An Efficient Framework and Techniques of Data Deduplication in Cloud Computing

[1]Amanpreet Kaur, [2]Sonia Sharma
[1,2]Dept. of CSE, Guru Nanak Dev University, Amritsar, Punjab, India

Abstract
With an increase in the usage of cloud storage, effective methods need to be employed to reduce hardware costs, meet the bandwidth requirements and to increase storage efficiency. This can be achieved by using Data Deduplication. Data Deduplication is a method to reduce the storage need by eliminating redundant data. Thus by storing less data you would need less hardware and would be able to better utilize the existing storage space. Currently usage of cloud storage is increasing and to overcome increasing data issue, Data deduplication techniques are used. Moreover the Cloud storage service is provided by third party cloud providers thus security of data is needed. Data Deduplication techniques cannot be applied directly with security mechanisms.

To understand the concept and framework of de-duplication process along with application, various methods and technologies involved during the each level implementation of this process. Different chunking algorithm such as fixed, variable and content aware chunking methods are used to decide the chunk size of deduplication. Digital data increase happens in all cloud deployment models and this requires more storage capacity, more costs, manpower and more time to handle data information like backup, replication and disaster recovery, more bandwidth utilization in transmitting the data across the network. If we handle the data effectively like remove the redundant data before storing into the storage device we can avoid data handling overhead and we can improve the system performance. Thus here in this paper we would be discussing data deduplication framework and data deduplication techniques along with securing techniques thus forming secure deduplication.

Keywords
Cloud Computing, Data Deduplication, Cloud Storage, Deduplication Framework

I. Introduction

The use of cloud for storing data by companies for backup and common people for sharing information among friends has increased drastically over the past few years. This has created a challenge to the cloud service providers to maintain all this massive data and to offer these services at lower price to the customers. In reality most of the data stored in the servers is often repeated. For example, a service may contain several instances of same data file, storing all these instances would require a large amount of storage space. This problem can be solved by using Data Deduplication technique. Data deduplication stores only one unique instance of the data type on the disk or tape. In this method redundant data is replaced with a pointer to the unique data copy. This reduces the hardware used to store data and the bandwidth costs required for transmitting and receiving purposes.

Basically Data has been classified into two types namely 'structured data' and 'unstructured data' which are playing a major role in the recent trend. Normally the structure data can be easily organized includes website log data, customer call detailed records etc., Due to the rapid increase of social media usage and mobile usage the unstructured data cannot be easily organized includes blog data, social media interaction data, videos etc., So, unstructured data should be managed in a cost effective way. Today in IT budgets, on an average of 13% of the money being invested on storage capacity1. Data to grow more quickly says IDC's Digital Universe study2. These impacts create more problems, like degradation of performance, compromise of quality, and more operational costs. So in order to overcome the above problems and handle system where the concept of Deduplication is derived. Deduplication technology looks into data either at a block level (sub file) or file level. The incoming data is split into smaller fixed or variable blocks or segments. Each of these smaller blocks is given a unique identifier which is created by several hashing algorithms or even a bit by bit comparison of the block. Common algorithms used for this process are MD5 or SHA-1. Also content aware logic, which considers the content type of the data and finalize the size of block and boundaries. As the deduplication system processes data, it compares the data to the already identified blocks and stores in its database. If a block already exists in the database, the new redundant data is discarded and a reference to the existing data is inserted into the repository. If the block contains new, unique data, then the block is inserted into the data store (file system), and a reference is added to that block in the de-dupe database. The primary benefit of deduplication is that it greatly reduces storage capacity requirements, drives several other advantages like lower power consumption, lower cooling requirements, longer disk-based retention of data (faster recoveries), and disaster recovery.

After the rapid development of cloud computing, users and enterprise would like to back up their data to cloud storage. According to prediction given by International Data Corporation the digital data will exceed 44 Zeta Bytes in 2020 [1-2]. The development of cloud storage encourage the service provides to make the data storage service been outsourced to third-party cloud providers [3]. Management of ever increasing data over the cloud storage is a important issue to be looked upon. With the explosive growth of digital data, deduplication techniques are used widely to backup data and minimize storage and network overhead by detecting and eliminating redundancy among data [4].

