Target Based Document De-Duplication for optimized Virtual Storage using Extended Frequency Based Chunking Algorithm(ExFBC).

[1]K.Geetha
Ph.D. Research Scholar(PT)
Guest Lecturer,
Department of Computer Applications
Govt. Arts College(Auto)
Salem-7, TamilNadu, India

[2]Dr.A. Vijaya
Research supervisor
Asst. professor& Head
Department of Computer Applications
Govt. Arts College (Auto)
Salem-7,TamilNadu.India

Abstract

Most of the mobile based services are generating trillion TB of data every second and are taken backup for future reference and recovery periodically. The existing storage techniques are becoming insufficient when they backup huge volume of data often. So there is a need to depend third party service providers like cloud to overcome these issues. The backup process increases duplicate generation of documents, that occupies most storage spaces unnecessarily. To manage this problem an intelligent technique of de-duplication has been used to remove duplicate information. In de-duplication the content based chunking methods have been used to break the input stream of file into blocks. This eliminates the redundant information based on the content and improves the proper utilization of storage space. In this paper, it has been proposed an Extended Frequency Based Chunking (ExFBC) algorithm. The existing most popular Basic sliding Window(BSW) algorithm and Two Threshold Two Divisor(TTTD) algorithm are creating issues in defining the average chunk size and number of chunks. ExFBC explicitly utilizes the chunk frequency information in the data stream to enhance the chunking size and running time of the algorithm. The ExFBC algorithm includes chunk frequency estimation algorithm for identifying the frequency of chunks, and a two-stage chunking algorithm which uses these chunk frequencies to obtain a better chunking result. To improve the effectiveness of de-duplication the documents has been clustered based on user index and also by categorizing the files based on its type.

Keywords: ***Cloud, De-duplication, BSW, TTTD, Frequency Based Chunking, Dedup gain ratio, ExFBC.***

1. Introduction

Today, a predominant portion of Internet services, like content delivery networks, news broadcasting, blogs sharing and social networks, etc., is data centric. A significant amount of new data is generated by these services each day. To efficiently store and maintain backups for such data is a challenging problem for current data storage systems.

In comparison to the compression technique which does not support fast retrieval and modification of a specific data segment, chunking based data de-duplication is becoming a prevailing technology to reduce the space requirement for both primary file systems and data backups. In addition, it is well known that for certain backup datasets, de-dup technique could achieve a much higher dataset size reduction ratio comparing to compression techniques such as gzip . The whole file chunking and fixed size chunking are becoming ineffective when we go for large sized documents. They are facing issues with boundary shifting problems. So, breaking up a file into chunks(blocks) and removing duplicated is effective when compare with whole file de-duplication. It will be optimized when we break the file based on the content as well as the number of occurrence of the chunks.

A good chunking algorithm should be capable of generating minimum chunks with effective processing time. Even though the existing Basic Sliding Window(BSW) and Two Threshold Two Divisor(TTTD) algorithm are for content based chunking, by its time consuming with boundary shifting problems they set back in data de-duplication.

The frequency based chunking (FBC) algorithms are good in producing chunks based on the occurrence of contents in input stream. This effectively reduces chunk size and number of chunks. So, there will be no issues in maintaining hash table and indexing the hash produced for chunks. The FBC itself adopt with existing CDC algorithm BSW for producing chunks with the help of Fixed size sliding widow. Our Extended FBC has two stages. In first stage it uses BSW to produces chunks of larger size than desired. Then the ExFBC checks the frequency of occurrence for chunks. This will eliminate the duplicate chunks in order to fine tune the resultant chunks. By this the storage and retrieval of information will be more efficient than the other techniques.

2. De-Duplication

This technique has been used to identify the duplicates from huge volume of datasets and provide effective usage of storage space [10]. The process involves breaking the input stream into fine grained blocks and calculates hash value for each block which is used to identify the unique blocks. The hash values are maintained in an index table to reassemble the content using mapping. For arrival of each new block the index table is updated with new hash id. This process is of two types as Offline and Online De-duplication. The duplicates are removed in offline after the data has been moved to storage space where in online the duplicates are removed before the transfer of data.

