RELIABILITY AWARE DISTRIBUTED DATA DEDUPLICATION

A Major Project Report

submitted in partial fulfillment of the requirements for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER ENGINEERING

Submitted By,

Aakriti Shukla [B204037]

Sohan Patil [B204034]

Under Guidance of,

Prof. Amar More

SCHOOL OF COMPUTER ENGINEERING & TECHNOLOGY

Alandi (D), Pune – 412105, Maharashtra (INDIA)

MAY 2021

MIT | Academy of Engineering

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

CERTIFICATE

This is hereby certified that the work which is being presented in the B.Tech. Major Project Report entitled "Reliability Aware Distributed Datadeduplication", in partial fulfillment of the requirements for the award of the **Bachelor of Technology in Computer Engineering** and submitted to the **School of Computer Engineering & Technology of MIT Academy of engineering, Alandi(D), Pune** is an authentic record of work carried out during an Academic Year 2020-2021, under the supervision of Prof. Amar More,**School of Computer Engineering & Technology.**

Aakriti Shukla	PRN No. 0120170180	Exam Seat No. B204037
Sohan Patil	PRN No. 0120170170	Exam Seat No. B204034

Date:

Signature of Project Guide
Prof. Amar More,
School of Computer Engineering and Technology,
MIT AOE, Alandi(D), Pune

Signature of School Dean
Prof. Ranjana Badre,
School of Comp. Engg & Technology,
MIT AOE, Alandi(D), Pune

Signature of External Examiner/s

Name:
Affiliation:

Signature of Director
Director,
MIT AOE, Alandi(D), Pune

Abstract

Flexibility and cost efficiency provided by cloud storage vendors like Amazon, Google Cloud Platform, etc. are attracting many organizations for migrating their data to the cloud storage . Also the number of people using social media like Facebook, Twitter, WhatsApp have already crossed the count of some billions. The amount of the data posted by the people on those social media is also increasing exponentially. The studies have shown that, among the data posted by people, more than 50 percent of the data is duplicate. If we think from cloud storage provider point of view, then storing duplicate data require more storage space and energy which actually is a waste. If we could detect this duplicate data and store only one copy of it, then lot of space and energy will be saved. For maintaining the reliability of the data, the storage providers will have to replicate data, thereby generating the duplicate data. Thus reliability and deduplication are two sides of one coin and if handled efficiently, will help in reducing the extra space and energy to store data and also provide the reliability. In this project, we propose energy efficient reliability aware distributed data deduplication for storing data. The algorithm will detect the duplicate data from the data stored on many servers, and will maintain only one copy of the data and to provide reliability, the algorithm will maintain the multiple copies of this to achieve both deduplication and reliability. We make use of content defined chunking to detect the duplicate data more efficiently and use distributed hash tables to reduce the read and write latency

---

Acknowledgment

Every orientation work has an imprint of many people and it becomes our duty to express deep gratitude for the same. During the entire duration of this seminar, we received endless help from a number of people and we feel that this report would be incomplete if we don't convey thanks to them. This acknowledgement is a humble attempt to thank all those who were involved in the project work and were of immense help to us. We want to express our gratitude towards our respected project guide Prof.Amar More for his constant encouragement and valuable guidance during the completion of this project work. We also want to express our gratitude towards respected School Dean Mrs. Ranjana Badre for her continuous encouragement. We would be failing in our duty if we do not thank all the other staff and faculty members for their experienced advice and evergreen co-operation.

Aakriti Shukla

Sohan Patil

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

In a cloud management system, with the explosive growth of digital data, data deduplication Techniques are widely employed to backup data. Data deduplication is a way for lowering the quantity ofstorage area and might assist businesses to growth performance ofstorage, backup and decrease operational costs. It removes duplicate data at both sub file level and file level and recognizes redundant content by calculating its hash collision resistant fingerprint which is secure and cryptographically hashed which accelerates the task computation as compared to the traditional compression approaches in large-scale storage systems also a key to backup data in industry trend of data deduplication. If we think from cloud storage provider point of view, then storing duplicate data require more storage space and energy which actually is a waste. If we could detect this duplicate data and store only one copy of it, then lot of space and energy will be saved. For maintaining the reliability of the data, the storage providers will have to replicate data, thereby generating the duplicate data. Thus, reliability and deduplication are two sides of one coin and if handled efficiently, will help in reducing the extra space and energy to store data and also provide the reliability. Data deduplication is a way that is used to track and dispose of the duplicate chunks (piece of facts) in a storage unit. So many carriers are using this period to implement for efficient data storage, but there may be separate

merits and demerits. Deduplication is extra essential at the shared storage degree, but, implementations in software program software and the database. The most suitable candidates for deduplication are platform virtualization and backup server, because of the truth every applications will use and produce some of identical/replica copies. However, few carriers offer in-vicinity deduplication, which deduplicates primary storage. Deduplication takes vicinity on the record degree and block degree. In record degree deduplication, it receives rid of replica or redundant copies withinside the identical record. This form of deduplication is called as single instance storage (SIS). In block degree deduplication, it receives rid of redundant or duplicated blocks of facts that is observed in unique files. Block-degree deduplication reduces extra vicinity than SIS, this form of deduplication is referred to as variable block or a variable duration deduplication. Since the word facts deduplication is used as a synonym for block-degree or a variable duration deduplication. Data deduplication is one of the well growing generation for optimizing the storage and it saves a hundreds of coins in organizations with the useful resource of the usage of lowering the storage and bandwidth cost. It is useful for cloud carriers, because of the truth this technique needs tons much less hardware to store the statistics.

1.2 Project Idea

With the exponential growth in digital data, Data deduplication has become fundamental technique in moving data to the cloud. It is method used to track and get rid of the duplicate chunks (piece of data) in a storage unit. Extra copies of same data are removed leaving single copy. Duplicate byte patterns are identified across the file and single instance of multiple copies is stored recording the meta data so that the file can be reconstructed when demanded by user. It also plays important role in backup systems. Only the incremental data between source file and backup file is uploaded optimizing the storage and saves a lot of money in companies by reducing the storage and bandwidth cost. It is helpful for cloud providers, because this technique needs less hardware to store, reduces disk I/O operations and increase space efficiency.

1.3 Motivation

Flexibility and cost efficiency provided by cloud storage vendors like Amazon, Google Cloud Platform, etc. are attracting many organizations for migrating their data to the cloud storage. Also, the number of people using social media like Facebook, Twitter, WhatsApp have already crossed the count of some billions. The amount of the data posted by the people on those social media is also increasing exponentially. The studies have shown that, among the data posted by people, more than 50 percent of the data is duplicate.

1.4 Project Challenges

The first challenge faced was to how to make the first chunk of variable size. The second challenge faced was how to delete the file because if two users upload the same file then, on deletion of the file the chunks will be deleted too, but the same chunks are necessary for the second user too, which would become a problem later. The third challenge was to how to keep a track of all the files uploaded by the user, which in turn means how to maintain the database tables. The fourth challenge was to determine and select which database to use which mean in-memory database orthe normal Disk Database. The next and to final challenge was to keep a track of the files updated by the user and reuploaded by hem. These were the key challenges faced while developing the project.

