

# System Software Instrumentation to Support the Visual Characterization of I/O System Behavior for High-End Computing

Pete Beckman<sup>1,2</sup>, Jason Cope<sup>1,2</sup>, Kamil Iskra<sup>1,2</sup>, Sam Lang<sup>1,2</sup>, Kwan-Liu Ma<sup>3</sup>, Chris Muelder<sup>3</sup>, Robert Ross<sup>1,2</sup>, Carmen Sigovan<sup>3</sup>

<sup>1</sup>Computation Institute, University of Chicago, Chicago, IL, 60637

<sup>2</sup>Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 60439

<sup>3</sup>Department of Computer Science, University of California-Davis, Davis, CA, 95616

## IOVIS Project Goals

Develop software infrastructure to provide end-to-end analysis and visualization of I/O system software. Specifically, the IOVIS project's objectives are to develop, improve, and deploy the following infrastructure and tools:

- ▶ End-to-end, scalable tracing infrastructure integrated into the I/O software stack components (high-level I/O libraries, MPI-IO libraries, I/O forwarding layers, and file systems)
- ▶ Information visualization tools for inspecting traces and extracting knowledge
- ▶ Testing components that drive systems to generate example patterns, including a fault injection instrumentation layer
- ▶ Tools to help system software developers, system architects, and system administrators incorporate this analysis and visualization system into their system design workflows

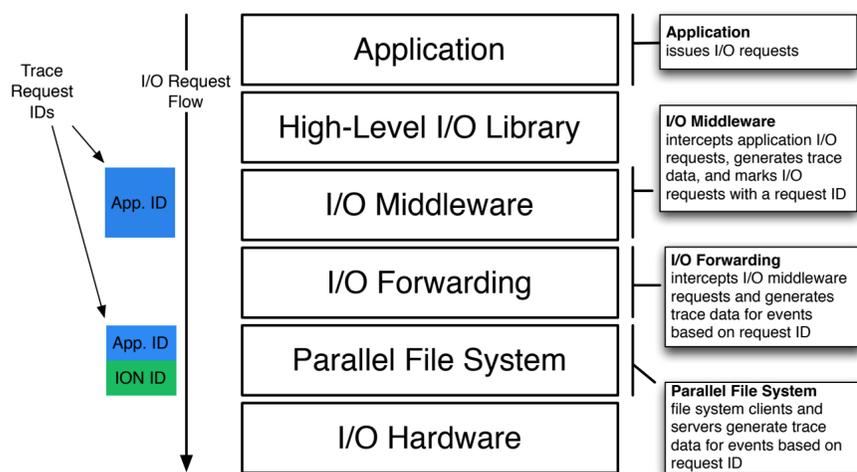
## IOVIS Software Infrastructure and Tools

IOVIS provides a suite of system software instrumentation tools to collect data for characterization of I/O systems. These components include:

- ▶ An I/O software instrumentation and tracing layer based on the University of Oregon's Tracing and Analysis Utilities (TAU)
- ▶ Application wrapper libraries to capture MPI-IO and POSIX I/O requests using the tracing layer software
- ▶ System software libraries to capture application I/O requests passing through the IOFSL I/O forwarding layer and PVFS2 parallel file system
- ▶ Tools to reformat the collected trace data

## IOVIS-Instrumented HPC I/O Software Stack

Using the IOVIS instrumentation layer, the I/O request start and end events are traced through the software layers (right). To identify the origin of requests from specific software components, a unique identifier is appended to the I/O request as it traverses the I/O software stack (left).



## IOVIS Data Collection

IOVIS-instrumented software collects several event types within each layer of the HPC I/O software stack.

