International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 10 Issue 6 ǁ 2022 ǁ PP. 1238-1243

De -Duplication to Enhance the Storage System Using File System Object

Mrs. D.SUJEETHA [1st], GOWSHIGA PRIYA L [2nd], HARINI N [3rd] , HARINI DEVI M [4th], INDUJAA SUBRAMANIAM [5th]
1st Assistant Professor, 2nd, 3rd, 4th, 5th UG Scholar(B.E), Department of Computer Science and Engineering, Mahendra Engineering College(Autonomous), Mahendhirapuri.

Abstract
In the explosive growth in data volume, the I/O bottleneck has become an increasingly daunting challenge for big data analytics in the Cloud. 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 relatively high temporal access locality associated with small I/O requests to redundant data. Based on these observations, we propose a performance oriented I/O deduplication, called POD, rather than a capacity oriented I/O deduplication, exemplified by iDedup, to improve the I/O performance of primary storage systems in the Cloud without sacrificing capacity savings of the latter. POD takes a two-pronged approach to improving the performance of primary storage systems and minimizing performance overhead of deduplication, namely, a request-based selective deduplication technique, called Select-Dedupe, to alleviate the data fragmentation and an adaptive memory management scheme, called iCache, to ease the memory contention between the bursty read traffic and the bursty write traffic. Experiments performed on the POD's lightweight prototype implementation show that the POD significantly outperforms iDedup by to 87.9%, averaging 58.8%, in I / O performance measurements. In addition, the evaluation result shows that POD achieves capacity savings equal to or greater than iDedup.
Keywords: *Cloud server, File system object, Data deduplication, iCache.*

Date of Submission: 06-06-2022 Date of acceptance: 21-06-2022

I. INTRODUCTION

Deduplication has received much attention from both academia and industry because it can greatly improves storage utilization and save storage space, especially for the applications with high deduplication ratio such as archival storage systems. Especially, with the advent of cloud storage, data deduplication techniques become more attractive and critical for the management of everincreasing volumes of data in cloud storage services which motivates enterprises and organizations to outsource data storage to third-party cloud providers, as evidenced by many real-life case studies. According to the analysis report of IDC, the volume of data in the world is expected to reach 40 trillion gigabytes in 2020. Today˝s commercial cloud storage services, such as Dropbox, Google Drive and Mozy, have been applying deduplication to save the network bandwidth and the storage cost with client-side deduplication.There are two types of deduplication in terms of the size: (i) file-level deduplication, which discovers redundancies between different files and removes these redundancies to reduce capacity demands, and (ii) block level deduplication,which discovers and removes redundancies between data blocks. The file can be divided into smaller fixed-size or variable-size blocks. Fixed size blocks simplify the calculation of block boundaries. Using variable size blocks (for example, based on the Rabin fingerprint) makes the more efficient in deduplication. Deduplication technology can save storage space for cloud storage service providers, but it reduces system reliability . If such a shared file / part is lost, all files that share that file / part will be unavailable and you will not be able to access the disproportionately large amount of data. If the chunk value is measured with respect to the amount of file data lost if a single chunk is lost, the amount of user data lost when the chunk is corrupted in the storage system is of the chunk. It increases with the number of commonalities. Therefore, ensuring high data reliability in a deduplication system is an important issue. Most deduplication systems prior to were considered only in a single server environment. However, many deduplication and cloud storage systems are designed by users and applications for reliability, so data is especially important for archive storage systems where needs to be retained for extended periods of time. This requires a deduplication storage system that offers comparable reliability to other high availability systems.

II. LITERATURE SURVEY

1. Similarity and Locality Based Indexing for High Performance Data Deduplication- IEEE Transactions on Computers (Volume: 64, Issue: 4, April 2015)

Data deduplication has gained increasing attention and popularity as a space-efficient approach in backup storage systems. One of the main challenges for centralized data deduplication is the scalability of fingerprint-index search. In this paper, we propose SiLo, a near- exact and scalable deduplication system that effectively and complementarily exploits similarity and locality of data streams to achieve high duplicate elimination, throughput, and well balanced load at extremely low RAM overhead. The main idea behind SiLo is to expose and exploit more similarity by grouping strongly correlated small files into a segment and segmenting large files, and to leverage the locality in the data stream by grouping contiguous segments into blocks to capture similar and duplicate data missed by the probabilistic similarity detection. SiLo also employs a locality based stateless routing algorithm to parallelize and distribute data blocks to multiple backup nodes. By judiciously enhancing similarity through the exploitation of locality and vice versa, SiLo is able to significantly reduce RAM usage for index-lookup, achieve the near-exact efficiency of duplicate elimination, maintain a high deduplication throughput, and obtain load balance among backup nodes.