1.5 Proposed Solution

In this project, we propose energy efficient reliability aware distributed data deduplication for storage clouds. The algorithm will detect the duplicate data from the data stored on many servers, and will maintain only one copy of the data and to provide reliability, the algorithm will maintain the multiple copies of this to achieve both deduplication and reliability. We make use of content dependent chunking to detect the duplicate data more efficiently and use distributed hash tables to reduce

the read and write latency

Chapter 2

Literature Review

1. This paper affords the historical past and key functions of facts deduplication concept. Also, the state-of-the art studies in facts deduplication in step with the important thing workflow of the facts deduplication technique is summarized. The precis and taxonomy of the country of the artwork on deduplication assist discover and apprehend the maximum essential layout issues for facts deduplication systems. The predominant programs and enterprise fashion of facts deduplication is mentioned similarly the observe offers a listing of the publicly to be had reassets for deduplication studies and. Secure strategies can shape the premise for growing strategies with greater extensive coverage.

2. This paper an strive has been the set of rules methods of the classical or kingdom of artwork chunking algorithms which includes Rabin Chunking Algorithm, LMC Chunking Algorithm, AE Chunking Algorithm, RAM Chunking Algorithm and the shortcomings of those algorithms in finding incremental facts are mentioned withinside the study. Paper proposes a singular facts chunking set of rules referred to as MII, that's specially applied to discover incremental facts in facts synchronization system. The key function of the MII set of rules is that a sturdy resistance towards the byte shifting trouble is received via way of means of sacrificing a few balance of bite size, which permits to locate incremental facts greater correctly in facts synchronization system.[The time and space complexity of the referred to algorithms is in comparison and analyzed w.r.t Minimal Incremental Chunking Algorithm.

3. In this paper an strive is made explaining the method which are extensively used to eliminate chunks of replica information. It evaluates the strategies primarily based totally on protection and protection of information and confidentiality as the important thing parameter of evaluation. Location primarily based totally, Time primarily based totally, goal strategies are mentioned as those strategies contain handling the users information, the information has the hazard to protection and may be breached at many levels.

Chapter 3

Problem Definition and Scope

3.1 Problem statement

To develop an algorithm to detect duplicate data and to maintain a single copy of the data using variable length chunking.

3.2 Goals and Objectives

Our purpose is to develop an algorithm for data deduplication. The proposed approach should accomplish the following tasks:

- To remove duplicate data chunks.
- To store only one copy of the duplicate data.
- To achieve the goal of saving storage space in storage backup systems.
- To increase storage efficiency and reduce storage costs.
- To minimize the transmission of redundant data in low bandwidth network environments.

3.3 Scope

The project scope is limited to the establishing working data deduplication model with maximum possible efficiency for text based files.

3.4 Hardware and Software Requirement

Hardware

- HDD: 1TB HDD or 500GB SSD or more
- RAM: 4GB
- Processor: Intel i5 or more

Software

- MYSQL 8.0 Command Line Client
- IntelliJ IDEA Community Edition 2019.3.3

3.5 Expected Outcomes

The primary outcome of the algorithm is to create variable sized chunks of the file been provided. The second outcome is if the same file has been uploaded with some variations in It then the algorithm must detect those updated chunks and store them and in the mean while ignore those chunks which already exist. This way it would help in reducing the storage space and at the same time decrease the backup time. The final outcome is to create the versions of the same file which has been uploaded with few variations in it so that the user can revert back if the present version of the file is not satisfactory.

Chapter 4

Systems Analysis and Design

4.1 Overall Description

The widespread subject matter at the back of that is to address record information as a whole. A chew is a fraction of record wherein information is saved with minimal redundancy to store garage area and make efficient use of it. Redundant occupies area and therefore, is wasteful. If variations of the information are in special stages of updating the system frequently offers conflicting information. It is fundamental however poorly-solved to locate the incremental information among backups and source information for incremental backup technology.To discover the incremental information at some point of the backup process, the supply information and backup information are chunked into a few small chunks withinside the equal manner with the variable length. Then, with the aid of using evaluating whether or not a bit of source information isn't like any of the chunks in backup information, we will evaluate whether or not the chew of supply information is incremental information.

4.1.1 Product Perspective

It is The Data Deduplication portal System gives clean mechanism for clients to shop and backup their files. The following are the number one capabilities which are included in Data Deduplication System Portal :

- User account: The system lets in the person to create their accounts withinside the system and provide functions of updating and viewing profiles
- Number of customers being supported with the aid of using the machine: Though the range is exactly not stated however the machine is capable of aid a big range of customers at a time.
- Versioning: Provides customers with a platform to keep variations of the identical record with few changes and assist person to revert again if needed.
- Data Retrieval : Allows person to retrieve documents that she or he has uploaded.

4.1.2 Product Function

The major functions the proposed system must perform or must let the user perform are as listed down below:

- Upload File
- Download File
- Update File
- Add User
- Delete User

4.1.3 User Characteristics

The various user classes that will anticipate and use the product are those in the field of Information Technology, companies dealing with huge amount of data. IT specialists and researchers working in cloud and backup technology domain.

4.2 Specific Requirements

4.2.1 User Requirements

The customers of the system are the groups who need to save their records at the cloud in an efficient an cost effective way. The customers are assumed to have fundamental understanding of dealing with records and backup technology. The directors of the system have to have greater understanding of the internals of the system and rectify issues withinside the instances of catastrophes on the way to keep the system. The consumer manual, on line assist and manual for installation, right UI are enough to help customers on a way to use the system. The admin provides users with:

- Forgot Password
- Data Backup and Recovery
- Maintaining Files

4.2.2 Functional Requirement

Description and Priority	• This function allows the user to backup their files. The algorithm would them create chunks of the backed up data and store hash codes on to the sever in form of hash code tables. • On making changes in the file and backing it up again the server who not backup the entire file. Rather it'd move reference the chunks wherein adjustments were made and eradicates the duplicates and backup simplest the adjustments. • By this means the amount and time required to backup would be reduced and help in efficient and effective performance.
Inputs	Two files • The Original file • Original files with few changes.
Source	All inputs are provided from the local storage.
Output	Chunks of the the original file will be create in the assigned directory and the size of the backup being taken will be lesser
Destination	The outputs are displayed at the display screen in addition to saved withinside the system.
Requires	The user provides files that needs to be backup.

Table 4.1: Functional Requirements

4.2.3 Performance Requirement

- Processor usage- The amount of processor resources that are used depends on how many client sessions or server processes are simultaneously active. Additionally, the amount of processor usage is increased because of other factors,

such as the size of the files that are backed up. When I/O bandwidth is available and the files are large, for example 1 MB, finding duplicates can use an entire processorduring a session or process. When files are smaller, other bottlenecks can occur. These bottlenecks can include reading files from the client disk or the updating of the database. In these bottleneck situations, data deduplication might not use all of the resources of the processor. You can control processor resources by limiting or increasing the number of client sessions for a client or a server duplicate identification processes. To take advantage of your processor and to complete data deduplication faster, you can increase the number of identification processes or client sessions for the client. The increase can be up to the number of processors that are on the system.

