1. List of Figures

2. List of Tables

3. List of Abbreviation

Sr. No.	Abbreviation	Meaning
1.	MII	Minimum Incremental Interval
2.	AE	Asymmetric Extremum
3.	LMC	Local Maximum Chunking
4.	RAM	Rapid Asymmetric Maximum

CHAPTER 01
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 technique for reducing the amount of storage space and can help organizations to increase efficiency of storage, backup and reduce 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 technique which is used to track and eliminate the duplicate chunks (piece of data) in a storage unit. So many vendors are using this technology to implement for efficient data storage, but there is separate merits and demerits. Deduplication is more important at the shared storage level, however, implementations in software and the database. The most suitable candidates for deduplication are platform virtualization and backup server, because both applications will use and produce a lot of identical/duplicate copies. However, few vendors offer in-place deduplication, which deduplicates primary storage. Deduplication takes place on the file level and block level. In file level deduplication, it eliminates duplicate or redundant copies in the same file. This type of deduplication is called as single instance storage (SIS). In block level deduplication, it eliminates redundant or duplicated blocks of data which is present in unique files. Block-level deduplication reduces more space than SIS, this type of deduplication is known as variable block or a variable length deduplication. Since the word data deduplication is used as a synonym for block-level or a variable length deduplication.

Data deduplication is one of the well growing technology for optimizing the storage and it saves a lots 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 the data.

The advantage of data deduplication are:

- Hardware costs is reduced
- Backup costs is less
- Storage efficiency is increased and
- Improved network efficiency and reduced bandwidth.

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 eliminate 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 or the 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 02
LITERATURE SURVEY

2. COMPARISONS, RESEARCH PAPER STUDIED/RELATED WORK

1. This paper presents the background and key features of data deduplication concept. Also, the state-of-the art research in data deduplication according to the key workflow of the data deduplication process is summarized. The summary and taxonomy of the state of the art on deduplication help identify and understand the most important design considerations for data deduplication systems. The main applications and industry trend of data deduplication is discussed in addition the study provides a list of the publicly available sources for deduplication research and. Secure methods can form the basis for developing methods with more extensive coverage.

2. This paper an attempt has been the algorithm processes of the classical or state of art chunking algorithms such as Rabin Chunking Algorithm, LMC Chunking Algorithm, AE Chunking Algorithm, RAM Chunking Algorithm and the shortcomings of these algorithms in finding incremental data are discussed in the study. Paper proposes a novel data chunking algorithm called MII, which is specifically implemented to find incremental data in data synchronization system. The key feature of the MII algorithm is that a strong resistance against the byte shifting problem is obtained by sacrificing some stability of chunk size, which allows to locate incremental data more accurately in data synchronization system.[The time and space complexity of the mentioned algorithms is compared and analyzed w.r.t Minimal Incremental Chunking Algorithm.

3. In this paper an attempt is made explaining the technique that are widely used to eradicate chunks of duplicate data. It evaluates the techniques based on safety and security of data and confidentiality as the key parameter of evaluation. Location based, Time based, target techniques are discussed as these techniques involve dealing with the users data, the data has the threat to security and can be breached at many levels.

CHAPTER 03
REQUIREMENTS

3. DETAILED 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 REQUIREMENTS

a. Hardware Specifications:

- HDD: 1TB HDD or 500GB SSD or more
- RAM: 4GB
- Graphics Card: If Necessary
- Processor: Intel i5 or more

b. Software Specifications:

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

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.

4. SYSTEM REQUIREMENT SPECIFICATION

4.1 OVERALL DESCRIPTION

The general theme behind this is to handle file data as a whole. A chunk is a fragment of file in which data is stored with minimum redundancy to save storage space and make efficient use of it. Redundant occupies space and therefore, is wasteful. If versions of the data are in different phases of updating the system often gives conflicting information. It is key but poorly-solved to find the incremental data between backups and source data for incremental backup technology. To find out the incremental data during the backup process, the source data and backup data are chunked into some small chunks in the same way with the variable length. Then, by comparing whether a chunk of source data is different from any of the chunks in backup data, we can evaluate whether the chunk of source data is incremental data**.**

4.1.1 PRODUCT PERSPECTIVE

It is a web-based system implementing client-server model. The Data Deduplication portal System provides simple mechanism for users to save and backup their files. The following are the main features that are included in Data Deduplication System Portal :

- User account: The system allows the user to create their accounts in the system and provide features of updating and viewing profiles.
- Number of users being supported by the system: Though the number is precisely not mentioned but the system is able to support a large number of users at a time.
- Versioning: Provides users with a platform to maintain versions of the same file with few modifications and help user to revert back if needed.
- Data Retrieval : Allows user to retrieve files that he or she 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 users of the system are the companies who want to store their data on the cloud in an efficient an cost effective way. The users are assumed to have basic knowledge of handling data and backup technology. The administrators of the system should have more knowledge of the internals of the system and rectify problems in the cases of catastrophes in order to maintain the system. The user manual, online help and guide for installation, proper UI are sufficient to assist users on how to use the system.

The admin provides users with:

- Forgot Password
- Data Backup and Recovery
- Maintaining Files

4.2.2 FUNCTIONAL REQUIREMENTS

★ **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 would cross reference the chunks in which changes have been made and eradicates the duplicates and backup only the changes. By this means the amount and time required to backup would be reduced and help in efficient and effective performance.
★ **Inputs:**	Two files 1. The Original file 2. Original files with few changes.
★ **Source:**	All inputs are provided from the local storage.
★ **Outputs:**	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 on the screen as well as stored in the system.
★ **Requires:**	The user provides files that needs to be backup.

