ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 10, Issue 1, pp: (35-37), Month: January - March 2022, Available at: www.researchpublish.com

Digital Entropy on Improving Data Reduction for Cloud Storage

Dr. Muchelule Yusuf Wanjala [1], Jacob Neyole Misiko[2], Shaaban Bakari[3]

[1,2,3] School of Business & Technology

[1,2,3] Umma University, Kenya

ymuchelule@umma.ac.ke, jneyole434@gmail.com, shabee.bakari@gmail.com

Abstract: **The next generation of storage systems will use new NVMe based devices to store huge amounts of data at a very high price. So, storage users will need to use the new flash media very efficiently to ensure a good return on investment. The efficiency of data reduction is also critical for new generation Cloud Storage as the amount of data explodes impacted by huge data sets used by AI systems. The increase in efficiency can be obtained in different ways but mostly by reducing the size of the data written to the storage arrays using novel deduplication and compression methods. But increased efficiency comes at a price of reduced IO performance of the array due to CPU used for data reduction. So, although the new information theory research defined very efficient data reduction methods, most of these methods are targeting archive and backup storage that are less sensitive to IO performance and IO latency than storage systems that perform data reduction in real time or in-line. But we found out that there are ways to use similar techniques, used for backup storage, but optimize them for IOPS and latency performance and achieve a balance between the data reduction and performance of the arrays.**

Keywords: **cloud storage, data reduction, latency and performance.**

I. INTRODUCTION

In recent work and implementation of data reduction new ideas of improving the efficiency of the algorithms are used, by using augmented standard technique and calculating a simple representation of each non-duplicate data block and evaluate if there are similarities between them. The representations are referred to as resemblance hashes. These are weak hashes of the data blocks with the property that if two same size blocks have the same simple representation, they are likely near-duplicates of each other. There are many pro-posed methods for detecting similarity between data blocks using weak hashes and defining a distance function to decide if there are identical sub-blocks that can be reduced by using variable block deduplication methods.

We propose to use such weak hashes that are fast to compute and with small cache footprints that will allow detection of similarities inside data blocks without full comparison of the block. We will use such hashes that preserve, with higher probability, the similarity functions between fixed size data blocks, based on comparing the Digital Entropy of the blocks. We are considering the Entropy difference between blocks to as a valid fast computing method of distance between hashes of the blocks. We propose to use Entropy of the entire block and prove it is a distance metric and will use it for detecting similarities inside data blocks. We proved its validity as a metric and show that Entropy is a good candidate and we can use parallelized multi-threaded algorithms for Entropy calculations.

II. IS ENTROPY A GOOD METRIC?

We prove that the Entropy is a good measure of similarity between the blocks and sectors of a block. If H be the Shannon Entropy function of a chunk, with respect to a data set it may be defined as follows:

where, Pi is the probability of occurrence of symbol Xi in the data set; N is the total number of unique symbols; and log2 is the base-2 logarithm.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 10, Issue 1, pp: (35-37), Month: January - March 2022, Available at: www.researchpublish.com

We will compute the Shannon Digital Entropy of each 4KB blocks using the following algorithm.

$$H = -\sum_{i=0}^{255} P_i \log_2(P_i)$$

III. COMPUTING ALGORITHM

Compute Entropy E of 4k chunk

If E <= 3.5 then compress the chunk; else

If E> 3.5 compute E1-E8 the entropies of the 8 sectors

If max(E1-E8) <= 2 then compress the chunk

Else start sector level deduplication

We calculate the Entropy of the 512 bytes sectors that indicates if they com-press to less than 64 bytes the deduplication is not needed as the compression reduces the size enough. For sectors compressed to more than 64 bytes, entropy less than 2, there is a higher probability for the sector with higher Entropy to be deduped. Experimental results validated the use of Entropy as a metric is improving the data reduction efficiency by up to 50%. The above algorithm will reduce the number of false positives and the computations to find similarities at sector resolution as well as reduce the digest cache size as we will not store the 8 hashes of the sectors in the database unless their Entropy is above a certain watermark. Of course, these 2 thresholds need additional tuning or use machine learning to adapt to different data sets and variability in time.

IV. CONCLUSION

In this work we introduce a new technique for improving the data reduction for flash storage arrays and cloud storage preserve and improve the performance for real time random IO applications. We introduced a distance metric approach to detect dedupe opportunities in data chunks without analyzing their entire byte content. We introduced a distance metric extending Shannon Entropy to be used to detect similarities between the data blocks. The purpose of this work is to show that inline deduplication without background deduplication with minimal cache and CPU utilization and maximum data reduction efficiency will achieve maximum IOPS performance. We proved that Entropy is a good similarity metric computing efficient tool requiring minimal resource usage and will not require background deduplication.

We assumed that we use SHA-1 hash for deduplication which doesn't reflect the similarities between sectors of the 4k chunks. We also proposed an algorithm for the sector level variable block dedupe and demonstrated that the data reduction rates obtained using this method are higher than for the method using sector deduplication and that the CPU usage is lower and IOPS are higher. We also discussed different hash algorithms that can preserve the similarity index and using Entropy of the hash we can decide to activate variable block dedupe at sector level with relatively low false positive rate. We left for future work to select the cheapest computation wise hash method that will allow to detect similarities between the data blocks and improve performance both latency and IOPS. We will further look to use more accurate similarity detection using sketches and also enabling background deduplication and garbage collection to improve the data reduction for background operations.

REFERENCES

[1] P. Shilane, M. Huang, G. Wallace, and W. Hsu. WAN optimized replication of backup datasets using stream-informed delta compression. In Conference on File and Storage Technologies, 2012.

[2] D. Bhagwat, K. Eshghi, D. D. Long, and M. Lillibridge. Extreme binning: scalable, parallel deduplication for chunk-based file backup. In Proceedings of the 17th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, Sept. 2009.

[3] D. R. Bobbarjung, S. Jagannathan, and C. Dubnicki. Improving duplicate elimination in storage systems. Trans. Storage, 2:424–448, November 2006.

[4] Broder. On the resemblance and containment of documents. In Proceedings of the Compression and Complexity of Sequences, page 21, 1997.

[5] Broder. Identifying and filtering near-duplicate documents. In Proceedings of the 11th Annual Symposium on Combinatorial Pattern Matching, pages 1–10, 2000.

[6] Z. Broder, M. Charikar, A. M. Frieze, and M. Mitzenmacher. Min-wise independent permutations (extended abstract). In Proceedings of the 30th annual ACM symposium on Theory of computing, pages 327–336, New York, NY, USA, 1998.

[7] P. Shilane, M. Huang, G. Wallace, and W. Hsu. Delta Compressed and Deduplicated Storage Using Stream-Informed Locality.

[8] M. O. Rabin. Fingerprinting by random polynomials. Technical report, Center for Research in Computing Technology, 1981.

[9] S. Faibish. SpaceSavingsSonar.py python tool for data reduction simulation DellEMC 2018.

[10] W. Xia, P. Shilane, et al. A Comprehensive Study of the Past, Present, and Future of Data Deduplication. Computer magazine No 7, July 2017.