2. Try Managing Your Deduplication Fine-Grained-ly: A Multi-tiered and Dynamic SLA- Driven Deduplication Framework for Primary Storage -2016 IEEE 9th International Conference on Cloud Computing (CLOUD)

Inevitable tradeoff between read performance and space saving always shows up when applying offline deduplication for primary storage. We propose Mudder, a multi-tiered and dynamic SLA-driven deduplication framework to address such challenge. Based on specific Dedup-SLA configurations, Mudder conducts multi-tiered deduplication process combining Global File-level Deduplication (GFD), Local Chunk-level Deduplication (LCD) and Global Chunk-level Deduplication (GCD). More importantly, Mudder dynamically regulates deduplication processes according to instant workload status and predefined Dedup-SLA during runtime. Data deduplication is an efficient technique used for eliminating redundant data, especially when the growth rate of data has far outpaced the dropping rate in hardware cost. Compared to secondary storage, primary storage is commonly characterized as "latency-sensitive" for being constantly and directly accessed by the end-users. A number of deduplication schemes designed or optimized [1], [2], [3] for primary storage emerge in recent years. Be that as it may, satisfactory solutions towards some critical challenges have not been provided by existing deduplication schemes. First, inevitable tradeoff between read performance and space saving makes it a sophisticated task to apply deduplication for distributed primary storage. Most schemes execute only one specific type of deduplication, namely, Global File-level Deduplication (GFD) [4], [5], Global Chunk-level Deduplication (GCD) [6], [7], or Local Chunk-level Deduplication (LCD) [8], [9]. Combinational schemes that deliver more fine-grained deduplication quality for distributed primary storage have rarely been researched. Second, most if not all existing schemes operate in a static mode, executing unchanged deduplication strategy during runtime regardless of the dynamic nature of primary storage workload. In this paper, we propose Mudder, a Multi-tiered and dynamic SLA-driven deduplication framework for primary storage. To begin with, we expand the Dedup-SLA proposed in our previous work [10] by classifying it into two types: latency-oriented (Dedup-SLA-L) and space-oriented (DedupSLA-S) Dedup-SLA, according to opposite preferences on performance/space tradeoff. Afterwards, we respectively establish different multi-tiered deduplication processes for DedupSLAs of both types. We coordinate LCD and GFD for DedupSLA-L to maintain high read performance with acceptable space saving. We combine GCD and LCD for Dedup-SLAS to eliminate as much redundant data as possible while restricting the impact on read efficiency. Meanwhile,

3. HPDV:A Highly Parallel Deduplication Cluster for Virtual Machine Images - 2018 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)

Data deduplication has been widely introduced to effectively reduce storage requirement of virtual machine (VM) images running on VM servers in the virtualized cloud platforms. Nevertheless, the existing state-of-the-art deduplication for VM images approaches can not sufficiently exploit the potential of underlying hardware with consideration of the interference of deduplication on the foreground VM services, which could affect the quality of VM services. In this paper, we present HPDV, a highly parallel deduplication cluster for VM images, which well utilizes the parallelism to achieve high throughput with minimum interference on the foreground VM services. The main idea behind HPDV is to exploit idle CPU resource of VM servers to parallelize the compute-intensive chunking and fingerprinting, and to parallelize the I/O-intensive fingerprint

indexing in the deduplication servers by dividing the globally shared fingerprint index into multiple independent sub-indexes according to the operating systems of VM images. To ensure the quality of VM services, a resource-aware scheduler is proposed to dynamically adjust the number of parallel chunking and fingerprinting threads according to the CPU utilization of VM servers. Our evaluation results demonstrate that

compared to a state-of-the-art deduplication system for VM images called Light, HPDV achieves up to 67% deduplication throughput improvement.

III. EXISTING METHOD

The existing data deduplication schemes for primary storage, such as iDedup and Offline-Dedupe, are capacity oriented in that they focus on storage capacity savings and only select the large requests to deduplicate and bypass all the small requests (e.g., 4KB, 8KB or less). The rationale is that the small I/O requests only account for a tiny fraction of the storage capacity requirement, making deduplication on them unprofitable and potentially counterproductive considering the substantial deduplication overhead involved.

