International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 02, February-2022

A Review Paper on Variable-Size Content-based Chunking

Tawanda Muradzikwa
Harare Institute of Technology

***Abstract*:- Data deduplication has been around for a while now with both companies and individuals looking for ways to save storage (on local machines/clouding computing sites) or bandwidth when required to transfer data over network. In modern day most of the advances have been done in variable-size content-based chunking, which is more effective on identifying duplicate records than fixed-size chunking agreeing to later ponders, and bargains with the issue of boundary-shift during the upload or deletion of files. Since the chunking stage has a direct impact on finding redundancy, Content-Defined Chunking (CDC) algorithm has proved to be more effective on performance and deduplication ratio. Many researchers have and continue to work on ways on how to fully utilize the CDC algorithm. This review will discuss on how various researchers developed their own unique algorithms based on variable-size content-based chunking.**

Key words: Content Defined Chunking, Deduplication, Variable-size content-based chunking

INTRODUCTION

"Deduplication method is a unique information compression strategy to remove the excess information and decrease transmission rate and loading space in the distributed storage frameworks[1]". Deduplication may be occurring either inline or post-process. In in-line the duplication process (hash calculations and lookup) done in the real-time. They used offline deduplication also known as post-process deduplication, it is a process where the whole data is sent to the storage and the deduplication process will be done later after the data has been stored. "[2]. Data deduplication involves three major processes: a) *chunking*, b) *hashing*, and c) *comparing* hashes to recognize redundancy." The process which breaks a dataset into many smaller files is called chunking and it produces chunks."[3]. Chunking is the breaking down of large blocks of data into smaller chunks, chunking improves the storage by storing unique file chunks by comparing it to in coming chunks of file." The keys are mapped through a process called hashing, values into the hash table by using **a** hash function. This enables fast access of elements. Since it provides a lot of benefits when dealing with a huge amount of data, it's no surprise that there are lot of techniques deduplication can be applied.

RELATED WORK

[4]proposed a different approach that balances duplicate elimination by using of large average size and small chunks, "the design is based on two mechanisms which switch from one querying data to the other that are already stored, this is made possible by 2 chunk size targets; the algorithm used small chunks from restricted regions to change duplicate to non-duplicate data, and elsewhere we use large chunks" the algorithms makes use of the ability to make a decision on already stored blocks by replying with their existing queries, in so doing, this makes it fast to perform computations for already stored blocks. Though this algorithm looks to improve the computational time, it is a little more complex than the basic content based chunking which uses unimodal (uses one chunking method) because it uses chunking build up a combination of small chunks and big chunks which gives it its name bimodal but this also gives the bimodal algorithm an advantage of emitting an already existing big chunk. Bimodal deduplication algorithm also struggles with in-line deduplication due to re-chunking big chunks into small chunks that may not be used in future.

Leap-based CDC algorithm was developed as a comparison to an already existing sliding-window-based CDC, leap-based CDC provides significant improvement in deduplication performance without compromising the deduplication ratio. The leap-based CDC algorithm adopts complicated judgment function." The new chunking algorithm with the leap technique in which the executing times of the judgment function are approximately 1/5 of those in the sliding-window-based CDC algorithm. The computation complexity of the judgment function of the new algorithm is less than 2.5 times that of the sliding-window-based CDC algorithm." [5] This method ensured that leap-based CDC algorithm reduced the computational complexity by roughly 50%.the leap-based CDC was able to deal away with the computational overheads problem faced by the sliding-window based CDC even though the complexity of the algorithm is fairly similar. It also seemed as if the experiments, designs and comparisons where treated as if the two algorithms exist in isolation from other technologies.

[6]"we propose a two-stage parallel content defined-chunking or in short SS-CDC which is mainly used for deduplication in storage data, the algorithm allows partially full parallelism on chunking of dataset without affecting deduplication ratio. SS-CDC takes advantage of instruction-level SIMD parallelism which is a technology available in modern day processors." The chunking process is separated into two task, the first one is for rolling window computation that generates potential chunk boundaries, this process is expensive. The second process selects boundaries to meet the minimum and maximum chunk size requirements. The algorithms gain advantage by making use of the parallel computing dependant on the underlying hardware for high chunking speed drawing no negative impact on deduplication ratio. The performance of

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 02, February-2022

SS-CDC can be affected by the underlying hardware which is one of its major draw backs.

Asymmetric Extremum(AE) CDC Algorithm for Fast and Bandwidth-Efficient is a new DE duplicating algorithm which "AE is driven from observing extreme values in asymmetric local range that are not likely to be replaced by a new extreme values to handle the boundaries-shift problem, the whole idea promoted the of AE instead of symmetric as in MAXP local range detects cut off points and simultaneously gain high chunking throughput and low chunk-size variance.[7]" The algorithm was developed to address the large chunking problem that decrease the deduplication ratio and performance degradation that leads to bottlenecks.

[8] came up with an idea of accelerating CDC algorithm to exploit parallelism in data deduplication, "we present P-Dedupe, a the algorithm was pipelined and parallelized data deduplication algorithm called P-dedupe which accelerates the deduplication process by splitting the process into four stages (i.e., chunking, fingerprinting, indexing, and writing), pipelining the above stages with chunks and datasets, CDC and secure hash based fingerprint were paralleled to avoid bottlenecks". The idea to parallelize the upper and lower chunk size was inspired by MapReduce model. Though the algorithm was different to most common content defined chunking algorithms it produces a similar deduplication ratio and has a fast performance, it also manages to alleviate the hash computation bottleneck. It is however expensive in terms of processing power required to run P-Dedu.

