Analyzing I/O Patterns and Dataflow in HPC Workflows

Meng Tang Luanzheng Guo Nathan R. Tallent Anthony Kougkas
Xian-He Sun
mtang11@hawk.iit.edu, lenny.guo@pnnl.gov, nathan.tallent@pnnl.gov,
akougkas@iit.edu, and sun@iit.edu

Introduction

Modern scientific discovery is increasingly data-driven, with AI-assisted workflows generating massive, complex data flows across simulations, analyses, and instruments [2]. These workflows rely on shared parallel file systems, which often become performance bottlenecks due to contention and inefficient data movement [1]. Existing I/O-aware schedulers often lack insight into workflow-level patterns like producer–consumer relationships and file reuse, limiting their ability to optimize end-to-end performance. To address this:

- We present a workflow-centric methodology to analyze I/O behavior by linking workflow context with I/O metrics.
- We apply this to two HPC workflows, identifying producer-consumer patterns and bottlenecks.
- We uncover workflow-specific I/O insights and optimization opportunities.

Approach

We combine workflow dataflow DAG with multidimensional visualization to uncover dataflow patterns critical for I/O-aware scheduling. It consists of the following three steps:

1. **Dataflow DAG**: We construct a task-to-file DAG by extracting input/output definitions from a workflow schema (e.g., Pegasus DAX or JSON), capturing data flow between tasks via shared files and highlighting producer–consumer dependencies.
2. **Multidimensional Visualization**: We visualize I/O metrics such as bandwidth and dataflow volume using operation count–centric plots, grouped by tasks, files, and producer–consumer pairs.
3. **Workflow Pattern Characterization and Analysis**: We map core dataflow patterns onto the workflow to enable pattern recognition and guide optimization strategies.

Workflow Centric I/O Metrics

- **Task Group**: Tasks of the same type (could be executed in parallel) are grouped to compare I/O statistics.
- **Producer-Consumer (P-C) Group**: A core metric capturing data dependencies between producer (write) and consumer (read) task groups, enabling insight into I/O behavior, data placement, and task coordination.
- **Workflow**: A set of interdependent tasks coordinated by a workflow system or script, typically run within a batch-scheduled HPC job with data dependencies.
- **Workflow Stages**: Logical units of execution in a workflow, where stage order reflects task and file access sequence.
- **File Group**: Files are grouped by similar types or extensions.
- **Dataflow**: The total amount of data (bytes) transferred between a task and a file.

Pattern	Description
$1 \to 1$	One producer writes data read by one consumer.
$1 \to n$	One producer's data is shared across multiple consumers.
$n \to 1$	Several producers write parts consumed by one task.
$n \to n$	Multiple producers and consumers with shared I/O.

Table 1. Core p-c I/O patterns in workflow interactions.

References

[1] Jean Luca Bez, Suren Byna, and Shadi Ibrahim.
I/o access patterns in hpc applications: A 360-degree survey.
ACM Computing Surveys, 56(2):1–41, 2023.

[2] Harsh Bhatia, Francesco Di Natale, Joseph Y Moon, Xiaohua Zhang, Joseph R Chavez, Fikret Aydin, Chris Stanley, Tomas Oppelstrup, Chris Neale, Sara Kokkila Schumacher, et al.
Generalizable coordination of large multiscale workflows: challenges and learnings at scale.
In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16, 2021.