The existing data deduplication schemes fail to consider these workload characteristics in primary storage systems, missing the opportunity to address one of the most important issues in primary storage, that of performance. Existing scheme focuses on improving the read performance by exploiting and creating multiple duplications on disks to reduce the diskseek delay, but does not optimize the write requests. That is, it uses the data deduplication technique to detect the redundant content on disks but does not eliminate them on the I/O path. This allows the disk head to service the read requests by prefetching the nearest blocks from all the redundant data blocks on disk to reduce the seek latency. They only select the large requests to deduplicate and ignore all smallrequests (e.g., 4KB, 8KB or less) because the latter only occupy a tiny fraction of the storage capacity. Moreover, none of the existing studies has considered the problem of space a. Most of them only use an index cache to keep memory, leaving the memory contention problem unsolved

IV PROPOSED SYSTEM

To address the important performance issue of primary storage in the Cloud, and the above deduplication-induced problems, we propose a Performance-Oriented data Deduplication scheme, called POD, rather than a capacity-oriented one (e.g., iDedup), to improve the I/O performance of primary storage systems in the Cloud by considering the workload characteristics. POD takes a two-pronged approach to improving the performance of primary storage systems and minimizing performance overhead of deduplication, namely, a request-based selective deduplication technique, called Select-Dedupe, to alleviate the data fragmentation and an adaptive memory management scheme, called iCache, to ease the memory contention between the bursty read traffic and the bursty write traffic. More specifically, Select-Dedupe takes the workload characteristics of small-I/O-request domination into the design considerations. It deduplicates all the write requests if their write data is already stored sequentially on disks, including the small write requests that would otherwise be bypassed from by the capacity-oriented deduplication schemes. For other write requests, Select-Dedupe does not deduplicate their redundant write data to maintain the performance of the subsequent read requests to these data. iCache dynamically adjusts and swaps these data between memory and back-end storage devices accordingly. 6 The extensive trace-driven experiments conducted on our lightweight prototype implementation of POD show that POD significantly outperforms iDedup in the I/O performance measure of primary storage systems without sacrificing the space savings of the latter. Moreover, as an application of the POD technology to a background I/O task in primary cloud storage, it is shown to significantly improve the online RAID reconstruction performance by reducing the user I/O intensity during recovery.

Reducing small write traffic:
By calculating and comparing the hash values of the incoming small write data, POD is designed to detect and remove a significant amount of redundant write data, thus effectively filtering out small write requests and improving I/O performance of primary storage systems in the Cloud.

Improving cache efficiency :
By dynamically adjusting the storage cache space partition between the index cache and the read cache, POD efficiently utilizes the storage cache adapting to the primary storage workload characteristics.

Guaranteeing read performance:
To avoid the negative readperformance impact of the deduplication-induced read amplification problem, POD is designed to judiciously and selectively, instead of blindly, deduplicate write data and effectively utilize the storage cache.

