A Review on Data Deduplication for Cloud Backup Services of Personal Storage

Sujata V. Randad[1] and V. R.Chirchi[2]

[1,2]Computer Science and Information Technology Department, MBES COE, Ambajogai, Maharashtra, India.

ABSTRACT

After opening a window of cloud storage for backup services of personal storage it is becoming a serious issue to protect it while sharing it with our personals. Also the problem of local and global source duplication arises due to the different and independent layers of personal storage present in our systems or devices. This will be making redundant and inefficient cloud storage for personal use and raising the cost to purchase it and also the fact that the applications and devices are running at background for cloud storage cannot be ignored. The only solution to this problem is to search a technique for a local and global source deduplication for cloud backup services of personal storage. So this paper gives a comprehensive study on deduplication for cloud backup services considering different aspects and algorithms with respect to mentioned studied papers.

Keywords – Data deduplication for backup services, local global source deduplication, cloud backup services, personal storage.

I. INTRODUCTION

As we know that the increasing use of smart phones other devices like cameras, microphones, radio-frequency identification(RFID) readers had increased global data traffic in such a way that there is various difficulties are arising while handling its variety, volume and velocity. And there is no doubt that this will be making a really big threat in personal space when it is a think of sharing it with our family or friends. The numerous applications are continuously starving for the data to increase the volume on cloud. In this way the data getting redundant when we consider its local and global storage presence. This has to be deduplicating for efficient cloud as well as personal storage. Usually the term deduplication stands for an emerging technology that introduces reduction of storage utilization and an efficient way of handling data replication in the backup environment.

Generally deduplication techniques are used in the cloud server for reducing the space of the server. It helps to manage the data growth, increase efficiency of storage and backup, reduce overall cost of storage, reduce network bandwidth and reduce the operational costs and administrative costs. The most common way for deduplication is to divide large data into the smaller chunks and create there references to further use. These references are called as a hash values. Then these values are used to determine if another block of the same data has already been stored or not. After this replace the duplicate data with a reference to the object already in the database. Similar patterns are easy to find in smaller chunks of data and then they are encrypted before outsourcing or sharing. And also the duplicated data effects on the storage and the performance of the cloud. The major duplication can be seen in backup systems. There are number of algorithms are found for deduplication. This data deduplication is also referred as intelligent compression or single instance storage [1].

This paper gives a comprehensive study on various deduplication algorithm like section II consists of Secure Hash algorithm, section III consists of Two Threshold Two Divisor Switch (TTTDS), section IV consists of Secure Deduplication With Efficient And Reliable Convergent Key Management, section V describes Hybrid Cloud Approach. Section VI describes Block level data Deduplication algorithm and section VII discusses on ALG-Dedupe algorithm in detail.

II. SECURE HASH ALGORITHM (SHA1)

This algorithm deals with the hash function that can process a message to produce a condensed representation. This takes two steps as preprocessing and hash computation. These hash functions should not create the same index for different data. In other words, an index is normally considered a hash key that represents data. Indexes should be saved to permanent storage devices like a hard disk, but to speed up the comparison of indexes, they are prefetched in memory. The indexes in memory should provide temporal locality to reduce the number of evictions of indexes from memory owing to filled memory as well as a decrease in the number of prefetches. In the same sense, to prefetch related indexes, the indexes should be grouped by spatial locality. That is, indexes of similar data are stored close to each other in storage [2].

III. TWO THRESHOLD TWO DIVISOR SWITCH (TTTDS)

In this algorithm the main focus is given to segment a input file into smaller chunks. But sometime chunking is the very time consuming process as we have to traverse till the end of file. it take four basic variables that determine its behavior (Min, Max, D and Ddash) parameter values. The smallest chunk size is maintained as possible as to reduce the chances of identifying duplicate data. TTTDS algorithm sets minimum and maximum threshold boundaries for every chunk of file. The schematic diagram of the algorithm is shown in figure 1. The TTTD algorithm is consists of the following steps [3]:

- Read file as one character at time.
- Skip first Min boundary.
- When reach Min value start to compute finger print value for last window size.
- If (finger print) mod D = D-1 then consider it as a chunk boundary and add breakpoint And go to 2
- Else if (finger print) mod Ddash =Ddash -1 then consider it as backup breakpoint and continue reading next character and update window size by deleting first character and append new one and compute new (finger print).