II. Data Deduplication

Data deduplication is a technique for eliminating duplicate copies of data, and has been widely used in cloud storage to reduce storage space and upload bandwidth [4]. When a data is uploaded its hash value is formed and then compared with the existing hash value, if duplicate value is found then that data is not uploaded and is replaced with pointer to the unique data else if no duplicate is found the data is uploaded to the server.

Various benefits of Data Deduplication technologies are:

- Increases network efficiency.
- Lower storage space requirements.
- Storage cost is reduced.
- Reduced upload bandwidth.

A. How Deduplication Works?

Data deduplication works by comparing objects (usually files or blocks) and removes objects (copies) that already exist in the data set. All the processes which are not unique are removed in this method. In Data deduplication method we divide the input data into blocks and a hash value is calculated for each of these blocks. Then using these hash values we can determine whether another block of same data has already been stored. If a similar data file is found then replace the duplicate data with a reference to the object already present in the database.

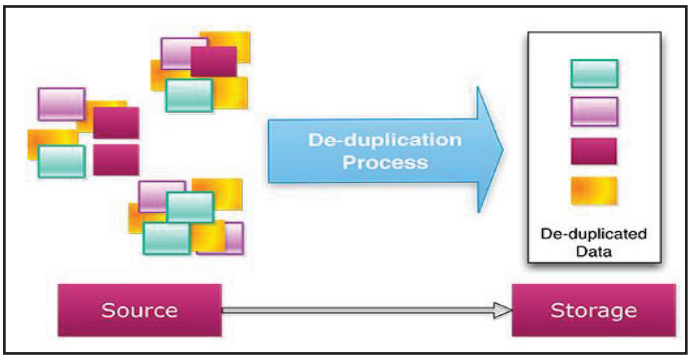
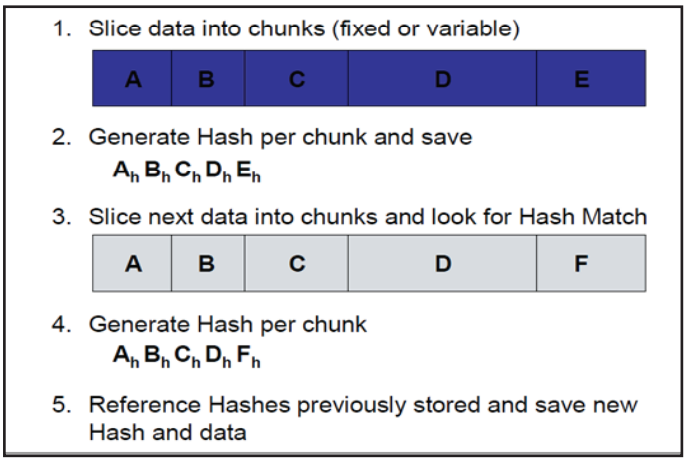


Fig. 1: Deduplication Process

1. Hash-based Algorithms

Hash based deduplicatoin methods use algorithms to identify chunks of data. If the hash is already created, the data is identified as a duplicate and is not stored. Commonly used algorithms are Secure Hash Algorithm 1(SHA1) and Message-Digest Algorithm 5(MD5).SHA-1: This was devised to create cryptographic signatures for security application. The 160-bit value created by SHA-1 is unique for each piece of data, it breaks data into "chunks" which are either fixed or variable in length. This processes the "chunk" with hashing algorithm to create a hash, If the hash already exists, the data is deemed to a duplicate and is not stored. If the hash does not exist, then the data is stored and the hash index is updated with the hash.



