first, identifying redundant data and next steps is for removing of identifying data.

The server - side process of removing data can be done by any of the succeeding methods: retrieving data in primary or copy storage volumes, Moving/Migrating data from primary storage to other primary volumes of data storage, or backing up data of primary storage to copy storage volumes. That server-side processed deduplicated data accessed by client-side. Likewise, deduplicated data can be accessed by the server that processed from client-side [17][21].

Comparatively, client-side data deduplication offers several advantages:

- Reduces the volume of data that is sent over LAN.
- Does not require any additional processing power for deduplicating the data from storage volumes of data.
- Instantly space saving process happens in server side [18].

3. COMPARATIVE ANALYSIS

There are quite a lot of server-side and client-side deduplication techniques and algorithms are available in existing. Since client-side is comparatively having more benefits than server-side deduplication, it has been applied in most of the existing deduplication tools. Here, the paper analyzed and discussed about few among the client-side data deduplication techniques that are available.

The Table.1 illustrates various techniques and mechanisms available for client-side data deduplication.

Table.1. Comparative Analysis of deduplication techniques

Referred Articles	Proposed Idea	Solution/ Algorithm	Problem Identified	Possible Enhancement
Mark W. Storer et al [2]	-generation of encryption keys from chunk data -keys can't be found from encrypted data	-a solution provides security of data, space efficiency in both single-server storage systems & distributed storage	-co-existence of security and deduplication incurs issues of data loss	-remains to create a real time dataset
Jan Stanek et al [8]	-based on popularity, data can be differentiated	-popular & unpopular file Strategy -multi layered crypto system	-privacy is maintained only for unpopular files owned users and not for popular file holders	-assumption and time consumption for extraction of results
Shai Halevi et al [9]	-proofs-of ownership (PoWs)	-proof of-retrievability protocol based on Merkle Tree	-client can prove efficiently that it holds a file to server	-incurs small overhead
Jia Xu et al [10]	-hash-as-a-proof method	-secure hash-as-proof based client-side deduplication scheme	-protects data confidentiality against adversaries	-leakage setting with respect to both outside adversaries and honest to be

				considered
Taek-Young Youn et al [11]	-authorized deduplication scheme. -considering data security, storage efficiency	-Ciphertext-Policy Attribute-Based Encryption (CP-ABE)	-provides confidentiality -prevents exposure of users' sensitive data	-time complexity -overhead
Keonwoo Kim et al [12]	-transfers only non-duplicate files -prevents duplicate-faking & erasure attacks	-a secure client-side deduplication primitive scheme	-reduces the amount of traffic -requires smaller amount of cryptographic operations	-server performance can be improved further
Shanshan Li et al [13]	-resists brute-force attacks	-Bloom filter-based proofs of ownership (PoW) mechanism	-increases I/O efficiency -resists illegal content distribution attacks	-integrity auditing, access control can be added
Xueqin Liang et al [14]	-privacy-preserving individualized discount based implementation algorithms, -game theory	-C-DEDU - individualized discount based incentive mechanism	-ensures the data privacy	-real dataset to be considered
Cheng Guo et al [15]	-ElGamal encryption technique -sharing a random value used to generate encryption key for users	-R-Dedup (randomized, cross-user, secure, deduplication scheme)	-integrity of data -provides user authentication	-computation overhead on the client side
Chao Yang et al [16]	-scheme is thorough, broad and zero-knowledge -proxy re-encryption based key distribution scheme.	-ZK-DE: zero-knowledge based client-side deduplication scheme	-great detection probability of the clients' misbehavior	-encrypted key size and time consumption

3.1 Benefits of data deduplication

Deduplication opens up a great deal of capacity especially when it is performed over huge volumes of data. More than a few benefits of deduplication are,

- Reduces the spending for extra disk or tape
- Reduces storage requirements to an extent
- Reduces the required bandwidth for backup process in a network
- Speeds up the reserve process and recovery process
- Saves time, storage and money
- Reduces the volume of data that is sent over LAN.

4. SUMMARY

Data Deduplication, also known as "Dedup", is an element that helps to decrease the influence of redundant data on volume costs. Repetition disposal or deduplication over network packets requires huge registering assets to discover fundamental units of rehashed substance, called pieces, by checking each byte in each packet. Deduplication opens up a great deal of capacity, especially when it is performed over huge volumes of data. Hence, this paper broadly discussed the various types and mechanisms of data deduplication, the importance of client-side deduplication and additionally comparative analysis of fewer among the different existing client-side deduplication schemes are done.

5. REFERENCES

[1]. M. K. Yoon, "A constant-time chunking algorithm for packet-level deduplication," ICT Express, vol. 5, no. 2, pp. 131–135, 2019, doi: 10.1016/j.icte.2018.05.005.

