Optimizing Memory management and Data Recovery Efficiency by Categorizing Backup Metadata

MSc Research Project
Research And Computing

Niketan Bothe
Student ID: x20180837

School of Computing
National College of Ireland

Supervisor: Prof. Aqeel Kazmi

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Niketan Bothe
Student ID:	x20180837
Programme:	Research And Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Prof. Aqeel Kazmi
Submission Due Date:	15/12/2022
Project Title:	Optimizing Memory management and Data Recovery Efficiency by Categorizing Backup Metadata
Word Count:	7619
Page Count:	19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	NiketanB
Date:	29th January 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Optimizing Memory management and Data Recovery Efficiency by Categorizing Backup Metadata

Niketan Bothe
x20180837

Abstract

Cloud computing is actually needed in the world of computing due to the enormous amount of data. In recent years, cloud storage has emerged as one of the finest solutions for keeping big amounts of data due to lower maintenance costs and ease of access from anywhere over the internet. Due to the increased demand for cloud storage, a variety of services are now offered by third-party organisations. However, while the cloud might be less expensive, the cost of significant services or storage can rise. After completing the study, it was discovered that the major reason for wasting the memory is storing the similar data on cloud. The study focuses on data deduplication algorithms such as CAP, HAR, and others. The CAP method is regarded as one of the excellent technique for deduplication check, although it has significant faults, which are highlighted in this work. This study focuses not only on data deduplication but also on enhancing the accuracy of deduplication checks and increasing backup and restore performance. The main focus of this research is how I can provide the security as well as reduce the time for taking backup and restoring the deduplicate data. This research will highlight several essential analyses relevant to parallel processing that will be covered in this study, along with the scope of the project.

1 Introduction

Cloud demand has increased significantly in recent years. Every business owner wants their data on the cloud due to security purposes, storage server administration, labour costs, etc, because it is challenging for small businesses to run a personal server, the cloud is typically the best option for everyone. In the domain of computing, it is feasible to save money if the user is aware of the architecture or if the cost of storage is estimated by the amount of data stored. In recent years, it has been seen in the cloud domain that unnecessary data has been stored, for example, data that is existing on cloud and again uploading the same data, resulting in unnecessary cloud utilization and a rise in the cost of storing data on cloud. Considering this problem statement, several researchers have been working on it for a few years. To avoid the deduplication Some researcher work on file level deduplication Mahesh et al. (2020) Jiang et al. (2017), and few worked on block level deduplication Suresh and Bharathi (2019) In both scenarios, there are certain advantages, such as doing file level deduplication, which can speed up the method, but the possibility of storing similar data is quite high.The ability of storing comparable data at the block level may be limited, but comparing each block with all of the blocks existing on the cloud will be time consuming.

Consider the data that is already existing on the cloud and yet try to backup and restore similar data that can raise the cost of restoration. Cost is based on the size of the file thus it is not worth restoring all the data without knowing what sort of data is going to be restored. So considering most research survey of data deduplicationYang et al. (2021) This is the most feasible approach for all circumstances, however there are certain difficulties that may be addressed in this research to improve not only the algorithm process but also the accuracy of data deduplication. In the case of boosting the algorithm the most feasible method is to upload the blocks using a parallel process, as per the survey of some research the issue of parallel processing is that files are not compared with each other while parallel processing, So in this experiment while doing parallel processing if file get compared the chances of storing the same file get less and file get stored more accurately. The research is dependent on metadata as compared to previous study of the flow of uploading process or applying the metadata lookup on existing cloud systems that can avoid the similar data backup or restoration process due to presence of unique data.

In this experiment when user upload the data or files, it will convert into chunk files and the chunk file get encrypted for encryption there is AES algorithm which is light weight and most powerful Ali, Ahmad and Rafi (2020) and assigned with hash value in the metadata lookup the value will get stored if the hash value is already present into lookup that chunk will get ignored and will not be stored on the cloud. Comparing hash value instead of comparing local blocks with all the blocks which are stored on the cloud.

By considering previous research and few factors the following point identifies the project stand alone and can implement exceptional implementation.

- Encryption while storing
- Deduplication with own data or shared storage
- Considering the previous study and experiment this implimentation will take less time as per the situation

In this research, there are a few algorithms that will compare and verify which will be the best alternative for data duplication check, in terms of speed and accuracy. Related work is being done with the most current articles, which provide a basic notion of how the data deduplication method works. The general architecture for implementation is explained in the third part, which mostly focuses on algorithm architecture. Furthermore, the subsequent implementation of algorithm functioning is given, which might provide an useful description of the algorithm's operation.

Getting familiar with that algorithm there is implementation which is implemented and all the result shown in the section of Implementation. It also includes output of the live project. after successfully implementation of project there are some test cases which is also called as experiment is implemented in the Evolution section there are few scenario which is perform as a experiment in the project.

In the terms of future scope there are some points which can be overcome and can be make model more accurate in proper way.

2 Literature Review

The main focus of this research is on enhancing data deduplication, improving accuracy, and accurately restoring data, so the research is divided into two parts in this implementation. According to the scenario, a few papers worked on improving the algorithm of data deduplication, but they are not accurate in deduplication. A few authors worked on improving the accuracy of data deduplication, but their algorithms were too time consuming. In recent survey few algorithm are best as per the scenario which will be discuss below. This study is not only depend on boosting the speed but also improving the accuracy of current algorithm.

2.1 Data Deduplication:-

Deduplication, commonly known as Dedup as abbreviation, seems to be a function that also can assist in eliminating a lot of duplicate data or cost of storage. While Daedup is active, it optimizes available space on a drive simply analyzing the information on the drive and searching for repeated sections. In terms of the research there are some papers which work on the file level data deduplication and some implementation is done with block level deduplication.

Yang et al. (2021)Considering both the condition research by (Yang.;2021) that shows the algorithm for data deduplication which can boost the process of uploading files with higher accuracy. Starting with investigating the memory characteristics of the recovery information. Surprisingly, having 10,000,000 data, the document metadata barely takes up roughly 0.34 GB of memory. Since becoming comfortable with such an idea, authors developed Classified metadata based restoration technique, which separates backups metadata between blocks and file metadata. Because document metadata uses very less memory, CMR preserves all file metadata in storage space, however chunks of information are aggressively prefetched.

In the survey of Shin et al. (2017) shows that cloud is adopting the data deduplication algorithm rapidly due to storage cost and efficiency of storage processing algorithm. To address particular vulnerabilities, safe dedup approaches for data in the cloud have already been developed, resulting in a wide range of techniques and transfer. In this article the author continues study on safe minimization for cloud systems in light of the most common vulnerabilities threats in cloud computing. Researchers investigate security concerns and exploit possibilities through both internal as well as external attackers based mostly on categorisation of the compression systems. Author survey and proposed on State-of-the-art secure deduplication solutions fig 1. But considering the survey of state-of-the-art which is mainly focused on security that can affect the processing time of deduplication algorithms. Discussing security concerns that can be resolved by access specifiers which cannot reduce the processing time.