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 processor during 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 is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.Cloud platforms are reliable because they create three copies of each file over three different databases and the above mentioned data helps us to retrieve data on catastrophic failure

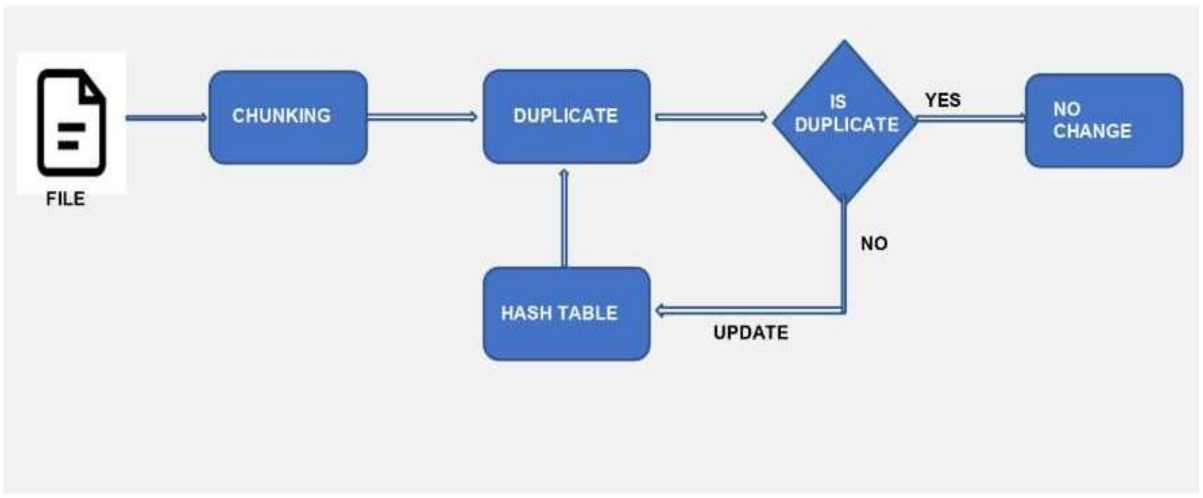
CHAPTER 04

SYSTEM DESIGN

.

4.3 SYSTEM ARCHITECTURE

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 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

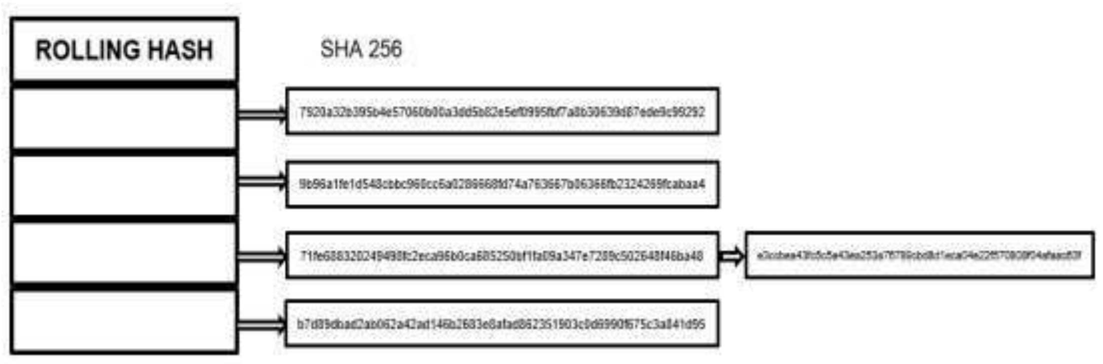

ROLLING HASH
SHA 256
7920a32b395b4e57060b00a3dd5b82e5ef0995fbf7a8b30639d87ede9c99292
9b96a1fe1d548cbbc960cc6a0286668fd74a763667b06366fb2324269fcabaa4
71fe68832024949 8fc2eca96b0ca685250bf1fa09a347e7289c502648f46ba48
b7d89dbad2ab062a42ad146b2683e8afad862351903c0d6990f675c3a841d95

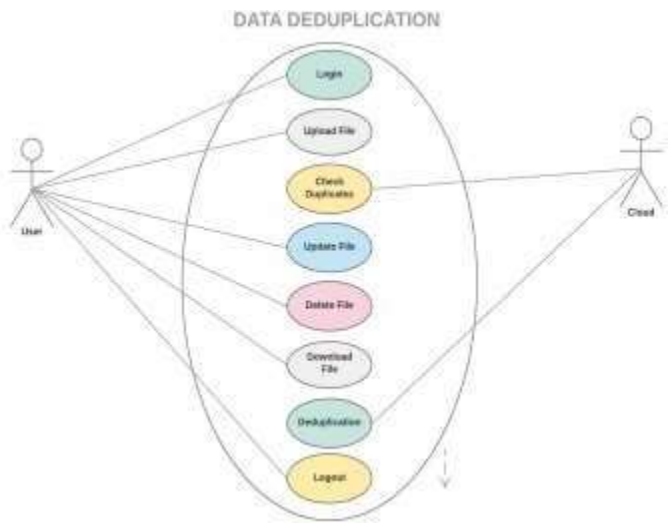
Figure 2 Hash Table Structure

Using the hashing function rolling hash is generated and 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.