MD5: This 128-bit has also designed for cryptographic uses. In this method the 128-bit state is divided into four 32-bit words, denoted A, B, C and D. These are initialized to certain fixed constants. The main algorithm then uses each of these messages in turn to modify the state. The processing of a message block consists of four similar stages, termed rounds.

$$F(B,C,D) = (B \wedge C) \vee (\neg B \wedge D)$$
$$G(B,C,D) = (B \wedge D) \vee (C \wedge \neg D)$$
$$H(B,C,D) = B \oplus C \oplus D$$
$$I(B,C,D) = C \oplus (B \vee \neg D)$$

$\oplus, \wedge, \vee, \neg$ Denote XOR, AND, OR and NOT operations respectively.

III. Improved Technique Deduplication

Deduplication is an effective technique for optimization of instances of data stored in cloud storage [5]. Deduplication can be classified into chunk level and file level deduplication. Chunk level deduplication method enforces the storage of unique chunks by comparing every incoming chunk for duplicate identification. This method achieves better deduplication efficiency because it does exact deduplication [6]. Based on how the incoming chunk is checked against duplicates, deduplication can be categorized into two types, namely, chunk[6] and file level.

A. Chunk Level Deduplication

Whenever a data stream has to be written, every chunk in the stream is checked for duplicates before writing. This is termed as chunk level deduplication. Since every incoming chunk is checked for possible duplication, only unique chunks occupy the cloud storage. Therefore, chunk level deduplication has better deduplication efficiency. However, as each incoming chunk is checked against a large list of chunk indices, the number of disk I/O operations is large. This has a significant impact on deduplication throughput. Storage of traditional backup workload demands good deduplication efficiency as it involves large data redundancy among different workloads. Hence, this deduplication approach is best suited for such workloads [6].

B. File Level Deduplication

Whenever a data stream has to be written, every chunk in the stream is checked against the chunks of similar files. This is termed as file level deduplication. This approach provides a scalable solution with the division of chunk index into two tiers namely Primary and Secondary index [6]. In this approach, all the Chunk_IDs that constitute a file and the minimum Chunk_ID among them are found. This minimum Chunk_ID is termed as representative Chunk_ID. According to Broder's Theorem, two files are said to be nearly similar, when the representative Chunk_IDs of both the files are same. Primary chunk index consists of representative Chunk_ID, whole file hash and the address of the secondary index or bin. Bin is made up of three fields namely, Chunk_ID, chunk size and the storage address of the chunk [5].

IV. System Architecture

The main aim in designing this scheme is to design an improved technique for storage in Cloud computing [3].The overall architecture of the technique is shown in fig. 3 Overall architecture is divided into four layers, Interface Layer - Interface layer provides the user interface to select the file for deduplication .It also provides interface to select the type of deduplication and also to specify the split length. Chunk Layer - Based on the split length different Segments of file are created. For these chunks hash values are computed by using MD5 algorithm.

Algorithm to Segment File:

Algorithm 1: File Segment
Input: File selected Split Size
Output: Files divided into segments based on split size
Procedure: Segment File
/*Enter the split size.*/
/*Read the input file and create the byte stream.*/
while(bytes != -1)
{Divide the file into number of bytes specified by split length and create the new file with .txt extension}
end while
/*Different segments of the file is created.*/
end

Algorithm to Upload File:

Algorithm 2: Upload File
Input: File
Output: File uploaded to Cloud
Procedure: Upload
begin upload
/*Select file to upload*/
/*Select Amazon S3 client*/
/* provide Aws credential properties that is secret key and access key of Amazon client.*/
/*upload the file to specific bucket using S3.putobject ( )*/
end.

V. Deduplication Framework

The deduplication framework has four important steps including data partitioning and extraction, finger print calculation and lookup, comparing the finger prints and write the finger prints into the database.

