Data Duplication Removal using File Checksum

Debashree Sadagar
MCA Scholar
School of CS and IT, Dept of MCA
Jain Deemed-to-Be-University,
560069
Bangalore, India
19mcar0070@jainuniversity.ac.in

Dr. Mir Aadil
Assistant Professor
School of CS and IT, Dept of MCA
Jain Deemed-to-Be-University, 560069
Bangalore, India
mir.aadil@inurture.co.in

Abstract- The project enables the user to check for any duplicates in the database by checking the hash value of the file uploaded. If the file already exists in the database, it won't be stored otherwise the file will be saved in the database.
The goal of the project is to develop software that uses file checksums to prevent data duplication. The project's main goal is to reduce the number of duplicates in the database, particularly the key-value store, to improve process performance so that the backup window is not impacted, and to design for horizontal scaling so that it can compete on a Cloud Platform.

Keywords: Database, Duplication, Entity, Data, Checksum, Redundant, User id.

I. Introduction

The amount of digital data created throughout the world is massive, and it is continually increasing. According to research, the amount of data produced each year will more than sixfold in the next decade, expanding at a rate of 57 percent every year. The storage infrastructure is being strained by the tremendous rise in data. Enterprise data includes images, audio, video, emails, and other sorts of data. As data grows at a rapid rate, traditional storage methods confront a slew of problems.
A high amount of data necessitates the utilisation of extra storage space. In truth, a significant size of the data in storage archives is redundant or has been slightly altered to another copy of data. There are a variety of approaches for removing redundancy from stored data. Data deduplication is currently gaining prominence in the research community. Data de-duplication is a sophisticated data compression way to eliminate redundant data and increasing storage utilization. Duplicate data will be deleted and not saved all through the de-duplication process, leaving a distinct copy of the data to be retained.
A web-based application can be designed to improve storage efficiency and reduce data redundancy by allowing the user to connect the application to the storage device or database and check if the file already exists in memory. If the data is already recorded in the database, the file will not be saved, saving memory space and increasing storage capacity.

II. Literature Survey

Many duplicate files of significant size will accumulate on the computer when maintaining and conducting file operations on it or on other storage media. The accumulation of this digital garbage might lead to a lack of storage space and a decline in computer performance. As a result, it's necessary to search for and delete duplicate files from your computer's hard disc. It is sometimes useful to have information about files that have several replicates. If your computer has multiple copies of a requested file, they will all be stored in RAM, slowing down your system. We employ a data deduplication technique in which the system checks the checksum every time a file is uploaded, and the checksum verifies checksum information kept in the database. If the file is already there, the section shall be refreshed; otherwise, a new entry will be made into the database. The Duplicate File Detector and Remover will assist you in reclaiming critical disc space and increasing data efficiency. Duplicates can be deleted to assist speed up indexing and minimize backup time and size. It can swiftly and safely locate any undesirable duplicate files on the system, delete them, or relocate them to a different location, depending on the user's preferences. Your system will be freed of duplicates. Data deduplicationIII. tools check data down to the block and bit level, and only the altered data is saved after the initial occurrence. The rest is discarded, and a pointer to previously saved data is substituted. Under the correct conditions, block- and bit-level deduplication algorithms can achieve compression ratios of 20x to 60x, or even more.

Different de-duplication techniques are investigated in this work, where deduplication tactics are performed on encrypted data in a large storage area. The majority of the solutions discussed here are based on Confluent encryption, which is a basic way for deduplication that works well with encoded data. According to the paper's authors, a methodology should be devised that will increase storage capacity without consulting an encryption strategy, by using a de-duplication system in data or information storage servers that scrambles the available data.[1]

In this paper, we looked at modern-day backup systems and applied a unique technique to help reduce fragmentation. The leading backup tool has sections of each backup that are presumably bodily scattered, causing a time-consuming fragmentation issue. Fragmentation promotes the upward movement of two types of fragmented containers: sparse and out-of-order boxes. On both the restore and garbage series, the device's sparse container exacerbates the device's average overall performance. During a repair, out-of-order packing containers interact with those that can be accessed often. The authors use an age-old set of guidelines and a stockpile aware filter out to reduce fragmentation. Those assist in determining the scant and out of order discipline.[2]

This paper developed a secure deduplication solution based on convergent and MECC algorithms in a cloud-fog scenario. The proposed method is tested in four different scenarios: a) when new users try to upload a new file, b) when the same user tries to post the same file, c) when several users try to upload the same file, and d) when many users try to download the file. The proposed system's performance was tested with

files ranging in size from 5 MB to 25 MB, with each iteration rising by 5 MB. The new system has a security rating of 96 percent, which is a promising result and higher than the other encryption methods already in use, according to the performance analysis.[3]

