Secure Authorized De Duplication for Primary Storage System in Cloud

YERRAVALLI SWATHI[1], H. ATEEQ AHMED[2]
[1] PG Scholar, Dept of CSE, Dr. K. V. Subba Reddy Institute of Technology, Kurnool, AP, India,.
[2] Assistant Professor, Dept of CSE, Dr. K. V. Subba Reddy Institute of Technology, Kurnool, AP, India.

ABSTRACT

Recent studies have shown that moderate to high data redundancy clearly exists in primary storage systems in the Cloud. Our experimental studies reveal that data redundancy exhibits a much higher level of intensity on the I/O path than that on disks due to the relatively high temporal access locality associated with small I/O requests to redundant data. On the other hand, we also observe that directly applying data deduplication to primary storage systems in the Cloud will likely cause space contention in memory and data fragmentation on disks. Based on these observations, we propose a Performance-Oriented I/O DE duplication approach, called POD, rather than a capacity-oriented I/O deduplication approach, represented by iDedup, to improve the I/O performance of primary storage systems in the Cloud without sacrificing capacity savings of the latter. The salient feature of POD is its focus on not only the capacity-sensitive large writes and files, as in iDedup, but also the performance-sensitive while capacity-insensitive small writes and files. The experiments conducted on our lightweight prototype Implementation of POD show that POD significantly outperforms iDedup in the I/O performance measure by up to 87.9% with an average of 58.8%. Moreover, our evaluation results also show that POD achieves comparable or better capacity savings than iDedup. We further extended by adding the concept of data dynamics, Ranking, and providing security to file sharing.

I.INTRODUCTION

Duplication of data in primary storage systems is very common because of the technological traits, which have been driving storage capability consolidation. The removal of duplicate content data at both the document and block stages for enhancing storage area utilization is an active region of research. Certainly, eliminating most duplicate content material is inevitable in ability-sensitive programs including archival storage for cost-effectiveness. However, there exist systems with a mild degree of content similarity in their primary storage consisting of email servers, virtualized servers, and NAS physical devices strolling file and version manage servers. the present data deduplication schemes for primary memory storage, consisting of iDedup and Offline-Dedupe, are potential-orientated, in that they attention on storage capacity cost savings and only select the big requests to deduplicate and pass all the small requests (like 4 KB, 8 KB or much less). The reason is that the small I/O requests best account for a tiny fraction of the data storage capability requirement, making deduplication on them unprofitable and doubtlessly counterproductive considering the enormous deduplication overhead concerned. but, preceding workload research have found out that small files dominate in primary storage systems (extra than 50 percent) and are at the basis of the gadget overall performance bottleneck. Furthermore, due to the buffer impact, primary storage workloads show off apparent I/O burstiness. From a performance angle, the prevailing information deduplication schemes fail to remember those workload characteristics in primary garage systems, lacking the possibility to cope with one of the maximum critical problems in primary storage, that of overall performance. In our proposed work, we introduce a performance-orientated Deduplication scheme, referred to as POD, instead of a capacity-orientated one (like iDedup), to enhance the I/O overall performance of primary storage systems within the Cloud by thinking about the workload traits. POD takes a two-pronged technique to enhancing the performance of primary storage systems and minimizing performance overhead of deduplication, particularly, a request-based selective deduplication method, known as select-Dedupe, to relieve the data fragmentation and an adaptive space management scheme, known as iCache, to ease the memory contention among the bursty read data traffic and the bursty write data traffic.We further extend by adding security

