© 2016 IJSRSET | Volume 2 | Issue 3 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Survey of Research on Chunking Techniques

Harshita Sharma
Department of Computer Science and Engineering, UIET KUK, , Kurukshetra, Haryana, India

ABSTRACT

The explosive growth of data produced by different devices and applications has contributed to the abundance of big data. To process such amounts of data efficiently, strategies such as De-duplication has been employed. Among the three different levels of de-duplication named as file level, block level and chunk level, De-duplication at chunk level also known as byte level is the most popular and widely deployed. Many chunking techniques are also available which are categorised as Whole File Chunking, Fixed Size Chunking (FSC) and Content Defined Chunking (CDC). The objective of this paper is to analyse the performance of different existing chunking techniques based on their characteristics. In this study the significance of each technique provides insight to enable researchers understand and select a technique for their research.

Keywords : Deduplication, Chunking, Boundary shift problem, Deduplication Ratio

I. INTRODUCTION

Today, deduplication has become very common n well known technique for space saving. It involves removal of redundant data by saving only one copy of input data stream. The input data can be in different forms such as structured data, semi-structured data and unstructured data. The process of redundancy removal involves chunking, hashing, index lookup and writing. Chunking is the technique of splitting data streams into chunks of non-overlapping data blocks. The data blocks can be of fixed size and variable size depending on chunking technique used. The chunking techniques have been categorized as Whole File Chunking (WFC), Fixed Size Chunking (FSC) and Content Defined Chunking (CDC). Whole file chunking is the simplest and fastest, but shows worst results regarding de-duplication ratio (DER). The Fixed size chunking method is used in case of fixed data blocks and the DER totally depends on what the fixed size is. The smaller the fixed size is, the better DER has. Boundary Shift Problem is the most important issue of these two chunking methods. A common method used to produce chunks of variable size is CDC which is also known as Variable Size Chunking (VSC). CDC determines chunk boundaries in the content by threshold breakpoints. Hence, it allows data modifications with most of the chunks remain unchanged preventing boundary shift problem. In hashing phase, hashing techniques such as MD5 and SHA1 is applied to the chunks produced by chunking phase to provide a unique identity to each chunk in form of hash value. The lookup table is an index that contains hash values of unique chunks. Index lookup process involves the checking of already stored chunks by comparing the stored hash values with the new hash values generated in hashing phase. The last phase of writing includes the writing of all unique chunks to the data store. This paper focuses on FSC and CDC techniques .Table 1 specifies the basic differences between FSC and CDC.

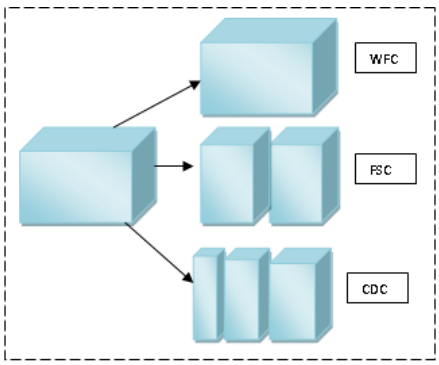

Figure-1. Chunking categories

1. Boundary Shift Problem

The limitation of the both whole file chunking and FSC is Boundary Shift Problem, occurs during data modification. When adding new data or one byte to a file, all subsequent blocks in the file will be rewritten. The rewritten blocks are likely to be considered as different from those in original file, even though most of the data in the file are unchanged. This problem is known as the boundary shift problem.

The paper is organised as follows; Section 2 presents the chunking techniques proposed by researchers along with the datasets on which these are implemented. Section 3 states the approach with their objectives and features. Section 4 concludes the paper with future work.

Table-1 Difference between FSC and CDC

FSC	CDC
Low deduplication	High deduplication
Consumes less time	Time consuming
Vulnerable to byte shifts in data	Robust against insertion of data
No bounds	Upper and lower bounds

II. METHODS AND MATERIAL

2. Datasets used by different chunking techniques

To evaluate performance accuracy researchers validate their notions by testing in experimental environments. For example, a large collection of both real and random datasets were collected to prove the effectiveness of TTTD [1] and BSW [2] technique for various types of datasets. Random files (.txt, .doc) of different sizes were used by the researchers to check the efficiency of the two hashing algorithms MD5 and SHA1 [3]. Three empirical datasets of sizes 0.92GB, 3.37GB and 0.48GB were preferred to evaluate the effectiveness and efficiency of the proposed technique FBC [4] in high and low degree of redundancy. The datasets includes text, images, binaries, video clips and mail repository of software engineers. Random files such as pdf documents of size 80.2 MB, and web image files of 2.6GB with high probability of duplication were used to provide the experimental environment for the testing of IFBC [5]. For byte index chunking [6] and multi-level byte index [7] chunking techniques, 1110MB sized two (.rar) files were used as inputs of the experiments. The effectiveness of multi core chunking MUCH [8] was examined in three different datasets: 2GB ISO image, mixture of different sized files of 200GB and Linux source tree of size 341 MB.10 datasets including office files, pdf documents, music and video files were collected to measure the performance of Leap based CDC [9]. Table 2 specifies datasets employed by researchers to assess the validity of their techniques. Datasets are identified by type of dataset they have.