4.3.1 UML DIAGRAM

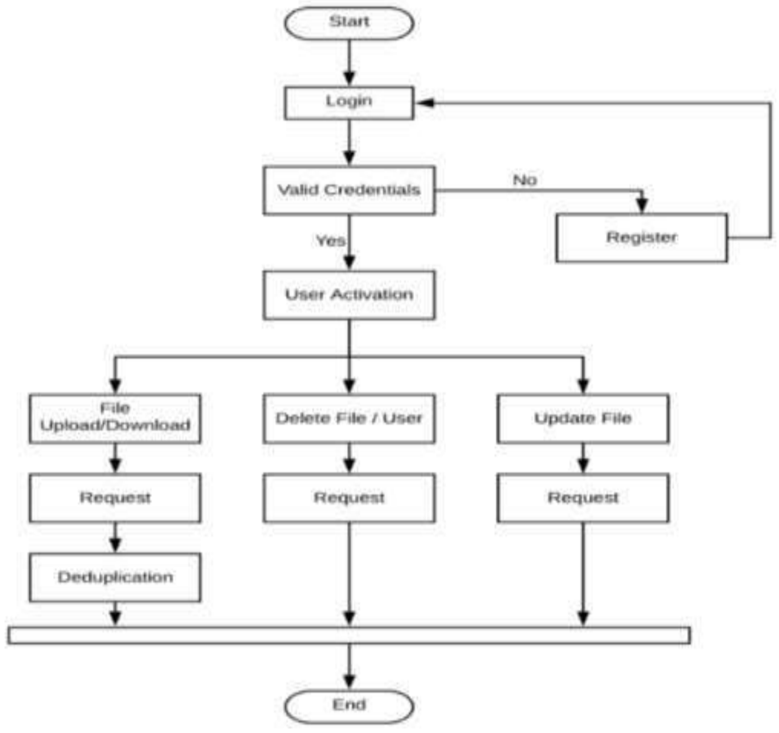
- Use Case Diagram

  Use case illustrates a unit of functionality provided by the system. The main purpose of the use- case diagram is to help development teams visualize the functional requirements of a system, including the relationship of "actors" to essential processes, as well as the relationships among different use cases. Use-case diagrams generally show groups of use cases, either all use cases for the complete system, or a breakout of a particular group of use cases with related functionality to Show a use case on a use-case diagram, you draw an oval in the middle of the diagram and put the name of the use case in the center of, or below, the oval. To draw an actor (indicating a system user) on a use-case diagram, you draw a stick person to the left or right of your diagram. Following diagram shows the relationships of the user or actors with the use cases which are shown in an oval shape.