- ▶ MPI-IO requests from the I/O middleware layer (MPI\_File\_write\_at)
- ▶ Network communication requests between the PVFS2 client and server (bmi\_server\_send, bmi\_client\_rcv)
- ▶ File I/O requests made by the PVFS2 server (dbpf\_write)

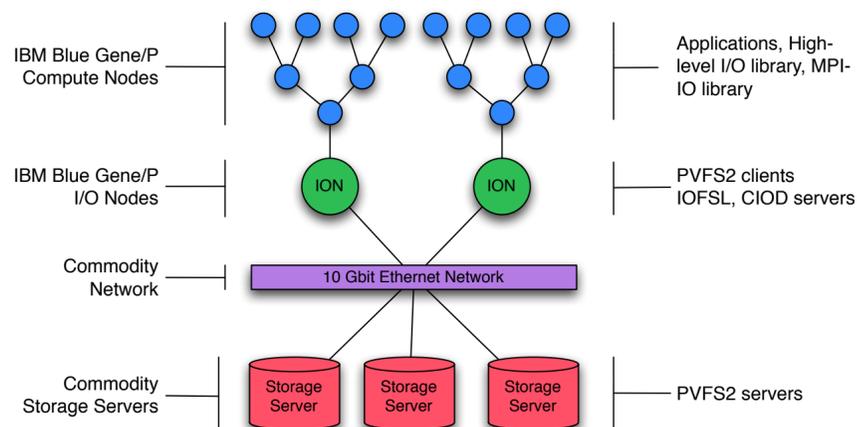
Additional data describing the operation is also tracked with each event.

- ▶ Event start and stop times
- ▶ Software layer IDs (application rank, file system client ID)
- ▶ I/O request payload size
- ▶ File handle (if available)

Hints and temporary files are used to pass IDs between the software layers and are used to associate the I/O requests at each software layer.

## IOVIS Deployment on IBM Blue Gene/P Systems

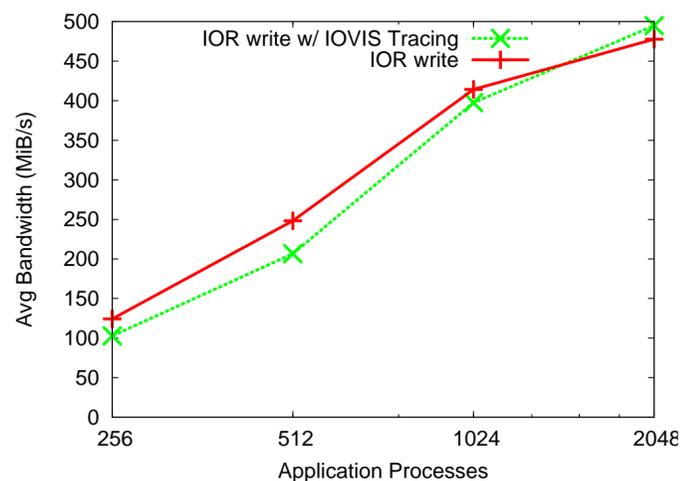
- ▶ The compute nodes (blue) run the instrumented application and generate the initial I/O request ID
- ▶ The I/O nodes (green) run a custom ZeptoOS profile that mounts the IOVIS-instrumented PVFS2 file system
- ▶ The 10 Gbit Ethernet network (purple) connects the I/O nodes and the IOVIS-instrumented PVFS2 file system
- ▶ The storage servers (red) hosts the IOVIS-instrumented PVFS2 file system



We have produced traces for the IOR benchmark, the mpi-tile-io benchmark, and the FLASH I/O kernel (using Parallel netCDF and HDF5) for the MPI-IO, PVFS2 client, and PVFS2 server software layers. We have scaled up our traces to 16,384 application processes on Argonne's ALCF IBM Blue Gene/P (Intrepid) and 32 storage servers on Argonne's ALCF Eureka cluster.

## IOVIS Scalability and Trace Data Management

We evaluated the IOVIS trace instrumentation overhead on the IBM Blue Gene/P using IOR and PVFS2.



Our largest PVFS2 client and server traces exceed 1.0 GB for short IOR and FLASH runs using 10% of ALCF's largest computing platform. We reduced trace sizes to 17% using a zlib-based reformatting tool and to 43% using a binary trace format.

## Current and Future Work

- ▶ Trace Publication
- ▶ I/O Software Instrumentation (Parallel netCDF, IBM's cioid)
- ▶ Trace Layer Portability, Generalization, Scalability and Data Management
- ▶ Fault and Congestion Instrumentation Layer

## Acknowledgements

Support for this project is provided by the National Science Foundation under Grant No. 0938114. Additional support is provided by the Office of Advanced Scientific Computer Research, Office of Science, U.S. Dept. of Energy, under Contract DE-AC02-06CH11357. This research used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under contract DE-AC02-06CH11357.