 If Max boundary reached, if there is any backup break point use it, else use max as a break point boundary; then go to 2.[4]

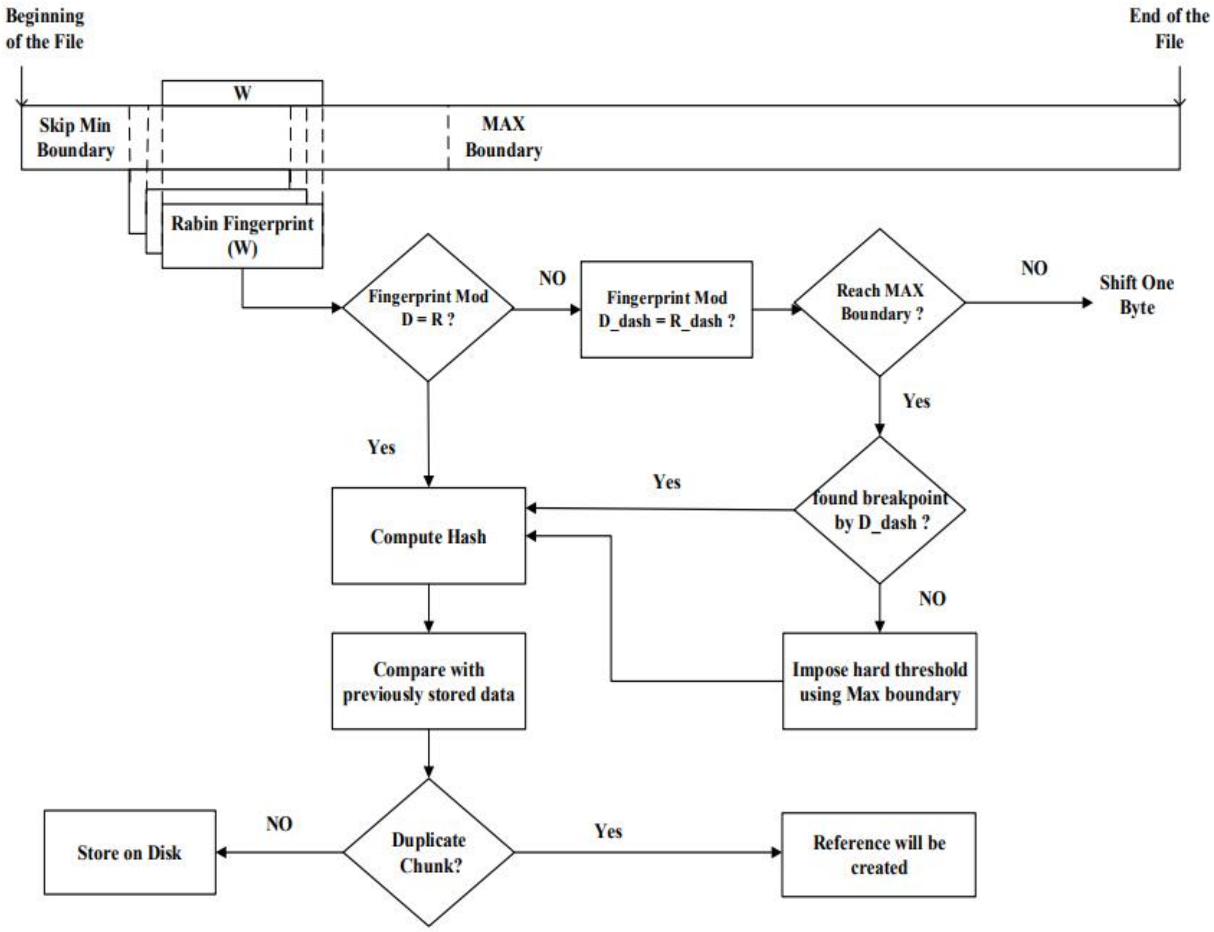

Figure 1- Two Threshold Two Divisor Switch (TTTDS)

IV. SECURE DEDUPLICATION WITH EFFICIENT AND RELIABLE CONVERGENT KEY MANAGEMENT

In [5] Jin Li, et al, proposed a reliable convergent key management scheme for secure deduplication among convergent keys and distributes convergent key shares across multiple key servers while preserving semantic security of convergent keys and confidentiality of outsourced data[3]. In this paper they have used two levels of deduplication as file level deduplication and block level deduplication. In this scheme the user first performs the file level duplicate check and if it is found duplicate, then its entire block must be duplicate as well; otherwise, the user goes for block level duplicate check. Dekey scheme constructs secret shares on the original convergent keys and distribute the shares across multiple key-management cloud service providers (KM-CSP). If same block is shared among multiple users then same corresponding convergent keys is accessed by them. So this approach reduces storage overhead, provides fault tolerance and allows the convergent keys to remain accessible even if any subset of KM-CSPs fails.

V. HYBRID CLOUD APPROACH

In hybrid cloud approach for secure authorized deduplication [6] Jin Li, et al, introduces a hybrid cloud architecture to solve the problem. This model uses three entities to achieve deduplication as users, private cloud and S-CSP in public cloud. The different privileges are set on the files according to various applications. Each file is given a unique file token with a specific set of privilege values. User computes and sends duplicate check tokens to the public cloud for authorized duplicate check.

The S-CSP is a data storage service in public cloud. It provides efficient cloud storage by applying deduplication. Data users are those who want to outsource data storage to the S-CSP and access it later. They are uploading only unique data to save upload bandwidth. Each file has given a convergent key and privilege key for encryption and security. At last private cloud provides an interface between the user and public cloud. In this scheme private keys are managed by the private cloud server. To get a file token, the user needs to send a request to the private cloud server. The private cloud server will also check the user’s identity and authorized duplicate check for this file can be performed by the user with public cloud. On the basis of results of duplicate check the user either uploads this file or runs POW. In short data deduplication was proposed to protect the data security by applying different privileges of users in the duplicate check.