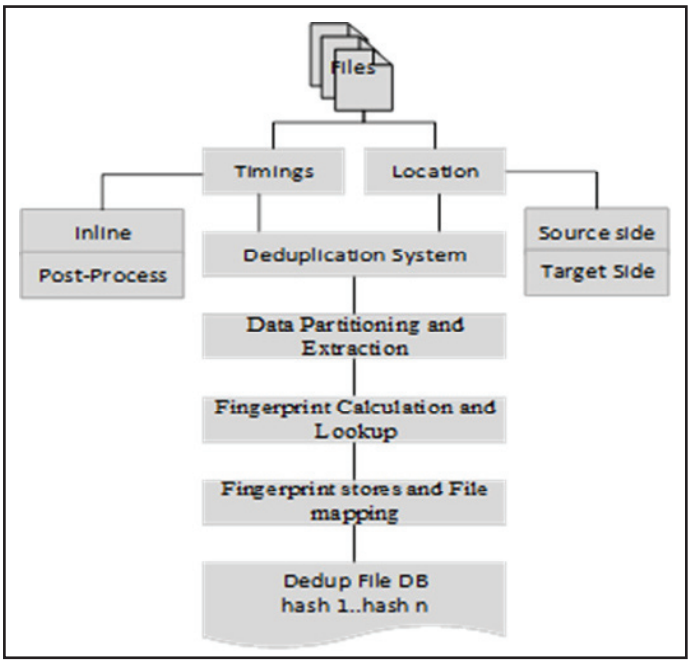


Fig. 4: Framework of Deduplication System

Fig. 4, explains the framework of deduplication. Deduplication process can be classified based on the timings, and location. Based on the timings deduplication can happen on the Post Process or Inline. In the Post Process techniques first data's are stored into the disk then it removes the redundancy data whereas Inline method first removes the redundancy data then stores it into the disk. The main advantages of the post process method is the deduplication algorithm completely understands the data which already stored in the storage system but this method consumes more disk spaces as it stores the complete data before start the deduplication. Compared with the post process method, the Inline method is adapting in to the most recent backup deduplication process, because extra disk space is not required as it is removing the redundant data on the fly.

Based on the location, deduplication can happen on the source side or target side. In the source side, redundant data will be detected and it will be moved to the target side, whereas target side deduplication redundant data will be detected on the target side not on the source side. In the below sections we have discussed each and every steps of deduplication framework in detail.

C. Data Partitioning and Extraction

In the first step, the incoming data's are partitioning or dividing using the chunking algorithm. The chunk size has implications on number of chunk entries and the hash lookup so it can decide the deduplication ratio and the performance of the system8. Dividing data's into smaller chunks or segments are happening into two different major levels such as File-Level deduplication and Block-Level deduplication.

In the File-Level deduplication, the hash value will be created for each file using cryptographic hash algorithm such as MD or SHA-15,10,14 and the same value will be stored in the hash table. As it is creating only one hash value for each file, it takes minimal time to look up the hash value from the hash table but it fails and provides more duplicated date if the sub set of the files changes by only a single byte also we can expect more delay when it handles a large files.

The Variable-Size Chunking divided the data into variable size chunks and the boundaries are finalized based on the content of the file not on the offset of the file by the finger print algorithm such as sliding window approaches26 rolling hashes27 Rabin fingerprints25, and bimodal chunking28. This method overcomes the boundary shifting problem which is occurring in the fixed size chunking method. Delta encoding is a one of the variable size chunking approach will record the change between a source file and a target file. As it stores only changed byte on storage, it will not store the file when files are identical or just small changes between almost same files. But it can be only performed on a pair of files as well as it has to remember the file or chunks that are used for delta encoding.

D. Fingerprint Calculation and Lookup

After data chunking has been done, the finger print creation and lookup has to be done with existing stored hash value to remove the de-duplication. Generally the identification of duplicate can be done by comparing data hash value or files bit by bit. These methods provide correct accuracy but taking some additional time. When we use the hash algorithm to find out the duplicate, the hash collision would be increased which depends on the hash algorithm. Thus choosing the hash algorithm is very important at this stage. Comparing the finger prints and writes the finger prints into the database are last steps in the deduplication process. The most important point to remember is when we deal with single node deduplication system.

