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[intel], José Pereira, João Paulo

INESC TEC and University of Minho [∅]AIST [*]TACC and UT Austin [intel]Intel

1 PROBLEM STATEMENT

EFFECTIVELY ENSURING STORAGE QOS GUARANTEES IN LARGE-SCALE HPC SYSTEMS IS NOT TRIVIAL:

⊗ Manual intervention

- In HPC facilities, sysadmins manually stop jobs with aggressive I/O behavior
- Reactive approach that is only triggered when concurrent jobs were already harmed

⊗ Intrusive to I/O layers

- Existing solutions are tightly coupled to core layers of the HPC stack (e.g., Parallel File System (PFS), job scheduler)
- Require profound system refactoring

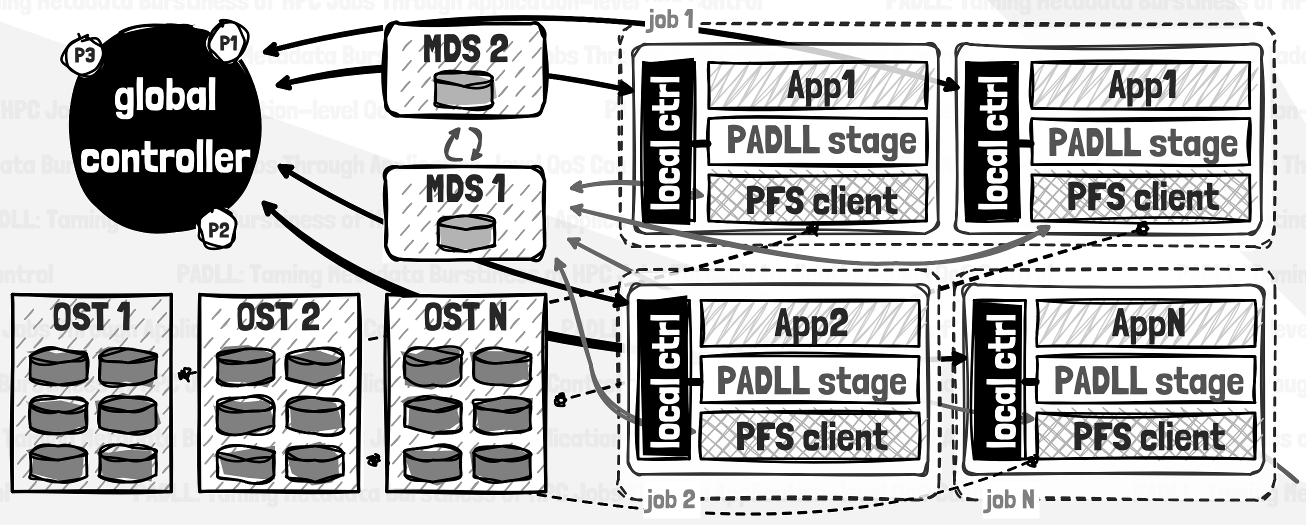
⊗ Partial visibility and I/O control

- Compute node-level solutions actuate in isolation (i.e., agnostic of other jobs)
- Unable to coordinate the I/O generated from multiple jobs that compete for shared storage

⊗ Metadata remains overlooked

- Existing proposals mainly focus on achieving QoS over data workflows
- Metadata operations of a single job can saturate the PFS metadata resources

2 PADLL STORAGE MIDDLEWARE



Software-Defined Storage architecture

- PADLL is organized in multiple data plane stages that differentiate and rate limit I/O workflows, and a hierarchical control plane that manages all stages to ensure storage QoS policies

Application and PFS agnostic

- PADLL sits between applications and the PFS, and does not require changing any core layer of the HPC I/O stack (LD_PRELOAD)
- Compatible with POSIX-compliant storage systems

Fine-grained I/O control

- Classifies, differentiates, and enforces I/O requests with different levels of granularity (e.g., operation type, class, and job)

Global visibility

- Coordinated control of all I/O workflows destined towards the PFS, preventing I/O contention and unfair usage of shared resources

PADLL IS AN APPLICATION AND FILE SYSTEM AGNOSTIC STORAGE MIDDELWARE THAT ENABLES QOS CONTROL IN HPC STORAGE SYSTEMS

3 CONTROL ALGORITHMS

Uniform rate distribution

- Jobs are throttled with a fixed rate throughout their execution, regardless of their size, duration, and workload
- Even distribution of resource shares

Priority-based rate distribution

- PFS resources are distributed based on a given priority
- Can lead to both under-provisioning (e.g., leftover I/O resources) and over-provisioning (e.g., resource shares larger than needed)

