Accelerating Data Deduplication by Exploiting Pipelining and Parallelism with Multicore or Manycore Processors

Wen Xia [†,‡,*] Hong Jiang [‡] Dan Feng [†] Lei Tian [‡]

[†] Huazhong University of Science and Technology, Wuhan, China
[*] Student, email:wx.hust@gmail.com
[‡] University of Nebraska-Lincoln, Lincoln, NE, USA

1 Introduction

As the amount of the digital data grows explosively, Data deduplication has gained increasing attention for its space-efficient functionality that not only reduces the storage space requirement by eliminating duplicate data but also minimizes the transmission of redundant data in data-intensive storage systems. Most existing state-of-the-art deduplication methods remove redundant data at either the file level or the chunk level (e.g. Fixed-Sized Chunking and Content-Defined Chunking). But the four stages of the traditional data deduplication process, chunking, fingerprinting, indexing, writing the metadata & unique data chunks, are time consuming in storage systems, especially the processes of chunking and fingerprinting take up significant CPU resources.

Since the computing power of single-core stagnates while the throughput of storage devices continues to increase steadily (e.g. flash and PCM), the chunking and fingerprinting stages of deduplication are becoming much slower than the writing stage in real-world deduplication based storage systems. More specifically, the Rabin-based chunking algorithm and the SHA-1- or MD5-based fingerprinting algorithms all need to compute the hash digest, which may lengthen the write process to an unacceptable level for the required write speed in high performance storage systems.

Currently, there are two general approaches to accelerating the time-consuming hash calculation and alleviating the computing bottleneck of data deduplication, namely, software-based and hardware-based methods. The former refers to employing a dedicated co-processor to minimize the time overheads of computing the hash function so that the deduplication-induced storage performance degradation becomes negligible or acceptable [1, 2, 4]. A good example of the hardware-based methods is called StoreGPU [5] that makes full use of the computing power of the GPU device to meet the computational demand of the hash calculation in storage systems. The software-based approaches exploit the parallelism of data deduplication instead of employing faster computing devices. Liu et al. [5] and Guo et al. [3] have attempted to improve the write performance by pipelining the Fixed-Sized Chunking (FSC) process. Because of the internal content dependency, it remains a challenge to fully exploit the parallelism in the chunking and fingerprinting tasks of the Content-Defined Chunking (CDC) based deduplication approaches.

In this report, we propose P-Dedupe, a deduplication system for high performance data storage that pipelines and parallelizes the compute-intensive deduplication processes to remove the write bottleneck. P-Dedupe exploits the pipelining among the deduplication data units (e.g. chunks and files) and parallelism among the deduplication functional units (e.g. the fingerprinting and chunking tasks) by making full use of the idle computing resources in a multicore- or manycore-based computer system. P-Dedupe aims to remove the time overheads of hashing and shift the deduplication bottleneck from the CPU to the IO, so as to easily embed data deduplication into a normal data storage system with little or no impact on the write performance.

2 Pipelining and Parallelizing

While hash calculations for deduplication are time consuming and CPU-intensive, modern computer systems based on multicore or manycore processor architectures are poised to provide increasingly more CPU resources. On the other hand, our study of the deduplication process indicates that the deduplication process can be viewed and organized in terms of data units (such as chunks and files) and functional units (such as chunking, hashing, indexing and writing, etc.) that are independent of one another. Thus we can fully utilize the idle CPU resources in multicore- or manycore-based computer systems to pipeline and parallelize the compute-intensive deduplication tasks (i.e., functional units) that are then fed by the deduplication data units, as is shown in Figure 1.

The most challenging issue in parallelizing deduplication task stems from the Rabin-based Chunking (CDC) algorithm that uses a sliding window on the data stream. Since the chunking task cannot be parallelized between two adjacent chunks because of content dependency be-