Similarly Ali, Ilyas Ahmad and Rafi (2020) Propose the block level deduplication in which author discuss how block level deduplication helps to boost deduplication algorithm in the practical way considering the content of the research the block level comparison can also be boost in more proper manner by comparing hash value of data which will boost the data deduplication process. To avoid data duplication, a comparison between store files and new files can be performed. As per the survey the metadata can be fit in the scenario by considering the situation, as per the survey metadata basically divide the file size and compare each file with stored file if the same file is present then file is not going to store or else file can be stored. The Venkatesh (2019) shows the proposed system discloses a technique for transforming a vms out of a first VHD or VHDX format to a specific hyper - v type using dedup information. Author concludes that this technique could comprise creating a clone of such a Vms via replicating decompression information

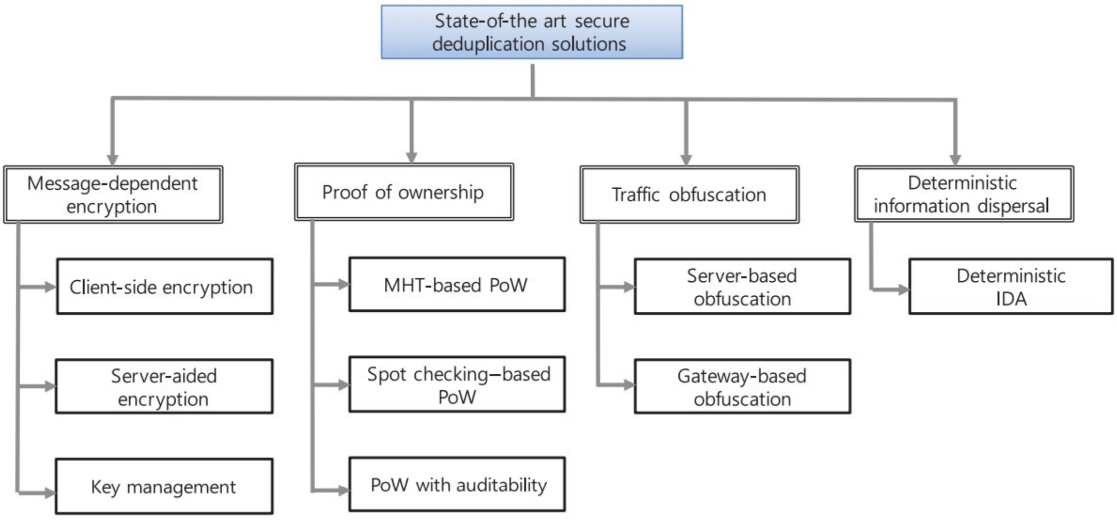

Figure 1: State-of-the art
Ali, Ilyas Ahmad and Rafi (2020)

with one or many virtual machine-related records. Additionally, the approach could comprise activating and duplicated optimization metadata copies of both the VM.

Another Wu et al. (2017) point out some issues the first major issue were In certain main memory applications, the environment of duplicate content uploads might not even exist, hence internal caching frequently fails to produce a decent compression ratio. Another point the author discussed is that regarding the pre-dedup permits duplicates to just be issued onto discs, it lacks the benefits of I/O decompression necessitates a large peak memory space. The author of this research introduces HP Deduplication, a Dynamic Differentiated with priority DataDedup method that combines a linear and a post-processing approach for accurate preservation to handle storage devices maintained by apps that run in cross vm or container. Kaur et al. (2018) The research explores compression strategies that focus on text-based and multimedia data, as well as their accompanying categories, because these strategies face various problems for repetitive data identification. This author's research may be used to find compression algorithms for document, picture and video files. In this study the algorithm refers to the block level comparison so in this case the block is compared with whole data fig 3,

which is stored on the cloud so comparing the single block with whole data can affect the processing algorithm of data deDuplication. In this case it's not feasible for large amounts of data. On the other side, the author keep the fix block size in the study by Shin et al. (2018) also keep the block size similar so that if the same block is present then it will compare each and every data of that block with the another block which is stored in a cloud storage. Let's take a example of document which is 10 MB the block size is fixed which is 100KB so in this case data deDuplication is easy but when there is large amount of data then In this case the processing time is quite high, it will affect to the data processing algorithm as well as algorithm can be time consuming. Considering the same situation Ali, Ahmad and Rafi (2020) proposed the Dynamic size block level deDuplication in which the block of size will be dependent on the size of file or data. This can avoid the unnecessary processing time for the deDuplication algorithm. But

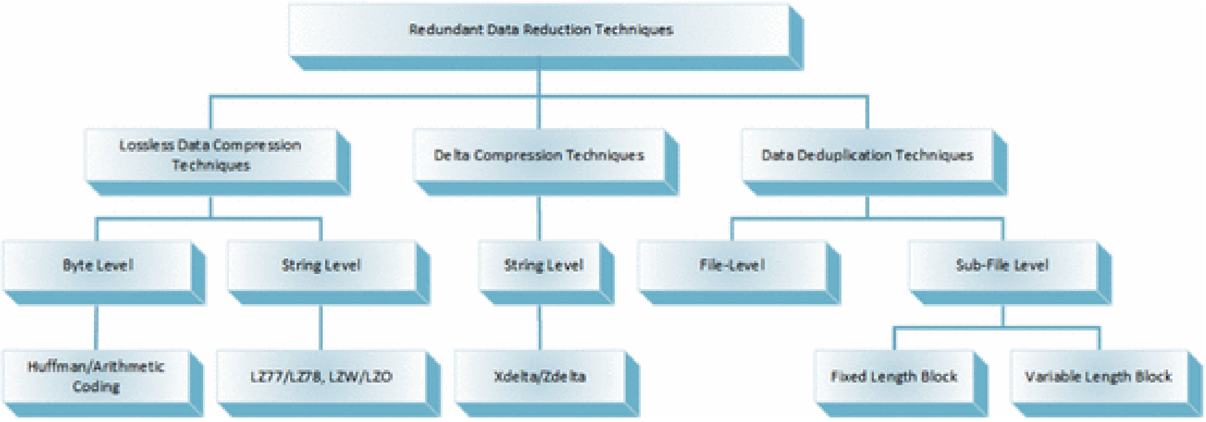
Redundant Data Reduction Techniques
Lossless Data Compression Techniques
Delta Compression Techniques
Data Deduplication Techniques
Byte Level
String Level
String Level
File-Level
Sub-File Level
Huffman/Arithmetic Coding
LZ77/LZ78, LZW/LZO
Xdelta/Zdelta
Fixed Length Block
Variable Length Block

Figure 2: Redundant Data Reduction Techniques

considering the Dynamic size block level deduplication the ratio of deduplication has not been mentioned which is the important part of data deduplication. So when the file size increases the Block size also increases and hence when the large block is getting compared with cloud storage block there may be the chances the same data get stored or ignored with the dedup algorithm. So considering all the data deduplication methods, let's move forward to the data backup and data restore. The Backup or restoring the data which is lost or stolen. In the previous research the encryption techniques are limited its either on file level or block level in most of the cases the enryption and decryption is done on the file level and the encryption technique is quite simple. There are number of tools in the market which can easily inject in the software and decrypt the data but most recent study Priya et al. (2022) explain the powerful encryption technique which can secure the file so this can be also helps to improve the system security.

