Diagnosing Parallel I/O Bottlenecks in HPC Applications

Peter Harrington -- University of California, Santa Cruz

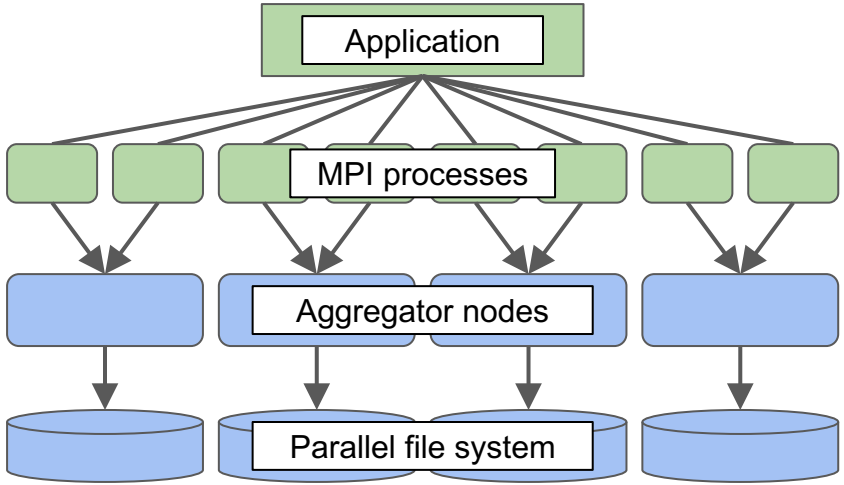
Advisors:
Wucherl Yoo, Alexander Sim, & Kesheng Wu -- Lawrence Berkeley National Laboratory

HPC Parallel I/O

Becoming a significant bottleneck as HPC data volumes grow, and is especially challenging in Adaptive Mesh Refinement (AMR) applications.

Multiple layers of software and hardware:

- Nodes communicate with Message Passing Interface (MPI)
- Aggregator nodes buffer data from MPI processes
- Aggregators write to file system (Lustre Object Storage Target) via POSIX

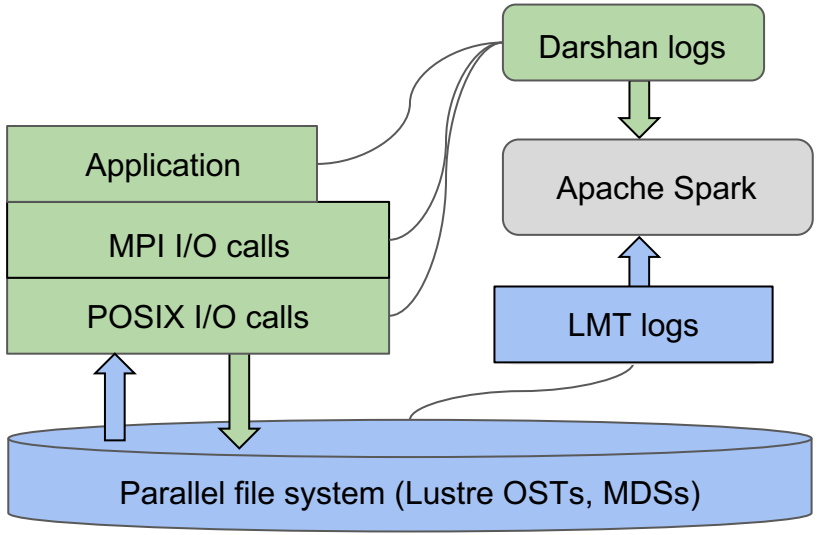


Parallel I/O Performance Analysis

Performance logs exist for some levels of the underlying parallel I/O subsystems but are independent of each other, so analyzing I/O performance remains challenging.