Workflow Analysis: Seismology

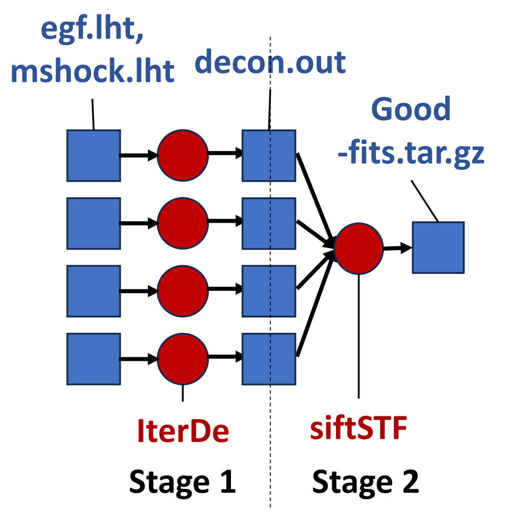

Figure 1: Seismology Dataflow DAG

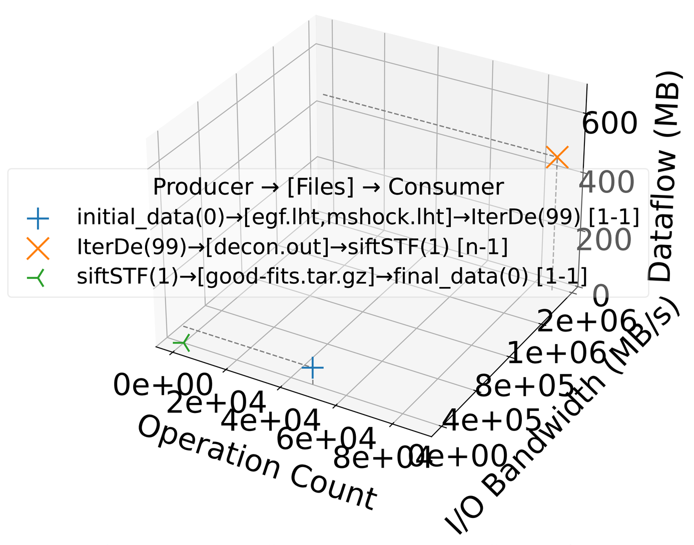

Figure 2: Group by P-C Pairs

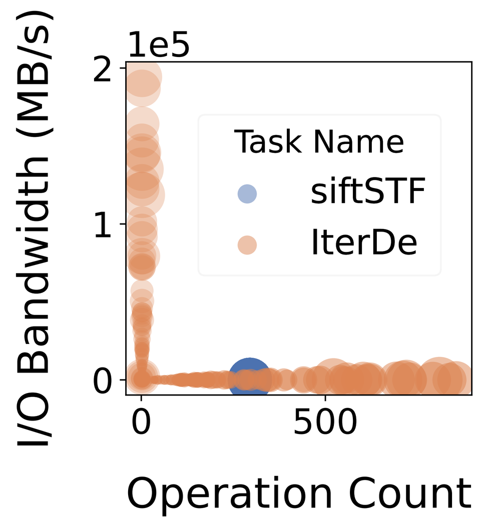

Figure 3: Group by Tasks

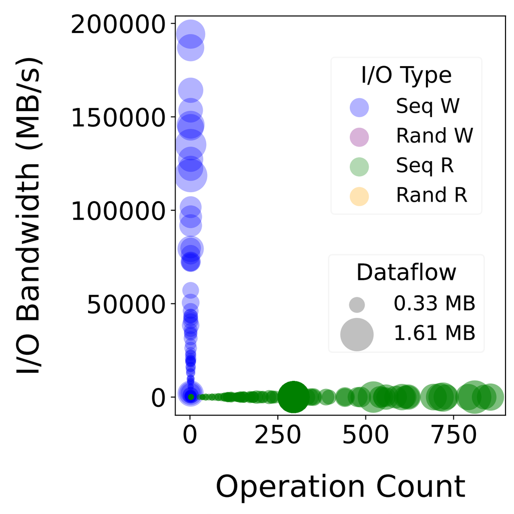

Figure 4: Group by OP Type.

