Understanding I/O Behavior in Scientific Workflows on High Performance Computing Systems

Fahim Chowdhury[1], Francesco Di Natale[2], Adam Moody[2], Elsa Gonsiorowski[2], Kathryn Mohror[2], Weikuan Yu[1]

[1] Florida State University, [2] Lawrence Livermore National Laboratory

Overview

- ❑ Leadership high performance computing (HPC) infrastructures empower scientific, research or industry applications
- ❑ Heterogeneous storage stack is common in supercomputing clusters
- ❑ **Project goals**
  - To extract the I/O characteristics of various HPC workflows
  - To develop strategies to optimize overall application performance by improving the I/O behavior
  - To leverage heterogeneous storage stack
- ❑ **Initial steps**
  - To understand I/O behavior in scientific application workflow on HPC
  - To perform systematic characterization and evaluation

I/O Workloads on HPC Workflow

- ❑ What is HPC Workflow?
  - Pre-defined or random ordered execution of set of tasks
  - Tasks performed by inter-dependent or independent applications
- ❑ Data transfer or dataflow in HPC Workflows can create bottleneck
- ❑ **Dataflow examples**
  - Huge **metadata overhead** on parallel file systems (PFS) by random tiny read requests in deep learning (DL) training workflow [1]
  - **Sequential write-intensive** applications, e.g., CM1
  - **In-situ and in-transit analysis** in applications, e.g., Montage
  - **Checkpoints and update-intensive** applications, e.g., Hardware Accelerated Cosmology Code (HACC) [2]