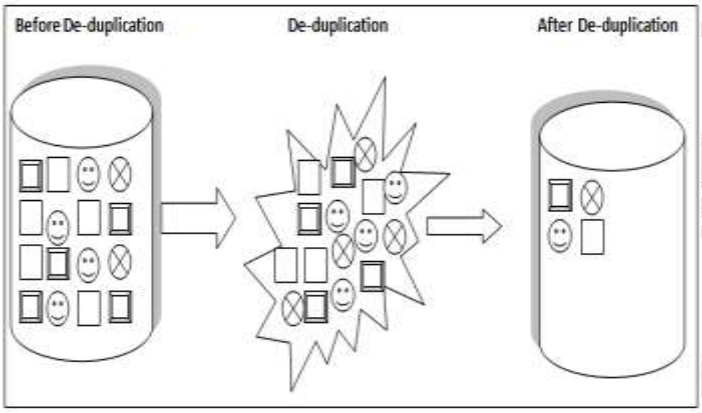


Fig 1:Data De-Duplication process

2.1 De-Duplication Types

The de-duplication algorithm and its implementations affect both how the duplicate chunks are stored and later how they are restored. Based on this, it is of two types as Source and Target De-duplication. In source the de-duplication is carried out where the data has been generated and in target the de-duplication is carried out where the data is stored. Target de-duplication removes data duplicates on the secondary store.

2.2 De-Duplication Levels

File level- In this method the whole file will be treated as a single unit. Here duplicate files are identified by comparing their hash values with the existing stored in hash index table. It is easy and simple to process. **Block Level-** Here the file is divided into blocks and each block will be assigned a unique hash value. Block with same hash value will be treated as duplicates and hence removed. Single copy of blocks will be maintained.**Byte Level-** It is same as block level. It focuses on the content or semantic of the file. It understands the content of the data. So it is more efficient than others.

3.Literature Review

The primary benefit of data de-duplication is that it reduces the amount of disk or tape that organizations need to buy, which in turn reduces costs. NetApp reports that in some cases, de-duplication can reduce storage requirements up to 95 percent[11]. De-duplication is sometimes confused with compression, another technique for reducing storage requirements. While de-duplication eliminates redundant data, compression uses algorithms to save data more concisely[11]. The latest cloud based storage services must provide storage reliability and efficient retrieval of large amount stored data without sacrificing time and cost.

The SAERS- a cloud based storage system integrates erasure coding and data de-duplication support efficient storage and reliable data storage with faster response for user requests[5]. Data de-duplication technique has three types of chunking categories like Whole file chunking, Fixed size chunking(FSC) and Variable sized chunking or Content Defined Chunking(CDC)[9]. The variable sized chunking is based on the content available in the data stream[6]. The whole file chunking plays predominant role in removing duplicate files in cloud storages. The Dynamic Whole file De-duplication (DWFD) provides dynamic space optimization in private clouds storage backup and increase the throughput and de-duplication efficiency[7]. The FSC is totally depend on the chunks with fixed size. Smaller the fixed size chunk has better de-duplication ratio. Moreover FSC algorithm has Boundary shifting problems[8]. The Content Defined Chunking(CDC) is Better than the remaining two in solving Boundary Shifting problem[8]. So, the Enhanced FBC adopt the content defined chunking and also the frequency of content occurrences.