Approach:

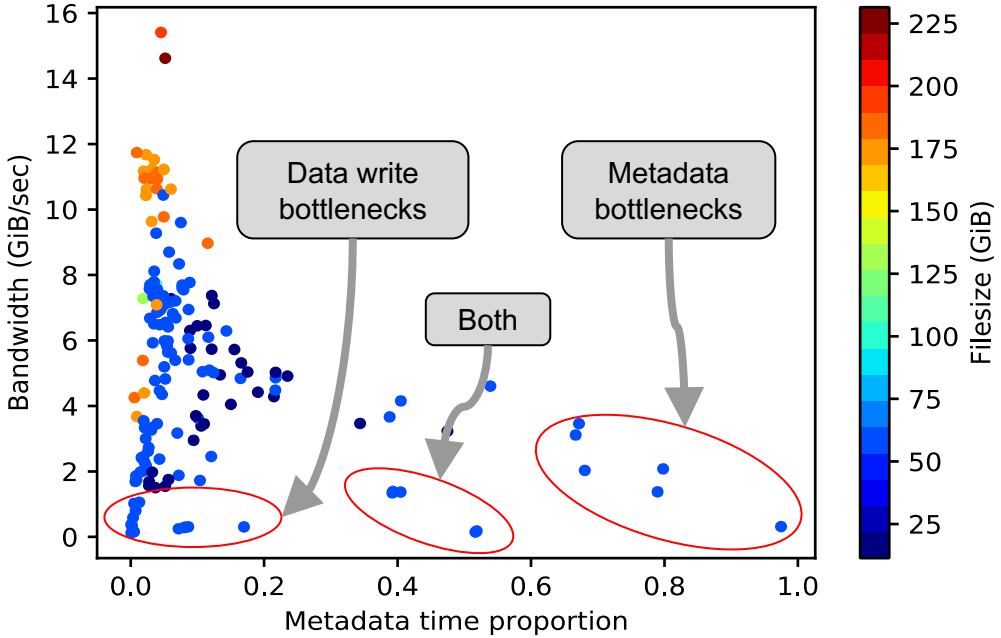
1. Track application-level I/O activity with Darshan profiling tool
2. Track system-wide Lustre file system activity with Lustre Monitoring Tool (LMT)
3. Parse performance logs and analyze in parallel with Apache Spark



Locating and Diagnosing Bottlenecks

Even with optimized file system stripe settings, some runs showed significant slowdowns that caused as much as a 40x reduction in I/O bandwidth.

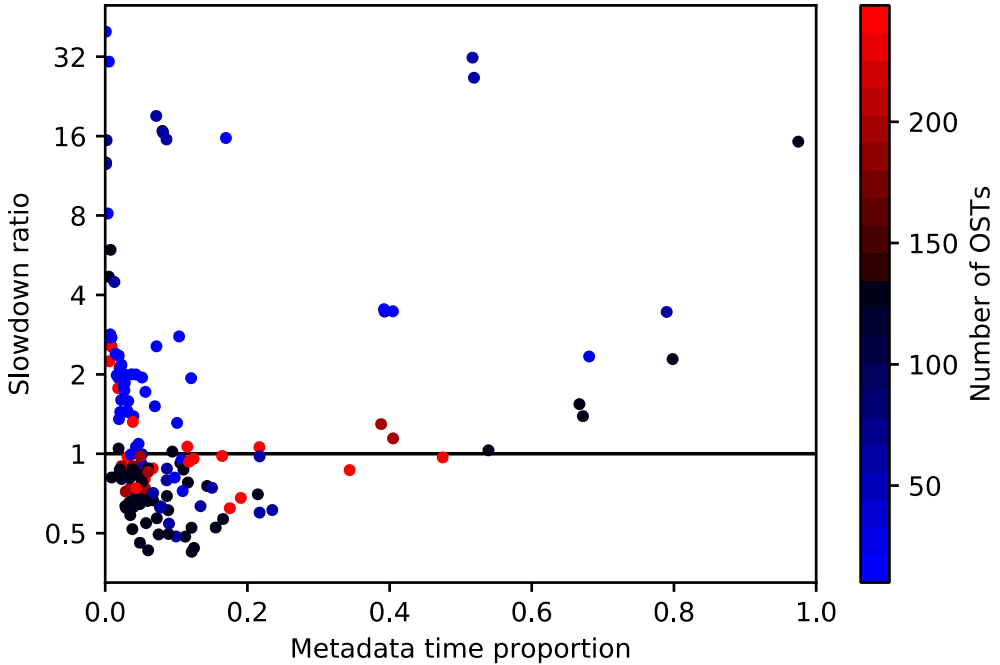
Write bandwidth for runs of the `HyperCLaw` *AMR application , colored by the output file size. The x-axis denotes the proportion of total I/O time consumed by metadata operations.*