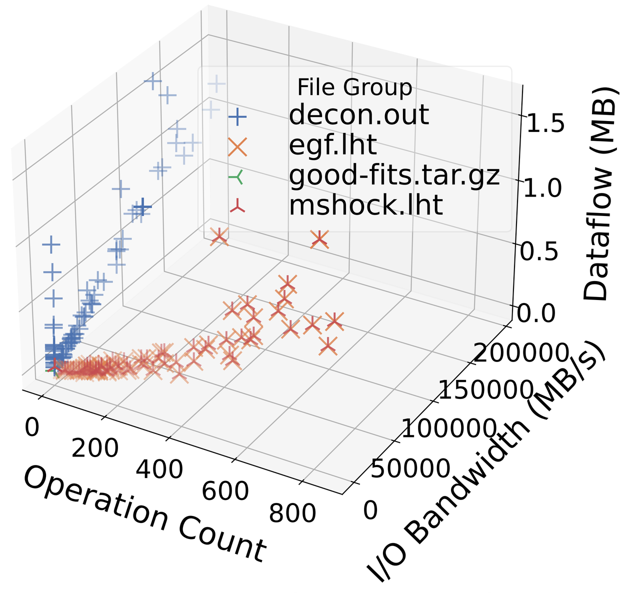

Figure 5: Group by File

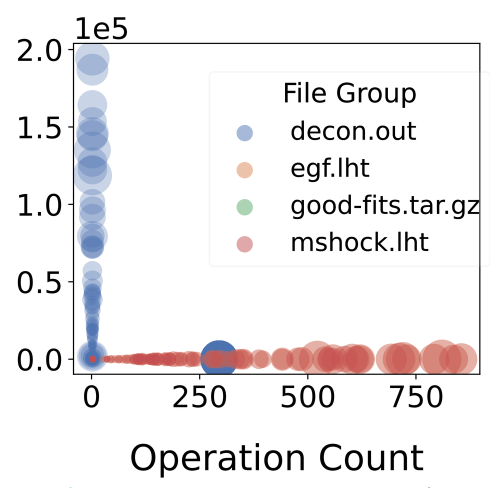

Figure 6: Group by File

1. IterDe (producer) and siftSTF (consumer): $n \to 1$ pattern.
2. Write performance variation under high parallelism.
3. Intra-task variation in I/O patterns and performance.

Optimization: Reduce write contention by distributing parallel tasks; improve intra-task read performance through prefetching of frequently accessed files.