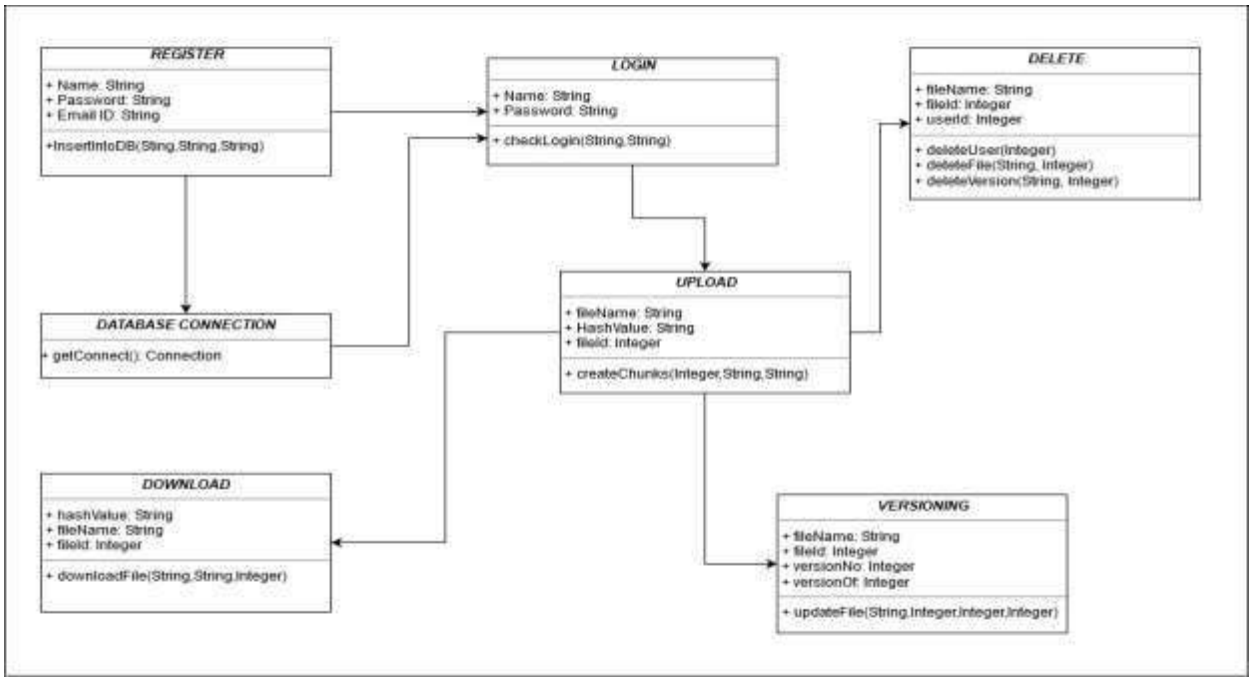
- Activity Diagram

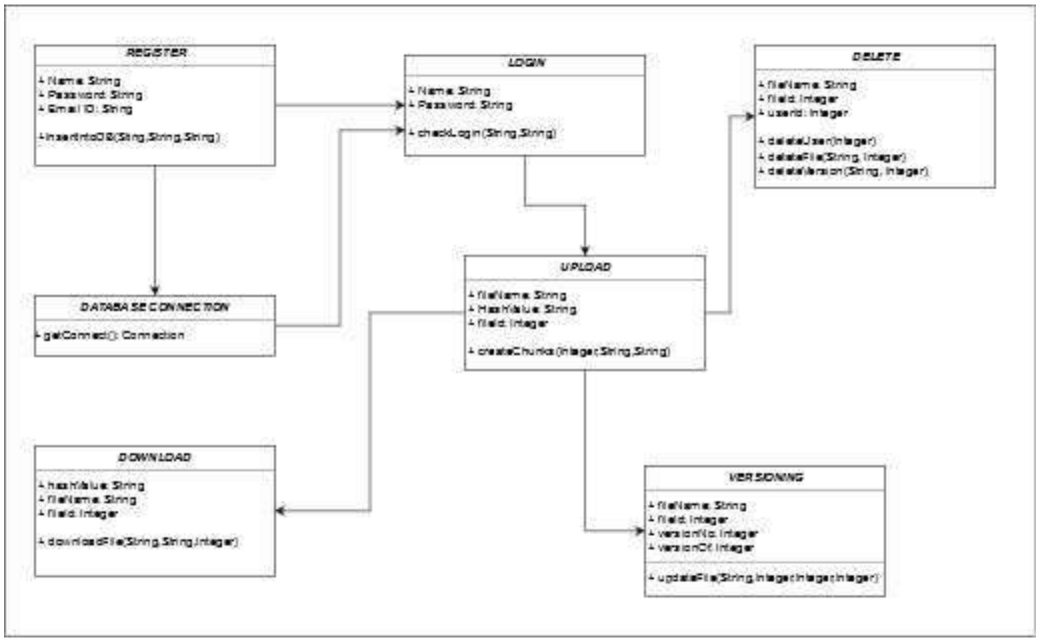
  Activity diagram is typically used for business process modeling, for modeling the logic captured by a single use case, or for visualizing the detailed logic of a business rule. Complicated process flows in the system are captured in the activity diagram. Similar to a state diagram, an activity diagram also consists of activities, actions, transitions, initial and final states, and guard conditions. However, difference is state diagrams are in context of simulation while activity gives detail view of business logic. Activity diagrams are "less technical" in appearance, compared to sequence diagrams, and business-minded people tend to understand them more quickly.



- Class Diagram

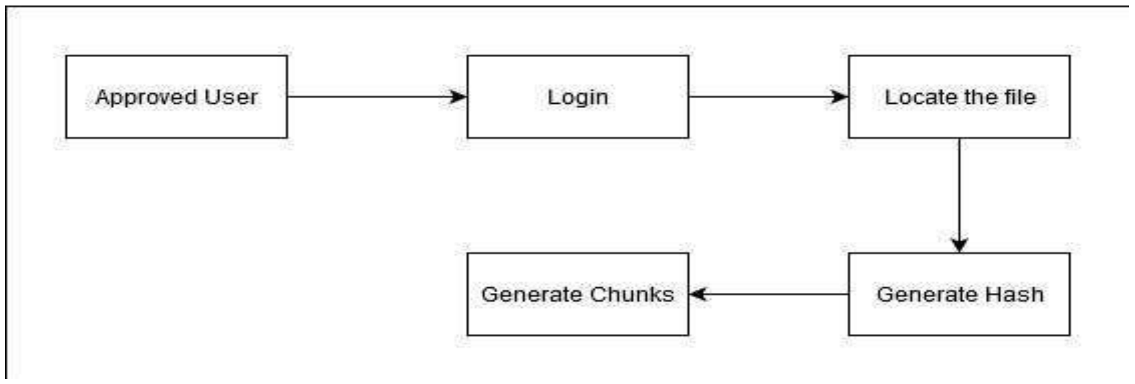
  In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.