Locating and Diagnosing Bottlenecks

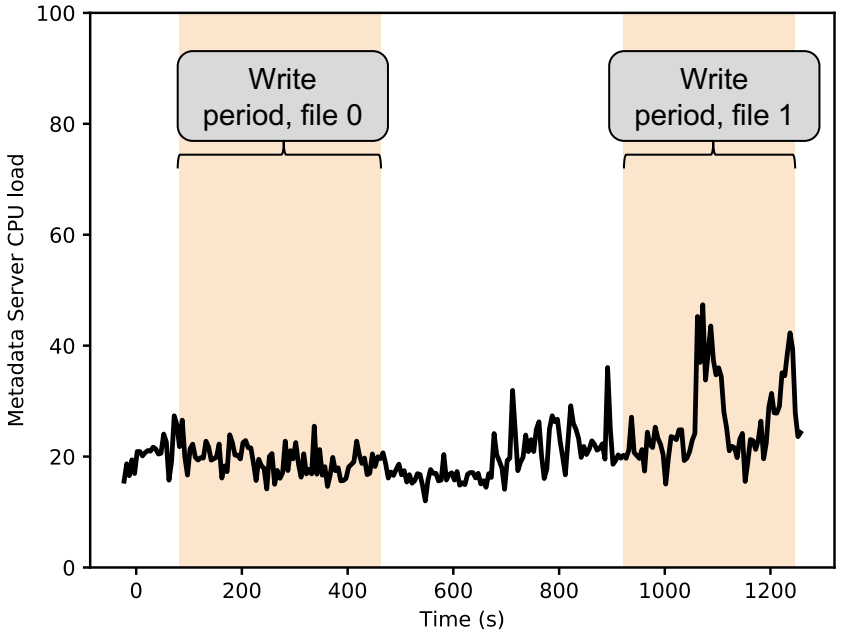
During bottlenecked runs, either metadata or data write operations took much longer than usual, and in some cases both did.

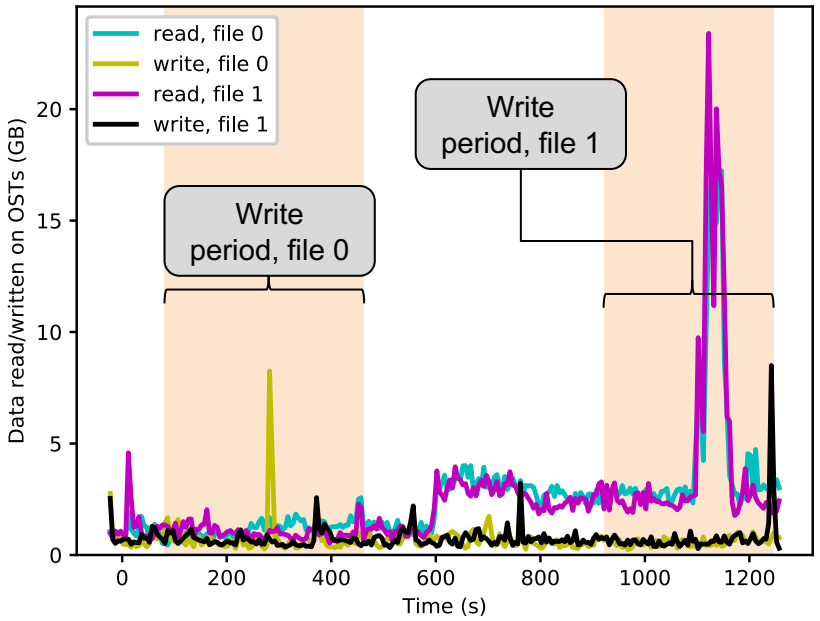
The slowdown ratio of each run's actual bandwidth relative to normal bandwidth (log scale), with respect to the proportion of time spent in metadata operations.



