I/O Traces of HPC Applications

Chen Wang[1]
chenw5@illinois.edu

Kathryn Mohror[2]
kathryn@llnl.gov

Marc Snir[1]
snir@illinois.edu

1. University of Illinois at Urbana-Champaign
2. Lawrence Livermore National Laboratory

Introduction

- A better understanding of I/O behaviors of an application can:
  - Help users optimize I/O performance and identify potential I/O bottlenecks.
  - Help file system and I/O library developers make better optimization decisions.

- The first release of the dataset contains I/O traces from 14 HPC applications.

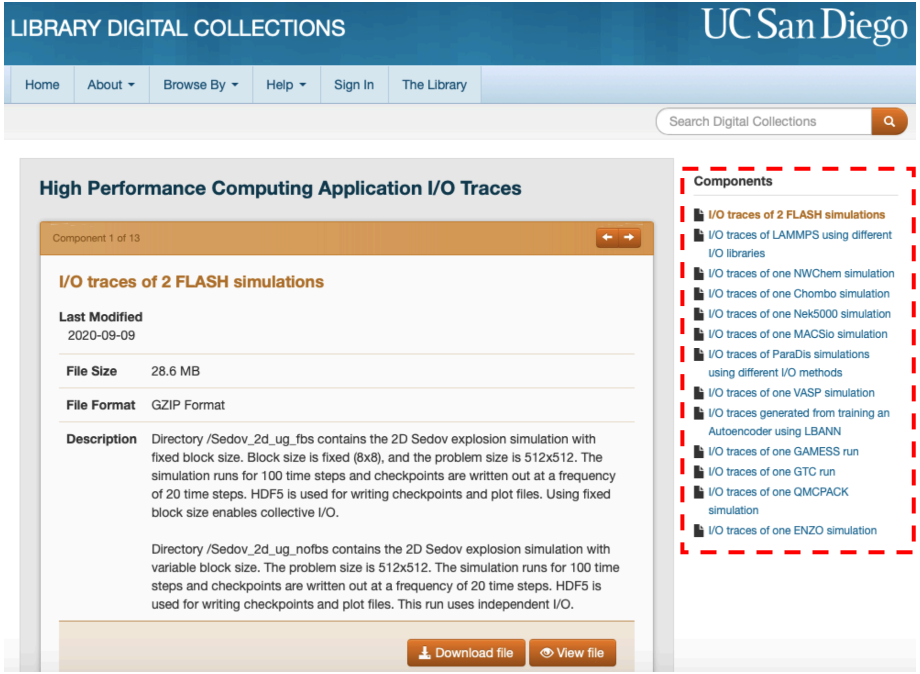
Introduction

- All traces were generated using Recorder[1], a multi-level tracing tool.
  - Trace record includes enter/exit timestamp of the function call, function name and all function parameters except the data buffer.
  - Currently includes POSIX, MPI-IO and HDF5.
  - I/O traces with more details take more time to collect. But once generated can be used repeatedly by researchers to perform different analysis tasks.
- Traces are in Recorder-specific binary format.
  - Tools are provided to read, decompress and decode the trace. These tools are written in C with python bindings provided for easy analysis.
  - More information on reading, visualizing and analyzing the traces can be found here: https://github.com/uiuc-hpc/Recorder

[1] Wang, Chen, Jinghan Sun, Marc Snir, Kathryn Mohror, and Elsa Gonsiorowski. "Recorder 2.0: Efficient parallel I/O tracing and analysis." In *2020 IPDPS Workshops (IPDPSW)*, pp. 1-8. IEEE, 2020.

I/O Traces of 14 HPC applications

- 14 HPC applications and I/O emulators.
- Applications perform I/O using POSIX, MPI-I/O and high level libraries include HDF5, NetCDF, Silo, and ADIOS.
- 8 nodes x 8 MPI ranks/node @ LLNL Quartz system.
- Traces and visualizations are available at https://doi.org/10.6075/J0Z899X4



High Performance Computing Application I/O Traces

I/O traces of 2 FLASH simulations

Last Modified
2020-09-09

File Size	28.6 MB
File Format	GZIP Format
Description	Directory /Sedov_2d_ug_fbs contains the 2D Sedov explosion simulation with fixed block size. Block size is fixed (8x8), and the problem size is 512x512. The simulation runs for 100 time steps and checkpoints are written out at a frequency of 20 time steps. HDF5 is used for writing checkpoints and plot files. Using fixed block size enables collective I/O. Directory /Sedov_2d_ug_nofbs contains the 2D Sedov explosion simulation with variable block size. The problem size is 512x512. The simulation runs for 100 time steps and checkpoints are written out at a frequency of 20 time steps. HDF5 is used for writing checkpoints and plot files. This run uses independent I/O.