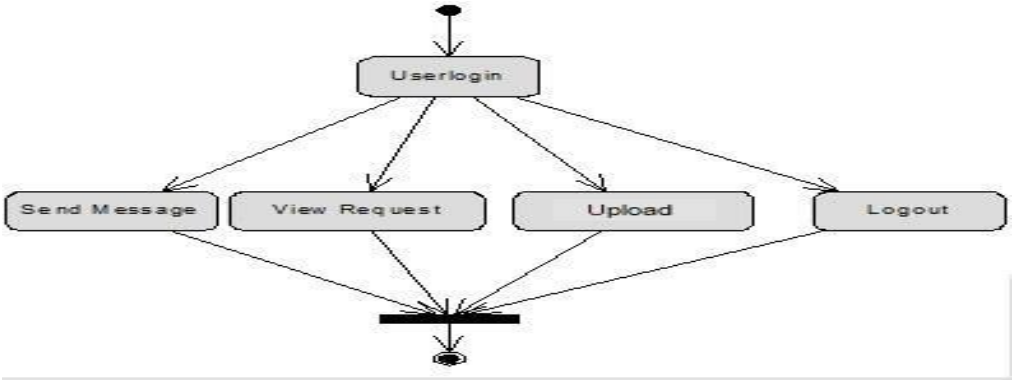

Fig 4.1 Activity Diagram for Sender

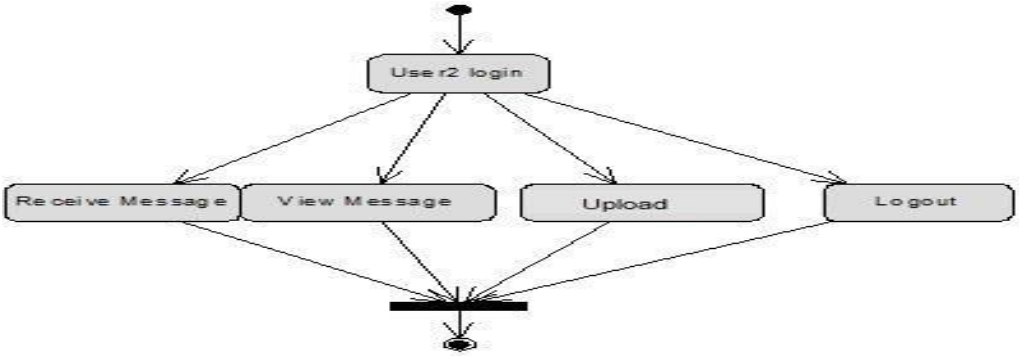

Fig 4.2 Activity Diagram for Receiver

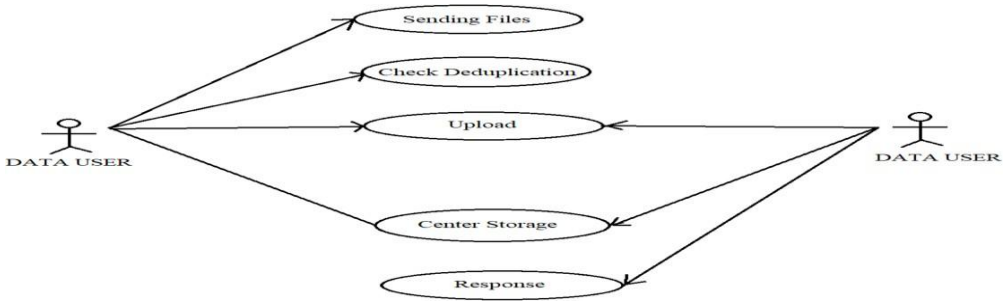

Fig 4.3 Use Case Diagram

V. KEY RESULTS

Fig 5.1 Welcome Page

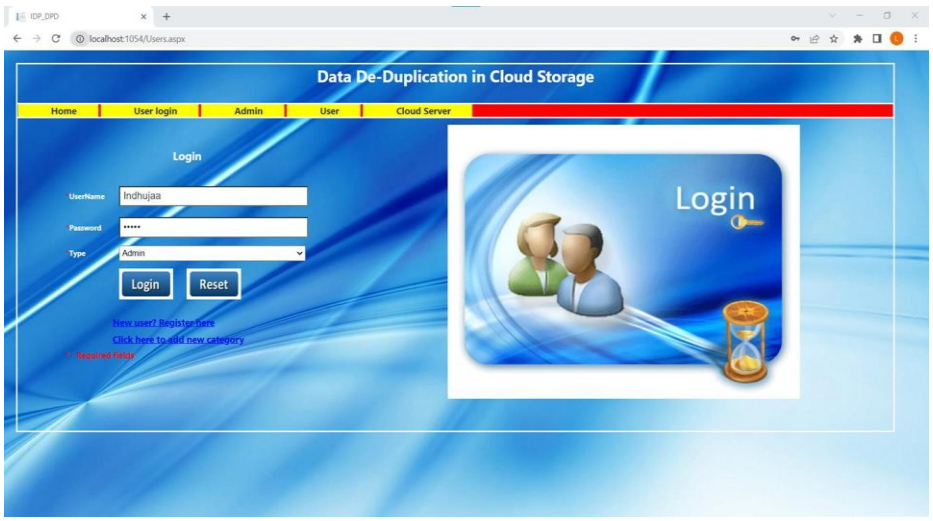

Fig 5.2 Login Page

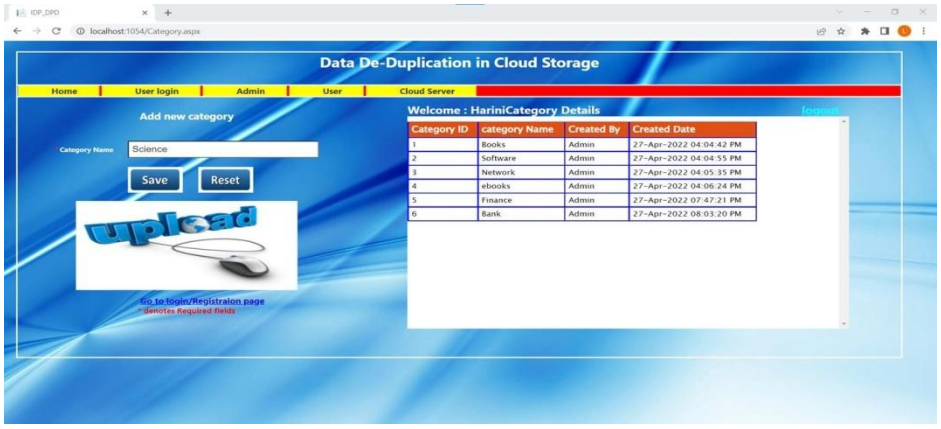

Fig 5.3 File Upload and Categorization

VI. FUTURE ENHANCEMENTS

We pointed out the potential risks of cross-user source based-deduplication. We described how such deduplication can be used as a side channel to reveal information about the contents of files of other users, and as a covert channel by which malicious software can communicate with the outside world, regardless the firewall settings of the attacked machine. Future work includes a more rigorous analysis of the privacy guarantees provided by our mechanism and a study of alternative solutions that maximize privacy while having minimal influence on deduplication efficiency. Furthermore, our observations give motivation to an evaluation of the risks induced by other deduplication technologies, and of cross- user technologies in general. The goal must be to ensure clients that their data remains private, by showing that uploading their data to the cloud has a limited effect on what an adversary may learn about them.

VII. CONCLUSION

Cloud storage using deduplication techniques and their performance and suggests a variation in the index of chunk level deduplication and improving backup performance and Reduce the system overhead, improve the data transfer efficiency on cloud is essential so presented approach on application based deduplication and indexing scheme that preserved caching which maintains the locality of duplicate content to achieve high hit ratio with the help of the hashing algorithm and improve the cloud backup performance. This proposed a novel variation in the deduplication technique and showed that this achieves better performance.