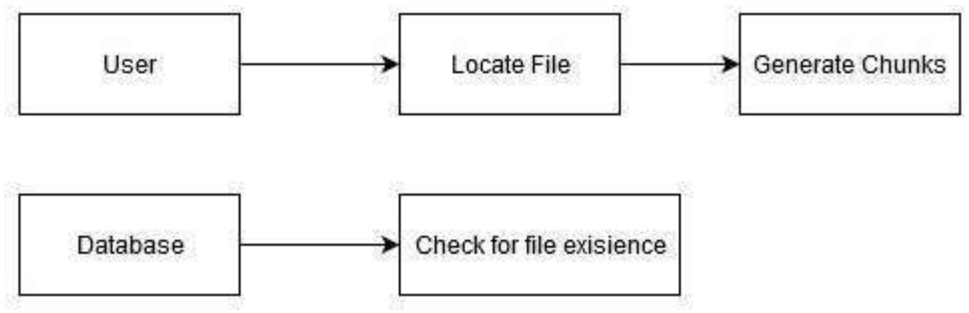


4.3.2 DATA FLOW DIAGRAM

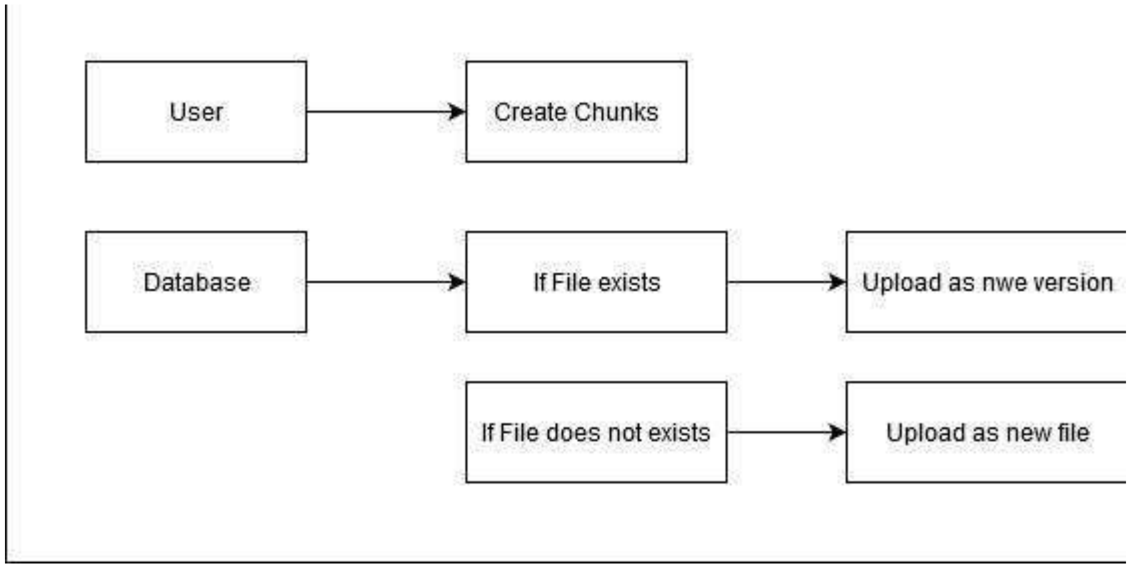
- Level 1



- Level 2



- Level 3

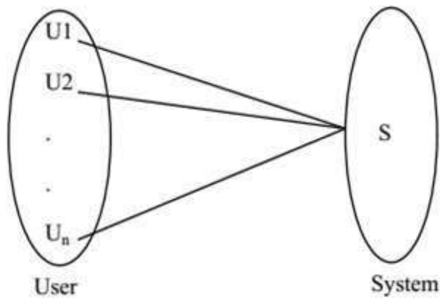


CHAPTER 05
METHODOLOGY

5. METHODOLOGY

5.1 MATHEMATICAL MODELING

A] Mapping Diagram:



Where,U1, U2…. Un =Users. S = System

B] Set Theory Input: File

Output: 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 with webpage.

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 € 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 files on system, Files are broken down into chunks such as, F = {FC1, FC2….Fcn}

Deduplication Checking:

H (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 multiple nodes. User gets result very fast according to 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 depends on following factors: time taken to upload file, time taken during file level and block level deduplication, delta calculating, storing deltas and non-duplicate data in encrypted format on different nodes.

5.2 OBJECTIVE FUNCTION

1. Upload File
    - This function uploads the file on the system.
2. Download File
    - This function downloads the file from the system.
3. Delete File
    - This function deletes the existing file on the system.
4. Delete User
    - This function deletes the existing user account on the system.
5. UpdateFile
    - This function updates the existing file as per the user requirement on the system.

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 an In-memory database which reduced the loss of data due to mechanical or software failure.

CHAPTER 06
IMPLEMENTATION

6.1 SYSTEM IMPLEMENTATION

For implementation we preferred Java language, IntelliJ IDEA CE framework and Windows O.S. Platform as it provides inbuilt server called IIS. Java provides inbuilt Message Digest class for SHA-256 hashing. The SHA (Secure Hash Algorithm) is one of the popular cryptographic hash functions. A cryptographic hash can be used to make a signature for a text or a data file. Data deduplication is referred to as a strategy offered to cloud storage providers (CSPs) to eliminate the duplicate data and keep only a single unique copy of it for storage space saving purpose.Data deduplication is one of the techniques which used to solve the repetition of data. The deduplication techniques are generally used in the cloud server for reducing the space of the server. To prevent the unauthorized use of data accessing and create duplicate data on cloud the encryption technique to encrypt the data before stored on cloud server. Cloud Storage usually contains business-critical data and processes; hence high security is the only solution to retain strong trust relationship between the cloud users and cloud service providers In this methodology we have to detect the duplicate copy of the file any type of file can be detect file .txt,.doc,.xls, ppt, .pdf. so we have to start with uploading the file when we upload the file we have to extract first 50 bytes from the file and last 50 bytes from the file. After extracting this 100 byte we have match this byte with existing byte. This comparing is done one by one byte i.e.one byte after another upon last byte. After completion of comparing this byte if this new file upload file is duplicate then we will discard the file. If this file is not duplicate, then we will upload this file. In this way the methodology is use.