- Network bandwidth - A primary reason to use client-side data deduplication is to reduce the bandwidth that is required to transfer data.The amount that the bandwidth is reduced by is directly related to how much of the data is duplicate that is already stored on the server. If an extent is found that was previously sent, it is not necessary to query the server again for that extent. Therefore, bandwidth and performance are not additionally reduced.

- Safety Requirements- If there may be enormous harm to a huge part of the database because of catastrophic failure, inclusive of a disk crash, the recuperation technique restores a beyond replica of the database that became sponsored as much as archival garage and reconstructs a extra modern nation with the aid of using reapplying or redoing the operations of dedicated transactions from the sponsored up log, as much as the time of failure.Cloud systems are dependable due to the fact they invent 3 copies of every record over 3 extraordinary databases and the above stated facts allows us to retrieve facts on catastrophic failure.

Chapter 5

Methodology/ Approach/ Techniques

5.1 Architecture of Systems

The system proposed deals with the process to eliminate redundant data so that a single copy is stored instead of multiple copies of same data. The diagram of system architecture is as follows: We have used Rabin Karp Fingerprinting Algorithm

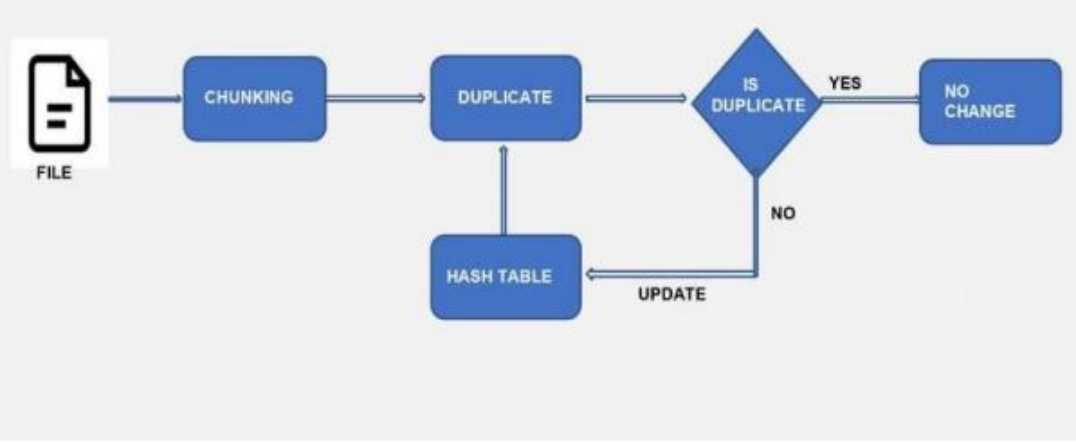


Figure 5.1: System Architecture

to divide file into chunks of variable length. In spite of the fact that fixed sized chunking is simple and easy, if data is inserted at the beginning or in middle subsequent data chunks are affected. In order to overcome this disadvantage, we use

the content defined chunking approach. The Rabin Chunking Algorithm takes the file data as byte stream. There is a predefined fixed length window which slides over the entire file data. Rolling hash is computed of the data that falls into this window and compared with the pre-set value. This is the condition to determine the cut point of the chunk. Also known as boundary condition. If match is not found then the fixed length moves head byte by byte. The detailed explanation is as follows:

A. Sliding Window and Rolling Hash Here we are using the concept of bitmask for calculation of rolling hash.

0000000000000 – Window start

0000000000001

...

1111111111111

0000000000000 – New window start

Initially 13 bits of the hash are all ones and we decrement the mask by one. We start a new window when the bits of the hash are all zeroes. If we are aiming for a 8k window size, we have used the lowest 13 bits of the hash to decide when to start a new window

Min window: 1

Max window: 81287

Average window: 8132

Median window: 5658

By using this method if modifications are done in some parts of the file, they will affect only the current and subsequent chunk at max, remaining chunks remain same. So, we get chunks of reasonable size using this approach. Keeping even the first chunk variable helped us achieve better results. The determined cut point conditions is if the lowest 13 bits of the hash are all zeroes i.e((hash mask) == 0)

B. Hash table structure Using the hashing function rolling hash is generated and

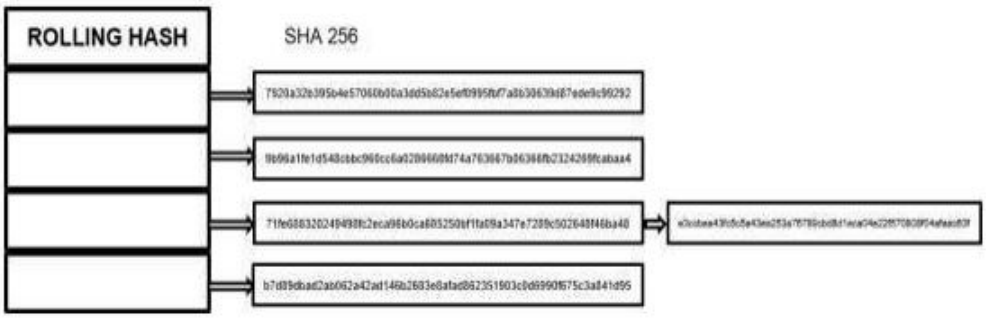


Figure 5.2: Hash Table Structure

stored in the hashmap.But rolling hash is more likely to have collision.So to handle this problem of collision we used SHA256 hash.SHA256 algorithm for calculation hash generates unique value even for a slightest change in the data and is less prone to the problem of collisions.If collision occurs the its corresponding SHA256 hash is calculated and is attached as arraylist as shown in the figure above.

5.1.1 UML Diagram

- Use Case Diagram

  Use case illustrates a unit of capability furnished via way of means of the system. The principal cause of the use- case diagram is to assist improvement groups visualize the purposeful necessities of a system, such as the connection of "actors" to important processes, in addition to the relationships amongst special use instances. Use-case diagrams typically display corporations of use instances, both all use instances for the entire system, or a breakout of a selected organization of use instances with associated capability to Show a use case on a use-case diagram, you draw an oval withinside the center of the diagram and positioned the call of the use case withinside the middle of, or below, the oval. To draw an actor (indicating a system consumer) on a use-case diagram, you draw a stick individual to the left or proper of your diagram. Following diagram indicates the relationships of the consumer or actors with the use instances which are shown in an oval shape.

