International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 07 | July 2021 www.irjet.net p-ISSN: 2395-0072

Removal of Duplicate Data using Secure Hash and Difference Hash Algorithm

Nidhi B Sankhe[1], Shweta Singh[1] , Sumedh Pundkar[2]

[1]*B.Tech, Department of Computer Science and Technology, Usha Mittal Institute of Technology, SNDT Women's University, Maharashtra, India*
[1]*B.Tech, Department of Computer Science and Technology, Usha Mittal Institute of Technology, SNDT Women's University, Maharashtra, India*
[2]*Professor, Department of Computer Science and Technology, Usha Mittal Institute of Technology, SNDT Women's University, Maharashtra, India*

Abstract - *Removal of duplicate data is a new way used to compress the data by removing duplicate copies of information. It ensures data management by efficiently reducing the storage space and maintaining the energy consumption. Deduplication improves storage utilization with higher reliability. Due to abundant data generation through various sources need of this system increases as it ensures data management by efficiently reducing the duplicate copies of data. This technique results in making a system more optimized by calculating hash value of the files. The following paper aims to achieve the above goal by detecting and eliminating the duplicate data. This proposed system is a simple framework to use, provides ease of retrieval of data from storage and calculate hash value by using SHA(secure hash algorithm) and dhash(difference hash algorithm). Data in the form of text, images, audio and video can be examined in this proposed paper.*

***KeyWords*: Duplicate Data, SHA-256(Secure Hash Algorithm), Dhash(Difference Hash).**

1. INTRODUCTION

In today's world, working with data includes the task of organizing large amount of data. It is important to note the data are not redundant during performing such task. Duplicate data are stored in system due to human error or often happens when the file with same content is saved as different name. This duplicate data consumes free space in the system and leads to the problem of inconsistency. The accuracy of data organization is not maintained. Redundant data occupy more storage and affect the system efficiency. To overcome such problems data deduplication can be used. It is used to detect and eliminate the redundant copies of data. It lowers the storage consumption and makes the system effective. It maintains data integrity and maximizes the performance. Data deduplication is performed on file level and block level. The file level deduplication approach examines the operation of files on the basis of multiple aspects like index, name, time-stamp, etc. If the file is different, it will update and store a new index of the specific file. Although this technique is not very efficient because it can consider files as unique on the basis of the different name and time-stamps in spite the content being similar. It may lead to the problem of saving the file repeatedly. The other approach is, block level. The data file is divided in terms of unique block and these blocks are further broken down into chunks of fixed sizes. It compares data in terms of chunks which are the contiguous block of data. These chunks are analysed and compared to other chunks by using different hash algorithms. The unique hash value generated for each file and is compared with the different files. If a similar hash value is found, then it is considered as repeated data. Deduplication has the ability to effectively manage storage allocation, significant cost savings as there is no need to buy extra storage space. Removal of duplicate data sustains network optimization. There is an enormous drop in both power and physical space requirements. The aim of this paper is to aid in removal of duplicate data by using hashing techniques and improve the efficiency. The flow of paper is organized as follow: Initial section I with introduction, section II background and related work ,section III details on proposed system , section IV on Methodology, Section V based on Advantages, continuing with Result and Analysis as Section VI and finally concluding with references in Section VII.

2. BACKGROUD AND RELATED WORK

2.1 Background

Data deduplication has different approaches listed below:

1. Source Based Data Deduplication.
It is removal of duplicate data before transmitting to the backup target (on a source side). Advantage of this approach is lower bandwidth and less use of storage space is done. But this approach is time consuming.

2. Target based Deduplication.
It is performed on the server side where data are supposed to be store. This approach requires higher bandwidth and extra hardware for a target size. But this can be very useful for large data sets as processing at the source may lead to degrade of performance.

3. Inline data deduplication.
The deduplication takes place on the client side where data are divided into chunks. Then the hash value of chunks is calculated and compared with the previously stored chunks. If the hash value is matched then the redundant chunk is removed and a reference to the original chunk is made. This technique results in minimizing the CPU overhead as it is performed on RAM but the only problem is the necessity of more resources for performing the task.

4. Offline Data deduplication.
In this approach, data is first stored in storage. Then the deduplication process reads the stored data by checking the hash value of the different chunk. If the similar hash value is detected then that chunk is removed and reference to the original chunk is made. This method lead to CPU over head as it requires space for storing the data and processing it.