2.2 Data Backup Restore:-

Backup of data since Data dedup has the capacity to save hard drive size, data deduplication is commonly applied in the cloud storage. A problem with dedup is that continuous data pieces in a segment might well be spread over many containers. Because of this, a restoration procedure may refer with several storage over multiple parts, that could lower restoration speed. In the survey Tamimi et al. (2019) discussed the data backup technique This paper mainly focused on cost control, data replication and security issues. The another backup technique is discussed in this study that is cold backup Abualkishik et al. (2020) also proposed this technique, When a service failure is identified, the cold backup restoration procedure begins.

When bringing the systems into an useful condition, the hardware components of the machine require a set of programs connected with only a set of data to be developed or retrieved. Whenever components are removed from storage or relocated into development and testing systems, there could be delays of many hours or even days. Cold backup servers are quite expensive. Several major disasters can result in significant harm to storage devices, demanding the usage of specialized and expert data retrieval procedures. Depending on the amount of the damages, restoration may involve the deployment of specialized software and hardware retrieval solutions also including switch data retrieval from damaged or broken discs.

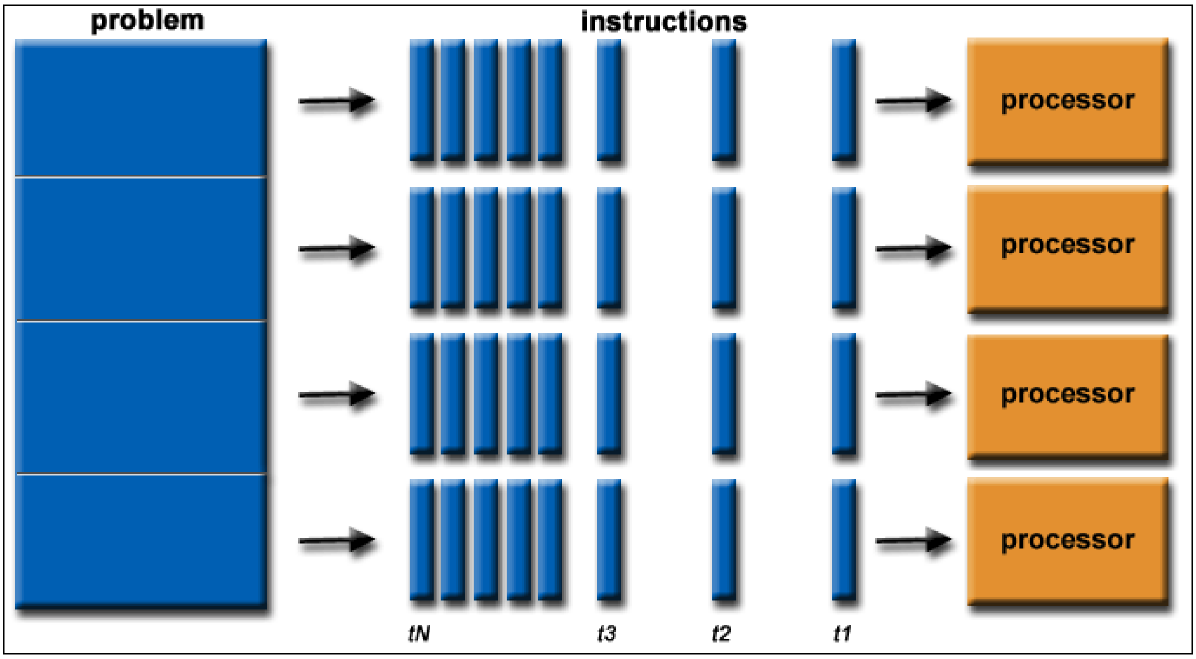

Figure 3: Parallel processing
Ji et al. (2019)

One such article Chang et al. (2021) shows the research setting and testing elements in the actual experiment also creates an attempt for retrieving. As per the experiment the backup process can be done more quickly by implementing the parallel processing but the accuracy of deduplication can get affected during the backup or restoration process so considering this implementation the data comparison is done between stored file and new file. So when parallel processing is done the data is not getting compared while parallel processing so there may be the chances to get the same file or same data restored or get uploaded on the server. In fig 3 the blocks are going through the algorithm the blocks which are in process are not getting compared with each other so it's not able to check whether the same data is going through the algorithm or not. So if data is getting compared while parallel prepossessing that can be helpful to get better deduplication accuracy.

After carefully evaluating all algorithms for data deduplication check or analysis, the CAP will be a suitable alternative to implement this architecture because the speed of processing or the accuracy of deduplication check can be increased by focusing on the important point. which are parallel processing comparisons between blocks or files, and the size of the file or chunk(block) depends on the data being sent to the server to store In this case, comparing hash values rather than whole blocks can help to improve the algorithm.

3 Methodology

Considering the previous studies most compatible algorithms are HAR, CAP, CMR and RARE considering the implementation the project is divided into two few different parts which includes Data deduplication, Metadata categorization, Metadata Prefetch, MetaData Lookup, Parallel Restoration. In the Data Deduplication various paper worked on the file level data deduplication similarly few researchers work on block level deduplication, So in this implementation experiment focused on both strategy file level and block level deduplication which can generate quite good accuracy of the data deduplication