[9] took a different approach which the called an unsupervised problem where the algorithm continual generalize the chunking process encompassing fixed and probabilistic chunks, discovery of chronological and causal structures and their recurrent variations. "The algorithm is called SyncMap which is capable of learning and adapting to problem by creating a dynamic map that preserves the correlation between variables." The results show that the system is able to learn almost optimal solutions, despite the presence of many variable and type of structures. It is a new approach to bring neurons learning to deduplicating systems but the system suffers from considering only one to one correlations most of the time. There is still some further work that needs to be done when it comes to dealing with noise problems, hierarchy and causal as well as tasks specific to language processing and image/action recognition.

"Multimodal CDC or in short MCDC splits a file into multiple size ranges and compression ratio into multiple compressibility ranges. It proceeds to make mapping tables which maps size range and compressibility range to its ideal chunk size using datasets. Using this mapping table, Multimodal CDC adapt using new information to reach desired results[10]." The system mainly uses two approaches, the first one work by dividing the data objectives into fixed size blocks and approximates the respective compression ratios using sampling. Then merges them with other adjacent blocks with similar compression ratio there by forming segments. Calculating fingerprints for deduplicating is the final stage. The major difference between the first and second approach is how the two estimate their compression ratio and chunk boundary selection. it generates expected chunks size first in a single scan of buffered data, after choosing one chunking scheme it uses that to calculate compression ratio of the rest of the chunks. However, only the second method was successful before the first method suffered from shifting boundary problems during chunk selection.

[11] developed an algorithm based on content defined chunking using low cost hashing function called byte frequency-based chunking (BFBC). the algorithm outperformed the common CDC algorithm in respect to deduplication elimination ratio. It was also superior to two thresholds two divisors and the basic sliding window. the proposed system defines chunk boundaries based on the bytes frequency of occurrence instead of the byte offset (as in the fixed-size chunking technique), so any change in one chunk will not affect the next one, and the effect will be limited to the changed chunk only. The system uses mathematical functions to generate three hashes that consume fewer computing resources. The algorithm proved to be considerably better than TTD,MD5 and SHA1, with so much novelty it is also normal that BFBC struggles with other aspects to be specific it faces performance degradation when dealing with large datasets due to its complex computing of the divisors

CONCLUSION

In conclusion, every algorithm is unique and takes its own approach to fully utilize variable-size content-based chunking, though they are all successful, there is a lot to be done on future works to achieve a more stable functional algorithm. It's the balance between being lightweight and providing deduplication ratio that is the major driving force behind these novelties, some managed to save more storage and less resources that are needed to run their algorithms these are Byte-Frequency Based-Chunking and Multimodal content based chunking, the former only facing challenges when it comes to large dataset and the latter failing to make its first approach successful but nevertheless their second approach was very successful because of these reasons this is why the research thinks that these two researches are worth a further researches.

REFERENCES

[1] M. Korre, "Security and Data De-Duplication Using Hybrid Cloud Technology," p. 22, 2017, [Online]. Available: https://repository.stcloudstate.edu/msia_etds/22.

[2] H. A. S. Jasim and A. A. Fahad, "New techniques to enhance data deduplication using content based-TTTD chunking algorithm," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 5, pp. 116–121, 2018, doi: 10.14569/IJACSA.2018.090515.

[3] M. K. Yoon, "A constant-time chunking algorithm for packet-level deduplication," *ICT Express*, vol. 5, no. 2, pp. 131–135, 2019, doi: 10.1016/j.icte.2018.05.005.

[4] E. Kruus, C. Ungureanu, and C. Dubnicki, "Bimodal content defined chunking for backup streams," *Proc. FAST 2010 8th USENIX Conf. File Storage Technol.*, pp. 239–252, 2010.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 02, February-2022

[5] C. Yu, C. Zhang, Y. Mao, and F. Li, "Leap-based Content Defined Chunking - Theory and Implementation," *IEEE Symp. Mass Storage Syst. Technol.*, vol. 2015-Augus, 2015, doi: 10.1109/MSST.2015.7208290.

[6] F. Ni, X. Lin, and S. Jiang, "SS-CDC: A two-stage parallel content-defined chunking for deduplicating backup storage," *SYSTOR 2019 - Proc. 12th ACM Int. Syst. Storage Conf.*, pp. 86–96, 2019, doi: 10.1145/3319647.3325834.

[7] Y. Zhang *et al.*, "AE: An Asymmetric Extremum content defined chunking algorithm for fast and bandwidth-efficient data deduplication," *Proc. - IEEE INFOCOM*, vol. 26, pp. 1337–1345, 2015, doi: 10.1109/INFOCOM.2015.7218510.

[8] W. Xia, D. Feng, H. Jiang, Y. Zhang, V. Chang, and X. Zou, "Accelerating content-defined-chunking based data deduplication by exploiting parallelism," *Futur. Gener. Comput. Syst.*, vol. 98, no. March, pp. 406–418, 2019, doi: 10.1016/j.future.2019.02.008.

[9] D. V. Vargas and T. Asabuki, "Continual General Chunking Problem and SyncMap," 2020, [Online]. Available: http://arxiv.org/abs/2006.07853.

[10] J. Wei, J. Zhu, and Y. Li, "Multimodal Content Defined Chunking for Data Deduplication.pdf," no. February, 2014.

[11] C. C. Using and B. Pair, "已读-2020-2区-参考好-基于基于字节对的内容定义分块的重复数据删除系统symmetry," 2020.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)