Proportional sharing (psharing)

- Traditional max-min fair share control algorithm (feedback loop) that enforces per-job rate reservations (IOFlow, Retro, PAIO)
- Suited for workloads with sustained I/O load, but suboptimal under volatile workloads (over-provisioning)

Proportional sharing without false resource allocation (psfa)

- New max-min fair share algorithm (feedback loop) that ensures storage QoS under volatile workloads
- Assigns resource shares based on the actual I/O usage of each job and their respective metadata demands

4 EXPERIMENTAL TESTBED

Goal

- Limit overall metadata load in the PFS (110 kops/s), while assigning different I/O priorities to jobs

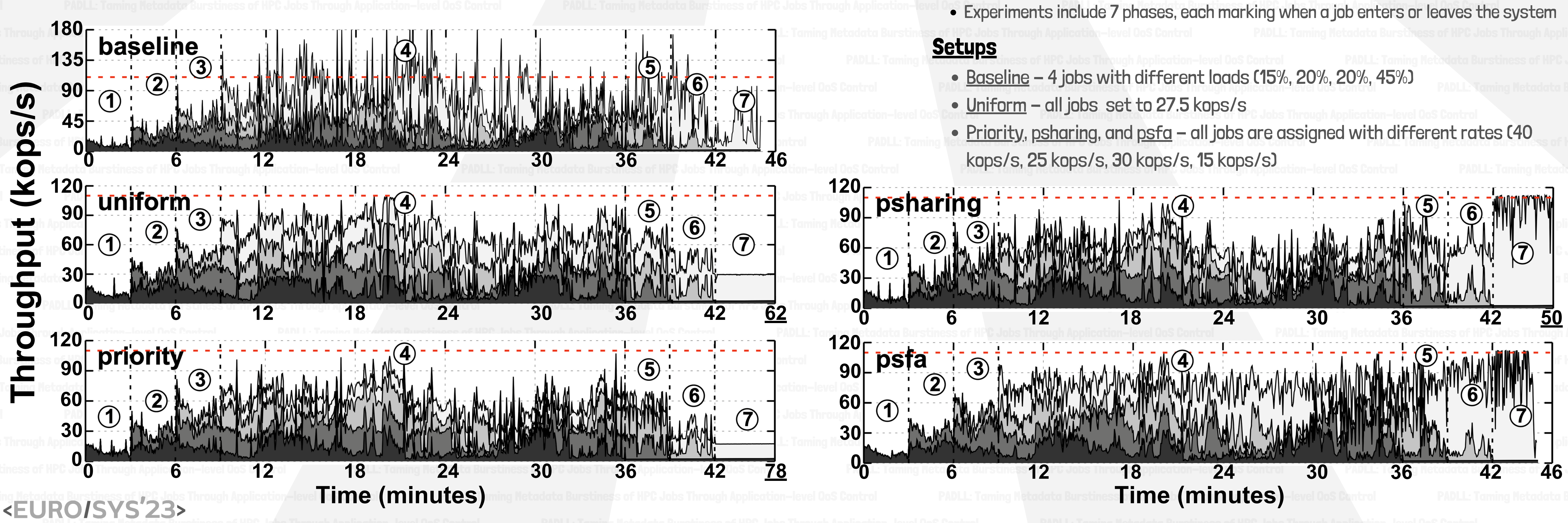
Experimental environment

- Per-job QoS control in the ABCI supercomputer, hosted by AIST
- Metadata traces from a production Lustre file system
- Experiments include 7 phases, each marking when a job enters or leaves the system

Setups

- Baseline – 4 jobs with different loads (15%, 20%, 20%, 45%)
- Uniform – all jobs set to 27.5 kops/s
- Priority, psharing, and psfa – all jobs are assigned with different rates (40 kops/s, 25 kops/s, 30 kops/s, 15 kops/s)

5 PER-JOB METADATA CONTROL



EURO/SYS'23

ACKNOWLEDGMENTS. WE THANK AIST FOR PROVIDING ACCESS TO COMPUTATIONAL RESOURCES OF ABCI. THIS WORK WAS SUPPORTED BY THE PORTUGUESE FOUNDATION FOR SCIENCE AND TECHNOLOGY AND THE EUROPEAN REGIONAL DEVELOPMENT FUND, THROUGH THE PHD FELLOWSHIP SFRH/BD/146059/2019, PD/BD/151403/2021 AND PROJECTS BIGHPC (POCI-01-0247-FEDER-045924 AND UTA18-001217).

dsrhaslab/padll
rgmacedo@inesctec.pt
https://arxiv.org/abs/2302.06418