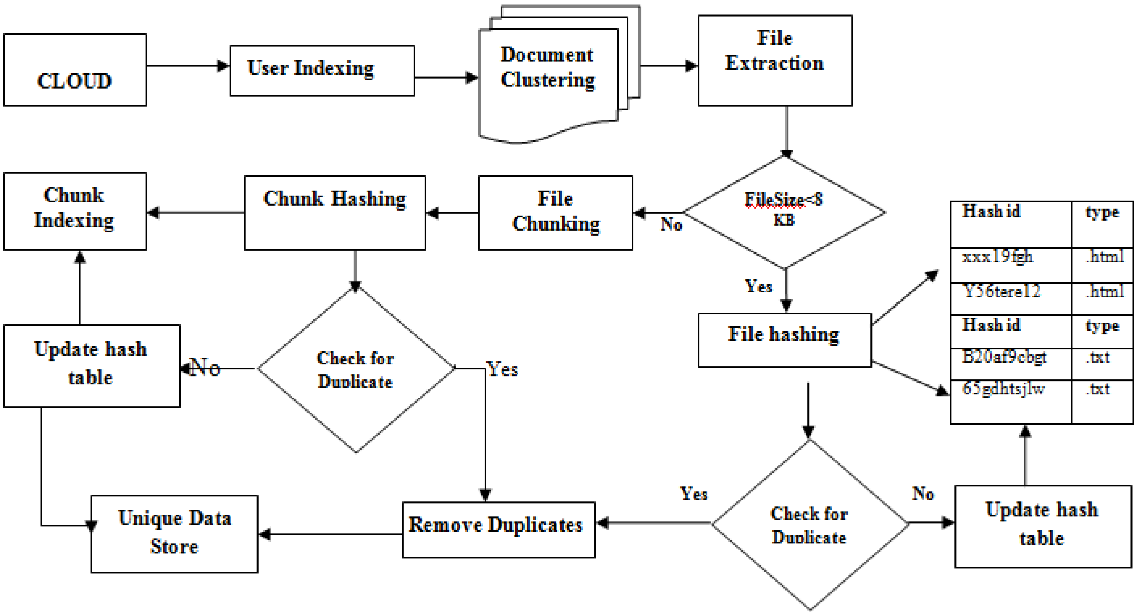
4. Proposed Design of work

Data are generated and updated very fast by every user all around the world through any devices in the form of big data. For data reliability and security everyone is need to take backup and restore their information periodically. Since their device capacity is considerably low and are in money motive they don't want to invest more on it. So, they go for third party storage providers called as Cloud(Google, Amazon, Dropbox, Etc.,). In cloud the periodic data backup by users create huge volume and the issues of shortage in storage space. This could be solved by identifying the redundant data and removing it.

In this proposed work the duplicate information is removed from the target level(from where the data is stored). So, it is called as target level De-duplication. In this proposed work, before de-duplication the input data from virtual storage will be pre processed in two levels. Initial level is grouping of data based on author(i.e. user id) , origin and in later the data will be extracted based on its type like .txt, .html, .pdf, etc.

The de-duplication will be performed by two stages. First one is File level and another one is block level de-duplication. For this, we collect data from cloud and do de-duplication based on the file size. If the file size is less than 8KB , the file will be considered as single chunk(block) and a hash value for it will be generated. This hash id is compared with the existing system hash table for duplicate identification. If collision occurs the file will be discarded and a reference to the old one will be created otherwise file will be maintained in the data store and hash table will be updated with new hash value. On the other hand if the file size is greater than 10KB, then the file will be broken into chunks based on the content using chunking based algorithm here the chunks are assigned with unique ID which has been generated by hashing algorithm(SHA-1) to identify the duplicates and also to restore from the data store. Here the chunk IDs are maintained in Index table to map the stored data with chunk IDs.

There is huge number of content based chunking algorithms like BSW and TTTD are available but they are set back with issues like boundary shifting, metadata overhead and chunk size variations. . The available Frequency Based Chunking(FBC) algorithm also not efficient with meta data overhead. The proposed Enhanced FBC uses the statistical approach in identifying the chunk frequencies under each user id from the input stream. This increases the de-duplication gain and produce less number of chunks. This reduces the meta data overhead. The proposed work is demonstrated in the following diagram.



Hash id	type
xxx19fgh	.html
Y56terel2	.html
Hash id	type
B20af9cbgt	.txt
65gdhtsjlw	.txt

Fig 2: Process of De-dupliaction using ExFBC

The below algorithm explains the step by step processing of effective de-duplication using Extended FBC with better **data dedup gain**. When we break a file "S" into 'n' non overlapping chunks S={x_1,x_2,x_3....x_n), each chunk will be assigned with an ID using SHA-1 denoted as ID(x_i). These IDs are maintained in a Meta Table(Index table) to refer the occurrences of chunks. The input file is now mapped with the chunk List as {ID(x_1),ID(x_2),...ID(x_n)}, which is used to rearrange the chunks from data store. If a chunk with same ID is found ,it stores only a single copy the other will be removed. The dedup gain is the difference between the number of duplicated eliminated and the metadata memory required to store the metadata.

```
Algorithm
 Input:  File stream for chunking
 Output: File or Chunks of file without duplicates.
 Data cleaning: Group the file based on its type
// Perform File Size filtering (Prefiltering)
If (Filesize<10KB)
            Perform Whole file chunking.
Elseif (Filesize >10KB & Filetype==.txt||.pdf||.html)
            perform content based chunking();
            Calculate hash();
If(File size<10KB)
              Perform RabinHash();
Else
             Perform SHA1Hash();
If Duplicate status (==1)
              Remove duplicate chunk()
              Update data store with unique chunks
              Update meta data for reference
Stop.
```

Let U(s) be the number of unique chunks to be U(s)= unique{x_1,x_2...x_n}of size X=|U(s)|. For any chunking algorithm A the dedup gain will be as follows:

$$\text{gain}_A(s) = \text{folding}_{\text{factor(s)}} - \text{metadata(s)} \quad \text{------ (1)}$$

$$\text{folding}_{\text{factor(s)}} = \sum_{xi \subseteq \text{unique(s)}} |x_i| \,.\, (f(x_i)-1) \quad \text{------ (2)}$$

From the above(1) the term gain$_A$(s) denotes the gain by removing the duplicates from the data store and the term folding $_{\text{factor(s)}}$ represents the number of repeated copies of each unique chunk. In (2) the (f(x_i)-1) represents the frequency of repeated chunks. This is multiplied with the length of the chunk. In (1) the metadata is called as metadata overhead.