Heterogeneous Storage Stack

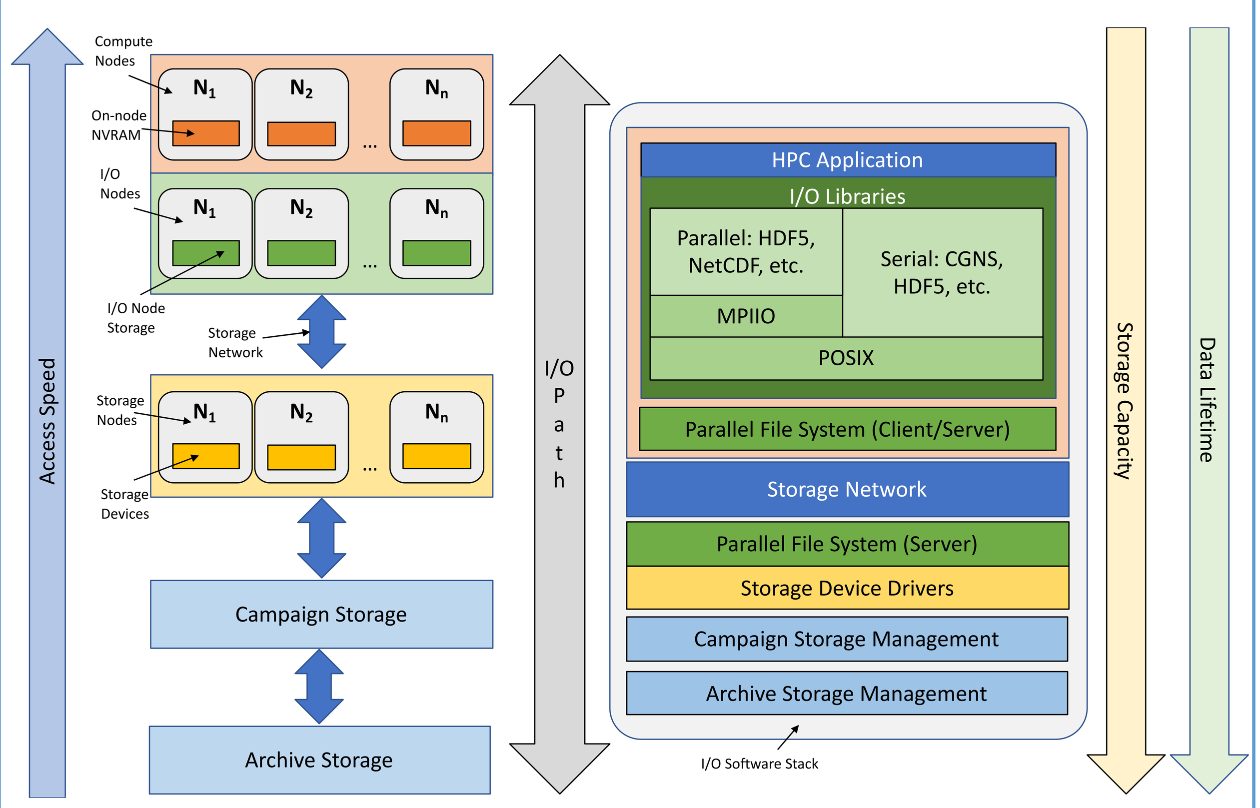


Fig. 1: Typical HPC I/O system architecture

Emulating Different Types of HPC I/O Patterns

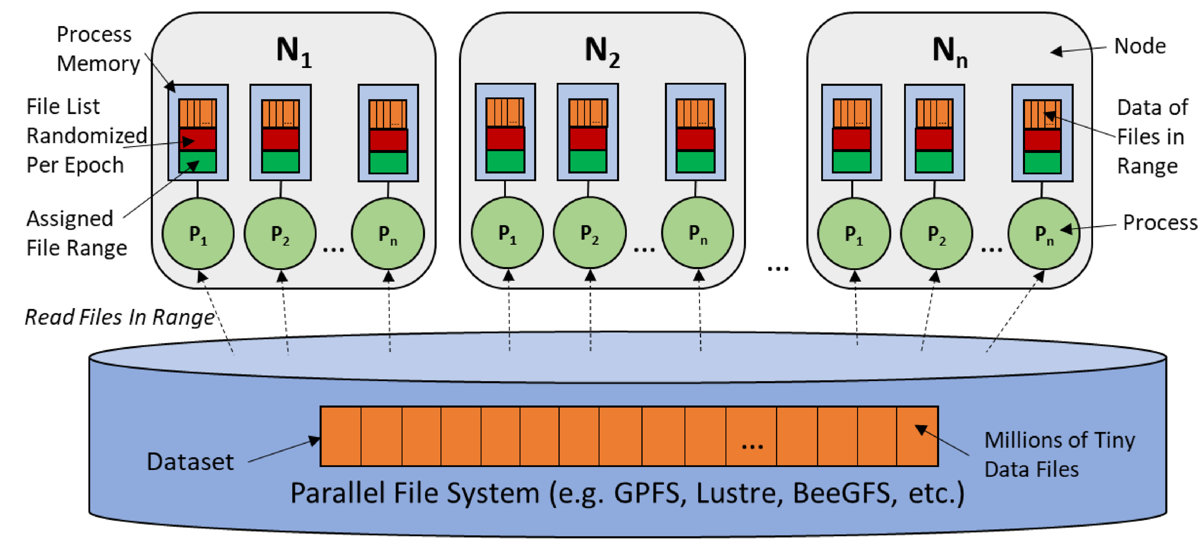