II.RELATED WORK

Study of Past Deduplication Strategies on IO: Most of the recent research that leverage the data deduplication technique to increase the I/O work ability of HDD dependent primary storage structures, the I/O Deduplication scheme specializes in improving the read performance by way of exploiting and constructing more than one duplications on disks to lessen the disk-seek postpone, but does no longer optimize the write request. That is, it makes use of the data deduplication method to locate the redundant data content on disks but does no longer remove them at the I/O route. This allows the disk head to service the read requests with the aid of pre-fetching the nearest blocks from all of the redundant statistics blocks on disk to reduce the seek latency. The write requests are nevertheless issued to disks even if their facts have already been saved on disks. Sequence-dependent deduplication and iDedup are two ability-orientated information deduplication schemes that focus on at primary storage structures by way of exploiting spatial locality to only selectively deduplicate the consecutive document data blocks. They only choose the big requests to deduplicate and forget about all small requests due to the fact the latter best occupy a tiny fraction of the storage capacity. but, preceding research and our workload analysis monitor that the big I/O requests handiest account for a small part of all requests while the small I/O redundancy on the I/O path is massive (Fig. 1). It means that it is the overall performance at the I/O path, instead of ability performance at the storage devices, which stands to doubtlessly benefit more from deduplication for number one storage structures. Furthermore, none of the prevailing research has taken into consideration the hassle of space allocation among the read cache and the index cache. Maximum of them only use an index cache to maintain the hot index in reminiscence, leaving the memory contention trouble unsolved. The iCache in POD is designed to deal with the memory competition and the study amplification problem, despite the fact that the idea of dynamic cache allocation is not generic.

Study on Optimization of Small Writes: commonly speaking, most of the existing small write optimizations for HDD-based primary storage structures make use of the logging approach that buffers the write information in temporal places, which includes disks, non-unstable RAM and flash, and efficaciously flushes them to their unique places ultimately. Parity Logging is brought to overcome the small write hassle of RAID5 through using the logging method. For a write request, the new information block is written in memory area, at the same time as the XOR end result of the very old data block and new information block is recorded sequentially in a log disk. Log-established Array (LSA) and DCD additionally log the updated information at the new disk as opposed to writing it in place to improve the write performance. AFRAID updates the write information immediately, but delays the parity replace operations to the following idle duration among the bursts of the patron activities. It basically sacrifices little reliability for performance improvement. Maximum of the aforementioned schemes aim to relieve the small-write problem of disks or parity-based disk arrays by using delaying the write and parity update operations to the machine idle period. POD, exceptional from them, improves the smallwrite overall performance with the aid of disposing of the write requests when their write facts is already saved on disks. As a result, with POD, many small-write requests can be processed in memory without disk IO operations, which extensively improves the system overall performance.

III.IMPLEMENTATION

we study the working of previous iDedup (Inline Deduplication) technique and also our proposed select-Dedup (Selective Deduplication) with iCache. We present architecture overview of both the schemes and we present the variations of both schemes.

A. iDedup for Minimization of IO workload: iDedup technique is primarily based on key insights from real world workloads: i) spatial locality exists in duplicated primary data statistics; and ii) temporal locality exists within the accessibility patterns of duplicated data. The usage of the primary insight, we selectively deduplicate simplest sequences of disk blocks. This reduces fragmentation and amortizes the seek resulting from deduplication. A second perception allows us to replace the costly, on-disk, deduplication metadata with a smaller, in-memory based cache. Those techniques allow us to trade off ability savings for overall performance, as established in our evaluation with actual-global workloads.

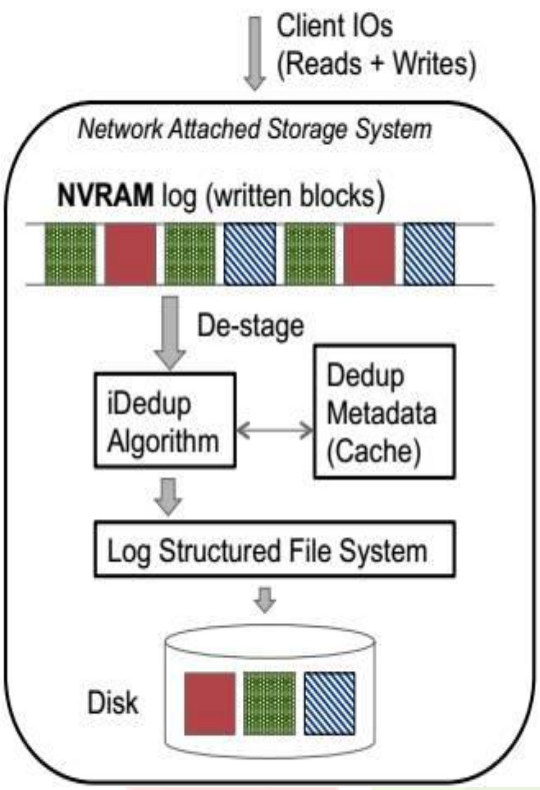

Fig.1. Architecture of iDedup.