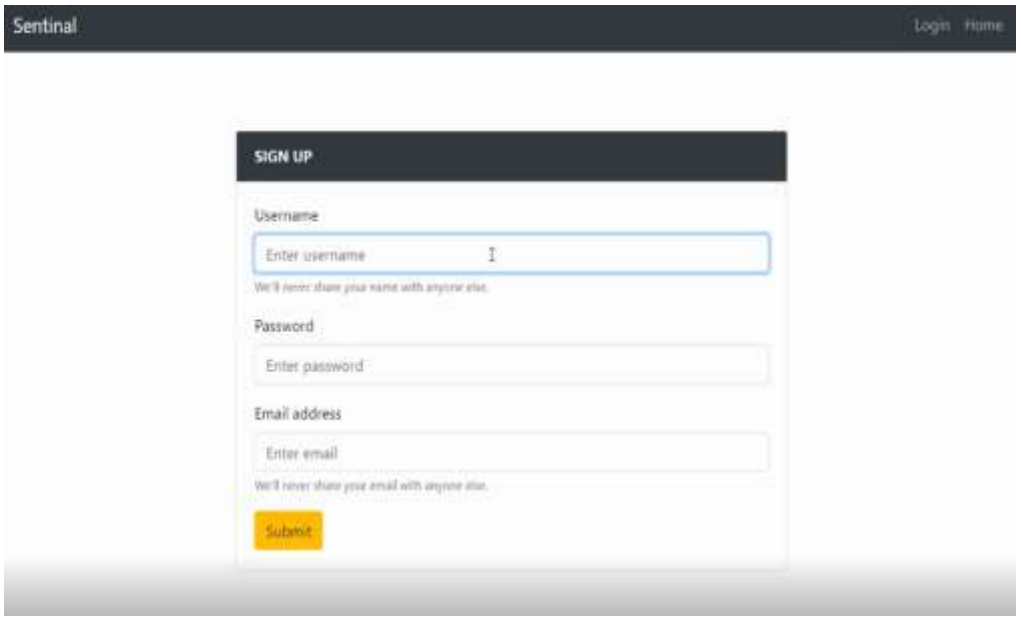
6.2 IMPLEMENTATION PARAMETERS

The data deduplication system interface has more than one parameter that controls its behavior. These implementation parameters are exposed to end users in the designed user interface, where they can also set values of the parameters at run time. These parameter settings are only of interest of the customers and can help customize an existing component to do something other than its original functionality.

6.3USER INTERFACE

User:

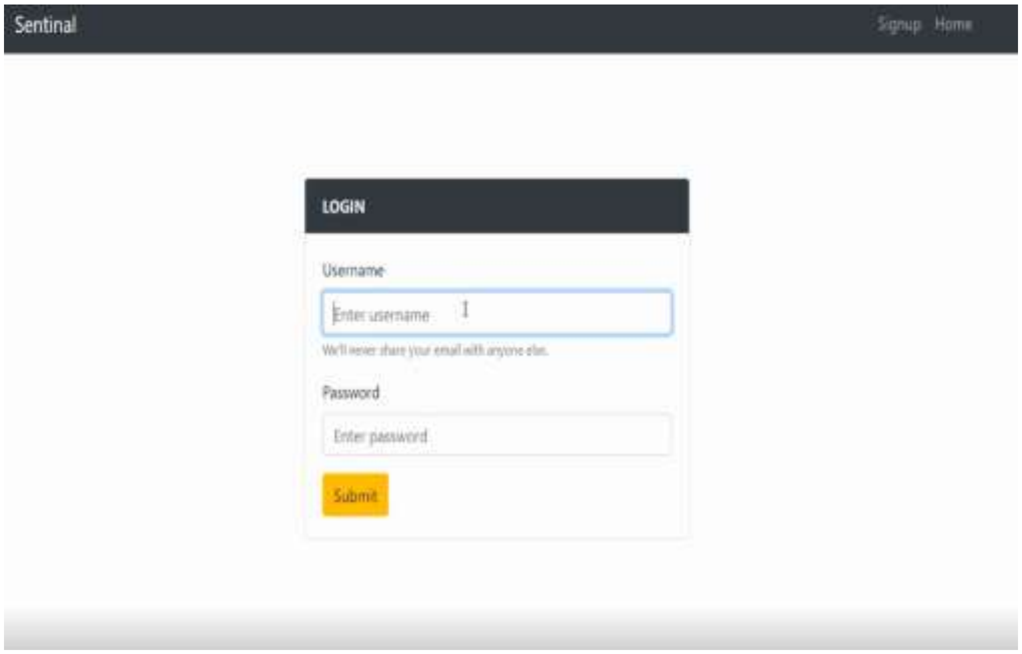
- Register



We'll never share your name with anyone else.

We'll never share your email with anyone else.

- Login



We'll never share your email with anyone else.

- Upload File
- Delete File
- Download File
- Chucks Count
- Update File
- Check Versions of the file
- Logout

6.4 DATA DESCRIPTION

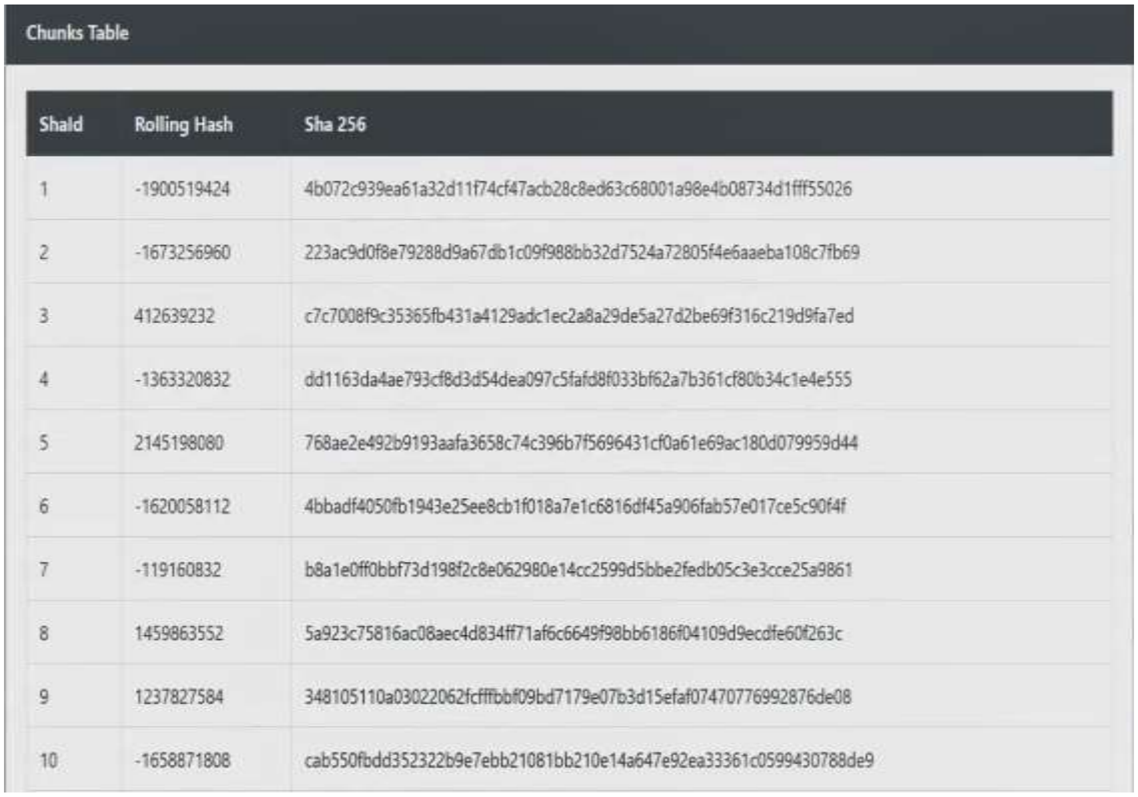
Currently the algorithm has the ability to deal with any type of text documents, to be specific it deals with txt, doc, odt and pdf. Not only does the algorithm deal with text documents but it can also split an image in to chucks and combine them back to recreate the same. The algorithm creates a window 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.

