Survey on Fragmentation for Deduplication in Backup Storage

Reshma A. Fegade[1], R. D. Bharti[2]

[1]ME Computer Student , D.Y.P.I.E.T., Savitribai Phule Pune University, Pimpri, Pune.

[2]Professor, Faculty of Computer Department, D.Y.P.I.E.T, Savitribai Phule Pune University, Pimpri, Pune

Abstract: ***In backup environments field deduplication yields major advantages. Deduplication is process of automatic elimination of duplicate data in a storage system and it is most effective technique to reduce storage costs. Deduplication effects predictably in data fragmentation, because logically continuous data is spread across many disk locations. Fragmentation mainly caused by duplicates from previous backups of the same backup set, since such duplicates are frequent due to repeated full backups containing a lot of data which is not changed. Systems with in-line deduplicate intends to detects duplicates during writing and avoids storing them, such fragmentation causes data from the latest backup being scattered across older backups. This survey focused on various techniques to detect inline deduplication. As per literature, need to develop a focused on deduplication reduce the time and storage space. Proposed novel method to avoid the reduction in restores performance without reducing write performance and without affecting deduplication effectiveness.***

Keywords: Chunking, Deduplication, Fragmentation, Recovery

1. Introduction

In advanced technology's world with variety of application leads to growth of stored digital information, duplicate data is receiving increased attention. From last few years archival and backup systems used automatic removal technique to remove duplicate data and recently have become a characteristic of several storage appliances. Since last few decades deduplication gained great popularity in field of storage. The effectiveness of such method in decreasing both time required completing backups and storage space is necessary to save them. Aside from its write performance, read performance of the deduplication storage has been attaining in importance with a wide range of its utilizations.

Deduplication is defined as a process of automatically eliminating coarse-grained and unrelated duplicate data. Main objective of deduplication is to eliminate both interfile and interfile redundancy over large datasets, stored at different times by uncoordinated users. Traditionally, performance of a deduplication system is measured by the data deduplication ratio, maximal write performance and restores bandwidth. The read bandwidth is usually very good for an initial backup saved to an empty system, but depreciates for subsequent backups. The occurrence of this issue is due to data fragmentation caused by in-line deduplication which results in data reasonably belonging to a recent backup distributed through multiple older backups. Fragmentation mainly caused by duplicates from previous backups of the same backup set, since such duplicates are frequent due to repeated full backups containing a lot of data which is not changed.

The remaining paper can be summed in next section studies some earlier developed methods for in-line deduplication.

2. Related Work

Y. J. Nam, D. Park, and D. H. Du[2], author proposed a novel indicator for dedupe scheme. They proposed scheme provides two-fold approach, first, a novel indicator for dedupe scheme called cache-aware Chunk Fragmentation Level (CFL) monitor and second selective duplication for improvement read performance. The CFL is consists of two parameters: optimal chunk fragmentation and cache-aware current chunk fragmentation.

Whenever the current CFL becomes worse than the demanded one, then selective duplication technique is activated to enhance read performance.

Proposed scheme assures demanded read performance of each data stream while completing its write performance at a practical level and also guaranteed a target system recovery time. Major drawback of selective duplication is that it requires extra memory space called in-memory temp container.

X. Lin, G. Lu, F. Douglis, P. Shilane, and G. Wallace [3], to improve the efficiency of traditional compressors in modern storage systems, author proposed Migratory Compression (MC), a coarse-grained data transformation. Relocation of similar data chunks gathers into one unit, to improve compression factors. After decompression, migrated chunks return to their previous locations. While compressing single files, MC paired with a typical compressor such as grip provides a clear improvement.

Proposed scheme improves compression effectiveness by 11% to 105%, compared to traditional compressors.

Deduplication process is used for eliminating duplicates in data, thus improving the effective capacity of storage systems. Single-node raw capacity is still mostly limited to tens or a few hundreds of terabytes, forcing users to resort to complex. In [4], author proposed new mechanisms called progressive sampled indexing and grouped mark and sweep, to address dedupe challenges and also to improve single-node scalability. Progressive sampled indexing removes scalability limitations by using indexing technique.