To ensure successful data processing and analysis, addressing rising storage needs is a difficult and time-consuming effort that necessitates a big computational infrastructure. Data deduplication is becoming increasingly important for cloud storage providers as the number of customers and the amount of their data grows at an exponential rate. Cloud providers save a lot of money on storage and data transfer by storing a unique copy of duplicate data. Cloud computing, cloud file services, accessibility, and storage are all included in this project. It also examines existing data de-duplication strategies, procedures, and implementations for the benefit of cloud service providers and cloud users. The project also presents an efficient way for finding and eliminating duplicates using file checksum algorithms by calculating the digest of files.[4]

Cloud service providers are enticed to utilize data de-duplication to save money on storage and bandwidth. Cloud users want to be able to use the cloud safely and discreetly in order to preserve the data they share there. As a result, before transferring data to the cloud, they encrypt it. The data deduplication feature becomes a difficult problem because the encryption goal competes with the de-duplication function. In terms of security and efficiency, existing de-duplication solutions are insecure and inefficient. They are either computationally expensive or vulnerable to brute force assaults, which allow the attacker to access files. This is what drives the authors to suggest an efficient and secure method for eliminating duplicate data. The authors begin by describing the implementations and functionality of deduplication techniques, then go on to a review of literature that suggests several ways to duplication removal as well as the security and efficiency issues that current systems face. The authors have recommended using the AES-CBC algorithm and hashing algorithms to boost the effectiveness and security of data deduplication for users. Without the involvement of a third party, users' keys are created in a consistent and secure manner. The authors illustrate the efficacy of the recommended approach by putting it into practise and comparing it to existing techniques.[5]

III. System Analysis

Problem Statement

As the amount of data created expands, so does the complexity of storing and managing it. Additional data needs more storage, which necessitates a rise in cost because the storage unit must be upgraded. Because we don't know how much storage we'll need, just upgrading the storage unit isn't a solution. The system becomes bulkier and more expensive as the number of storage units increases.

Proposed System:

The data duplication removal system is a web-based application that identifies redundant data quickly and correctly by using file checksum technique. In the proposed data duplication removal procedure, the file checksum technique is employed to quickly discover

duplicate or redundant data. The technique calculates a file's checksum and compares it to the checksums of other files in the database when it is uploaded. If the file already exists, the user will be notified that the file already exists; otherwise, a new file entry will be created. The MD-5 hash algorithm will be used in this system to detect duplicate files. A 128-bit hash algorithm is the Message Digest algorithm (MD-5).

IV. Scope of Project

Data de-duplication can reduce data by fifty to one. Storage costs are lowered because there is less storage required. Less data implies less time spent backing up and recovering data, as well as faster recovery times. As a result, storage systems that use data duplication elimination technology may handle larger amount of data in the same amount of space.

V. Methodology

I. Admin Module

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

a) Login: To gain control of the system, the administrator must first log in with proper credentials.

b) View User: Once the administrator has logged in, he can view all of the users.

c) Block / Unblock User: Admins can block and unblock users based on their activities.

d) View Files: The admin can view but not alter the files that users have uploaded.

2. User Module

In order to gain access to the system, the user must first register. The user must submit some information during the registration step, such as a username, email address, and password. After completing the registration process, the user can access the system using the credentials provided during the registration process.

a) Upload a File: This component allows users to upload files to the system, which they can later share with others.

b) Download a File: Once the file has been loaded, if the user wishes to download it, he or she can do so by selecting the download file option.

c) Password Change: The user can update his or her password at any moment.