Fig. 2: Deep Learning Training I/O Pattern

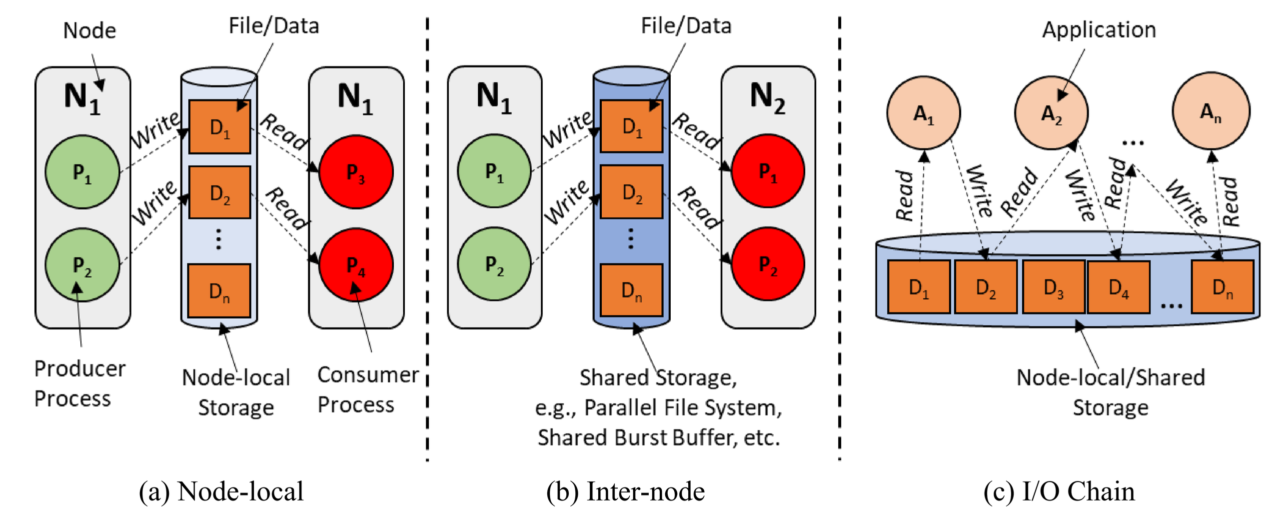


Fig. 3: Producer-Consumer I/O Pattern

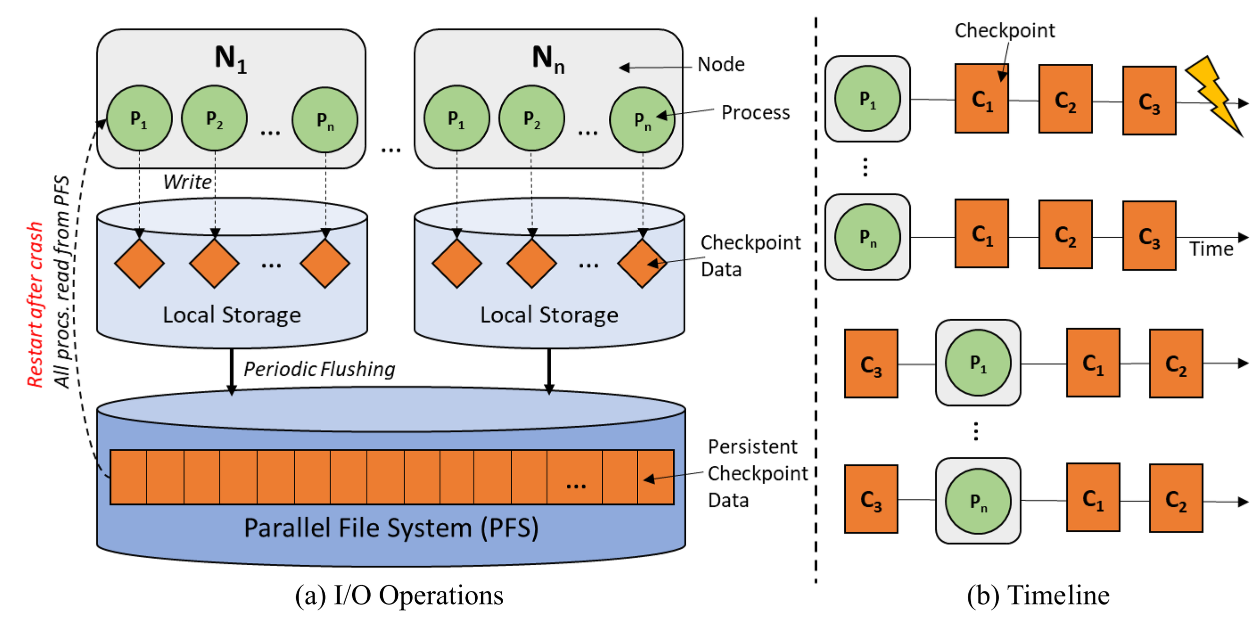


Fig. 4: Checkpoint/Restart I/O Pattern

✓ MPI enabled C++ Application to Emulate HPC I/O