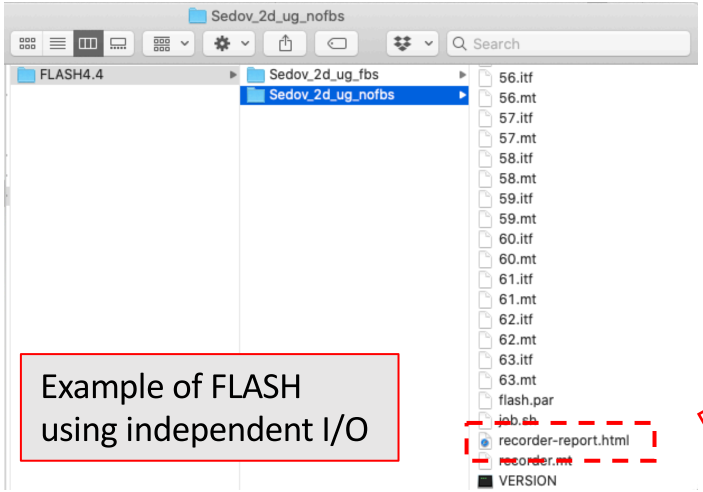
Components

- I/O traces of 2 FLASH simulations
- I/O traces of LAMMPS using different I/O libraries
- I/O traces of one NWChem simulation
- I/O traces of one Chombo simulation
- I/O traces of one Nek5000 simulation
- I/O traces of one MACSio simulation
- I/O traces of ParaDis simulations using different I/O methods
- I/O traces of one VASP simulation
- I/O traces generated from training an Autoencoder using LBANN
- I/O traces of one GAMESS run
- I/O traces of one GTC run
- I/O traces of one QMCPACK simulation
- I/O traces of one ENZO simulation

I/O Traces of 14 HPC applications

Following files are included for each application run:

- 64 trace files (*.itf) and 64 (*.mt) metadata files, one for each MPI rank.
- recorder-report.html: visualization report generated by Recorder.
- Input files if used.
- Other files like logs written by the application.

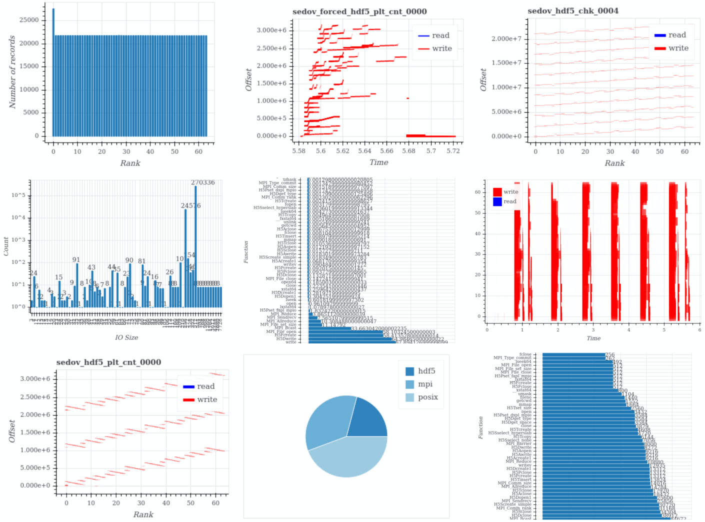


FLASH4.4

Sedov_2d_ug_fbs
Sedov_2d_ug_nofbs

56.itf
56.mt
57.itf
57.mt
58.itf
58.mt
59.itf
59.mt
60.itf
60.mt
61.itf
61.mt
62.itf
62.mt
63.itf
63.mt
flash.par
job.sh
recorder-report.html
recorder.mt
VERSION

Example of FLASH using independent I/O



Thank You

Contact:

Chen Wang (chenw5@illinois.edu)

Kathryn Mohror (kathryn@llnl.gov)

Marc Snir (snir@illinois.edu)