2.2. Related Work

Many research works have been implemented in this subject. Few of them have contributed significant findings for the topic. In this research paper, An overview on cloud computing, cloud file services, accessibility and storage is given. It studies existing deduplication technique and considers storage optimization for the benefit of cloud service providers. The project also proposes an efficient method for detecting and removing duplicates using file check sum algorithms by calculating the digest of files which takes less time than other pre-implemented. To remove redundant data and to maintain the unique instance of data. Along with removing the duplicate data it uses checksum algorithm for error-correcting and cyclic redundancy check(CRC). [1] In other paper, Two data cleaning algorithms are presented. The first algorithm, suitable for cleaning data when a data warehouse is being built, while the second algorithm designed for incremental cleaning of an existing data warehouse. The results were, Token-based algorithm outperforms the other two comparable algorithms. It has a recall close to 100 percent as well as negligible false positive errors. [2] A system is designed which achieves confidentiality and enables block-level deduplication. On top of convergent encryption, system is put together. The paper concluded that it was worth performing block-level deduplication instead of the file-level. [3] The subsequent paper used Proxy re-encryption technique. Where it was discovered that encryption and decryption both are efficient in Proxy re-encryption algorithm. Time taken to upload data for generating data token is stable. Proxy re□encryption provides high security and ease of maintenance. [4] In further paper, content level duplication check is performed in which all content of the file are matched and check based on hash function. To generate the hash value of the data, MD5 massage digest version 5 algorithm is used. Advantages of using this paper were: (a) Malicious users were not able to upload data. (b) It solved more critical part of the cloud data storage which is tolerated by only few methods. [5] In the next paper, certain observation was made. Chunking time required with switch divisor algorithm was decreased approximately to 25 %. Deduplication by implementing chunking algorithms resulted in maximum efficiency and less time. Successfully able to comment that variable size chunking gives better deduplication ratio as compared to fixed size chunking. [6]

3. PROPOSED SYSTEM

The application focuses on detecting and removing duplicate data which are in the form of different extensions. Removal of duplicity is from storage as well as on folders stored in local desktops. The core idea is fingerprinting the data and to generate hash values are stored in a linked list for detecting duplicates. Different Hashing algorithms like difference hashing, secure hashing, etc., fixed chunking method are used depending on type of files.

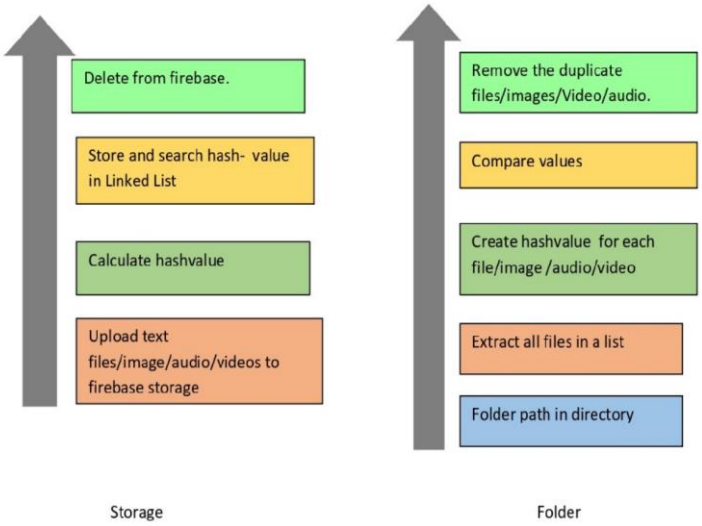

Fig. 1. Proposed System for deduplication.

4. METHODLOGY

The Algorithms used for the project are SHA-256(Secure Hash Algorithm) and Dhash(Difference Hash) algorithm. Both of them are considered as secure and modern hashing algorithm. SHA-256 is used for files of any extension and Dhash is particularly used only for detecting duplicate in Images . The reason for implementing Dhash algorithm for images is any other cryptographic algorithm like SHA-256 and Message Digest take minor changes(not visually evident to a human eye)into consideration and generate different hash values for same looking images. This property may degrade performance for deduplication. To overcome this problem, dhash is used for images. All the algorithms and the respective steps needed to perform for the implementation are as mentioned in the table. 1.