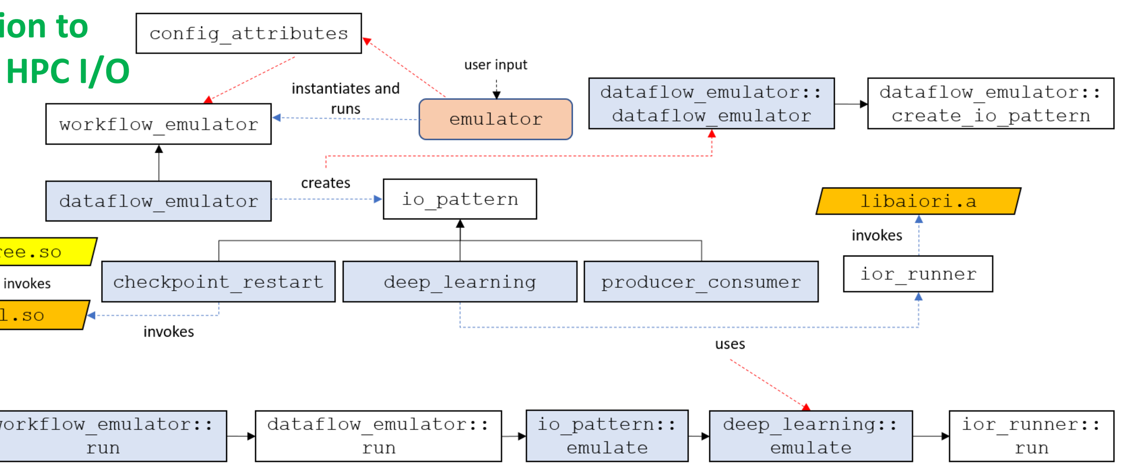


Fig. 5: Class Diagram of Dataflow Emulator

- ✓ Usage: *./emulator --type data --subtype dl --input_dir <dataset_dir>*
- ➢ Currently under active development

