PADLL: Taming Metadata Burstiness of HPC Jobs Through Application-level QoS Control

Ricardo Macedo, Mariana Miranda, Yusuke Tanimura[†], Jason Haga[†]
Amit Ruhela[★], Stephen L. Harrell[★], Richard Todd Evans[‡], José Pereira, João Paulo
INESC TEC & University of Minho [†]AIST [★]TACC & UT Austin [‡]Intel

1 PROBLEM STATEMENT

Contrary to long-lived assumptions about high-performance computing (HPC) where applications were predominately compute-bound, modern applications (*e.g.,* deep learning training) are data-intensive and generate massive bursts of metadata operations [2]. While these workloads demand scalable, high throughput, and low latency storage, having multiple concurrent jobs submitting large amounts of metadata operations can easily saturate the shared Parallel File System (PFS) metadata resources, leading to I/O contention, overall performance degradation, and I/O unfairness. However, efficiently controlling I/O workflows of large-scale HPC storage systems poses unique challenges of which existing approaches have been unable to address.

Intrusiveness to I/O layers. While existing solutions aim at mitigating I/O contention and variability of shared storage, these are tightly coupled to the implementation of core layers of the HPC I/O stack, including the shared file system and job scheduler. Such an approach requires deep understanding of the system's internal operation model and profound code refactoring, limiting overall maintainability and portability.

Partial visibility and I/O control. Existing solutions that enable QoS control at the compute node level and do not require changes to core I/O layers, actuate in isolation (*i.e.,* agnostic of other jobs in execution), being unable to holistically coordinate the I/O generated from multiple jobs that compete for shared storage, leading to I/O contention.

Metadata remains overlooked. While several works have focused on achieving QoS over data workflows (I/O bandwidth), the metadata counterpart has not received the same level of attention. This is problematic given that several HPC centers are observing a surge of metadata operations in their clusters and expect this to become more severe over time.

2 DESIGN OVERVIEW

In this poster we present PADLL, an application and file system agnostic storage middleware that enables QoS control of I/O workflows in HPC storage systems. It allows system administrators to proactively and holistically control the rate at which POSIX requests are submitted to the PFS. PADLL adopts ideas from Software-Defined Storage [1], following a decoupled design made of a data plane and a control plane. The data plane is a multi-stage component that actuates at the compute node level, and transparently handles applications' requests by intercepting POSIX calls and dynamically rate limiting those that are destined towards the PFS. Stages are then controlled by a hierarchical control plane that defines how workflows should be handled. It acts as a global coordinator with system-wide visibility that continuously monitors and adjusts the I/O rate of each stage. It does so by dynamically allocating storage resources (*i.e.,* metadata rate, bandwidth) to jobs upon workload and system variations.

PADLL enables system administrators to specify QoS policies through control algorithms, which can be as simple as statically rate limiting a specific type of request (*e.g.,* `open`) of a given job, to more complex ones, as achieving proportional sharing of metadata resources across active jobs.

3 RESULTS AND ONGOING RESEARCH

To validate the performance and feasibility of our approach, we implemented a PADLL prototype, including multiple control algorithms to enforce different storage QoS policies, namely *uniform* and *priority-based* rate distributions, *proportional sharing*, and a new *max-min fair share algorithm* that prevents over-provisioning under volatile workloads.

Experiments conducted using traces of metadata operations collected from a production file system of the ABCI supercomputer demonstrate that (1) PADLL enables enforcing complex storage QoS policies over distributed, metadata-aggressive jobs holistically; and (2) when configured with our new control algorithm, it maximizes the use of metadata resources, accelerating the performance of resource-hungry jobs without degrading over-provisioned ones.

Future work. The work presented in this poster opens the path to interesting research directions: (1) design control algorithms that manage conflicting QoS policies; (2) investigate the control plane's scalability and dependability requirements for managing Exascale infrastructures.

REFERENCES

[1] Ricardo Macedo et al. 2020. A Survey and Classification of Software-Defined Storage Systems. *ACM Computing Surveys* 53, 3 (May 2020).

[2] Tirthak Patel et al. 2019. Revisiting I/O behavior in large-scale storage systems: The expected and the unexpected. In *International Conference for High Performance Computing, Networking, Storage and Analysis.*