Advantages of proposed scheme are, improves scalability, provide good deduplication efficiency and improvement in throughput.

H. L. Goh, K. K. Tan, S. Huang, and C. W. d. Silva [5], author proposed three fold approaches, first they discuss sanitization requirements in the context of deduplicated storage, second implemented a memory efficient technique for managing data based on perfect hashing, third they design sanitizing deduplicated storage for EMC data domain.

Proposed approach minimizes memory and I/O requirements. Perfect hashing requires a static fingerprint space, which conflicts with proposed scheme desire to support host writes during sanitization

Data deduplication has recently gain importance in most secondary storage and even in some primary storage for the storage purpose. A read performance of the deduplication storage has been gaining great significance. In [6], author introduced called Chunk Fragmentation Level (CFL) as an indicator for read performance.

Proposed approach is very effective to indicate the read performance of a data stream with deduplication storage.

H. Dong, F. K. Hussain, and E. Chang, in paper [7], proposed the context based rewriting algorithm which rewrites selected few duplicates. Proposed algorithm recovers restore speed of the latest backups, while resulting in fragmentation increased for older backups, which are rarely used for restore. Main drawback proposed algorithm is that it reduces write performance a little bit.

In this paper [8], author proposed an iDedup, inline deduplication solution for primary workloads. Proposed algorithm is based on two key insights from real world workload that is spatial locality exists in duplicated primary data and second temporal locality occurs in the entrance forms of duplicated data. Proposed algorithm minimizing extra Input/Outputs and seeks.

3. Client Server Architecture

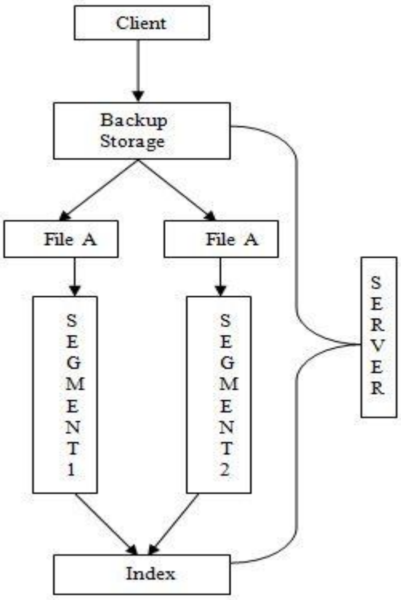

Figure 1: Client-Server Backup Storage [4]

In client server architecture interaction client wants to take backup of storage system on the server side. So clients send request to the server. When server approves client request for backup then connection get formed between client and server. Server acquires backup of client storage and proceed for storage purpose. In this process we should assure that performance measures and restore should be good during client request for restoration. Server should be available and very quick to give access to that backup whenever client or user wants to access data from backup storage on server side. At the server side, server takes backup and proceeds for fragmentation. File level fragmentation technique fragments the backup set into small files so that it will be easy to analysis and manage for storage and restore purpose. Those fragmented files get divided into segments and get stored into containers. Segment index get stored in index module for restore purpose. Duplicate data index value do not get stored, if same data value get detected during hashing that time same data value get referenced to already existing data value index.

Deduplication technique gives following benefits:
i) Read data from files
ii) Stored into segments
iii) Calculate index values
iv) Stores those indices.

Throughput and I/O performance should be good during restoration process. So it is need to develop new techniques which are more capable for changing technology and faster data process with large data set [4].

4. Comparison

Assuring demanded read performance of data deduplication storage with backup datasets , in this paper they proposed scheme provides two-fold approach, first, a novel indicator for dedupe scheme called cache-aware Chunk Fragmentation

Level (CFL) monitor and second selective duplication for improvement read performance. Proposed scheme assures demanded read performance of each data stream while completing its write performance at a practical level and also certain a target system recovery time. Major drawback of selective duplication is that it requires extra memory space called in-memory temp container [2].