Experimental Results from Emulator

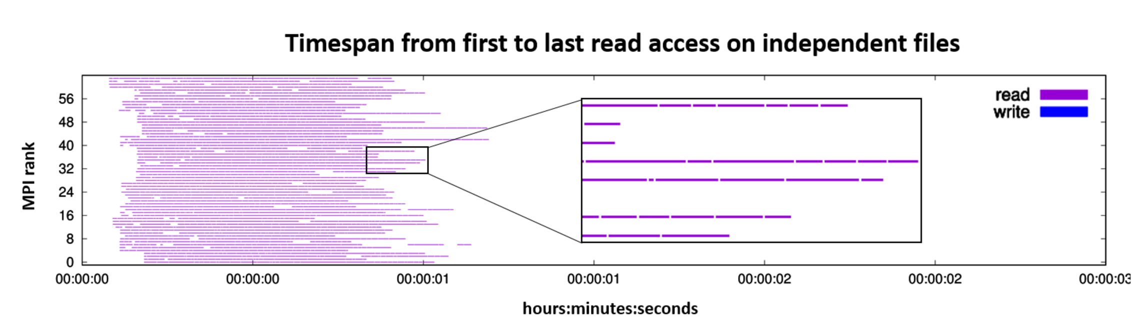


Fig. 6: Deep Learning Training I/O Timeline

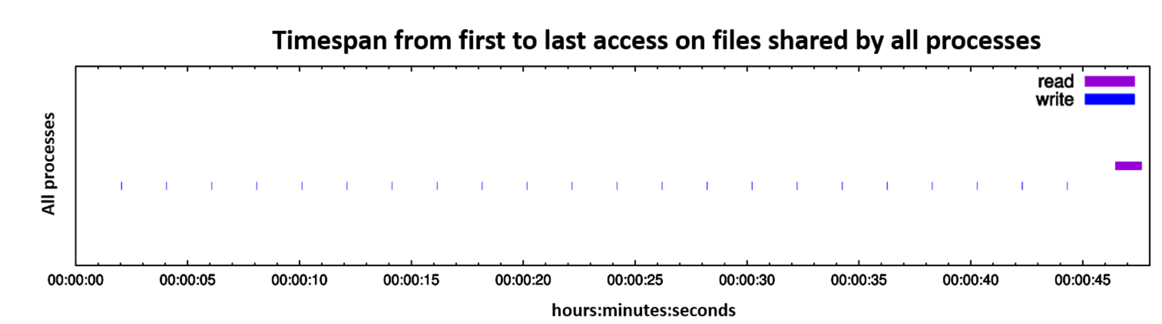
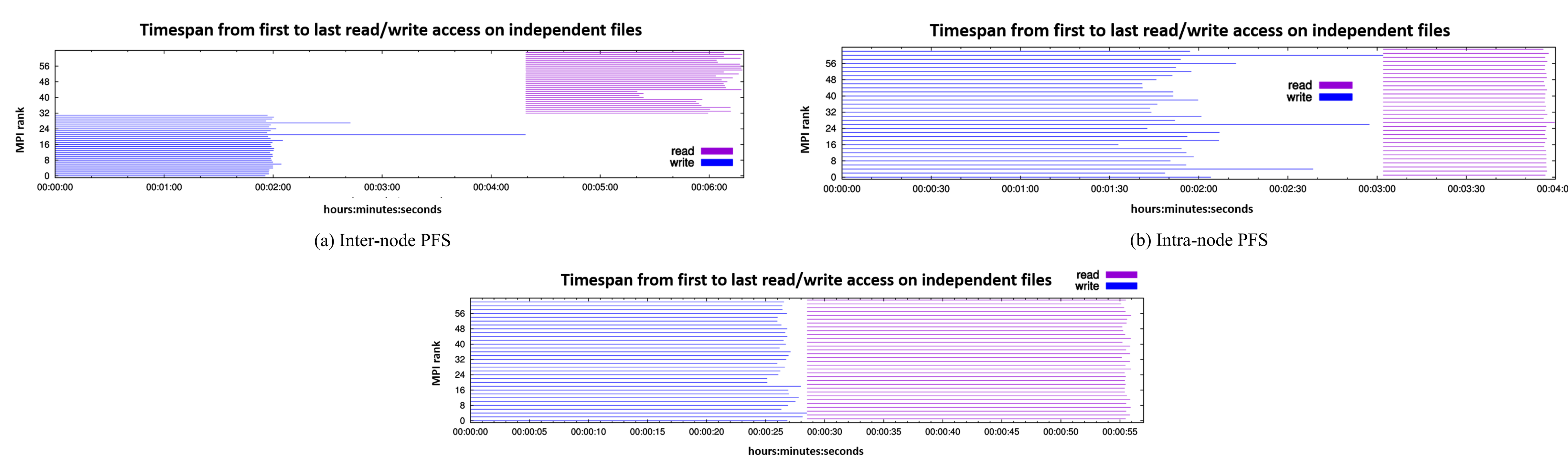


Fig. 7: Checkpoint/Restart I/O Timeline



(a) Inter-node PFS

(b) Intra-node PFS

(b) Intra-node Burst Buffer

Fig. 8: Producer-Consumer I/O Timeline

I/O Demands of Cancer Moonshot Pilot-2

- ❑ Cancer Moonshot Pilot2 (CMP2) [3] project
  - Aims at using HPC for cancer research
  - Seeks to simulate RAS protein and cell membrane interaction
- ❑ I/O behavior in CMP2
  - Producer-consumer pattern between different submodules