As we observe in figure 1, the system makes use of a log structured file management combined with non-volatile RAM (NVRAM) to buffer user writes to decrease response latency. these writes are periodically flushed to disk in the course of the destage section. Allocations of new disk blocks occur at some stage in this segment and are finished successively for each document written. Each and every disk blocks are diagnosed by means of their specific disk block numbers (DBNs). Report metadata, containing the DBNs of its blocks, is saved in an inode format. Given our goal to perform inline deduplication, the newly written (dirty or repeated) blocks want to be deduplicated in the course of the destage section. By way of acting deduplication at some stage in destage, the working system benefits by no longer deduplicating small time-lived information this is overwritten or deleted while the buffered in NVRAM. Adding inline deduplication modifies the write route drastically.

B. POD for Maximizing Performance of Primary Storage and IO workload:

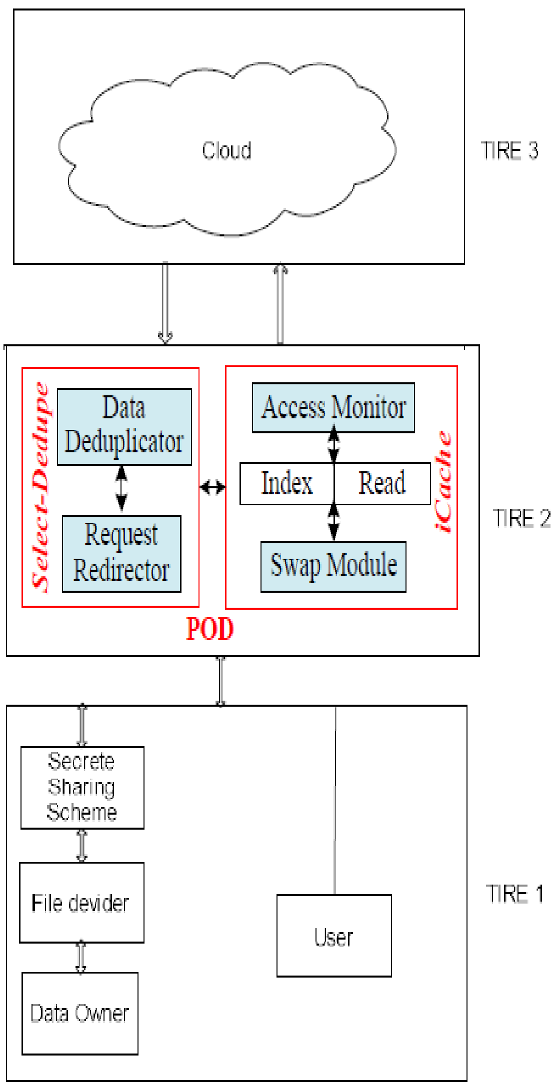

Fig.2. System Architecture

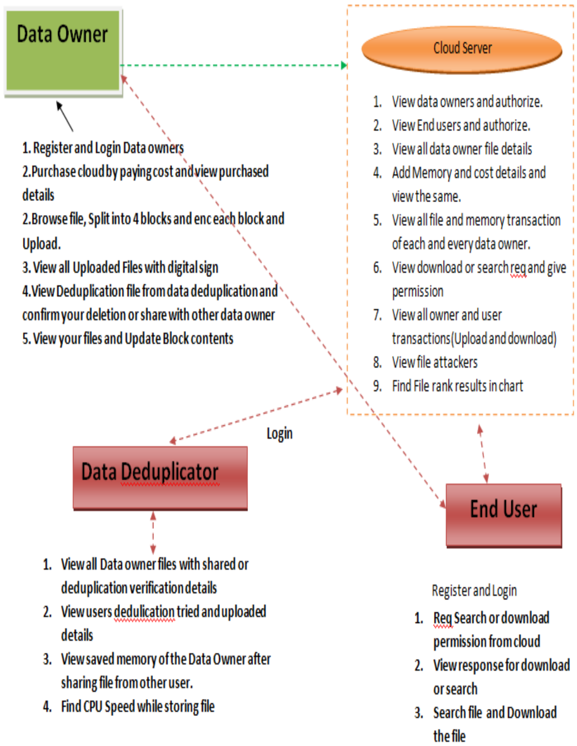

Fig 3.Experimental View

As we see in Fig. 5, POD is rested within the storage node and interacts with the file management systems through the standard read/write interface. As a consequence, POD will be effortlessly incorporated into any HDD-dependent primary storage systems to boost up their system overall performance. Furthermore, POD is impartial of the top file management systems, which makes POD greater flexible and also portable than entire-document deduplication [4] and iDedup. It may be deployed in several of environments, which includes virtual device images which are broadly speaking identical but fluctuate in a few information blocks.

POD has two important components: select-Dedupe and iCache. The request-primarily based select-Dedupe consists of two different modules: data Deduplicator and Request Redirector. The data Deduplicator module is accountable for splitting the incoming write records into data chunks, calculating the hash value of each records chunk, and figuring outwhether an information chunk or block is redundant