[2]. M. W. Storer, K. Greenan, D. D. E. Long, and E. L. Miller, "Secure data deduplication, " Proc. ACM Conf. Comput. Commun. Secur., no. October, pp. 1–10, 2008, doi: 10.1145/1456469.1456471.

[3]. T. R. Burramukku, "Available Online through A Comparative Study n Data Deduplication Techniques In Cloud Coden : IJPTFI Research Article," no. October, 2018.

[4]. C. Vinothini, P. Balasubramanie, M. Jayanthi, J. Priya, and P. Anitha, "Swarm Intelligence Algorithms in cloud Computing : A Survey," vol. 29, no. 7, pp. 105698–105706, 2020.

[5]. C. Vinothini, P. Balasubramanie, and K. S. Arvind, "Hybrid Fuzzy C Means Clustering (Fcm) And Improved Bat Optimization Algorithm For Multi-Servers Load Balancing In The Cloud Environment Department of Computer Science & Engineering , MVJ College of Engineering College ," vol. 29, no. 12, pp. 841–851, 2020.

[6]. X. L. Liu, R. K. Sheu, S. M. Yuan, and Y. N. Wang, "A file-deduplicated private cloud storage service with CDMI standard," Comput. Stand. Interfaces, vol. 44, pp. 18–27, 2016, doi: 10.1016/j.csi.2015.09.010.

[7]. X. Xu and Q. Tu, "Data Deduplication Mechanism for Cloud Storage Systems," Proc. - 2015 Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discov. CyberC 2015, pp. 286–294, 2015, doi: 10.1109/CyberC.2015.71.

[8]. J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl, "A secure data deduplication scheme for cloud storage," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 8437, no. ii, pp. 99–118, 2014, doi: 10.1007/978-3-662-45472-5_8.

[9]. S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs of ownership in remote

storage systems," Proc. ACM Conf. Comput. Commun. Secur., pp. 491–500, 2011, doi: 10.1145/2046707.2046765.

[10]. J. Xu, E. C. Chang, and J. Zhou, "Weak leakage-resilient client-side deduplication of encrypted data in cloud storage," ASIA CCS 2013 - Proc. 8th ACM SIGSAC Symp. Information, Comput. Commun. Secur., pp. 195–206, 2013, doi: 10.1145/2484313.2484340.

[11]. T. Y. Youn, N. S. Jho, K. H. Rhee, and S. U. Shin, "Authorized Client-Side Deduplication Using CP-ABE in Cloud Storage," Wirel. Commun. Mob. Comput., vol. 2019, 2019, doi: 10.1155/2019/7840917.

[12]. K. Kim, T. Y. Youn, N. S. Jho, and K. Y. Chang, "Client-side deduplication to enhance security and reduce communication costs," ETRI J., vol. 39, no. 1, pp. 116–123, 2017, doi: 10.4218/etrij.17.0116.0039.

[13]. S. Li, C. Xu, and Y. Zhang, "CSED: Client-Side encrypted deduplication scheme based on proofs of ownership for cloud storage," J. Inf. Secur. Appl., vol. 46, pp. 250–258, 2019, doi: 10.1016/j.jisa.2019.03.015.

[14]. X. Liang, Z. Yan, and R. H. Deng, "Game theoretical study on client-controlled cloud data deduplication," Comput. Secur., vol. 91, 2020, doi: 10.1016/j.cose.2020.101730.

[15]. C. Guo, X. Jiang, K. K. R. Choo, and Y. Jie, "R-Dedup: Secure client-side deduplication for encrypted data without involving a third-party entity," J. Netw. Comput. Appl., vol. 162, no. February, 2020, doi: 10.1016/j.jnca.2020.102664.

[16]. C. Yang et al., "Zero knowledge based client side deduplication for encrypted files of secure cloud storage in smart cities," Pervasive Mob. Comput., vol. 41, pp. 243–258, 2017, doi: 10.1016/j.pmcj.2017.03.014.

[17]. http://www.tsmtutorials.com

[18]. https://www.ibm.com/support/knowledgecenter

[19]. Xu, X.; Sun, Y.; Krishnamoorthy, S.; Chandran, K. An Empirical Analysis of Green Technology Innovation and Ecological Efficiency Based on a Greenhouse Evolutionary Ventilation Algorithm Fuzzy-Model. Sustainability 2020, 12, 3886.

[20]. N.K. Karthikeyan, K. Venkatachalam and R. Prabhakaran, 2012. Study and Implementation of Environmental Monitoring System (EMS) Using WSN. Asian Journal of Information Technology, 11: 216-224.

[21]. S. Ramamoorthy, G. Ravikumar, B. Saravana Balaji, S. Balakrishnan, and K. Venkatachalam, "MCAMO: multi constraint aware multi-objective resource scheduling optimization technique for cloud infrastructure services," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-8, 2020.