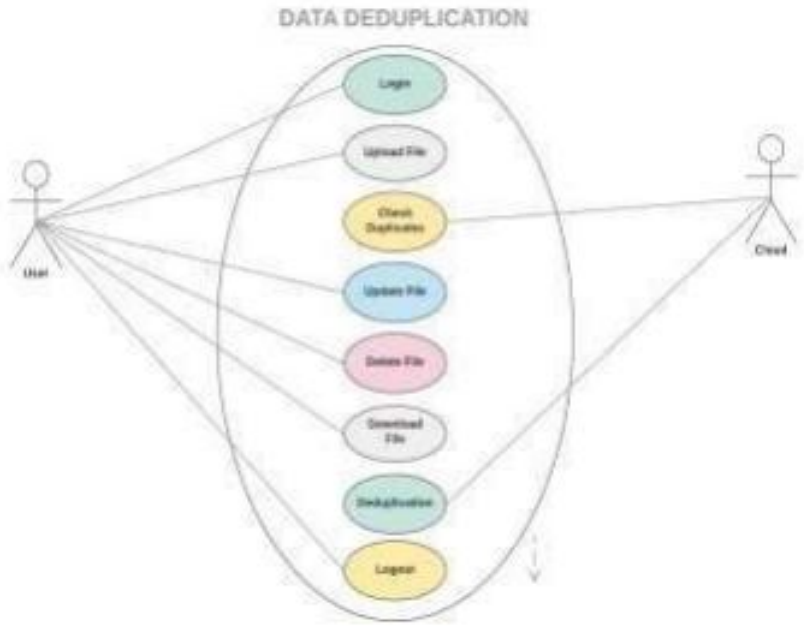


Figure 5.3: Use Case Diagram

- Activity Diagram

  Activity diagram is commonly used for enterprise technique modeling, for modeling the logic captured via way of means of a single use case, or for visualizing the unique logic of a enterprise rule. Complicated technique flows withinside the system are captured withinside the activity diagram. Similar to a kingdom diagram, an activity diagram additionally includes activities, actions, transitions, preliminary and very last states, and protect conditions. However, distinction is kingdom diagrams are in context of simulation whilst activity offers element view of enterprise logic. Activity diagrams are "much less technical" in appearance, as compared to sequence diagrams, and enterprise-minded humans have a tendency to recognize them greater quickly

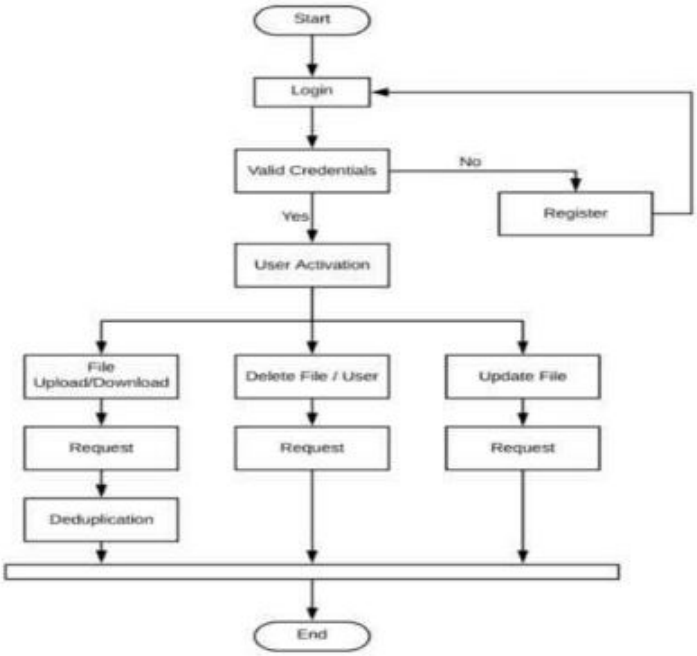
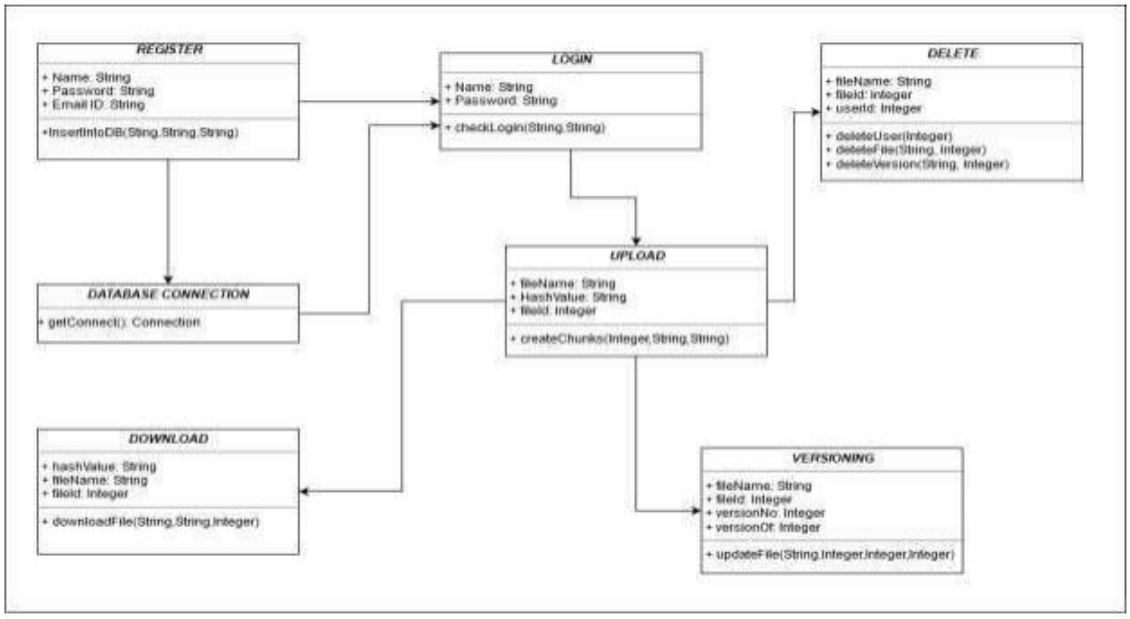


Figure 5.4: Activity Diagram

- Class Diagram

  In software engineering, a class diagram withinside the Unified Modelling Language (UML) is a form of static shape diagram that describes the shape of a system by displaying the system's classes, their attributes, operations (or methods), and the relationships amongst objects.