and famous. According to the information, the Request Redirector module makes a decision whether the write request have to be deduplicated, and keeps data consistency to avoid the referenced information from being overwritten and up to date. The iCache module also includes different modules: access monitor and swap Module. The accessibility monitor module is accountable for monitoring the depth and hit charge of the incoming reading and writes requests. Primarily based on this statistics, the swap module dynamically adjusts the cache space partition among the index cache and read cache. Furthermore, it swaps in/out the cached records from/to the backend storage system. iCache facilitates request-primarily based Select-Dedupe deduplicate as many redundant records

blocks as possible and improves the read performance via increasing the read cache size in face of read bursts.

View of iCache:

The layout of iCache is primarily based on the motive that the I/O workload of primary storage changes often with combined read and write burstiness. As located from the initial consequences, we need to dynamically alter the storagecache area partition among the index cache and examine cache adapting to the traits of consumer accesses to reap the best universal performance

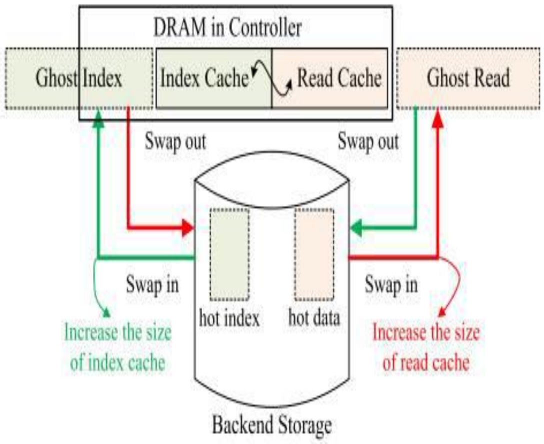

Fig.4. Structure of iCache

To maximize the performance of the storage cache in deduplication based primary storage systems, the form of facts thatoffers the most important performance advantage need to be stored in storage cache. Fig. 4 indicates the structure of iCache. The DRAM length within the storage controller is fixed whilst the index cache size and the read cache length canbe dynamically resized. The most size of an actual cache and its ghost cache is ready to be same to the full size of theDRAM within the garage controller. iCache continues the ghost index and ghost read caches that shop only metadatawhose actual statistics are saved on the back-end storage devices. Whilst a victim information item is flushed from theindex cache or the read information cache, its metadata is inserted into the corresponding ghost cache and the actual datais flushed to the back-end storage tool.