Locating and Diagnosing Bottlenecks

Bottlenecked performance sometimes corresponded with elevated levels of file system traffic, but the limited granularity of the traffic data could not conclusively assign responsibility to specific servers.



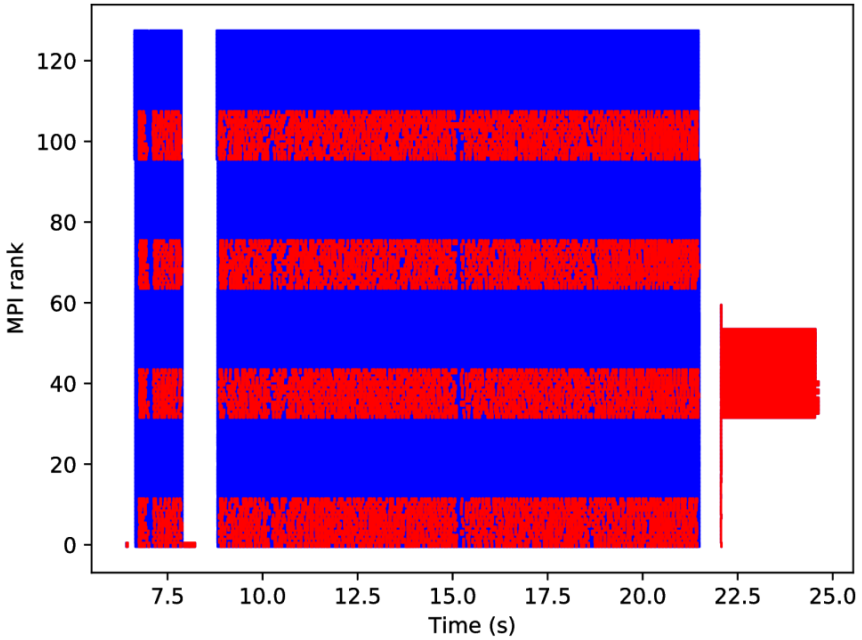


Additional Observations

An additional bottleneck was observed in the applications with HDF5 collective writes, showing several trailing writes that occurred after the main write output phase.

This behavior degraded:

- Efficiency
  Duration of trailing writes lasted up to 5.5 seconds, despite only writing less than 5 kB of data (a small fraction of the total output).
- Scalability
  Time taken by the writes increased when the total file size and number of processes increased, so such behavior does not scale.



Conclusions

- **Slowdowns that reduce I/O bandwidth by up to 40x can occur during metadata operations and/or data writing, and are not the application's fault.**
- **For 90% of runs, metadata operations consumed up to 23% of the total I/O time. Higher proportions of time spent in metadata operations resulted in bottlenecks.**
- **Repeating similar analysis using more detailed file-system measurement data will help ascribe metadata bottlenecks to specific servers within the file-system.**

Acknowledgements

Surendra Byna (LBNL) -- guidance and insights on HPC I/O

Vincent Beckner (LBNL), Ann Almgren (LBNL) -- AMReX `HyperCLaw` application

Brian Van Straalen (LBNL) -- `Chombo` application