3.1 System Architecture:-

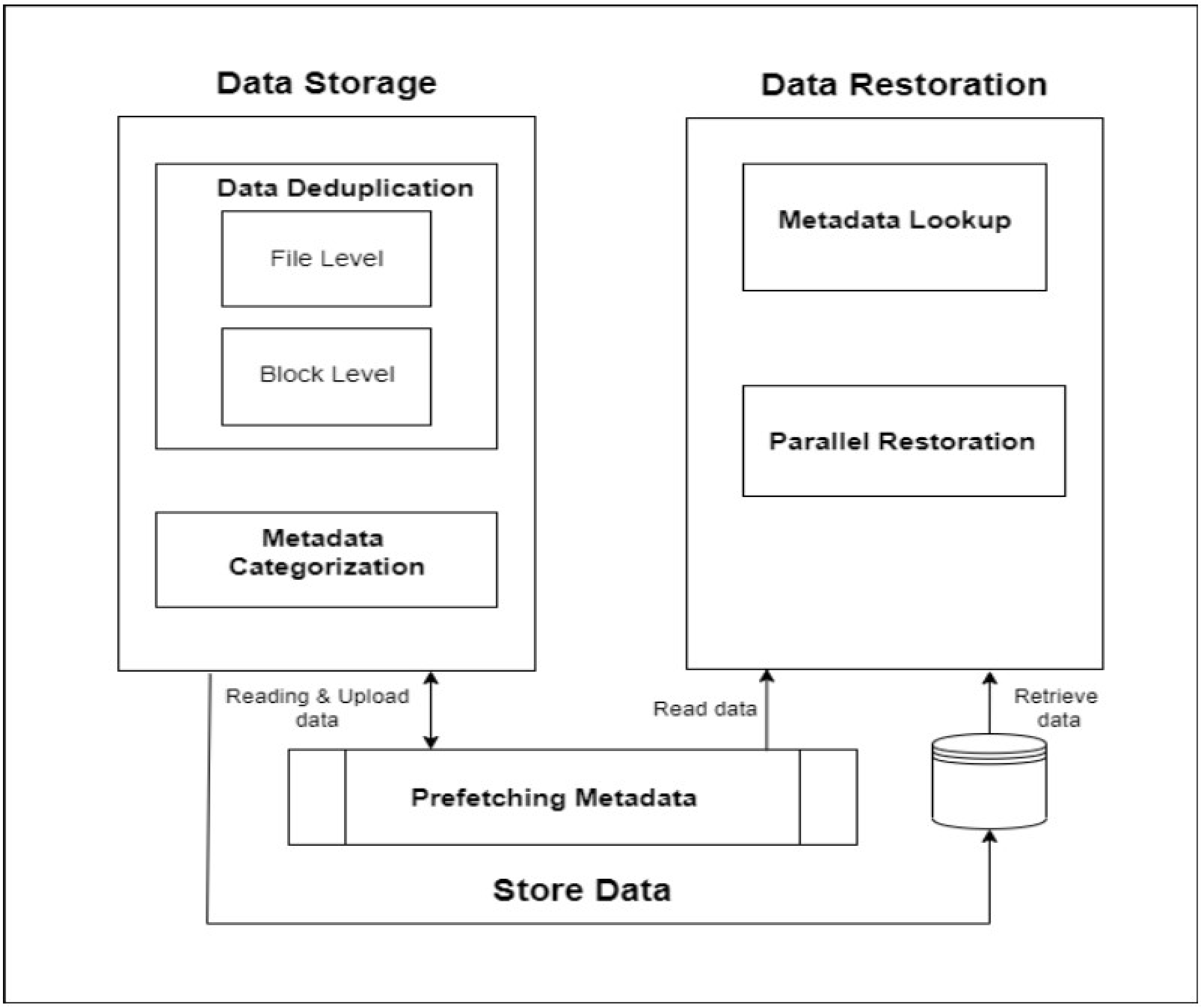

Figure 4: System Architecture

Above architecture is divided into two different parts Data Storage and Data Restoration. Data restoration contain the technique of data deduplication algorithum, parallel processing Metadata categorization, Pre-fetching Meta Data.

In the process of Data Restoration, backup is stored on cloud so when user request the data for restoration algorithm only recover or restore the data which is unique or similar type of data will not get restored.

This method is entirely dependent on a constant that is split into two cases. Let's take a deeper look at both instances based on the circumstance.

1. Block level duplication is determined by the file and block sizes. Only file level deduplication is performed if the file size is less than the block size. Let's take an example: if the constant size is set to 12KB and User A uploads a 10KB doc file and User B likewise

uploads a 10KB file that is already in the cloud, the algorithm will not partition the file into blocks and will only do file-level deduplication.

2. Block-level deduplication should always be used when dealing with duplication since the algorithm produces a greater duplication ratio when files are separated into blocks because backup data may only contain partial files that are identical. The block size or constant are important limitations. Tiny block sizes will be effective for small files if we define them just in terms of small files, but processing costs and overhead would increase, and latency would undoubtedly be impacted in this scenario for bigger files. Block size must be determined according to the data that will be uploaded to the system since file size delay and block size are transfer.

3.2 Data Storage

This module contain the Data deduplication and metadata categorization. Data Deduplication as per the related work few research worked on file level deduplication few experiment is done in block level so in this experiment algorithm is focused on both file level and block level data deduplication. In the experiment first file level module is get compared with file only which may very small in size. comparing small files or converting them in a blocks can be slowdown the process of deduplication after certain size files will not get convert in block.

3.2.1 Data deduplication at the file level

Suppose if four persons (AB.txt) are uploading documents to the server, while the server simply keeps single copy for every file at such a time, which all users would access to. Whenever person 3 upload a document (AC.txt), the host keeps it confidential and only person 3 has access to this now; all other users have access to document only (AB.txt). This is exactly what took place during file level dedup.

3.2.2 Data deduplication at the block level

Block size is also known as chunk, Block file data deduplication will take place if the size of data is small. The chunk size is fixed and cannot be altered either by user in later. Imagine the following scenario of multiple users. When user 1 upload a file, it is separated into several unique pieces, ranging from the original file (AB.txt) to blocks (a,b,c,d). and user 2 upload the file (BC.txt) which get convert in (b,c,e,f). so in this case block (e,f) will get stored only and block(b,c) will get link with user two which is already uploaded by user 1.

3.2.3 Metadata Prefetch

A couple of memory or SSD actions will be required to retrieve metadata. At this point, the required data would be put into Storage. After backup completion, all information is updated while also being analyzed for backup. CAP approach, which is based on performance.

3.2.4 Metadata Categorization

In the metadata structure It is necessary to merge data. but without variable definitions, data frame, or dimensions names The statistics numbers are ambiguous. Whenever a data get uploaded on a cloud, an algorithm creates a user id and backup id each routeing that could be used as a references for data retrieval and backup systems, as shown in Fig.5.