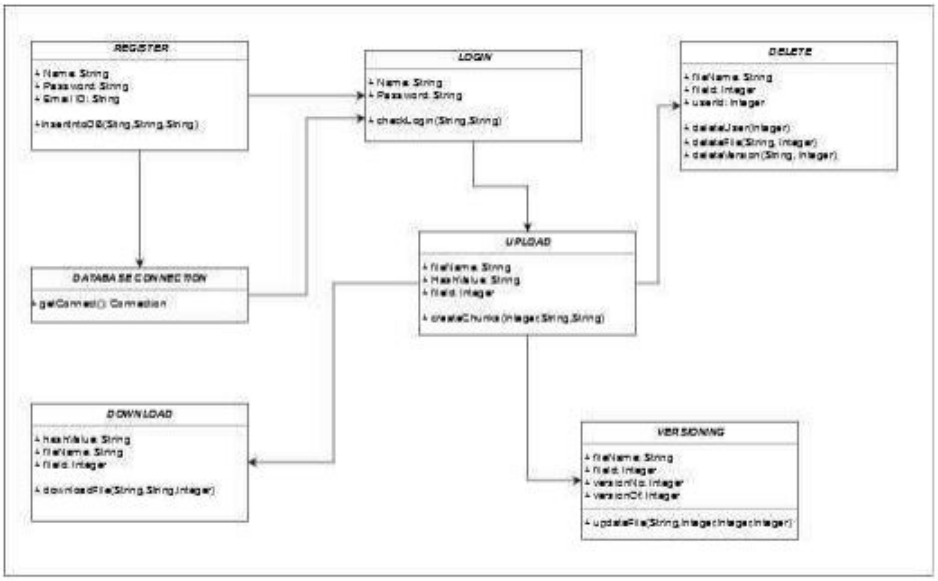


Figure 5.5: Class Diagram

5.1.2 Data Flow Diagram

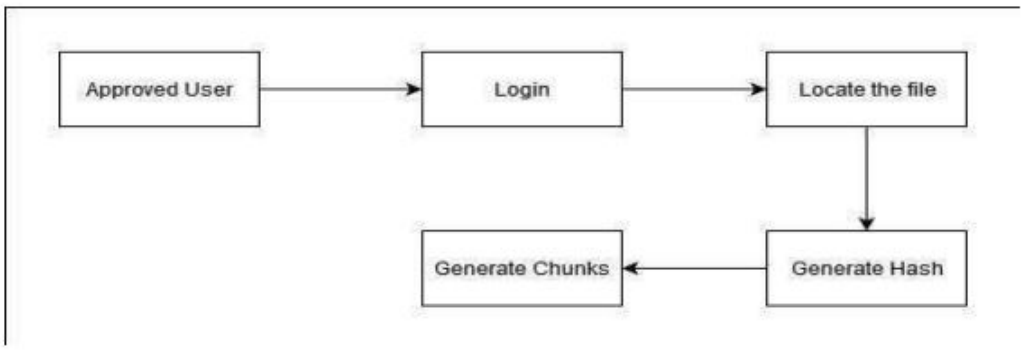


Figure 5.6: Level 1

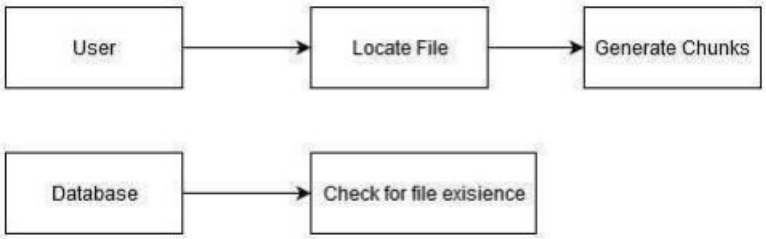


Figure 5.7: Level 2

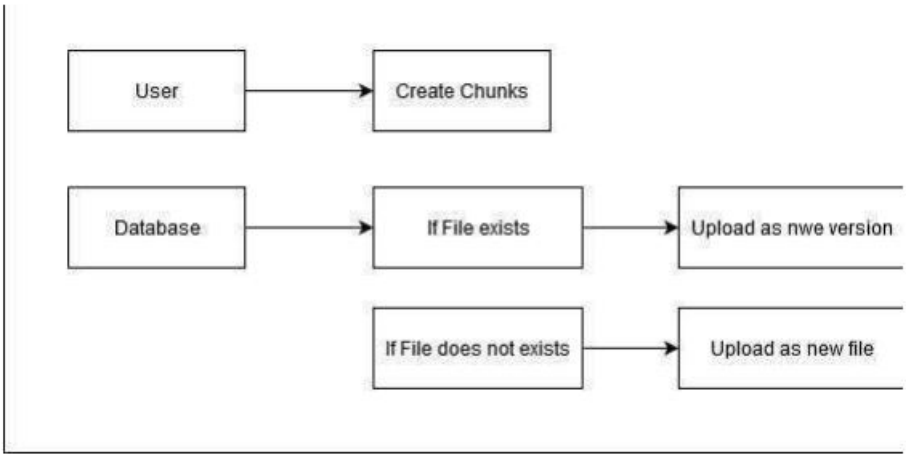
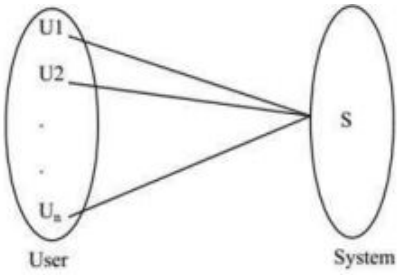


Figure 5.8: Level 3

5.2 methodology



Where,U1, U2.... Un =Users. S = System Whenever user wants to upload the file on system, then check or test the duplication. Process:

- Step 1: Open account.
- Step 2: Upload file on storage.
- Step 3: System checks for the duplicate file available on storage system.
- Step 4: If found then remove the duplication and maintains index co.
- Step 5: On non-duplicate data, check for deltas.
- Step 6: Store unique and deltas on system in encrypted form.

Mathematical model contains five tuples –

S = {s, e, X, Y, ϕ }

Where, the following conditions are satisfied-

s = Start of the program Log in.

To access the facilities of system such as store on system, user has to log into system.

Upload text Files on system.

Upload files on system in text format.

X = Input of the program. Input should be any text file.

Y = Output of the program.

e = End of the program.

ϕ = Success and failure conditions File will be first fragmented then it is encoded and the fragments are allocated.

{X, Y Ξ U}

Let U be the Set of System.

{U} = {Client, F, S, T, M, D, R, DC}

Where,

Client, F, S, T, M, D, R, DC are the elements of the set.

{Client} = Data Owner, User.

{F} = Fragmentation

{T} = Generates fingerprints for file and blocks.

{D} = Check for duplicate file or block.

{R} = Detects similarity by using existing information of a deduplication system.

{DC} = Delta compression module takes each of the blocks detected previously, and reads its base-chunk, and then delta encodes their differences.

Chunking:

Before storing the documents on system, Files are broken down into chunks such as, F = FC1,FC2....Fcn

Deduplication Checking:]

(New chunk) = h H (Old n chunks)

If H (New chunk) == H (Old n chunks [])

Chunk is duplicate and do not store it, instead provide link.

Else Chunk is not duplicate, and then stores it.

Success Condition

File splitting and storing it on a couple of nodes. User receives end result very

rapid in keeping with their needs.

Failure Condition

Hardware failure.

Software failure.

Maintaining indexing leads to more time consumption to get the proper file stored on system.

Space Complexity

More the storage of data more is the space complexity.

Time Complexity

Time complexity of system relies upon on following factors: time taken to add document, time taken for the duration of file level and block level deduplication, delta calculating, storing deltas and non-reproduction records in encrypted layout on specific nodes.