VI. Various Data Deduplication Approaches

Data Deduplication can be applied in various forms as follows:

Based on Granularity

- File Level Deduplication
- Block Level Deduplication

Based on Time of Application

- Inline Deduplication
- Post Process Deduplication

Based on point of Application

- Source Based Deduplication
- Target Based Deduplication

A. Based on Granularity

1. File Level Deduplication Approach

File Level Deduplication is also referred to as single-instance storage (SIS). It compares a file to be backed up or archived with those already stored, by checking its attributes against an index. If the file found to be unique then it is stored and the index table is updated, but if the file is not unique then a pointer to the presented file is stored. The outcome is that only one occurrence of the file is saved and consecutive copies of the files are replaced with a pointer to the original file. [7]

2. Block Level Deduplication Approach

Block-level data deduplication operates on the sub-file level. As per the name, the file is normally broken down into segments, chunks or blocks that are checked for redundancy vs. previously stored information. [7]. Block Level Deduplication is further divided into Fixed Chunk Level Deduplication and Variable Chunk Level Deduplication. In Fixed chunk level deduplication the blocks are divided into fixed chunk size of say 4 KB, 8KB and so on. Then check for deduplication. And in Variable block size the blocks are divided into various size blocks and then checked for Deduplication.

B. Based on Time of Application

1. Inline Deduplication Approach

Deduplicating the data before it is written to disk thus it reduces the storage requirement. The inline deduplication only checks the incoming raw blocks and it does not have any knowledge of the files. This forces it to use the fixed-length block approach. Extent of deduplication is less, and only fixed-length block deduplication approach can be used [8].

2. Post Process Deduplication Approach

In this approach data is first written to the storage device and then checked for deduplication. It can be applied on file-level or sub-file levels. Whole file data checksum can be easily compared with the existing checksums of previous backuped files and thus full file level duplicates can be eliminated easily [8].

C. Based on Point of Application

1. Source Based Deduplication Approach

Deduplication is applied when data is on the source i.e. when data is created. [8] Then the non-duplicate data is backup to the cloud. It helps in better and optimized utilization of resources. It is also helpful in incremental backup of new blocks in the user's instances.

2. Target Based Deduplication Approach

Deduplication occurs after data is been stored. Process of removing Deduplication occurs when data was not generated at that location [8]. User is unaware of the deduplication process occurrence. Thus this approach helps in storage utilization but does not helps in saving upload bandwidth [8].

VII. Results and Analysis

A. Chunk Level Deduplication:

Table 1: Before Deduplication

File Name	File Size in Bytes
Dee.txt	3734
Deepu.txt	3736
File.txt	3005
File1.txt	3007
Graph.txt	3291
Grapho.txt	5398
Graph1.txt	309
Total size	25266

Table 2: After Deduplication

File Name	File Size in Bytes
Dee.txt	3742
Deepu.txt	536
File.txt	3011
File1.txt	607
Graph.txt	3301
Grapho.txt	4610
Graph1.txt	2301
Total size	1088

Total space saved=25266-18108=7158=6.99 KB

Results shown in Table 2 is the file size after deduplication it corresponds to results shown in Figure 3.Seven files of different size of .txt and .java files are chosen for deduplication. By applying deduplication approach on these files we are able to save the Cloud storage space up to 50%.

B. File Level Deduplication

Table 3: After File level Deduplication

File size	File Size in Bytes
Dee.txt	3208
Deepu.txt	3208
File.txt	2406
File1.txt	2406
Graph.txt	3208
Grapho.txt	4812
Graph1.txt	2406
Total Size	21654

Total space saved=25266-21654=3612=3.52 KB

Results of Chunk level and File level deduplication techniques are shown in Table 2 and Table 3. Deduplication technique was carried out on set of sample files. For the same set of files, space saved in Chunk level is 6.99 KB and in File level is 3.52 KB. This is illustrated in the graphs shown above. From these results we can illustrate that Chunk level deduplication achieves better deduplication efficiency because it does exact deduplication [5]. However, the throughput is low as it checks every incoming chunk for duplication. File level deduplication achieves better throughput as it compares every incoming chunk only with chunks of similar files. However, the deduplication efficiency is comparatively low as some duplicate chunks may be found across different groups. Hence, this technique performs only approximate deduplication.