REFERENCES

[1]. M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee,
[2]. D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, „„A View of Cloud Computing,"" Commun. ACM, vol. 53, no. 4, pp. 49-58, Apr. 2010
[3]. A. Katiyar and J. Weissman, „„ViDeDup: An Application-Aware Framework for Video De-Duplication,"" in Proc. 3rd USENIXWorkshop Hot-Storage File Syst., 2011, pp. 31-35.
[4]. D. Bhagwat, K. Eshghi, D.D. Long, and M. Lillibridge, „„Extreme Binning: Scalable, Parallel Deduplication for Chunk Based FileBackup,"" HP Lab., Palo Alto, CA, USA, Tech. Rep. HPL-2009-10R2, Sept. 2009.
[5]. K. Eshghi, „„A Framework for Analyzing and Improving Content Based Chunking Algorithms,"" HP Laboratories, Palo Alto, CA,USA, Tech. Rep. HPL- 2005-30 (R.1), 2005.
[6]. B. Zhu, K. Li, and H. Patterson, „„Avoiding the Disk Bottleneck in the Data Domain Deduplication File System,"" in Proc. 6thUSENIX Conf. FAST, Feb. 2008, pp. 269-282.
[7]. M. Lilli bridge, K. Eshghi, D. Bhagwat, V. Deolalikar, G. Trezise, and P. Camble, „„Sparse Indexing: Large Scale, Inline Deduplication Using Sampling andLocality,"" in Proc. 7th USENIXConf. FAST, 2009, pp. 111-123.
[8]. P. Anderson and L. Zhang, „„Fast and Secure Laptop Backups With Encrypted De-Duplication,"" in Proc. 24th Int"l Conf. LISA,2010, pp. 29-40.
[9]. P. Shilane, M. Huang, G. Wallace, and W. Hsu, „„WAN Optimized Replicationof Backup Datasets Using Stream-Informed Delta Compression,"" in Proc. 10th USENIX Conf. FAST, 2012, pp. 49-64.
[10]. F. Douglis, D. Bhardwaj, H. Qian, and P. Shilane, „„Content-Aware Load Balancing for Distributed Backup,"" in Proc. 25thUSENIX Conf. LISA, Dec. 2011,pp. 151-168.