5.3 Approach

The approach used to develop data deduplication is variable sized chucking which is an upcoming method. Using this method every chunk generated would be of a different size due to which the drawbacks of fixed sized chunking can be overcome. Added to variable sized chunking the algorithm also consists of file versioning so that on uploading the same file again after changes the user can differentiate between the previous file and the new file which has been uploaded. The entire project has been developed using anIn-memory database which reduced the loss of data due to mechanical or software failure

Chapter 6

Implementation

6.1 System Implementation

For implementation we desired C++ language, For message digest we've got used outside sha256 C++ library. The SHA (Secure Hash Algorithm) is one of the famous cryptographic hash functions. A cryptographic hash may be used to make a signature for a textual content or a facts record. Data deduplication is known as a approach provided to Cloud Storage Providers (CSPs) to cast off the duplicate data and preserve handiest a single unique replica of it for storage space saving purpose. Data deduplication is one of the strategies which used to resolve the repetition of data. The deduplication strategies are normally used withinside the cloud server for decreasing the gap of the server. To save you the unauthorized use of facts having access to and create replica facts on cloud the encryption method to encrypt the facts earlier than saved on cloud server. Cloud Storage commonly consists of business-vital data and processes; therefore excessive protection is the handiest approach to keep robust accept as true with courting among the cloud customers and cloud provider carriers In this system we need to discover the duplicate copy of the record any sort of record may be discover record .txt,.doc,.xls, ppt, .pdf. so we need to begin with importing the record while we add the record we need to extract first 1024 characters from record if INT Hash or sha256 value matches with present INT Hash or sha256, we are able to discard that precise chunk i.e. we're stopping duplicate data from

being keep in cloud. Also we are able to be keeping the shaCount of the chunk in order that deduplication may be manipulate efficiently. If INT Hash or sha256 value doesn't matches, then we are able to keep that chunk.

6.2 Data Description

Currently the algorithm has the ability to deal with any type of text documents, to be specific it deals with txt, doc, odt and pdf. Not only does the algorithm deal with text documents but it can also split an image in to chucks and combine them back to recreate the same. The algorithm creates a window which keeps moving and as soon as the window size ends the chunk is created. The code is first fed with the file and the chunks are generated and kept in a folder.

6.3 Functional Implementation

Propose a system that gives the capacity create variable sided chunks of the given records and keep them accordingly, The system additionally possesses the ability to keep most effective the ones chunks that have been up to date now no longer the ones which already exists.

- Performance The performance of the system will provide faster chunking of the files.
- Capacity: Capacity of project according to data or the number of files being uploaded.
- Availability: User has allowed for login after activation of user's account. User gets result after uploading the file.
- Reliability: System is reliable for maintaining the privacy and security of the sensitive information of the user and their files.
- Security: The system is secure because information of the user's personal details in an account is not leaked or spread anywhere