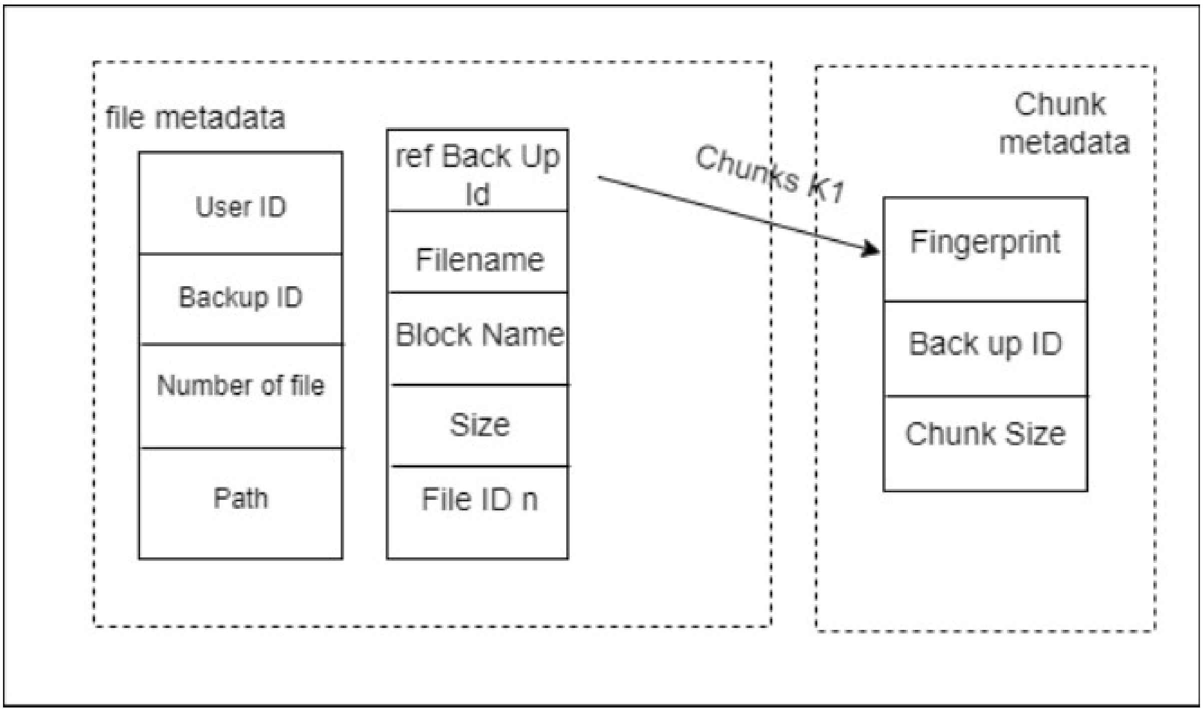

Figure 5: Metadata Structure

3.2.5 Metadata Lookup

The database lookup in just this section separates the meta records among all metadata dependent on the restoring requests user id and such quantity of backups. This table contain all the user id with hash value linking according to their data which is being uploaded or may the data which in backup/restore process.

3.3 Data Restoration

Data restoration was one of the biggest challenge because considering about the backup and restore, either user can choose what user wants to take as a backup or user can select all backup file option in that case there may be the chances to similar data get backup so in this scenario that can be overcome without storing the similar data on the cloud.

In the case of boosting the speed and accuracy of duplication can be improve while taking the backup of files using Parallel restoration With technique of metadata Lookup.

This solution might necessitate the use of 2 distinct plugins. Cases of data processing and data restoring are shown below.

4 Design Specification

A primary objective of the project is always to minimise storage volume and transmission costs while also developing a data-dedup system. The system is based on a user-server model. at the end of AWS-cloud, data-dedup will get verified.

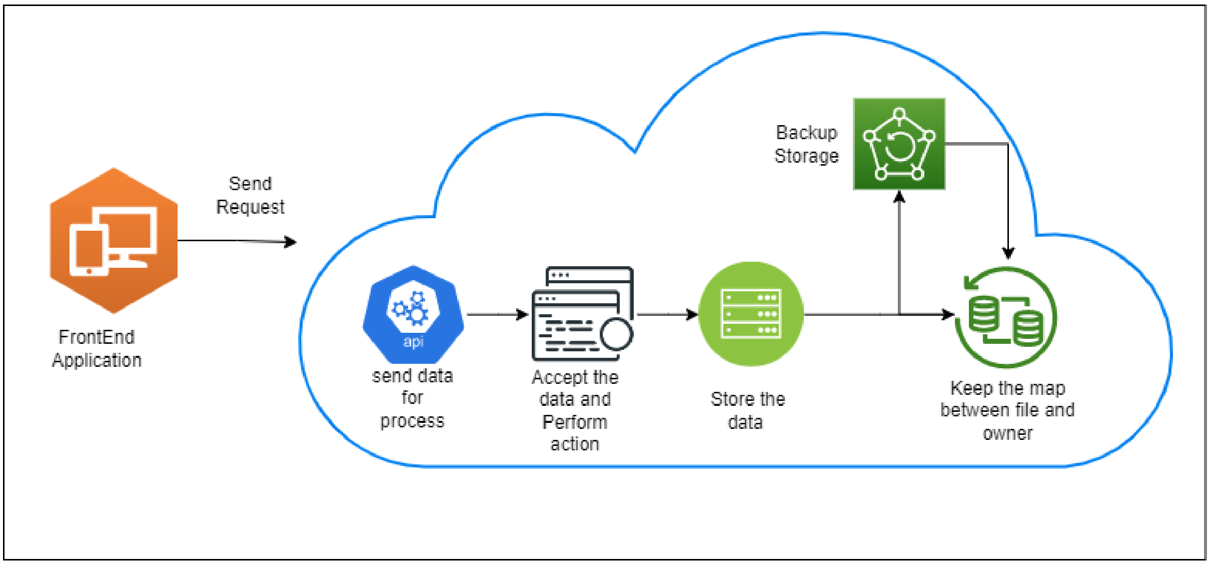

Figure 6: Implementation Architecture

as shown in above figure 6 This algorithm is going to work on cloud side for the implementation when user upload any file with the software which is present on the local machine or user side software file will get convert in small block which is also called as chunk. For each chunk the unique ID will get generate that is hash value. whenever the same chunk is get locate at that time algorithm will ignore the similar hash value and will store only unique hash value. In the terms of multiple user if 2 user uploading same file on cloud in that case algorithm will store only unique file instead of keeping different files and will map the hash value as per the user request.

for better understand the flow of algorithm is given below for

4.1 Data-DeDup Flow

data duplication check module is divided in two different parts Data duplication at file level and block level. in file level deduplication if the ownership and data is not present on the cloud server algorithm will store the data and map data or file as per ownership.

When it comes to block level deduplication, It occurs when there is a negative result from file level duplication; as previously stated, when a user uploads a large file, the file is distributed based on its size; if a specific block is not present on the server, it takes time to compare each and every block with the block present on the cloud; therefore, metadata will compare the hash value of each block with the block which is stored on the cloud.

- LM: Load metadata M in cache
- ULM: Filter user backup metadata cache.
- for each file in fi in f
- H(fi): Calculate file hash