Table 2- Datasets used by techniques

Chunking method	Name of Chunking technique	Datasets
Variable	TTTD	Large collection of real and random datasets
Variable	BSW	Real and random datasets
Fixed	MD5	Random files
Fixed	SHA1	Random files
Fixed	FBC	Random empirical datasets
Fixed	IFBC	Random datasets
Fixed	Byte index	Real datasets
Fixed	Multi-level byte index	Real datasets
Variable	MUCH	Random datasets

III. RESULTS AND DISCUSSION

3. Performance of chunking techniques

Researchers have done brilliant work by providing chunking techniques such as TTTD, BSW, FBC, Byte-index chunking etc. Every technique has its own working methods of producing chunks with their significances. With the enhancement in the working of chunking techniques, MD5 and SHA-1 calculate unique hash value of each chunk after producing chunks of an input data stream. Hash values are like the identifiers of produced chunks. Byte-index and Multi level Byte-index chunking technique maintain chunk index table of their hash values so as to transfer only unique blocks of data between two nodes. FBC and IFBC use the frequency of chunks for the working of their methods. Multithread chunking apply concurrent chunking techniques to enhance the chunking performance as in Multithread FBC. Table 3 specifies the list of approaches with their feature and description.

Table 3 : List of techniques

Chunking technique	Description	Features
TTTD	Impose maximum size limit on chunk's size called threshold breakpoints.	Stable under modification property.
BSW	Fixed width sliding window moved across the file by making chunk boundaries. Chunk boundaries are determined by the local contents of file. Rabin fingerprint is used for generation of fingerprints of chunks.	Stable under local modifications. Boundaries not affected by the modification
MD5	Produce fixed size chunks by taking input of any size. Encrypt chunks by calculating their hash values	Prevent tampering by generating unique message digest. Faster execution. Length of hash value: 128bit.
SHA1	Verify the integrity of data and encrypt message.	Higher security. Length of hash value: 160bit
FBC	Uses frequency of chunk to eliminate redundant data	50% higher de-duplication than CDC. Produces 2.5~4 time less number of chunks than CDC.
IFBC	Improve two metrics: time and space consuming	Faster than FBC. Improved time and space consuming.
Byte-Index	Provide efficient de-duplication capability with high performance in rapid time. Transfer only non-overlapping chunks of files between client and server.	Reduced speed of file processing. High data de-duplication.
Multi-level byte index	Detect duplicate blocks of data in low bandwidth network. Produce two types of Index table for a file, each chunk sizes are 32KB and 4MB.	More accurate de-duplication rate. Better processing time than other FSC algorithms.
MUCH	Apply content based chunking techniques concurrently to improve chunking performance.	Improved time performance. Reduced computing overhead with same DER.

IV. CONCLUSION

In this paper we have presented a review on chunking techniques in their performances. First we have presented some important key features and differences of the FSC and CDC techniques. We have presented the techniques on the basis of size of chunks and datasets used by the researchers to prove the accessibility of their techniques. The description and their features of techniques are also presented in this paper. In future, researchers can try to reduce the number of chunks with high DER and can improve the time complexity.

V. REFERENCES

[1] KaveEshghi, HsiuKhuernTang,"A Framework for Analyzing and Improving Content-Based Chunking Algorithms", Hewlett-Packard Laboratories, pp. 1-10, February 25, 2005.

[2] A.Muthitacharoen, B.Chen, D.Mazieres,"A low bandwidth network file system",In proceedings of the 18th ACM Symposium on Operating SystemsPrinciples(SOSP'01), pp. 174-187, Chateau Lake Louise,Banff, Canada, October 2001.

[3] Zhenqi Wang, Lisha Cao, "Implementation and comparison of Two Hash Algorithms", International Conference on Computational and Information Sciences, IEEE, pp. 721-725,2013

[4] Guanlin Lu, Yu Jin, David H.C. Du, "Frequency Based Chunking for Data De-Duplication", 18th Annual IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, pp. 287-296,2010

Yunhe Zhang, Weiling Wang Ting Yin, Jiang Yuan,"Novel Frequency Based Chunking for Data Deduplication",Applied Mechanics and Materials Vols. 278-280, pp. 2048-2053,2013

[5] IderLkhagvasuren, Jung Min So, Jeong Gun Lee, Jin Kim, Young WoongKo,"Design and Implementation of Storage System Using Byte-index Chunking Scheme",International Journal of Software Engineering and Its Applications Vol.8, No.1, pp.33-42, 2014

[6] IderLkhagvasuren, Jung Min So, Jeong Gun Lee, Jin Kim, Young WoongKo, " Multi-level Byte Index Chunking Mechanism for File Synchronization"International Journal of Software Engineering and Its Applications Vol.8, No.3 , pp.339-350, 2014

[7] Youjip Won, Kyeongyeol Lim, Jaehong Min," MUCH: Multithreaded Content-Based File Chunking", Transactions on Computers, IEEE, VOL. 64, NO. 5, MAY, pp. 1375-1388, 2015

[8] Chuanshuai Yu, Chengwei Zhang, Yiping Mao, Fulu Li," Leap-based Content Defined Chunking--Theory and Implementation", IEEE,31st Symposium on Mass Storage Systems and Technologies(MSST), pp.1-12, 2015