6.4 Expected Output

Figure 6.1: Registration

Figure 6.2: User Entry in database after registeration

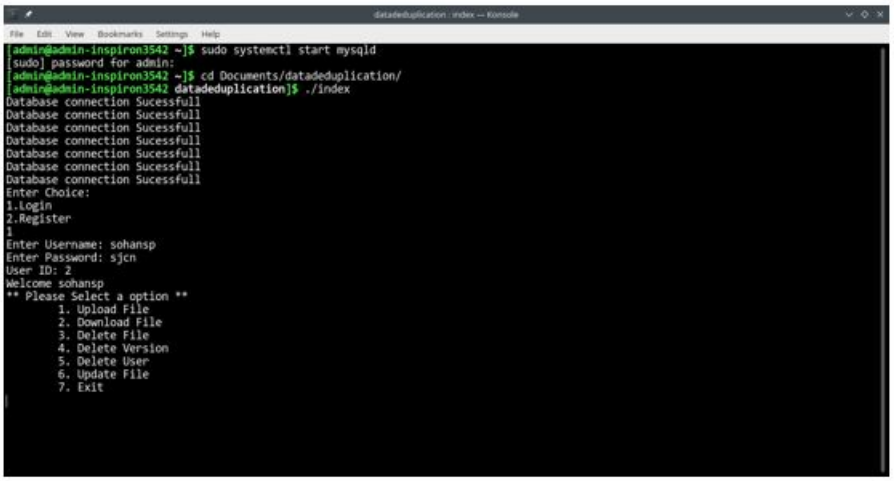


Figure 6.3: Login

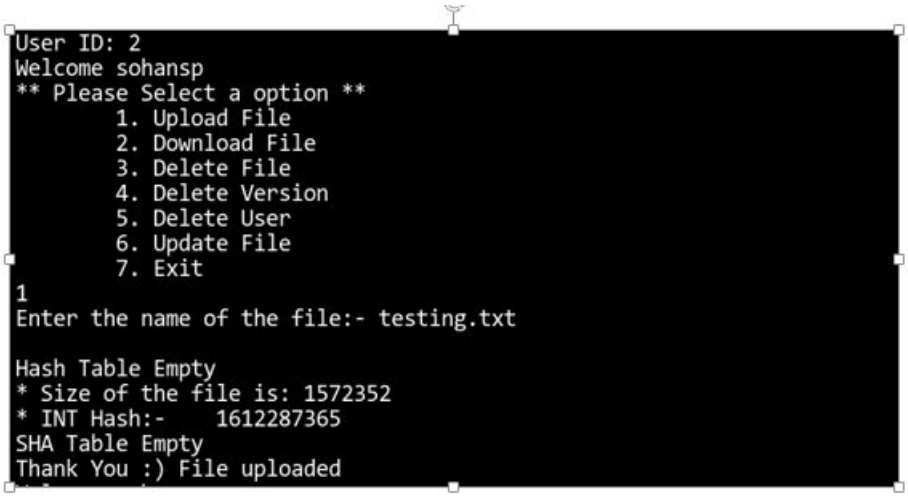


Figure 6.4: File Upload

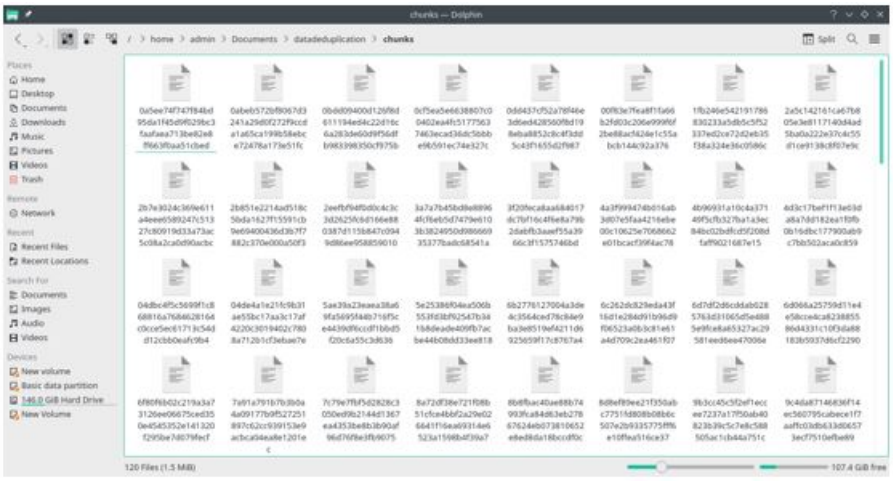


Figure 6.5: Generated Chunks(Files) in system

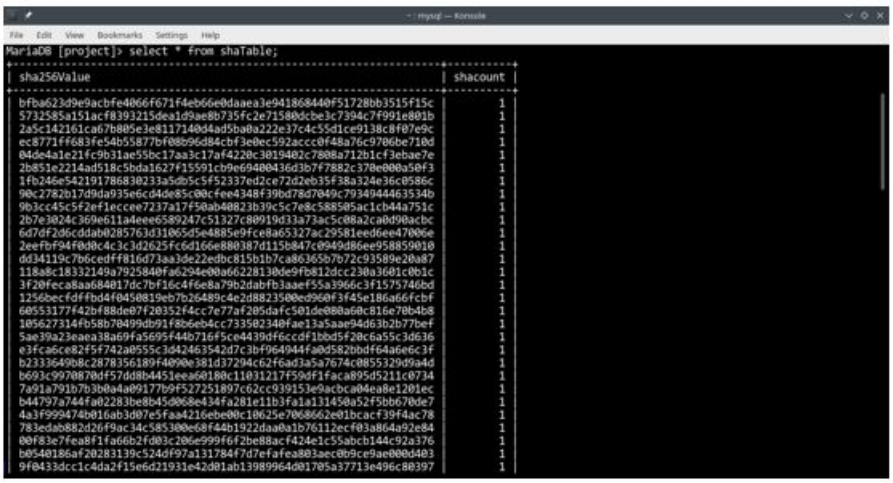


Figure 6.6: Sha256values in ShaTable

```
MariaDB [project]> select * from userFile;
+------------+--------+-------------+---------------------+-----------+-----------+----------+
| userFileId | userId | fileName    | fileDateTime        | versionNo | versionOf | fileSize |
+------------+--------+-------------+---------------------+-----------+-----------+----------+
|        965 |      2 | testing.txt | 2021-04-29 17:44:46 |         1 |       965 |  1572353 |
+------------+--------+-------------+---------------------+-----------+-----------+----------+
1 row in set (0.000 sec)
```

Figure 6.7: UserFile entry in database

```
Welcome sohansp
** Please Select a option **
       1. Upload File
       2. Download File
       3. Delete File
       4. Delete Version
       5. Delete User
       6. Update File
       7. Exit
2
---------------------------Download File---------------------------
Files you have:-
FileID  FileName
965     testing.txt
Enter FileID: 965

File downloaded sucessfully..
```

Figure 6.8: Download File

```
Welcome sohansp
** Please Select a option **
       1. Upload File
       2. Download File
       3. Delete File
       4. Delete Version
       5. Delete User
       6. Update File
       7. Exit
6
Enter name of the file : testing.txt

* Size of the file is: 1572352
* INT Hash:-   1612287365
0 => YES hash   YES 256   =>  -1597071360       sha256  5732585a151acf8393215dea1d9ae8b735fc2e71580dcbe3c7394c7f991e801b
1 => YES hash   YES 256   =>  -1913675776       sha256  2a5c142161ca67b805e3e8117140d4ad5ba0a222e37c4c55d1ce9138c8f07e9c
2 => YES hash   YES 256   =>  -141737984        sha256  ec8771ff683fe54b55877bf08b96d84cbf3e0ec592accc0f48a76c9706be710d
3 => YES hash   YES 256   =>  -922705920        sha256  04de4a1e21fc9b31ae55bc17aa3c17af4220c3019402c7808a712b1cf3ebae7e
4 => YES hash   YES 256   =>  667295744 sha256  2b851e2214ad518c5bda1627f15591cb9e69400436d3b7f7882c370e000a50f3
5 => YES hash   YES 256   =>  -1083604992       sha256  1fb246e5421917868308233a5db5c5f52337ed2ce72d2eb35f38a324e36c0586c
6 => YES hash   YES 256   =>  1477656576        sha256  90c2782b17d9da935e6cd4de85c00cfee4348f39bd78d7049c7934944463534b
7 => YES hash   YES 256   =>  -1022263296       sha256  9b3cc45c5f2ef1eccee7237a17f50ab40823b39c5c7e8c588505ac1cb44a751c
8 => YES hash   YES 256   =>  2137389184        sha256  2b7e3024c369e611a4eee6589247c51327c80919d33a73ac5c08a2ca0d90acbc
9 => YES hash   YES 256   =>  -537214976        sha256  6d7df2d6cddab0285763d31065d5e4885e9fce8a65327ac29581eed6ee47006e
10 => YES hash  YES 256   =>  1434222592        sha256  2eefbf94f0d0c4c3c3d2625fc6d166e880387d115b847c0949d86ee958859010
11 => YES hash  YES 256   =>  -664059904        sha256  dd34119c7b6cedff816d73aa3de22ed0c815b1b7ca863653b7b72c93589e20a87
12 => YES hash  YES 256   =>  1615290368        sha256  118a8c18332149a7925840fa6294e00a66228130de9fb812dcc230a3601c0b1c
13 => YES hash  YES 256   =>  -815570944        sha256  3f20feca8aa684017dc7bf16c4f6e8a79b2dabfb3aaef55a3966c3f1575746bd
14 => YES hash  YES 256   =>  649158656 sha256  1256becfdffbd4f0450819eb7b26489c4e2d8823500ed960f3f45e186a66fcbf
15 => YES hash  YES 256   =>  203309056 sha256  60553177f42bf88de07f20352f4cc7e77af205dafc501de080a60c816e70b4b8
16 => YES hash  YES 256   =>  1284284416        sha256  105627314fb58b70499db91f8b6eb4cc73350234 0fae13a5aae94d63b2b77bef
17 => YES hash  YES 256   =>  -1629806592       sha256  5ae39a23eaea38a69fa5695f44b716f5ce4439df6ccdf1bbd5f20c6a55c3d636
18 => YES hash  YES 256   =>  1302593536        sha256  e3fca6ce82f5f742a0555c3d42463542d7c3bf964944fa0d582bbdf64a6e6c3f
```