VI. Architecture

a) System Architecture

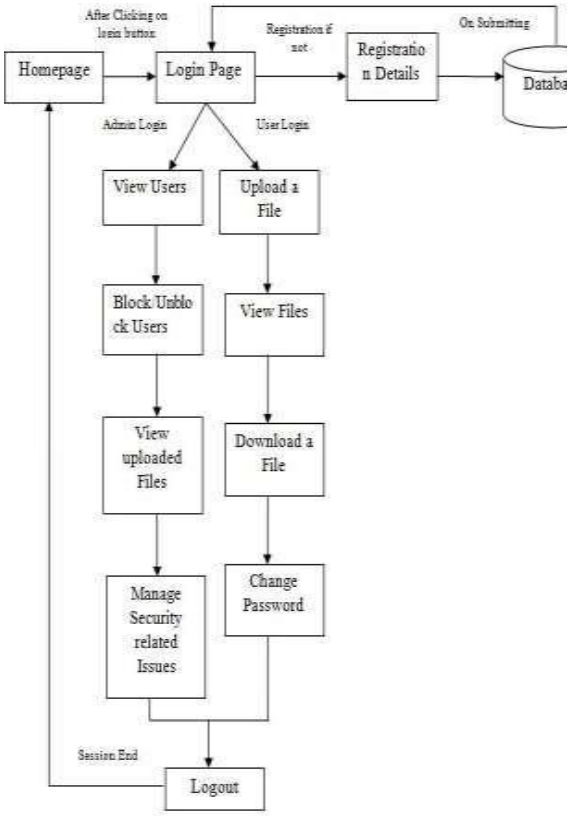

Fig 1: System Architecture

Admin has access to the files in the database as well as the user information, which he or she stores in the database. Admin has the ability to create new users as well as block and unblock existing users. The user has the option of uploading or downloading files.

The user has the option to login or register. The user can then upload files, view and download them. If the file already exists in the database, it will alert the user about the file being already there. If it doesn't exist the file will be saved in the database. The user can then end the session by simply logging out.

b) Sequence Diagram

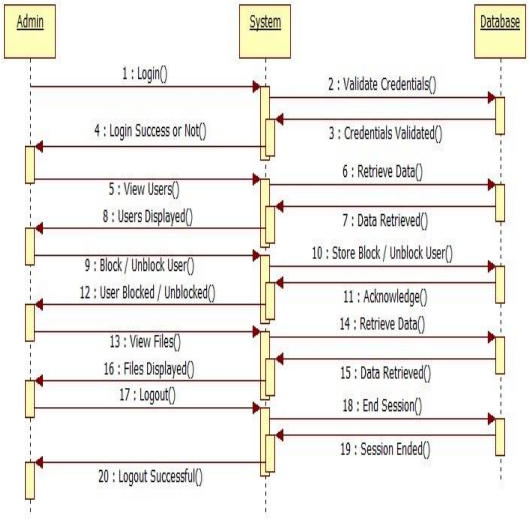

Fig 2: Sequence diagram of admin

This diagram explains the sequence of the login process of the admin into the system. The admin inserts the login credentials, which are verified from the database. Once logged in the admin can manage the files in the database as well as block or unblock any user.

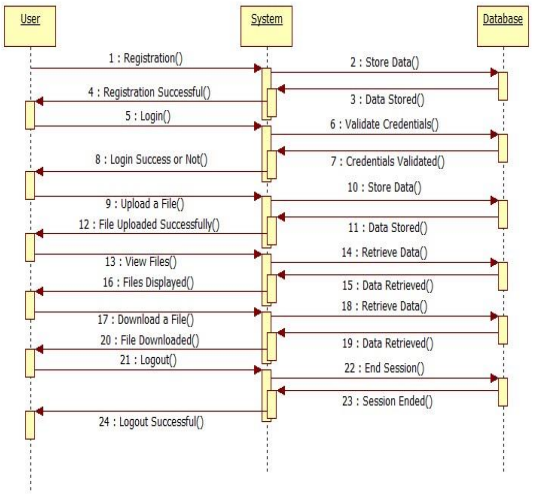

Fig 3: Sequence Diagram of User

This diagram explains the sequence of the login process of the user into the system. The user tries to login into the system with his login credentials, if the credentials are verified from the database if found correct, he logins successfully if not he can use the option

forgot password to reset the password and then login with the new password.

c) Class Diagram

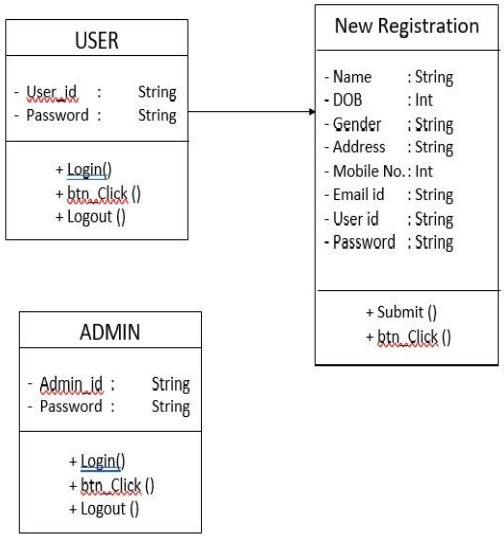

Fig 4: Class Diagram

VII. Results and screenshots

Following snapshots shows the implementation results of the proposed system.