5. Conclusion

Every internet based transactions are periodically taken backup and restore for future reference. It contains redundant information that to be reduced to optimize the storage requirements. The most of the available storage spaces are occupied by redundant information. It could be solved by removing the duplicate information periodically. The coarse grained chunking of files provide better de-duplication gain. The BSW and TTTD algorithms are not that much efficient in dedup gain ratio as well with chunk size optimization, the proposed ExFBC gives better dedup gain ratio also reduce the meta data overhead by minimizing the number of chunks and its hash table size.

References

[1] MIT 6.006: Introduction to Algorithms 2011- Lecture Notes -Rolling Hash (Rabin-Karp Algorithm).

[2] M.O. Rabin. Fingerprinting by random Polynominals, Centre for Research in computing technology. Harward University, Cambridge.

[3] K. Eshghi and H.K. Tang . A Framework for Analyzing and Improving Content-Based Chunking Algorithms. Hewlett-Packard Labs Technical Report, TR 2005-30. URL: http://www.hpl.hp.com/techreports/2005/HPL-2005-30R1.html

[4] A. Muthitacharoen, B. Chen, and D. Mazieres, A low-bandwidth network file system. In Symposium on Operating Systems Principles, 2001, page 174-187, 2001.

[5] Ying Li, Katherine Guo, Xin Wang, Emina Soljanin, Thomas Woo," SEARS: Space Efficient And Reliable Storage System in the Cloud"

[6] P.Neelaveni, M. Vijayalakshmi," A survey on Deduplication in Cloud Storage",Anna University Chennai, Asian Journel of Information Technology 13(6), 2014.

[7] M. Shyamala Devi, V. Vimal Khanna, and A. Naveen Bhalaji, "Enhanced Dynamic Whole File De-Duplication (DWFD) for Space Optimization in Private Cloud Storage Backup", International Journal of Machine Learning and Computing, Vol. 4, No. 4, August 2014

[8] Bing Chun Chang, "A Running Time Improvement for Two Thresholds Two Divisors Algorithm",Master's Projects, ,2009.

[9] G.Lu, Y.Jin, D.H.Du,"Frequency Based Chunking for Data de-Duplication", In Proceedings of the 2010, IEEE International Symposium on Modeling, Analysis and Simulation of computer and Telecommunication Systems, MASCOTS'10, pages 287-296, Washington, DC, Computer society,USA,2010.

[10] Wenfei Fan, Floris Geerts,"Foundations of Data Quality Management",Synthesis Lectures on data management,ISBN:9781608457779, 2012.

[11] http://www.webopedia.com/TERM/D/data_deduplication.html.

[12] Steven snyder,"The Basics of Deduplication: Data Type, Chunk Size, Source/Target, Re-hydration", http://www.storagecraft.com/blog/basics-deduplication,26 Feb'2015.

AUTHORS

1. **K.GEETHA**, is currently working as Guest Lecturer in Computer Applications Department, Govt. Arts College (Autonomous), Salem-07, TamilNadu, INDIA. She is handling both UG and PG classes. She received her M.Phil. in Computer Science from PRIST University, Thanjavur. She also has completed State Level Eligibility Test for Assistant professors. She guiding PG students for the academic project completion. She is also pursuing Ph.D., in Computer Science under Periyar University, Salem-11. She presented three papers in International, National and State level conferences. Her area of interest is DataMining in virtual Environments.

2. **Dr.A.VIJAYA KATHIRAVAN** is working as an Assistant Professor and Head in Computer Applications Department, Govt. Arts College (Autonomous), Salem-07, TamilNadu, INDIA. She received her M.Phil. in Computer Science from Bharathiar University, Coimbatore and she awarded her doctoral degree in Computer Applications from University of Madras, Chennai. She has published 6 Books, 3 papers in National Journal, 30 papers in International Journal, 35 Papers in National Conference Proceedings, 38 Papers in International Conference Proceedings and a total of 112 publications. Her research interests include data structures and algorithms, data/text/web mining, search engines, web communities, social network mining, machine learning, Natural Language Processing, Organizational leadership and human resource management.