Migratory Compression: Coarse-grained Data Reordering to Improve Compressibility in this paper author improved the efficiency of traditional compressors in modern storage systems author proposed Migratory Compression (MC), a coarse-grained data transformation. Proposed scheme improves compression effectiveness by 11% to 105%, compared to traditional compressors. Disadvantage is overhead to perform standalone compression [3].

Building a high performance deduplication system in this paper author proposed new mechanisms called progressive sampled indexing and grouped mark and sweep, to address dedupe challenges and also to improve single-node scalability. Advantages of proposed scheme are, improves scalability, provide good deduplication efficiency and improvement in throughput. The indexing metadata size grows linearly with the capacity of the system [4].

Memory efficient sanitization of a deduplicated storage system in this paper author proposed three fold approaches, first they discuss sanitization requirements in the context of deduplicated storage, second implemented a memory storage efficient technique for managing data based on perfect hashing, third they design sanitizing deduplicated storage for EMC data domain. Proposed approach minimizes memory and I/O requirements. Using perfect hashes requires a static fingerprint space, which conflicts with proposed scheme desire to support host writes during sanitization [5].

Reducing impact of data fragmentation caused by in-line deduplication author proposed the context based rewriting algorithm which rewrites selected few duplicates. Proposed algorithm recovers restore speed of the modern backups, while resulting in fragmentation increased for older backups, which are rarely used for restore. Main drawback proposed algorithm is that it reduces write performance a little [7].

5. Conclusion

This paper presented an all-inclusive survey on the in-line deduplication reduction. The main features, the advantages and disadvantages of each detection technique are described. In backup environments field called deduplication yields major advantages and also results certainly in data fragmentation, because rationally constant data is spread across many disk locations.

As per survey, there is strong need of a fast, robust and innovative method to detect in-line deduplication to reduce time needed to perform backups and storage space required to save them.

.

References

[1] Min Fu, Dan Feng, Member, IEEE, Yu Hua, Senior Member, IEEE, Xubin He, Senior Member, IEEE, Zuoning Chen, Jingning Liu, Wen Xia, Fangting Huang, and Qing Liu, "Reducing Fragmentation for In-line Deduplication Backup Storage via Exploiting Backup History and Cache Knowledge",2015, IEEE Transactions on Parallel and Distributed Systems.

[2] Y. J. Nam, D. Park, and D. H. Du, "Assuring demanded read performance of data deduplication storage with backup datasets," in Proc. IEEE MASCOTS, 2012.

[3] X. Lin, G. Lu, F. Douglis, P. Shilane, and G. Wallace, "Migratory compression: Coarse-grained data reordering to improve compressibility," in Proc. USENIX FAST, 2014.

[4] F. Guo and P. Efstathopoulos, "Building a high performance deduplication system," in Proc. USENIX ATC, 2011.

[5] H. F. C. Botelho, P. Shilane, N. Garg, and W. Hsu, "Memory efficient sanitization of a deduplicated storage system," in Proc. USENIX FAST, 2013.

[6] Y. Nam, G. Lu, N. Park, W. Xiao, and D. H. Du, "Chunk fragmentation level: An effective indicator for read performance degradation in deduplication storage," in Proc. IEEE HPCC, 2011.

[7] M. Kaczmarczyk, M. Barczynski, W. Kilian, and C. Dubnicki, "Reducing impact of data fragmentation caused by in-line deduplication," in Proc. ACM SYSTOR, 2012.

[8] K. Srinivasan, T. Bisson, G. Goodson, and K. Voruganti, "iDedup: Latency-aware, inline data deduplication for primary storage," in Proc. USENIX FAST, 2012.

[9] B. Zhu, K. Li, and H. Patterson, "Avoiding the disk bottleneck in the data domain deduplication file system," in Proc. USENIX FAST, 2008.