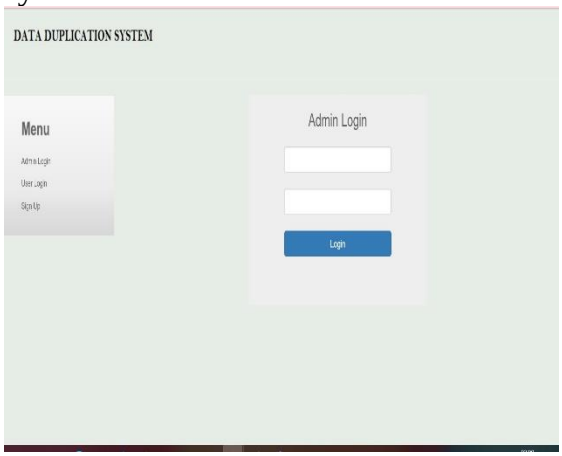

Fig 5: Homepage

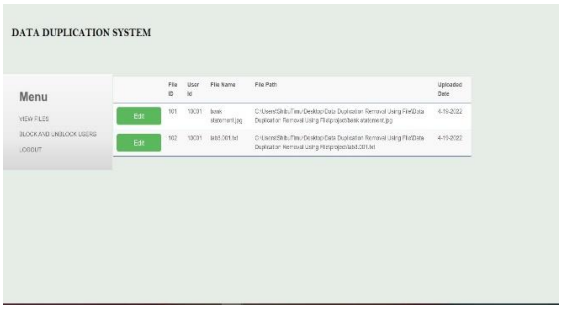

Fig 6: Admin Dashboard

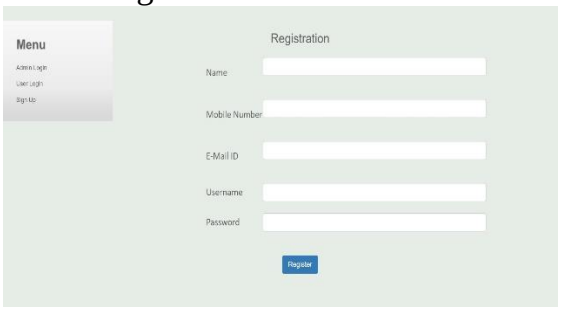

Fig 7: New User Registration Page

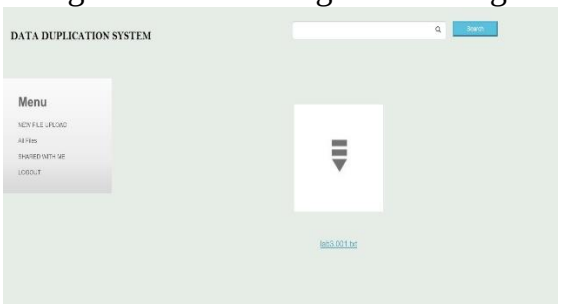

Fig 8: User Dashboard

Fig 9: New file Upload

Fig 10: File already uploaded.

VIII. Conclusion

Using the file checksum technique, this technique aims to create a web-based application that can quickly and

easily detect redundant data. The checksum of both existing and new files is calculated using the Message Digest (MD-5) algorithm. To calculate the checksum and provide stronger safety and encryption for users' valuable files, the MD-5 algorithm is used. As a result of its enhanced security, this system removes duplicate files quickly and easily.

References

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

[2] V.Sathiya Suntharam, Sheo Kumar, Chandu Ravi Kumar, "Research Method of Data Deduplication Backup System" International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8, Issue-11S2, September 2019.

[3] Shynu P.G, Nadesh R.K, Varun G. Menon, Venu P., Mahdi Abbasi & Mohammad R. Khosravi, "A secure data deduplication system for integrated cloudedge networks" Journal of Cloud Computing 9, article number 61(2020).

[4] Osuolalea.Festus, "Data Finding, Sharing and Duplication Removal in the Cloud Using File Checksum Algorithm" International Journal of Research Studies in Computer Science and Engineering (IJRSCSE), Volume 6, Issue 1, 2019, PP 26-47, ISSN 2349-4840 (Print) & ISSN 2349-4859 (Online).

[5] Nourah Almrezeq , Mamoona Humayun , A. A. Abd El-Aziz,and NZ Jhanjhi "An Enhanced Approach to Improve the Security and Performance for Deduplication", Turkish Journal of Computer and Mathematics Education , 2866 Research Article Vol.12 No.6 (2021), 2866-2882.