6.5 FUNCTIONAL IMPLEMENTATION

Propose a system that provides the ability create variable sided chunks of the given data and store them accordingly, The system also possesses the ability to store only those chunks which have been updated not those 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.

6.6 OUTPUT



Chunks Table

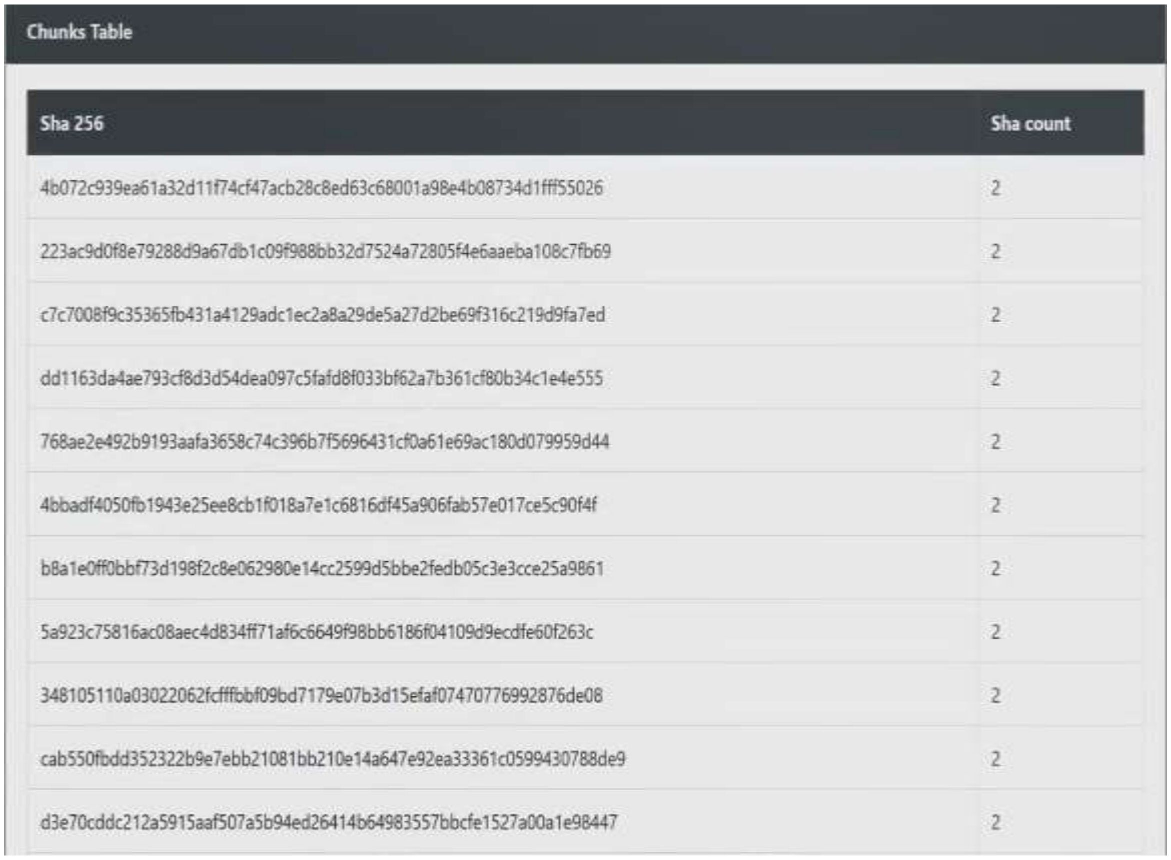
ShaId	Rolling Hash	Sha 256
1	-1900519424	4b072c939ea61a32d11f74cf47acb28c8ed63c68001a98e4b08734d1fff55026
2	-1673256960	223ac9d0f8e79288d9a67db1c09f988bb32d7524a72805f4e6aaeba108c7fb69
3	412639232	c7c7008f9c35365fb431a4129adc1ec2a8a29de5a27d2be69f316c219d9fa7ed
4	-1363320832	dd1163da4ae793cf8d3d54dea097c5fafd8f033bf62a7b361cf80b34c1e4e555
5	2145198080	768ae2e492b9193aafa3658c74c396b7f5696431cf0a61e69ac180d079959d44
6	-1620058112	4bbadf4050fb1943e25ee8cb1f018a7e1c6816df45a906fab57e017ce5c90f4f
7	-119160832	b8a1e0ff0bbf73d198f2c8e062980e14cc2599d5bbe2fedb05c3e3cce25a9861
8	1459863552	5a923c75816ac08aec4d834ff71af6c6649f98bb6186f04109d9ecdfe60f263c
9	1237827584	348105110a03022062fcfffbbf09bd7179e07b3d15efaf07470776992876de08
10	-1658871808	cab550fbdd352322b9e7ebb21081bb210e14a647e92ea33361c0599430788de9

Here, the table we have created displays rolling hash and its corresponding SHA hash of the each chunk created as the result of proposed algorithm. This algorithm is successfully able to create the chunks of variable length based on the content it comprises.