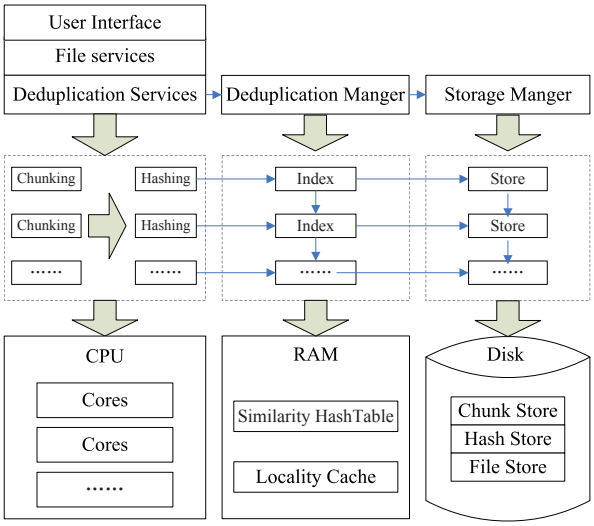

Figure 1: P-Dedupe system architecture. The deduplication pipeline of the four stages are chunking (S1), fingerprinting (S2), indexing the fingerprints in Hash Table and writing chunk data (S3), and writing file metadata (S4).

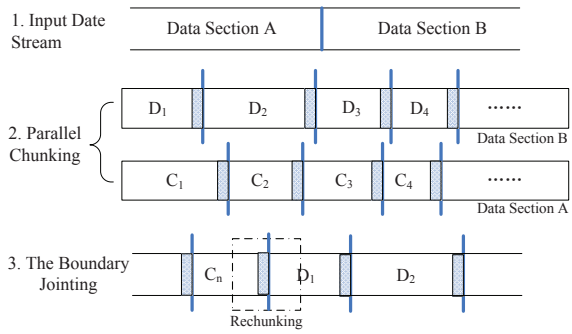

Figure 2: The parallel CDC algorithm runs with two threads. The data stream is divided to Section A and Section B. The boundaries of Sections A and B need to be re-chunked at the end of the chunking process.

tween them, we propose to parallelize on two different substreams of the data stream, which we call data "sections", of appropriate length (e.g., 128KB) that may contain a large number of chunks. P-Dedupe divides a data stream into multiple data sections, where each section is used to chunk a corresponding portion of the data stream and the size of each section must be larger than the maximum chunk size defined in the CDC algorithm. It then applies the CDC algorithm on the data sections in parallel. When the data sections are chunked individually, the boundaries of the sections need some subtle modifications as depicted in Figure 2.

P-Dedupe is an ongoing research project and we are currently exploring several directions and improving its performance with the increasing number of cores.

- Memory and cache management. Booting the performance of parallelizing deduplication with increasing numbers of the cores by retaining the access locality of memory and cache.
- Choices of section size and chunk size. The sizes of chunk and section are important to the efficiency of the deduplication pipeline and parallelism.
- Async & Sync. The asynchronization or synchronization of deduplication-based computing is also the challenging issue with the increasing numbers of cores.

3 Conclusion

In this report, we propose P-Dedupe, an efficient and scalable deduplication system that exploits pipelining and parallelism in deduplication based storage systems. P-Dedupe divides the deduplication tasks into four stages and pipelines the four stages with the data units of chunks and files. P-Dedupe also proposes parallel CDC chunking and fingerprinting algorithms to further remove the hash calculation bottleneck of deduplication in primary storage systems. P-Dedupe's general philosophy of pipelining deduplication and parallelizing hashing is well poised to fully embrace the trend of multicore and manycore processors. With the removal of the performance bottlenecks of data deduplication, it may not be long before data deduplication becomes a necessity in designing file systems for primary storage in addition to backup or archiving systems.

References

[1] CHEN, F., LUO, T., AND ZHANG, X. CAFTL: A content-aware flash translation layer enhancing the lifespan of flash memory based solid state drives. In *FAST11: Proceedings of the 9th Conference on File and Storage Technologies* (2011), USENIX Association.

[2] GHARAIBEH, A., AL-KISWANY, S., GOPALAKRISHNAN, S., AND RIPEANU, M. A gpu accelerated storage system. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing* (2010), ACM, pp. 167–178.

[3] GUO, F., AND EFSTATHOPOULOS, P. Building a High-performance Deduplication System. In *Proceedings of the 2011 conference on USENIX Annual technical conference* (2011), USENIX Association.

[4] GUPTA, A., PISOLKAR, R., URGAONKAR, B., AND SIVASUBRAMANIAM, A. Leveraging Value Locality in Optimizing NAND Flash-based SSDs. In *FAST11: Proceedings of the 9th Conference on File and Storage Technologies* (2011), USENIX Association.

[5] LIU, C., XUE, Y., JU, D., AND WANG, D. A novel optimization method to improve de-duplication storage system performance. In *2009 15th International Conference on Parallel and Distributed Systems* (2009), IEEE, pp. 228–235.