VI. BLOCK LEVEL DATA DEDUPLICATION

In [7] V. Badge and R. Deshmukh applied a block level deduplication approach to reduce the data redundancy and for secrecy of data SHA-256 a cryptographic hash algorithm is implemented in this technique consist of two parts client side and cloud side. The client side generates the fingerprint index and cloud side fingerprints of the received index are stored in the global index. The system uses file size filter to sort out small size files (size less than 10KB) and large size files. Then small files are sent over to the data deduplicator and hashing unit. Then large files sent to the data chunking unit. Then this unit splits large files into small size files (maximum chunk size considered is of 256KB). After data chunking process, the data deduplication and SHA-256 hashing unit deduplicates the data and generates fingerprint of the data. To achieve data deduplication efficiency duplicate fingerprints are matched and if they are found same then they are not uploaded to the cloud. In this way only unique data chunks and the fingerprints are sent to cloud storage. In chunk compression unit DEFLATE compression algorithm [8] is implemented to compress the chunks of larger files and then uploaded to the cloud. Lastly in index generator the fingerprints works as the key element to identify the similar data chunks in the index. By using SQL index computational overload of matching of fingerprint for every session is reduced. By checking the index at first client side with local index and then at cloud side global index with already stored fingerprint the decision of uploading is taken. In this way the bandwidth is properly utilized by not sending the duplicate chunks.

VII. APPLICATION-AWARE LOCAL-GLOBAL SOURCE DEDUPLICATION FOR CLOUD BACKUP SERVICES OF PERSONAL STORAGE

In this approach an application-aware local-global source-deduplication scheme for cloud backup in the personal computing environment to improve deduplication efficiency. An intelligent deduplication deduplication strategy in ALG-Dedupe is designed to exploit file semantics to minimize computational overhead and maximize deduplication effectiveness using application and global deduplication to balance the effectiveness and latency of deduplication. The algorithm proposes a central index and divides it into many independent small indices to optimize look performance. This algorithm increases the performance to improve the deduplication efficiency. To achieve this the application aware deduplicator first detects duplicate data in the application-aware local index corresponding to the local dataset with low deduplication latency in the PC client, and then compares local deduplicated data chunks with all data stored in the cloud by looking up fingerprints in the application-aware global index on the cloud side for high data reduction ratio.[9] The architecture of ALG-Dedupe consists of file size filter which filters tiny files for efficiency reasons, and hen broken into chunks by intelligent data chunker. Chunks are then deduplicated in the application-aware deduplicator by generating chunk fingerprints in hash engine and performing data redundancy check in application-aware indices in both local client and remote cloud. Then their fingerprints are matched in disk for local redundancy check. If a match is found then metadata for the file containing that chunk is updated to point to the location of the existing chunk. [9]

VIII. CONCLUSION

From the above discussion we can conclude that there is very much need of data deduplication will arise as per the pervasive nature of cloud storage and there are varios algorithms are present to perform the deduplication. The ALG- dedupe is one of the deduplication algorithm which improves the efficiency of process in personal computing. Also this algorithm designed to minimize computational overhead and maximize deduplication effectiveness using application awareness.

REFERENCES

[1] Sulbha Ghadling, et al, “Data deduplication using optimized fingerprint lookup method for cloud storage using android,” IJECS, vol. 4, pp. 1071-10720, March 2015.

[2] ,D. Kim et al., Springer, “Data Deduplication for Data Optimization for Storage and Network Systems”, International Publishing Switzerland, 2017, DOI 10.1007/978-3-319-42280-0_2.

[3] Kruus, E., Ungureanu, C. and Dubnicki, C. 2010. Bimodal Content Defined Chunking for Backup Streams. Fast, pp. 239-252.

[4] Hala Abdulsalam and Assmaa A. Fahad, “Evaluation of Two Thresholds Two Divisor Chunking Algorithm Using Rabin Finger print, Adler, and SHA1 Hashing Algorithms”, Iraqi Journal of Science, 2017, Vol. 58, No.4C, pp: 2438-2446.

[5] Jin Li, et al, “ Secure deduplication with efficient and reliable convergent key management”, IEEE transactions on parallel and distributed systems, june 2014, vol. 25, pp. 1615-1625.

[6] Jin Li, et al., “A hybrid cloud approach for secure authorized deduplication”, ”, IEEE transactions on parallel and distributed systems, May 2015, vol. 26, pp. 1206-1216.

[7] V.Badge and R. Deshmukh,”Block level data deduplication for faster cloud backup”, IJSER, vol. 6, pp.998-1002.

[8] http://www.zlib.net/feldspar.html

[9] Y. Fu, et al., “Application-aware local-global source deduplication for cloud backup services of personal storage”, IEEE transactions on parallel and distributed systems, May 2014, vol. 25, pp. 1155-1165.