CHAPTER 07
RESULT

7 RESULT ANALYSIS

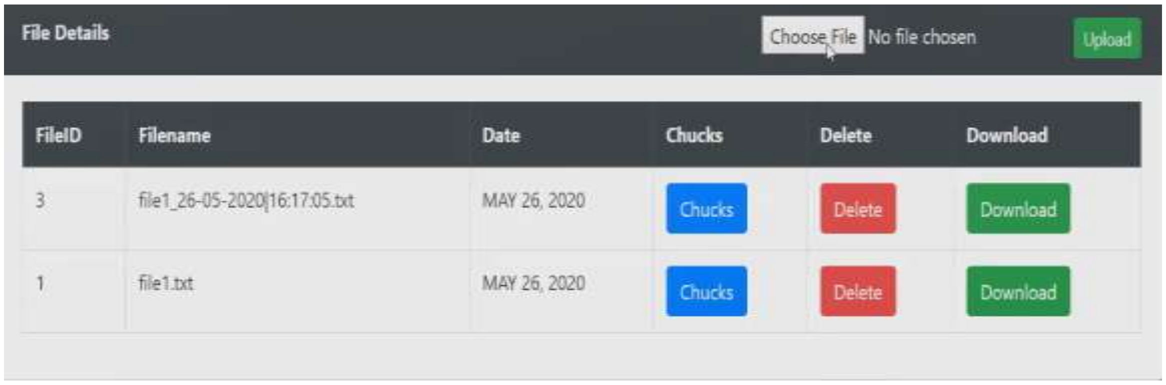
7.1 STORE ONLY SINGLE COPY OF DATA



Chunks Table

Sha 256	Sha count
4b072c939ea61a32d11f74cf47acb28c8ed63c68001a98e4b08734d1fff55026	2
223ac9d0f8e79288d9a67db1c09f988bb32d7524a72805f4e6aaeba108c7fb69	2
c7c7008f9c35365fb431a4129adc1ec2a8a29de5a27d2be69f316c219d9fa7ed	2
dd1163da4ae793cf8d3d54dea097c5fafd8f033bf62a7b361cf80b34c1e4e555	2
768ae2e492b9193aafa3658c74c396b7f5696431cf0a61e69ac180d079959d44	2
4bbadf4050fb1943e25ee8cb1f018a7e1c6816df45a906fab57e017ce5c90f4f	2
b8a1e0ff0bbf73d198f2c8e062980e14cc2599d5bbe2fedb05c3e3cce25a9861	2
5a923c75816ac08aec4d834ff71af6c6649f98bb6186f04109d9ecdfe60f263c	2
348105110a03022062fcfffbbf09bd7179e07b3d15efaf07470776992876de08	2
cab550fbdd352322b9e7ebb21081bb210e14a647e92ea33361c0599430788de9	2
d3e70cddc212a5915aaf507a5b94ed26414b64983557bbcfe1527a00a1e98447	2

7.2 MULTIPLE VERSIONS OF THE SAME FILE ARE MAINTAINED WITHOUT DUPLICATION



FileID	Filename	Date	Chucks	Delete	Download
3	file1_26-05-2020|16:17:05.txt	MAY 26, 2020	Chucks	Delete	Download
1	file1.txt	MAY 26, 2020	Chucks	Delete	Download

CHAPTER 08
CONCLUSION

8.1 CONCLUSION

In this proposed system, we have mainly surveyed the various deduplication techniques. Among them, it has been concluded that variable size data deduplication is well and good when compared to other strategies by comparing the hash of each and every chunk. Hence, this technique improves storage efficiency and thereby improve the performance by enabling storage resources to transfer and handle more data.

8.2 FUTURE SCOPE

In future, more research works could be focused on variable size chunking method to reduce processing time, and optimize of large scale data storage. And also to develop an efficient method to reduce fragmentation and obtain high write and read throughput.

REFERENCES

[1] J. Wen Xia, Member, IEEE, Hong Jiang, Fellow, IEEE, Dan Feng, Member, IEEE, Fred Douglis, Senior Member, IEEE, Philip Shilane, Yu Hua, Senior Member, IEEE, Min Fu, Yucheng Zhang, and Yukun Zhou, “A Comprehensive Study of the Past, Present, and Future of Data Deduplication”,in 2015

[2] E. Manogar, S. Abirami,” A Study on Data Deduplication Techniques for Optimized Storage”, 2014 Sixth International Conference on Advanced Computing(lCoAC)

[3] Changjian Zhang , Deyu QI, Zhe Cai, Wenhao huang,Xinyang wang, Wenlin li1 and Jing guo, “MII: A Novel Content Defined Chunking Algorithm for Finding Incremental Data in Data Synchronization”, 2019, published on July 1, 2019.

[4] W. Xia, D. Feng, H. Jiang, Y. Zhang, V. Chang, and X. Zou,”Accelerating content-defined-chunking based data deduplication by exploiting parallelism,’’ Future Gener. Comput. Syst., vol. 98, pp. 406–418, Sep. 2019.