C. Deduplication Result

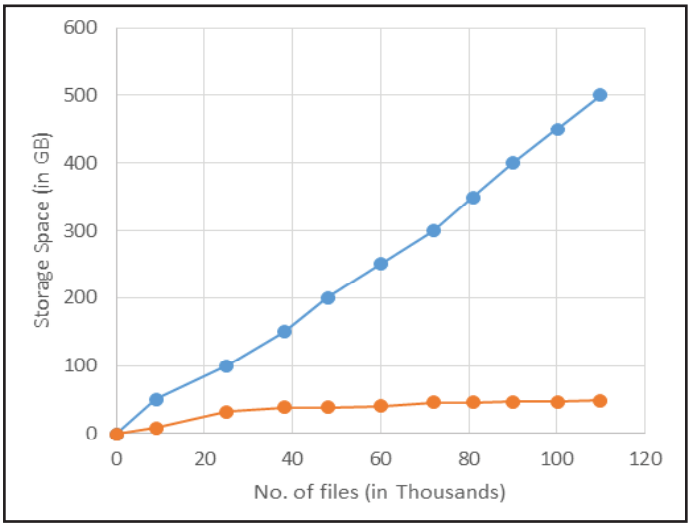


Fig. 5: Comparison of Without Deduplication and with Deduplication

Using deduplication concept in the backup storage system we can save up to 75% storage space. Figure 2, compares without deduplication and with deduplication space savings efficiency41. If we store 500 GB data without deduplication, the traditional storage system requires 500 GB storage space but using with deduplication we need only 40 – 50 GB storage space instead of 500 GB. So we can save lot of spaces using the deduplication.

VIII. Conclusion

This paper discusses the information about data deduplication for the cloud based systems. It includes the methods that are used to achieve cost effective storage and effective bandwidth usage by deduplication. The core concept involves eliminating the duplicate copies of the repeated data by using hashing algorithms. The future challenge therefore lies in identifying more effective hashing algorithms for improving the speed of storing data and security.In our study we have explained the complete framework of the deduplication. Inline deduplication method avoids the extra storage space compare than the Post Process method. The data partitioning and extraction is very crucial steps in the deduplication process, so the choosing of chunking algorithm can decide the entire deduplication performance as well as the correct chunk size can improve the deduplication ratio and the throughput. There are various studies to create the fingerprint value for each chunk and enormous effort to reduce the fingerprint lookup timings. Also when we handle the huge amount of data in the cloud storage the distributed system with cluster concept can avoid the single point failure.

References

[1] Yukun Zhou, Dan Feng, Wen Xia, Min Fu, Fangting Huang, Yucheng Zhang, Chunguang Li,"SecDep: A User-Aware Efficient Fine-Grained Secure Deduplication Scheme with Multi-Level Key Management", IEEE Mass Storage Systems and Technologies (MSST) 2015 31st Symposium, Year - 2013

[2] "The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things", http://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm, April 2014, EMC Digital Universe with Research & Analysis by IDC.

[3] Li, X. Chen, M. Li, J. Li, P. Lee, W. Lou,"Secure deduplication with efficient and reliable convergent key management", IEEE Transactions on Parallel and Distributed Systems, Vol. 25(6), Year – 2014

[4] Jin Li, Xiaofeng Chen, Xinyi Huang, Shaohua Tang, Yang Xiang, Mohammad Hassan, AbdulhameedAlelaiwi,"Secure Distributed Deduplication Systems with Improved Reliability", IEEE Transactions on Computers Volume: PP, Year – 2015

[5] Amazon Inc., "Amazon Elastic Compute Cloud," http://aws.amazon.com/

[6] Amrita Upadhyay, Pratibha R Balihalli, Shashibhushan Ivaturi and Shrisha Rao,"Deduplication and Compression Techniques in Cloud Design", 2012 IEEE

[7] http://searchdatabackup.techtarget.com/tip/The-pros-and-cons-of-file-level-vs-block-level - data-deduplication-technology

[8] http://www.druva.com/blog/understanding -data-deduplication/