e-ISSN: 2582-5208

International Research Journal of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume:06/Issue:07/July-2024 Impact Factor- 7.868 www.irjmets.com

DATA DUPLICATION REMOVAL USING FILE CHECKSUM

Ruchitha M*1, Mrs. Veena B*2

*1P.G. Student, Department Of Master Of Computer Applications, University B.D.T College Engineering, Davanagere, Karnataka, India.

*2Assistant Professor, Department Of Master Of Computer Applications, University B.D. T College Of Engineering Davanagere, Karnataka, India.

ABSTRACT

While managing and performing file operations on computer systems and different types of storage devices, a large number of redundant files will accumulate, affecting the performance of the machine. The accumulation of these digital trash levels is often be the main cause for a lack of storage space and become reason for decreasing in the performance of computer. The project's objective is to create console application that prevents data duplication by using file checksum. The file will not be saved in the directory if it already exists there; else, it will be saved in it. Reducing the amount of duplicate file in the system is the primary objective of the project. In particular, the key value store should be optimized for better process speed to avoid affecting the backup window and should be designed for horizontal scaling to enable competition on the cloud platforms.

Keywords: Data Reduplication, Storage, Raw Data Analysis, File Checksum, Administrator, User.

I. INTRODUCTION

The globe has produced a larger volume of digital data, and this number is only expanding. Research indicates that during the next ten years, the large volume of data generated annually would increase more than six times, growing at an annual pace of 57 percent. The massive increase of data is placing a pressure on the storage infrastructure. Emails, pictures, music, video, documents and other types of data all are included in enterprise data. Traditional storage techniques face numerous issues as data accumulates quickly. Excessive data volumes need the use of more storage capacity.There are numerous methods for getting rid of redundant data that has been stored. In the research community, data reduplication removal using file checksum is currently becoming more and more popular.

In today's technological world, organizing and manipulating files on a computer or other storage devices like android, become a really challenging task. The gradual increasing of such digital garbage levels might be the main reason for a lack of storage space, and also decreases the internal storage capacity and may slowdown in the performance of hardware system.

Therefore, we utilize the file checksum technique using Python to solve these issues and get rid of the duplicate data. The checksum analyzes if there are any duplicate files in the database every time a new file is uploaded. The initial files will remain in our directory even if a duplicate file is discovered; the checksum will automatically remove the duplicate file using a proper algorithm. Thus this technique assists in reducing backup size and time, and accelerates indexing. The software benefits end users as well as business organizations also.

II. RELATED WORK

When maintaining and performing file operations on the computer or other storage media, a large number of duplicate files of considerable size will build up on it. This results in digital waste, a loss of computer performance, and a need for a lot of space to store information.

In paper [1] Data reduplication adds more unwanted data to the storage unit by keeping multiple copies of the same file. In duplication of data elimination, the file checksum technique is used to quickly identify redundant and identical information.

The procedure calculates a file's checksum upon upload and compares it to the checksums of previously saved files within the database. It will make modifications if the file already exists; if not, an additional entry will be made for the file. This system will detect duplicate files using the MD-5 hash algorithm. MD-5 is the name of the 128 bit hash algorithm also referred to as the Message Digest algorithm.

In paper [2] A cloud-fog situation, this article created a safe deduplication solution based on convergent and MECC algorithms.

A) When a new user tries to share a file;

B) When the same user wants to share the same file;

C) When multiple users try to upload the same file; and

D) When numerous users try to access the file are the four scenarios in which the suggested approach is tried.

The efficiency of the suggested system was evaluated using folders with sizes that vary from 5 MB to 25 MB, increasing by 5 MB with each iteration. According to the outcomes of the analysis, the latest system has a privacy grade of 96%, which is better than the other encryption techniques currently in use and a promising result.

In paper [3] Because of private data storage creates new hurdles for data duplication in the cloud, enormous storage space becomes challenging. Here is the system framework for the job they have recommended. To eliminate duplicate records, the ORD [Optimal Removal of Deduplication] technique has been recommended. When the owners of data are unavailable for de-duplication control, it might be used. The ability to control file upload and authorize uploaded files is granted to both the admin and the user. After reduction of data, assets from the cloud and server were examined and divided into distinct units.

In paper [4] They examined contemporary backup methods and used a special method that minimized fragmentation. The top backup utility includes segments of every backup that are likely physically dispersed, leading to a laborious fragmentation problem. Sparse and out-of-order boxes are two forms of fragmented containers that are encouraged to travel upward via fragmentation. The gadget's sparse container contributes to the average functioning of the device on both the garbage and brings back series. Out-of-order wrapping containers interact with frequently accessed containers during a repair. To lessen fragmentation, the authors employ an abundance aware filter out in addition to an antiquated set of principles. This amis to identify the insufficient and improper regulation.