Table -1: List of Algorithms used

Start		
Steps:	SHA 256	Dhash
1	Divide input file into chunks.	Convert image to grayscale
2	Append padding bits and length bits.	Downsize to 9*9 thumbnail.
3	Initialize buffers.	Calculate row hash and column hash
4	Apply compression function	Combine both values to get one hash value for image.
END		

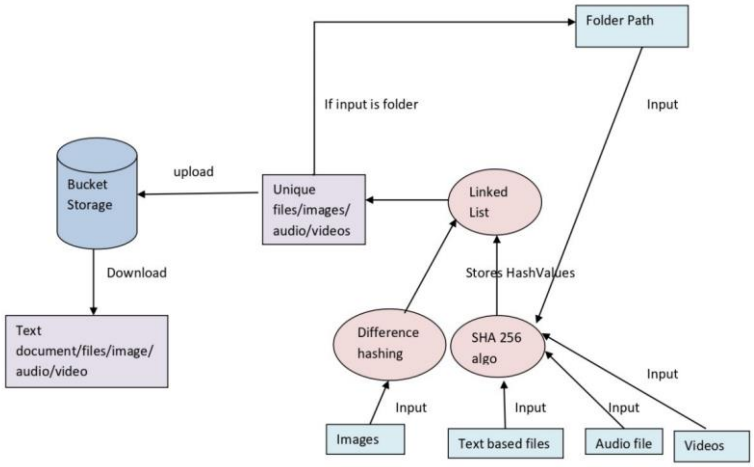

Fig- 2: Architecture

The architecture design of the project is explained in Figure-2. Initially, the user will need to upload the input file that is to be check for duplicate record. The inputs that are text based files, audio ,video and system folder with files of different extensions will follow SHA 256 algorithm. If the input is image then dhash algorithm is used on the input. After processing of inputs through these algorithms, hash value is calculated and stored in linked list. These hash values are compared and duplicate file is detected. The duplicate files are removed. Unique files are given as output. The unique files from the folder are saved in the folder itself and the rest files are stored in bucket. The Project also provides an option to upload as well as download the files from bucket storage.

For implementation of this application, libraries required are pyrebase for connecting application to firebase storage, Hashlib for incorporating hashing algorithm, PyQt5 designer for graphical user interface, Pillow for preprocessing images. The hash values of different kinds of files are calculated and store in a linked list for the future searching.

5. ADVANTAGES

1) Adept replication: Unique data are returned to the disk and hence there is no need to make a copy of data again.

2) Cost-effective: Storage requirement is decreased which leads to fewer demands for the disk.

3) This framework helps in easy detection and removal of duplicate data.

4) A greener environment can be attained as fewer cubic feet of space required.

6. RESULT AND ANALYSIS

6.1. Analysis for Storage Optimization and deduplication ratio

For the analysis, set of sample files are taken. These set of files are arranged in a folder of different extensions. The folder have duplicate copies of files inside it. Sample set of 14 folders are considered. Table below corresponds to the size of folder before and after deduplication. These information are used to find out the storage optimization and deduplication ratio as well.

Deduplication Ratio : =

Size before deduplication / Size after deduplication .

Table -2: Storage optimization and deduplication ratio analysis dataset table

Sample Set Table- Folder Size in MB			
Folder No.	Before Deduplication(MB)	After Deduplica-tion(MB)	Deduplication Ratio.
1	1.31	0.672	1.949404762
2	3.72	1.86	2
3	3.84	2.66	1.443609023
4	4.23	1.6	2.64375
5	4.77	2.38	2.004201681
6	4.82	2.41	2
7	5.33	2.66	2.003759398
8	15.9	6.89	2.307692308
9	17.4	6.47	2.689335394
10	5.47	4.44	1.231981982
11	24.8	12.4	2
12	26	13	2
13	31.7	15.58	1.996904025
14	34.6	17.3	1.988795518

Analysis result for Storage optimization:
After performing the analysis in figure 3, It is concluded that the storage is much more released after removing the duplicate files from the folder hence optimizing the storage for the system.

Analysis result for deduplication ratio:
Deduplication ratio for the above dataset refers to the number of duplicate files present in the folder. In the Figure 4, folder with files having one or multiple duplicate copies of it has highest deduplication ratio and the folder with minimum number of duplicate files has lowest deduplication ratio.