Figure 6.9: Update File

```
Welcome sohansp
** Please Select a option **
       1. Upload File
       2. Download File
       3. Delete File
       4. Delete Version
       5. Delete User
       6. Update File
       7. Exit
3
---------------------------Delete File---------------------------
Files you have:-
FileID  FileName
966     testing_29-4-2021|17:51:52.txt
Enter FileID: 966

File 1a963906351437cb6dfa7c3aabdd6fd011f560183c6774c496d8fb30abefe3a1 Deleted Sucessfully
File 5732585a151acf8393215dea1d9ae8b735fc2e71580dcbe3c7394c7f991e801b Deleted Sucessfully
File 2a5c142161ca67b805e3e8117140d4ad5ba0a222e37c4c55d1ce9138c8f07e9c Deleted Sucessfully
File ec8771ff683fe54b55877bf08b96d84cbf3e0ec592accc0f48a76c9706be710d Deleted Sucessfully
File 04de4a1e21fc9b31ae55bc17aa3c17af4220c3019402c7808a712b1cf3ebae7e Deleted Sucessfully
File 2b851e2214ad518c5bda1627f15591cb9e69400436d3b7f7882c370e000a50f3 Deleted Sucessfully
File 1fb246e5421917868308233a5db5c5f52337ed2ce72d2eb35f38a324e36c0586c Deleted Sucessfully
File 90c2782b17d9da935e6cd4de85c00cfee4348f39bd78d7049c7934944463534b Deleted Sucessfully
File 9b3cc45c5f2ef1eccee7237a17f50ab40823b39c5c7e8c588505ac1cb44a751c Deleted Sucessfully
File 2b7e3024c369e611a4eee6589247c51327c80919d33a73ac5c08a2ca0d90acbc Deleted Sucessfully
File 6d7df2d6cddab0285763d31065d5e4885e9fce8a65327ac29581eed6ee47006e Deleted Sucessfully
File 2eefbf94f0d0c4c3c3d2625fc6d166e880387d115b847c0949d86ee958859010 Deleted Sucessfully
File dd34119c7b6cedff816d73aa3de22ed0c815b1b7ca863653b7b72c93589e20a87 Deleted Sucessfully
File 118a8c18332149a7925840fa6294e00a66228130de9fb812dcc230a3601c0b1c Deleted Sucessfully
File 3f20feca8aa684017dc7bf16c4f6e8a79b2dabfb3aaef55a3966c3f1575746bd Deleted Sucessfully
File 1256becfdffbd4f0450819eb7b26489c4e2d8823500ed960f3f45e186a66fcbf Deleted Sucessfully
File 60553177f42bf88de07f20352f4cc7e77af205dafc501de080a60c816e70b4b8 Deleted Sucessfully
```

Figure 6.10: Delete File

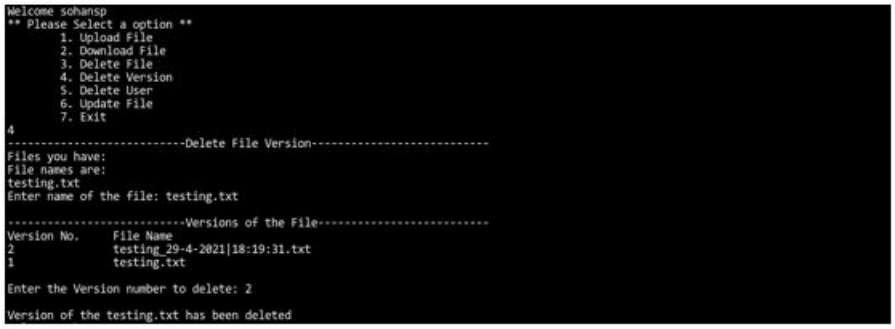


Figure 6.11: Delete Version

Figure 6.12: Delete User

Chapter 7

Results Analysis and Performance Analysis

```
MariaDB [project]> select * from shaTable order by shaCount desc;
+------------------------------------------------------------------+----------+
| sha256Value                                                      | shacount |
+------------------------------------------------------------------+----------+
| 36a9e7f1c95b82ffb99743e0c5c4ce95d83c9a430aac59f84ef3cbfab6145068 |        2 |
| 329036f693029adad5d9db4c99169e5ca86f6af52d4dfaeb220cc0e703995e3d |        2 |
| ea8c671fb21ad650fdcd9524aaff7f2844e746df4f62e1c53468b2a84ecba51a |        1 |
| ad4afa8508cda0fcb42f4ee195143b3f56eda8837f622fc0a3a8e16fe3754a70 |        1 |
| 3bedb61ed9553a56731d9ac4447b8402cd44010acf42fea688adacdfabf50fce |        1 |
| f2ec381674fb47d4e685393616b181f8ee2e2d77be15948445ee3703e4e92d02 |        1 |
| 92da39162dd0e0c36c4f3c3cbabf5082e22c7bd7fcba91c6198d45472a271d10 |        1 |
| fc1aca4a54b24324ca221906d5e02d5450e8f54af26aa1cdbe09a5268e04c085 |        1 |
| 35874d75360985977435c1b2f2dc925e5dd9f7a5d5b8753da9545bbbc1263cb7 |        1 |
| e361d29cb3ce5c8d37b107c187ffb51eef4e22b4a4f7e6a67622a5942853832a |        1 |
| df419b159b8aefee84524ebfadaad1f940495fb3f5f6ae11659efb58faac5cdc |        1 |
| 45e2d131cac62531f5c438ce90e796a5fc60d4b0256ff26904b17b437b632e05 |        1 |
| e3b98a4da31a127d4bde6e43033f66ba274cab0eb7eb1c70ec41402bf6273dd8 |        1 |
| b24990f1e609a939ae22cbb89ccc953973285be84c79481ddb23cc8eca5c39b4 |        1 |
| 810e01cb045147daf57ee8dbea80d60db4048543473ea3614f0e6e4346fbdb39 |        1 |
| 42796b9c3f22fe9ef1cfe2635711ba5321c42d973c835ddf2a1d67f943c6ce86 |        1 |
| 677fae73b7340bca6cfe6a996f81fbf92378348f1ec85108201084c674fce8d4 |        1 |
| f27204ce51e97d3825c465d0b2a39336263f5db4f87da12c6842fd079b56a124 |        1 |
| 14f2066cae57a8d3cc18e04cb60d5383021b484daa368fc416471b44ae68aa55 |        1 |
+------------------------------------------------------------------+----------+
19 rows in set (0.001 sec)
```

Figure 7.1: Avoiding duplication by managing ShaCount

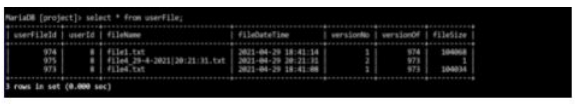


Figure 7.2: Maintaining versions of a file while maintaining duplication

Chapter 8

Conclusion & Future Work

8.1 Conclusion

In this proposed system, we've got specifically surveyed the numerous deduplication techniques. Among them, it's been concluded that variable length information deduplication is nicely and top while in comparison to different techniques with the aid of using evaluating the hash of every and each chunk. Hence, this method improves storage performance and thereby enhance the overall performance with the aid of using allowing storage sources to switch and take care of extra information.

8.2 Future Scope

In future, greater research works may be targeted on variable length chunking technique to lessen processing time, and optimize of big scale data storage. And additionally to expand an efficient technique to lessen fragmentation and acquire excessive write and read throughput.

Appendices

Appendix A

Sponsorship Certificate

Appendix B

Work Completion Certificate

Appendix C

Certificates

Appendix D

Papers presented/published

Appendix E

User Documentation

Appendix F

Special mathematics

Appendix G

Drawings/CAD Sheets

Appendix H

Photographs

Appendix I

Project Team Photo with Advisor

Appendix J

Project Photographs