Workflow Analysis: 1000 Genome

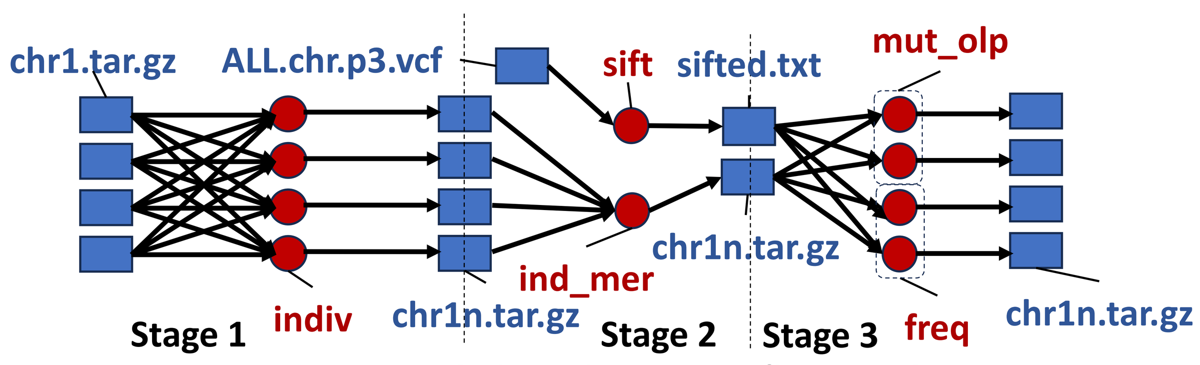

Figure 7: Seismology Dataflow DAG

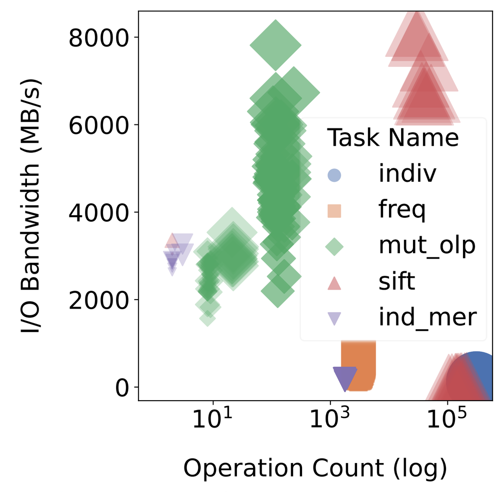

Figure 8: Group by Task

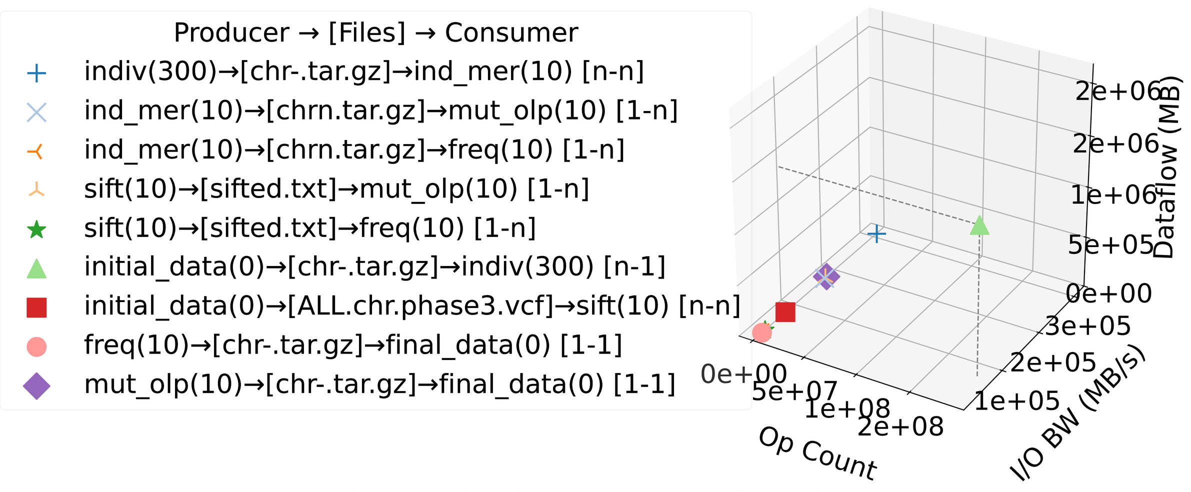

Figure 9: Group by P-C Pairs

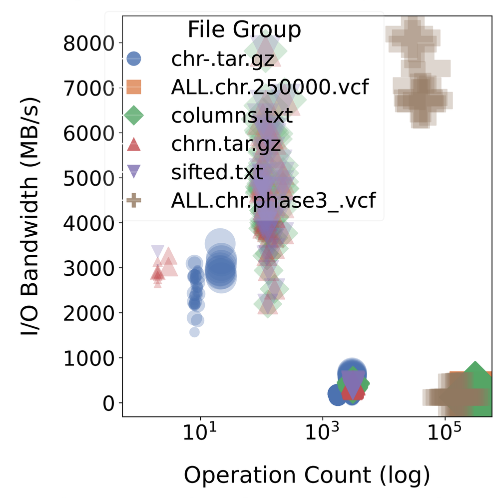

Figure 10: Group by File

1. Good Performance I/O patterns: $n \to n$, $1 \to n$, and $n \to 1$.
2. Inter-stage variation in I/O patterns and performance.
3. Same files different I/O patterns and performance.

Optimization: Forecast I/O demands for workload planning and storage selection; use storage systems optimized for small, random I/O operations.

Conclusion

Converged workflows integrate simulation, analytics, and AI, creating complex dataflows with diverse I/O demands. Supporting them requires storage systems tuned to these varied needs. Our producer–consumer analysis revealed insights that are otherwise difficult to uncover within workflow I/O, highlighting the need to co-design storage and scheduling for workflows.

Acknowledgments

This research is supported by the U.S. Department of Energy (DOE) through the Office of Advanced Scientific Computing Research's "Orchestration for Distributed & Data-Intensive Scientific Exploration"; the "Cloud, HPC, and Edge for Science and Security" LDRD at Pacific Northwest National Laboratory; and partly by the National Science Foundation under Grants no. CSSI-2104013 and OAC-2313154.