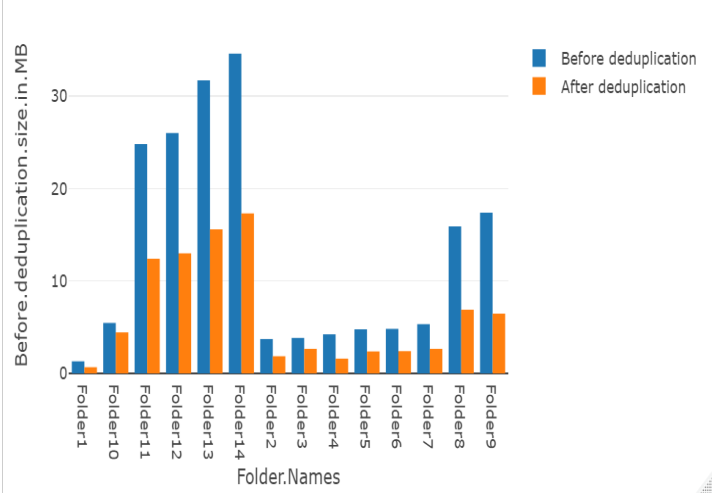

Fig. 3. Storage Optimization

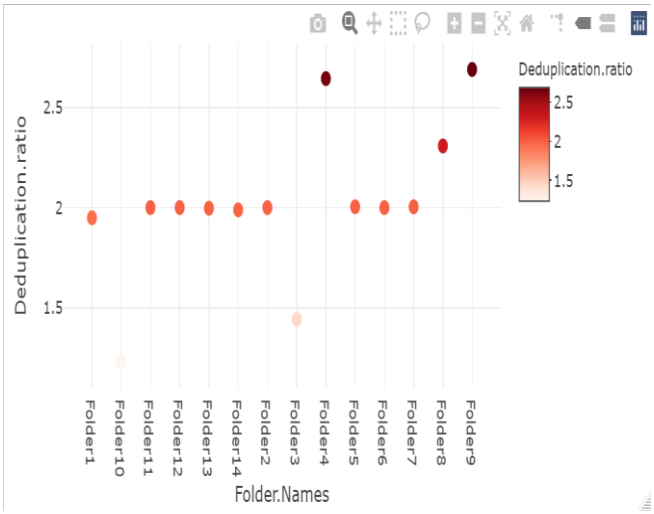

Fig. 4.Deduplication ratio

6.2. Time Analysis

The below table consists of dataset values for time analysis. This table has file size, time of files when hash value is generated and time at which the duplicate file is detected as well as removed.

Table -3: Time analysis table

Sample Set Table			
SR.No.	File Size(MB)	Value generation Time(seconds)	Duplicate Time(seconds).
1	34.6	0.86	0.89
2	25.1	0.85	0.88
3	18.9	0.88	0.9
4	16.3	0.84	0.94
5	8.99	0.86	0.87
6	5.92	0.92	0.96
7	2.37	0.89	0.98
8	1.91	0.83	0.87
9	1.84	0.81	0.8
10	1.49	0.97	0.98
11	1.1	0.82	0.87
12	1.03	0.7	0.9

Analysis Result for Correlation: In Figure 5, file size is slightly correlated to the time at which hash value is generated. Also, time of hash value generation is highly correlated to the time at which duplicate file is detected and removed.

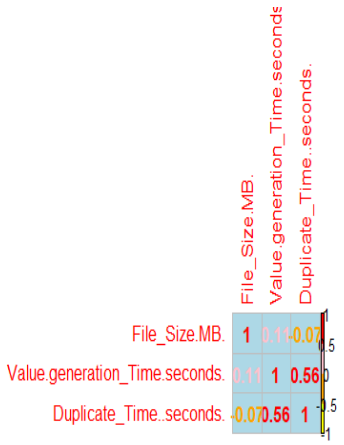

Fig. 5. Correlation

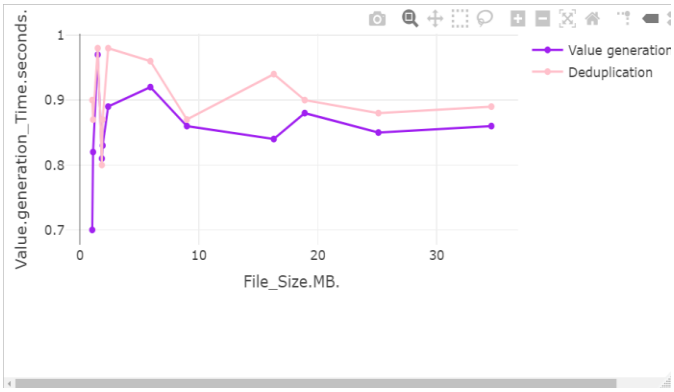

Fig. 6. Time required for deduplication

The time required to deduplicate the files is shown in figure 6. It is in comparison to the time of hash value generation. According to the project, hash value is generated first and then the duplicate files are detected and removed. Hence, figure 6 is the exact depiction of the above sentence.