IV.CONCLUSION

We know that data deduplication is most popular technique for data security and reducing data redundancy in memorystorage. We also make use such a convenient technique toward improving primary storage work performance. In theway of our proposed methodology, we introduce our technique referred to as "Performance Oriented Deduplication" by utilizing data deduplication on IO path eliminate the repeating write request while reducing and controlling the memory space. It takes a request-based selective deduplication technique (called select-Dedupe) in order to deduplicate the IO redundancy on the difficult IO path in this sort of manner that it minimizes the data fragmentation trouble. Meanwhile, a smart cache control (known as iCache) is embedded in POD to similarly enhance read overall performance and boom area saving, through adapting to I/O burstiness. Our significant trace-based evaluations prove that POD drastically improves the overall performance and saves capability of primary storage systems within the Cloud. We further improvised by adding the concept of data dynamics, Ranking, and providing security to file sharing.

V.REFERENCES

[1] A. T. Clements, I. Ahmad, M. Vilayannur, and J. Li, "Decentralized deduplication in SAN cluster file systems," in Proc. Conf. USENIX Annu. Tech. Conf., Jun. 2009, pp. 101–114.

[2] K. Jinand and E. L. Miller, "The effectiveness of deduplication on virtual machine disk images," in Proc. The Israeli Exp. Syst. Conf., May 2009, pp. 1–12.

[3] R. Koller and R. Rangaswami, "I/O Deduplication: Utilizing content similarity to improve I/O performance," in Proc. USENIX File Storage Technol., Feb. 2010, pp. 1–14.

[4] D. T. Meyer and W. J. Bolosky, "A study of practical deduplication," in Proc. 9th USENIX Conf. File Stroage Technol., Feb. 2011, pp. 1–14.

[5] K. Srinivasan, T. Bisson, G. Goodson, and K. Voruganti, "iDedup: Latency aware, inline data deduplication for primary storage," in Proc. 10th USENIX Conf. File Storage Technol., Feb. 2012, pp. 299–312.

[6] A. El-Shimi, R. Kalach, A. Kumar, A. Oltean, J. Li, and S. Sengupta, "Primary data deduplication-large scale study and system design," in Proc. USENIX Conf. Annu. Tech. Conf., Jun. 2012, pp. 285–296.

[7] S. Kiswany, M. Ripeanu, S. S. Vazhkudai, and A. Gharaibeh, "STDCHK: A checkpoint storage system for desktop grid computing," in Proc. 28th Int. Conf. Distrib. Comput. Syst., Jun. 2008, pp. 613–624.

[8] D. Meister, J. Kaiser, A. Brinkmann, T. Cortes, M. Kuhn, and J. Kunkel, "A study on data deduplication in HPC storage systems," in Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal., Nov. 2012, pp. 1–11.

[9] X. Zhang, Z. Huo, J. Ma, and D. Meng, "Exploiting data deduplication to accelerate live virtual machine migration," in Proc. IEEE Int. Conf. Cluster Comput., Sep. 2010, pp. 88–96.

[10] J. Lofstead, M. Polte, G. Gibson, S. Klasky, K. Schwan, R. Oldfield, M. Wolf, and Q. Liu, "Six degrees of scientific data: Reading patterns for extreme scale science IO," in Proc. 20th Int. Symp. High Perform. Distrib. Comput., Jun. 2011, pp. 49–60.

[11] V. Vasudevan, M. Kaminsky, and D. G. Andersen, "Using vector interfaces to deliver millions of IOPS from a networked key-value storage server," in Proc. 3rd ACM Symp. Cloud Comput., Oct. 2012, pp. 1–12.

[12] B. Mao, H. Jiang, S. Wu, and L. Tian, "POD: Performance oriented i/o deduplication for primary storage systems in the cloud," in Proc. IEEE 28th Int. Parallel Distrib. Process. Symp., May 2014, pp. 767–776.