III. METHODOLOGY

1. Admin Module:

Admin can handle a variety of tasks, including security issues maintaining the system server and granting varying levels of access to users. Admin is the person who has the full access to the system.

The fallowing tasks are performed by Admin:

a. Login

b. View User

c. View Files

2. User Module:

In order to gain access to the system the user first register with some login credentials (username, email, password) then the user can access the system.

The user can perform the fallowing task once they gain access to the system:

a. Upload a File .

b. Download a File.

3. File Checksum:

File Checksum is an interface where a user can browse the directory to delete redundant amount of data present in the directory.

a. Select a Directory

b. Delete a Directory

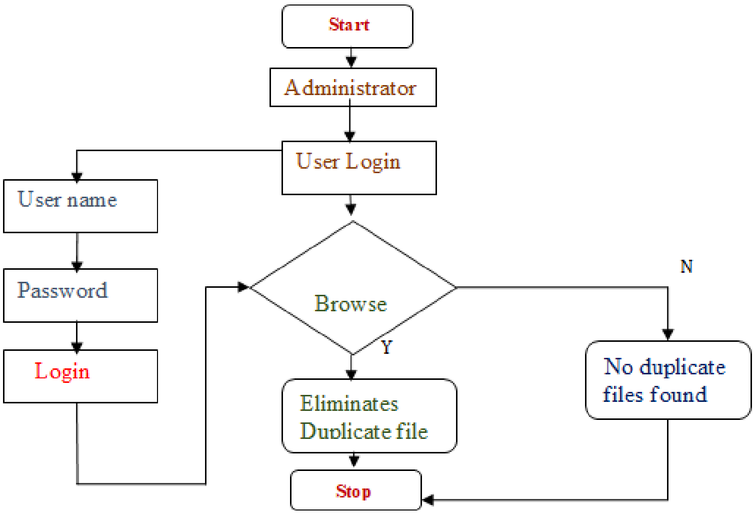

Fig 1: Data Flow Diagram

In the proposed system it works based on the raw data analysis using python with SHA-256 algorithm. Before going to identify the duplicate file in the system there is module called admin who provide access to the user with a proper login credentials like username and password, so the particular user can enter to the system. Then to remove the duplicate data in a particular directory we use the method called checksum.

Checksum is a technique of removing redundant amount of data and a duplicate file in the system. Whenever a user uploads a particular directory of the file to the checksum, the checksum checks for duplicate data present in a file, if it found any copy of the same files it will eliminate all the duplicate files present in the existing directory of the file system. So that finally we can have original and necessary files in the system.

To calculate the checksum SHA-256[Secure Hash Algorithm] algorithm is used which ensures that only truly identical files are identified as duplicates, regardless of their filenames. The hash value provides a unique value that corresponds to the content of the file. By using SHA-256 algorithm we can achieve enhanced security, though the system removes duplicate files quickly and easily.

IV. RESULTS

As a result, the approach generates storage facilities that use redundant data elimination technology to manage more data in the same amount of storage space. Data deduplication will reduce data by fifty percent to half. Less data indicates a shorter period of time is spent in backing up and retrieving data in a system, In addition it will consume less recovery time. Furthermore, the results will rapidly process huge files and directory structures by utilizing an effective secure hash technique, which allows us to manage enormous amount of data with out significantly degrading speed and performance of the system.

V. CONCLUSION

The project successfully illustrates an effective and efficient strategy to identify and to remove unnecessary numbers of files that are duplicates using the SHA-256 algorithm with the checksum method. Using the file checksum approach, the goal is to construct a console application that can rapidly and simply dSiscover duplicate files in the storage infrastructure. The checksum of both existing and new files is determined using the SHA-256 approach. This could ensures that memory utilization remains minimal even with a high number of files. As a result, this will serves as a solid platform for additional enhancements and applications in a variety of fields for both corporations and end users concerned about data duplication.

VI. REFERENCES

[1] S.Usharani, K.Dhanalakshmi, N.Dhanalakshm,"De-Duplication Techniques: International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878,Volume-7, Issue-6S5, April 2019

[2] Assistant Professor, Department of Computer Science and Engineering, Nagarjuna College of Engineering and Technology, Bangalore, India B.E. Student, Department of Information Science and Engineering, Nagarjuna College of Engineering and Technology, Bangalore, India.

[3] https://arstechnica.com/civis/threads/finding-duplicates-using-sha256-and-getting-inconsistent-results.1484042/

[4] https://www.sagacitysolutions.co.uk/about/news-and-blog/data-deduplication/#:~:text=Duplicate%20data%20occurs%20when%20storing,once%20within%20the%20same%20database

[5] https://github.com/Vatshayan/Data-Duplication-Removal-using-Machine-Learning