SCALABLE COMPUTING
SOFTWARE LABORATORY

ILLINOIS INSTITUTE OF TECHNOLOGY

Scalable Computing Software Laboratory
Technical Report
Department of Computer Science
Illinois Institute of Technology

Harmonia: An Interference-Aware Dynamic I/O Scheduler for Shared Non-Volatile Burst Buffers

Anthony Kougkas, Hariharan Devarajan, Xian-He Sun Illinois Institute of Technology, Department of Computer Science
{akougkas, hdevarajan}@hawk.iit.edu, sun@iit.edu
November 2017

Technical Report No. IIT/CS-SCS2017-2

http://www.cs.iit.edu/~scs/research/t-reports.html
10 West 31st Street, Chicago, IL 60616

Harmonia: An Interference-Aware Dynamic I/O Scheduler for Shared Non-Volatile Burst Buffers

Anthony Kougkas, Hariharan Devarajan, Xian-He Sun
Illinois Institute of Technology, Department of Computer Science
{akougkas, hdevarajan}@hawk.iit.edu, sun@iit.edu

Abstract—Modern HPC systems are adding extra layers to the memory and storage hierarchy to reduce the I/O performance bottleneck. New hardware technologies, such as NVRAM and SSD, have been introduced and are being used in burst buffer installations to reduce the peak I/O requirements for external storage and deal with the burstiness of I/O in modern scientific applications. In most new supercomputers, these I/O buffering resources are shared between multiple applications that run concurrently. This leads to severe performance degradation due to contention, a phenomenon called inter-application interference. In this paper, we first explore the negative effects of interference at the burst buffer layer and we present two new metrics that can quantitatively describe the slowdown applications experience due to interference. We, then, introduce a new dynamic I/O scheduler, called Harmonia. Our proposed scheduler is aware of interference, adapts to the underlying system, implements a new 2-tier decision-making process and employs several scheduling policies to maximize the system efficiency and applications' performance. Our evaluation shows that Harmonia, through better I/O scheduling, can outperform existing state-of-the-art burst buffer management software and can lead to better resource utilization.

1. Introduction

Modern HPC applications generate massive amounts of data. However, the improvement in the speed of disk-based storage systems has been much slower than that of memory, creating a significant I/O performance gap. In a large scale environment, the underlying file system is usually a remote parallel file system (PFS) with Lustre, GPFS, and PVFS2 being some popular examples. However, as we move towards the exascale era, most of these file systems face significant challenges in performance, scalability, complexity, and limited metadata services, creating the so called I/O bottleneck which will lead to less scientific productivity. To reduce the I/O performance gap, modern storage subsystems are going through extensive changes, by adding multiple levels of memory and storage in a hierarchy. Newly emerging hardware technologies such as High-Bandwidth Memory (HBM), Non-Volatile RAM (NVRAM), Non-Volatile Memory Express devices (NVMe), Solid-State Drives (SSD), and dedicated shared buffering nodes (e.g., burst buffers) have been introduced to alleviate this issue. Several new supercomputers employ such low-latency devices to deal with the burstiness of I/O, reducing the peak I/O requirements for external storage. For example, Cori system at the National Energy Research Scientific Computing Center (NERSC), uses CRAY's

Datawarp technology. Los Alamos National Laboratory Trinity supercomputer uses burst buffers with a 3.7 PB capacity and 3.3 TB/s bandwidth. Summit in Oak Ridge National Lab will also employ fast NVMe storage for buffering, based on the first developer machine already deployed. As multiple layers of storage are introduced into HPC systems, the complexity of data movement among the layers increases significantly, making it harder to take advantage of the high-speed or low-latency storage systems.

Another characteristic of modern supercomputers that contributes to challenges regarding I/O access is multitenancy (i.e., multiple concurrent jobs). Systems like Sunway TaihuLight, the top supercomputer in the Top500 list, have million of cores and run multiple applications concurrently. Due to the sharing of resources such as compute nodes, networks, remote parallel file systems, performance variability is observed. This phenomenon is called inter-application interference and is common in most HPC sites. The interference generally originates from concurrent access by multiple applications to shared resources. While computing and network resources can be shared effectively by state-of-the-art job schedulers, the same cannot be said about the storage resources. In fact, recent research suggests that I/O congestion, within and across independent jobs, is one of the main problems for future HPC machines. A lot of work has been done in PFSs to mitigate the effects of I/O interference. However, with the wider adoption of extra layers in the memory and storage hierarchy like burst buffers, extra care needs to be applied to coordinate the access to this new layer by multiple applications that shared it.

There are several characteristics of burst buffers (and in general I/O buffering nodes) that make scheduling I/O to this new layer quite challenging. First, burst buffer nodes are placed either inside or very close to the compute-node network fabric. The lower access latency and the higher bandwidth of these networks change the way applications perform I/O compared to a traditional remote PFS that uses a slower network. Second, burst buffer nodes are typically equipped with large amounts of RAM and with flash-based storage devices such as NVMe and SSDs. In contrast, a typical PFS server is disk-based. The difference of the medium (i.e., flash-based vs spinning drives) dictates different access concurrency, device bandwidth and latency, sensitivity to random access, and other performance variabilities such as garbage collection and data fragmentation. Third, burst buffers can be used in several ways: as a cache on top of the PFS (e.g., stage-in, stage-out, write-through, write-back), as a fast temporary storage for intermediate results or out-of-core applications (data may or may not need to be persisted), and as an in-situ/in-transit visualization and analysis. These use cases are fundamentally different from a file system where all I/O requests are persisted and strongly consistent. Lastly, burst buffers are presented to the applications through specific allocations and reservations made to the central batch scheduler (e.g., Slurm) whereas I/O access to PFS is performed via a mounting point and interleaved requests

are serviced by the file system scheduler in a first-come-first-serve fashion. The above characteristics constitute traditional PFS I/O schedulers not suitable for this new storage layer and special attention in scheduling the I/O is needed to mitigate the negative effects of inter-application interference.

In this paper, we propose Harmonia (Greek word meaning “in agreement or concord”), a new dynamic I/O scheduler tailored for systems with shared I/O buffering nodes. Harmonia is a scheduler that is interference-aware, operates in a finer granularity, and is adaptive to the current system status. Our proposed 2-tier design allows Harmonia to make scheduling decisions by collecting information from applications (i.e., intention to perform an I/O phase) and the buffering nodes (i.e., available or busy device status) at the same time. We also present a novel metric, called Medium Sensitivity to Concurrent Access (*MSCA*), that captures how each type of buffering medium (i.e., NVMe, SSD, HDD) handles concurrent accesses. This metric helps Harmonia make better decisions when scheduling I/O by tuning the concurrency. We investigate how interference affects the application's execution time and we model this performance degradation by introducing a metric called Interference Factor (I_f). Harmonia uses a dynamic programming algorithm to optimize the scheduling of individual I/O phases on available buffers. By over-provisioning buffer nodes and by overlapping computation with I/O, Harmonia is able to reduce the scheduling cost in several metrics such as maximum bandwidth, minimum stall time, fairness, and buffer efficiency. The contributions of this work are:
a) we introduce two new metrics that model performance degradation due to concurrent accesses and interference (i.e., resource contention),
b) we present the design and implementation of a new burst buffer I/O scheduler called Harmonia,
c) we present three techniques to perform I/O Phase Detection,
d) we propose five new scheduling policies that aim to optimize the performance of the buffering layer, and
e) we evaluate Harmonia’s design and scheduling policies showing that our solution can grant better performance compared to the state-of-the-art buffering platforms.