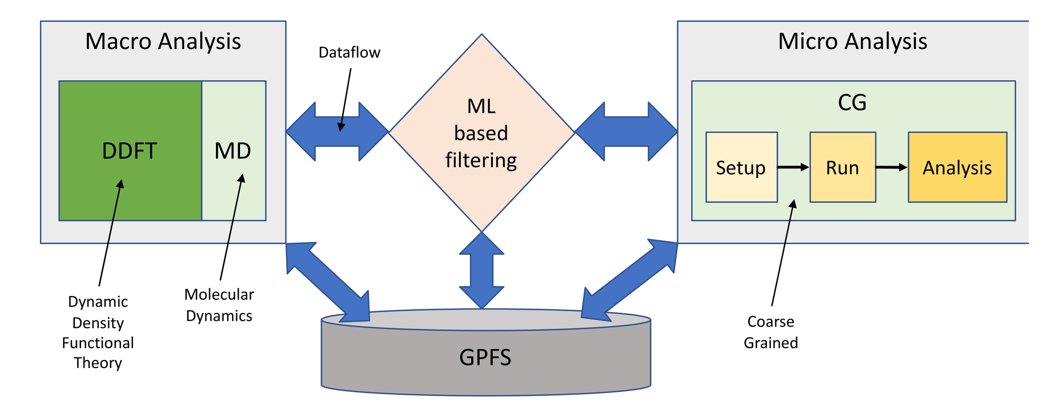


Fig. 9: Dataflow in CMP2 HPC implementation

Conclusion and Future Plans

- ❑ Handling dataflow in scientific applications' workflows is critical
- ❑ Effective I/O management system is necessary in workflow manager like MaestroWF [4]
- ❑ Intelligent data transfer among different units of heterogeneous storage resources in leadership supercomputers can improve performance
  - Third party API libraries like Asynchronous Transfer Library (AXL) [5] can be leveraged
- ❑ **Current Efforts**
  - ✓ Developing a Dataflow Emulator to generate different types of HPC I/O workloads
  - ✓ Analyzing the CMP2 project to detect possible I/O vulnerabilities
- ❑ **Future Plans**
  - ➢ To detect I/O behavior and dataflow by profiling CMP2 workflow
  - ➢ To develop data management strategies to properly handle the dataflow in complicated and composite HPC workflows like CMP2

References

[1] F. Chowdhury, Y. Zhu, T. Heer, S. Paredes, A. Moody, R. Goldstone, K. Mohror, and W. Yu. I/O Characterization and Performance Evaluation of BeeGFS for Deep Learning. In Proceedings of the 48th International Conference on Parallel Processing (ICPP 2019), Kyoto, Japan, August 5–8, 2019.

[2] Anthony Kougkas, Hariharan Devarajan, Jay Lofstead, and Xian-He Sun. 2019. LABIOS: A Distributed Label-Based I/O System. In Proceedings of the 28th International Symposium on High-Performance Parallel and Distributed Computing (HPDC '19). ACM, New York, NY, USA, 13-24.

[3] D. H. Ahn et al., "Flux: Overcoming Scheduling Challenges for Exascale Workflows," 2018 IEEE/ACM Workflows in Support of Large-Scale Science (WORKS), Dallas, TX, USA, 2018.

[4] MaestroWF. https://github.com/LLNL/maestrowf

[5] Asynchronous Transfer Library. https://github.com/ECP-VeloC/AXL

Acknowledgements and Contacts

This work is performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. LLNL-POST-784299.
This work is also supported in part by the National Science Foundation awards 1561041, 1564647, 1763547, and 1822737. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Contact Information

Email: fchowdhu@cs.fsu.edu; ftc17@my.fsu.edu; **Phone:** (786) 406-2617