

# Leveraging Data Deduplication to Improve the Performance of Primary Storage Systems in the Cloud

Bo Mao<sup>\*</sup>, Hong Jiang<sup>§</sup>, Suzhen Wu<sup>\*</sup>, and Lei Tian<sup>§</sup>  
<sup>\*</sup>Xiamen University, <sup>§</sup>University of Nebraska-Lincoln

## Abstract

Recent studies have shown that moderate to high data redundancy exists in primary storage systems, such as VM-based, enterprise and HPC storage systems, which indicates that the data deduplication technology can be used to effectively reduce the write traffic and storage space in such environments. However, our experimental studies reveal that applying data deduplication to primary storage systems will cause space contention in main memory and data fragmentation on disks. This is in part because applying data deduplication introduces significant index memory overhead to the existing system and in part because a file or block is split into multiple small data chunks that are often located in non-sequential locations on disks after deduplication. This fragmentation of data can cause a subsequent read operation to invoke many disk I/O requests, thus leading to performance degradation.

The existing primary data deduplication schemes, such as iDedup[1], are to leverage spatial locality in that they only select the large requests to deduplicate and exclude the small requests (*e.g.*, 4KB, 8KB or less) because the latter only account for a tiny fraction of the storage capacity requirement[2]. Moreover, these schemes tend to overlook the importance of cache management, leading them to manage the index cache and the read cache separately. However, previous workload studies on primary storage systems have revealed that small I/O requests dominate in the primary storage systems (more than 50%) and are at the root of the system performance bottleneck. Furthermore, the accesses in primary storage systems exhibit obvious I/O burstiness. The existing primary-storage data deduplication schemes fail to consider these workload characteristics in primary storage systems from the performance's perspective. We argue that, primary-storage data deduplication schemes should take the workload characteristics of primary storage into the design considerations.

*Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author. Copyright is held by the owner/author(s). SOCC '13, Oct 01-03 2013, Santa Clara, CA, USA ACM 978-1-4503-2428-1/13/10. <http://dx.doi.org/10.1145/2523616.2525939>*

To address the two problems and take the primary-storage workload characteristics into considerations, we propose a Performance-Oriented I/O Deduplication approach, POD, to improving the I/O performance of primary storage systems in the Cloud. POD takes a two-pronged deduplication approach to improve primary storage systems, a request-based I/O and data deduplication scheme, called Select-Dedupe, aimed at alleviating data fragmentation and an adaptive memory management scheme, called iCache, to ease the main memory contention. More specifically, the former 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 excluded 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 takes the I/O burstiness characteristics into the design considerations. It dynamically adjusts the cache space between the index cache and the read cache according to the workload characteristics, and swaps these data between memory and backend storage devices accordingly. During the write-intensive bursty periods, iCache enlarges the index cache size and shrinks the read cache size to detect much more redundant write requests, thus improving the write performance. The read cache size is enlarged to cache more hot read data to improve the read performance during the read-intensive bursty periods.

The prototype of the POD scheme is implemented as an embedded module at the block-device level with the fixed-size chunking method. Preliminary evaluations driven by the real traces conducted on our lightweight POD prototype implementation show that POD significantly outperforms iDedup in improving the performance of primary storage systems in the Cloud.

**Categories and Subject Descriptors** D.4.2 [Operating Systems]: Storage Management; D.4.8 [Operating Systems]: Performance

**Keywords** Performance, Data Deduplication, Cloud

**Acknowledgement:** This work is supported by the NSF of China under No. 61100033, the US NSF Grant No. CNS-1116606, CNS-1016609, IIS-0916859, and CCF-0937993. Most of the work was done while Bo Mao was working at UNL.

## References

- [1] K. Srinivasan, T. Bisson, G. Goodson, and K. Voruganti. iDedup: Latency-aware, Inline Data Deduplication for Primary Storage. In FAST'12, Feb. 2012.
- [2] D. Frey, A. Kermarrec, and K. Kloudas. Probabilistic Deduplication for Cluster-Based Storage Systems. In SOCC'12, Nov. 2012.