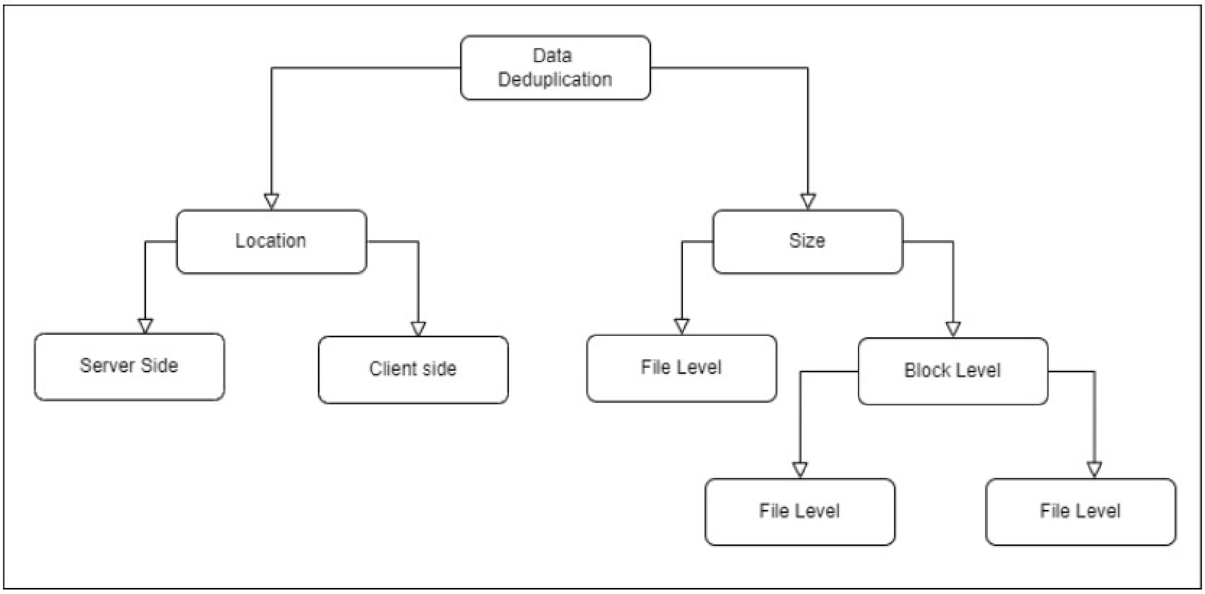

Figure 7: Metadata Structure

.

```
• if(H(fi) Present in in user UI old backup metadata ULM)
    1. Update Metadata MI
• Else
    1. B: Create File blocks
    2. for each block BI in B
        (a) H(bi): calculate file hash
        (b) if(H(BI) present in user UI old backup metadata ULM)
            i. update metadata MI
        (c) Else If(H(FI) present in other backup metadata(LM-ULM)
            i. update metadata MI
        (d) Else If(H(BI) present in other backup metadata(LM-ULM)
            i. update metadata MI
        (e) Else
            i. Upload file blocks and save metadata
            ii. Update metadata in cache ULM
```

4.2 Data-Restoration Flow

In terms of data restoration, when a user uploads a file, the dedup algorithm only keeps unique files or blocks, therefore backups on the server will only take backups of unique files. if user lost any file at the local machine User can recover that file from server

1. ULM: load user backup metadata cache

2. Filter Backup metadata files using UID and Backupid: I

3. Create parallel restoration process: P
4. Distribute Metadata files among parallel processes Pi
5. Each Process Pi in P
6. Read file id and its respective block information with offset
7. Read blocks from memory and recreate file
8. Save file to the defined file location

5 Implementation

In this study, the module is separated into multiple sections, running from the front end implementation, through which users may provide input, to the back end server, where all algorithms and file storage will take place. This part will look closely at algorithm implementation as well as front end and back end implementation on cloud storage. In this module there is detailed explain about the flow of application and algorithm.

5.1 Application Frontend

The application is being developed in Java and run on JDK 11. User architecture is used for system implementation. In the system design, users must register before uploading data. Once registered, users may use the upload, download, backup, and restore features. When user get into upload section they can select single file or multiple file for upload on the cloud. When it comes to backup, the user has the option of backing up a single file or numerous files. when a user creates a backup of a file that may be restored later using the restore files option, even if the file is removed from the server. All request are from user application are sent to cloud using the API which is the part of backend.

5.2 Application Backend

Apache Tomcat-7 and MySQL 8.1 are used to set up a server-side environment. The HTTP protocol is used by this application to connect with the cloud server. Eclipse JEE Neon is used for server-side application development. The server-side software is hosted on a t2.2xlarge Amazon EC2 machine. To run the backend application fluently the project setup with higher configuration of cloud which is given below in the table 1.

Component	Requirement
OS	Windows
vCPU	8
RAM	32GB
Storage	100GB

Table 1: System Requirement

5.3 Overall Application Flow

This Module contains the general flow for the architecture, including the detailed workings of each module, and will describe the file upload, download, backup, and restoration processes.

5.3.1 Upload File :-

In the client or user application, the user login into the system and uploads a file to the cloud via the application. The file name and file owner were get store in the database, which is implemented in MYSQL. On the cloud, the algorithm is implemented using API and developed in Java to generate files into hash tag values that are unique for all files that will also be upload in the future. After generation of tag that file is get convert into blocks(number of parts of the file), Block size is dependent on the size of file, in this block size is kept 2KB(which can increase as per choice). Each and every block is encrypted once again using hash code, making it more secure and, in comparison, a really helpful feature that can speed up the process.

In the terms of Duplication if there is minor changes in the file and any user again uploading the file this time hash tag will get change because there is small amount of change in file. when file proceed for blocks and each block have the unique value algorithm will not store the similar hash value as mentioned earlier. In this situation, the algorithm will not compare one block to another, only the hash values. As a result, comparing hash values takes less time than comparing blocks. This is the exact working of Upload process.

5.3.2 Download File :-

In the Download file which can be done by user application will maintain the session for user identification. When user send the request for download the file the request contain name of file user id which is handled in HTTP protocol. Using API it will get check the availability of file if it is available and which blocks are associated with the owner of file that can be fetch from map table algorithm will decrypt the blocks using AES decryption algorithm which is the most power full algorithm for now. After decryption each block which is associated with the request file will get merge and will send the acknowledgment to user and it will get download in user local machine using the software.

5.3.3 Delete File :-

If there is no backup of the file, the delete option accessible in the download file module might destroy it permanently. Tn the term of flow when user request for the delete operation API will send the request to the backend server. algorithm of backend server will check the blocks which is associated with that file if the same blocks is associated with another file owner than algorithm will not delete the block but it will delete the mapping of the user who requested for the delete file operation. In exceptional situations, if the user who performs the delete action has partially erased files, just the block that is not linked with any other user will be deleted from the server. So that it has no impact on other user files.

5.3.4 Backup And Restore File :-

There were challenges in the backup and restoration process where any user asks for backup, system has to verify whether backup is already accessible or not if any similar file of backup is taken then system would look into blocks If there is a unique block, the system will not preserve the file or its mapping. The functionality of this module is separated into two sections, which are listed below.

- Back-Up:- In this scenario, when a user wants to take a backup of a file to keep it safe, the user will request the backup, and the cloud server thread will get the requested user's information and use thread to check whether the file is present or not. If the file is already present with all the blocks, the backup will be mapped to the user who already requested the backup instead of making new backup file. If the file is unique at first time it is requested for backup, the mapping of the block is saved; yet, there is a risk that a few blocks already exist; thus, instead of keeping comparable block mapping, the system keeps only unique file mapping and get stored in the backup storage or cloud, which may be restored in the future.
- Restore:- When a user loses a file from the computer and requests that it be restored from a backup, the system will check the file mapping and assign that mapping of block to the user, as well as apply the owner rights to the mapped file.

this is the working of overall architecture

6 Evaluation

This Research is implemented with the consideration of time, security and duplication ratio. All the result were compared with previous result in which there is comparison between time taken by number of user for upload file, different block size as per different size of file. There are 3 experiment which will show the processing time taken for file duplication check, block duplication check and encryption duplication check. According to the scenario, the probability of duplicate file submission is maximum with a same user trying to upload same file.