6.3. Algorithm Time Analysis:

The following table below consists data for checking algorithm time analysis. The different algorithm considered here are SHA-256,MD5 and Dhash algorithm.

Table -4: Algorithm time analysis

Sample Set Table - Time in seconds				
SR.No.	File Size(MB)	Time(SHA-256)	Time(MD5)	Time(dhash).
1	1.1	0.82	0.87	0
2	1.84	0.81	1.21	0
3	1.49	0.97	1.4	0
4	2.37	0.89	1.13	0
5	1.91	1.06	1.14	0.83
6	5.92	1.1	1.38	0.91
7	1.39	1.07	1.02	1.03
8	2.54	0.91	1.33	1.26

Analysis Result of Algorithm Time: The graph in figure 7, depicts that SHA-256 algorithm takes the lowest time in comparison with MD5 algorithm for all the files. Difference hashing algorithm is an image specific algorithm which has taken minimum time amongst all and doesn't generate value for text files or audio video.

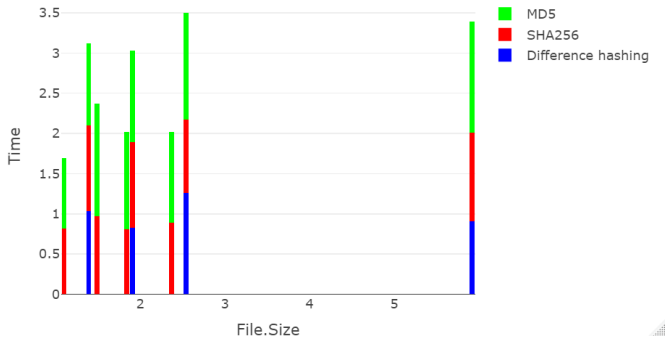

Fig. 7. Algorithm time analysis

7. CONCLUSIONS

There is a rapid increase in size of data which lead to heat-energy consumption, duplicate data, and inconsistent data organization. In this paper, we propose a framework in order to fulfil a balance between changing storing efficiency and performance improvement in system. The proposed system is capable of handling scalability problem by removing duplicate data. Deduplication aids in saving the storage space. This project helps in easy maintenance of data so that no duplicate files are saved. It works for text, images, audio and video. With the evolution, storage resources of commodity machines can be efficiently utilized.

8. ACKNOWLEDGEMENT

Any accomplishment requires work and effort. Working on Major project titled "Removal of Duplicate Data using Secure Hash and Difference Hash Algorithm" was a source of immense knowledge to us. We would also like to express our sincere gratitude to our guide Prof. Sumedh Pundkar on their guidance throughout the course of this project. We express our sincere gratitude to Dr. Sanjay Pawar, principal of Usha Mittal Institute of Technology, HOD of computer science and technology, to give us the opportunity to work on this project.

9. REFERENCES

[1] O. A. FESTUS, "Data finding, sharing and duplication removal in the cloud using file checksum algorithm."

[2] C. I. Ezeife and T. E. Ohanekwu, "The use of smart tokens in cleaning integrated warehouse data," International Journal of Data Warehousing and Mining (IJDWM), vol. 1, no. 2, pp. 1–22, 2005

[3] P. Puzio, R. Molva, M. Onen, and S. Loureiro, "Cloudedup: secure ¨ deduplication with encrypted data for cloud storage," in 2013 IEEE 5th International Conference on Cloud Computing Technology and Science, vol. 1. IEEE, 2013, pp. 363–370

[4] M. Maragatharajan and L. Prequiet, "Removal of duplicate data from encrypted cloud storage," in 2017 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS). IEEE, 2017, pp. 1–5.

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

[6] E. Manogar and S. Abirami, "A study on data deduplication techniques for optimized storage," in 2014 Sixth International Conference on Advanced Computing (ICoAC). IEEE, 2014, pp. 161–166.