6.1 Experiment / Case Study 1

In the first scenario, data duplication is checked with different file sizes, such as (100MB, 200MB, 300MB, and 500MB); in this situation, the block size is the same for all file sizes, which is 10MB. This analysis is compared to earlier studies, and according to the research, this implementation produces excellent results.8. As shown in result Existing system produce the output just considering the file level deduplication, But In this research the proposed algorithm takes quite higher time as compare to existing system, However Proposed system is not only creating and blocks but also it encrypt the file on file level duplication and block level duplication. If encryption time and block level duplication check will get remove it will give much better output than existing system. Existing system in some cases lots of time eg., if file is greater or minor changes are already available in the file then it will create large file. In this research user can keep the block size as per need and can avoid the unnecessary storage use.

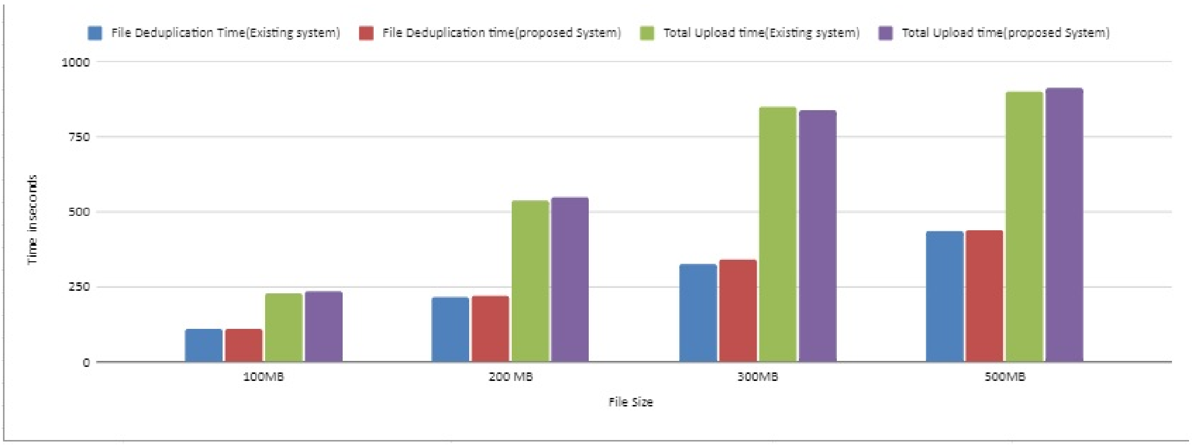

Figure 8: Same Block Different size

6.2 Experiment / Case Study 2

In the Experiment will check 4 conditional duplication check which is given below

- Time for 0 percent duplication check:- In this condition file will be new because there is 0 percent similarity so lets look how much time it will take. In all the cases there is 100Mb file.
- 25 percent duplication check:- In this case 25 percent of data will be already uploaded on the server so as per this algorithm remaining 75 percent of data should get proceed.
- 50 percent duplication check:- Lets consider 50 percent of data is already been available on server so what if user put 100 percent data again through which 50 percent is already available
- 100 percent duplication check:- similar condition will be check in the terms of 100 percent if 100 percent file is already present on the server still how much time algorithm will take for process the data. after considering the above scenario the result will look like below figure9

As shown in the figure the block size is 25MB in the first case 100MB of file when uploaded on the server the total time for upload file were 218.44 sec.there is a single unique file so it doesn't take much time for upload In the file duplication check it took 109.2 sec and for block dedup check algorithm took 108.522 sec and for the encryption process is done within 0.747 sec so in this case blocks will get divided into 4 different parts ie.,

(100MB = 25MB+25MB+25MB+25MB). In the second case there is again upload process of 100MB of file but this time 25 MB of file is already been store on the cloud and after that entire file is sent to the server. Now algorithm will check whether file is already present or not because of changes file will be unique than it will get split in block. Algorithm will again check block is available or not if present then algorithm will map that block to the owner who recently uploaded the same file in the terms of results total count of data deduplication process is 221.4040 sec, 111.002 sec is achieved by file duplication, considering the block algorithm it is 109.878, Total encryption is shown 0.53.

lets move forward to the another test this again upload the 100 Mb file from which 50 percent (50 MB) file is already present, In this case total time is 220.46299 sec taken which is less as compared to second situation so algorithm will check how many blocks already present if there is again similar block they get map with owner remaining will get store on cloud, considering the time taken for file dedup check that is 109.921Sec and for block Check that is 110.069 sec with the 0.484 sec encryption time.

The last condition which is uploading same file that already present on the cloud as shown in fig9 there is only total upload time is present which is 110.7610Sec. If file is already present on the server it will just check on file level and mapped with the owner.

By performing this experiment there is clearly see that File level and block level deduplication is working depending upon the situation.

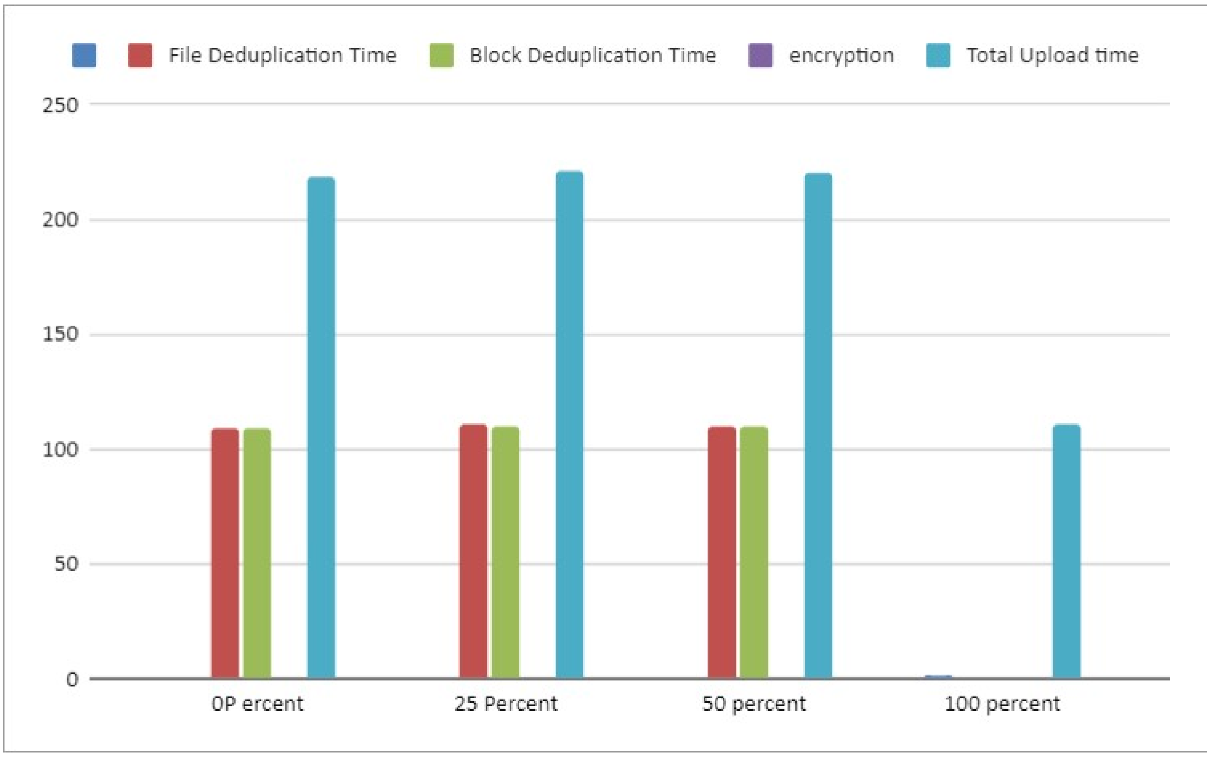

Figure 9: Same Block Different size

6.3 Discussion

This experiment demonstrates how this programme can manage huge file data duplication. This research not only shows the data deduplication approach, but also how to properly enhance the algorithm with a decent deduplication ratio. In terms of storage, this approach can save a significant amount of memory simply because the mapping is done with cache memory so that data can be retrieved rapidly with CPU threads.The main reason for adopting thread in this project is that this technique can check duplicated files of the same owner while data is being transferred to the cloud. one of the major objective of this implementation is security, In this implementation when user upload the data on server the data get encrypted and store with the hash value in data base, In the terms of block the hash value of file again get converted in to number of block and that blocks again get encrypted. The previously described AES technique is used for encryption and decryption. In this research, the block size is not the same for all cases, however in previous systems, the majority of the system block size is kept as static size, which takes a long time to process, thus this problem is also solved in this system. The following points were achieved at the end of the study in this study.

- fetching Encryption time while storing the data
- Deduplication with own data or shared storage
- Considering the previous study and experiment this implementation will take less time as per the situation

In the experiment In every execution there was small change in processing time just because of the background threads the time was nearly equal that means that varies. I executed the system for every size for 5 times and later on I took the average of that particular size depending upon the constant. When the user uploads data using software, I return the processing time on the console. So, in the console, the first algorithm will display file duplication time, if the file is smaller than block size then proceed to file encryption. In the case of large files, it will first print the file deduplication check time, then the block deduplication check time, and when there is an encryption process, it will also print the encryption file time in the console log. Whatever time it took for the entire process was added and displayed on the final result. Total Execution time = filelevel execution time + block level execution time + encryption block time So, in every case, if the data is already existing, it will first check the file deduplication time if comparable data is already, which means it will not take time for the block level or encryption technique that is already present on the server. Similarly, if the file is unique but certain blocks are already existing on the server, the block deduplication check time will be reduced and just the unique block encryption would require time.

7 Conclusion and Future Work

In the future, having a significant variety of gadgets will be necessary for everyone. Rapidly increasing technologies generating unbelievable data so most of the previous study focuses on cloud deduplication so that it can save the storage but security was always the concern. In that case this research not only improve the security but also boost the duplication algorithm with Higher Accuracy. In the terms of cost this will be the best solution for cloud user and discussing about the backup process instead of storing the data it get mapped so backup process get more quick, For now restoration process only take place when there is a backup of file if there is no backup file cant be recover. In the implementation there is no other option if backup is deleted from the server so to keep this backup of server secure that can be implement on server side that could be the better option for future scope for better understanding of data duplication algorithm there are lots of algorithm in process to developed but till now enhancement of CAP probably a good option. This implementation is fully functional each and every scenario and values were test with proper approach.

References

Abualkishik, A. Z., Alwan, A. A. and Gulzar, Y. (2020). Disaster recovery in cloud computing systems: An overview, *International Journal of Advanced Computer Science and Applications* **11**(9).

Ali, G., Ahmad, M. I. and Rafi, A. (2020). Secure block-level data deduplication approach for cloud data centers, *2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, IEEE, pp. 1–6.

Ali, G., Ilyas Ahmad, M. and Rafi, A. (2020). Secure block-level data deduplication approach for cloud data centers, *2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pp. 1–6.

Chang, D., Li, L., Chang, Y. and Qiao, Z. (2021). Cloud computing storage backup and recovery strategy based on secure iot and spark, *Mobile Information Systems* **2021**.

Ji, Y., Zhang, X., Zhang, G. and Jingjing, H. (2019). Research on fine grained software radio communication algorithm based on gpu parallel processing technology, *Cluster Computing* **22**: 1–10.

Jiang, S., Jiang, T. and Wang, L. (2017). Secure and efficient cloud data deduplication with ownership management, *IEEE Transactions on Services Computing* **13**(6): 1152–1165.

Kaur, R., Chana, I. and Bhattacharya, J. (2018). Data deduplication techniques for efficient cloud storage management: a systematic review, *The Journal of Supercomputing* **74**(5): 2035–2085.

Mahesh, B., Pavan Kumar, K., Ramasubbareddy, S. and Swetha, E. (2020). A review on data deduplication techniques in cloud, *Embedded Systems and Artificial Intelligence* pp. 825–833.

Priya, S., Karthigaikumar, P. and Teja, N. R. (2022). Fpga implementation of aes algorithm for high speed applications, *Analog Integrated Circuits and Signal Processing* **112**(1): 115–125.

Shin, H., Koo, D., Shin, Y. and Hur, J. (2018). Privacy-preserving and updatable block-level data deduplication in cloud storage services, *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, IEEE, pp. 392–400.

Shin, Y., Koo, D. and Hur, J. (2017). A survey of secure data deduplication schemes for cloud storage systems, *ACM computing surveys (CSUR)* **49**(4): 1–38.

Suresh, L. and Bharathi, M. (2019). Analysis of block-level data deduplication on cloud storage, *Ambient Communications and Computer Systems*, Springer, pp. 401–409.

Tamimi, A. A., Dawood, R. and Sadaqa, L. (2019). Disaster recovery techniques in cloud computing, *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, IEEE, pp. 845–850.

Venkatesh, C. (2019). Method and apparatus for conversion of virtual machine formats utilizing deduplication metadata. US Patent 10,353,872.

Wu, H., Wang, C., Fu, Y., Sakr, S., Zhu, L. and Lu, K. (2017). Hpdedup: A hybrid prioritized data deduplication mechanism for primary storage in the cloud, *arXiv preprint arXiv:1702.08153* .

Yang, R., Deng, Y., Zhou, Y. and Huang, P. (2021). Boosting the restoring performance of deduplication data by classifying backup metadata, *ACM/IMS Transactions on Data Science* **2**(2): 1–16.