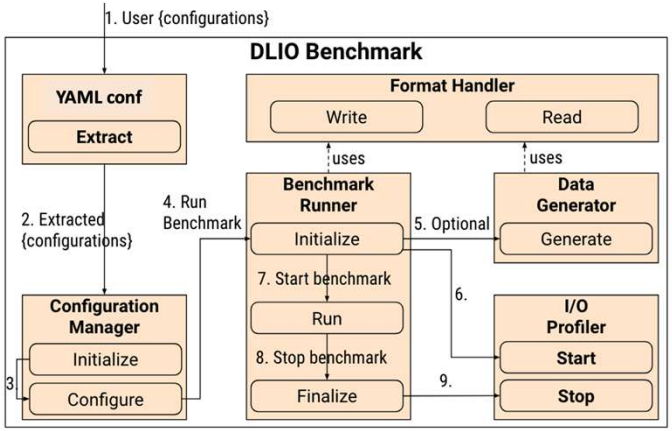
ML WORKLOADS

SIMULATION I/O VS AI I/O

- Simulation:
  - Collective reads/writes
  - Could be irregular, small, but never random
  - Standardization: MPI and libraries
- AI
  - Repeated reads of training model
  - No coordination among processes
  - No “middleware for AI” (yet)
  - Sometimes part of a workflow, not a single application

DLIO: A DEEP LEARNING BENCHMARK

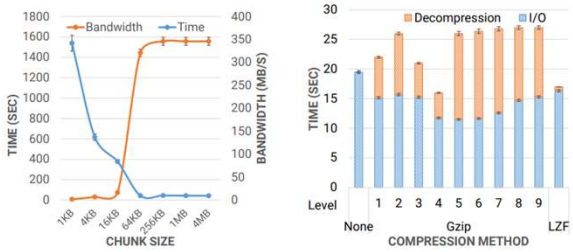
- Framework for evaluating deep learning I/O performance
- Replaces computationally intensive phases with sleep
- But performs I/O the way deep learning framework would
  - Calls “torch DataLoader” or “tensorflow.data” loaders with synthetic data
- More information:
  - https://argonne-lcf.github.io/dlio_benchmark a
  - https://github.com/argonne-lcf/dlio_benchmark



DLIO: A Data-Centric Benchmark for Scientific Deep Learning Applications, Harihan Devarajan et al, CCGrid21

DLIO TRAINING: RAPID EXPERIMENTATION

- Able to sweep across a wide range of parameters quickly
- Lessons learned in DLIO simulations apply directly to actual ML framework/workflow
- Helps improve Darshan
- Future work: integrating DLIO instrumentation with Darshan's py-darshan interface for improved reporting



Experimenting with access ("chunk") size and compression; from *DLIO: A Data-Centric Benchmark for Scientific Deep Learning Applications*, Harihan Devarajan et al, CCGrid21

DLIO tracing of workflow

ADDITIONAL TOPICS

- Helpful tools:
  - Ltrace and strace
    - Confirming behavior of I/O libraries
  - Gdb
    - “why is everyone stalled in this collective?”
- Technologies
  - GPU programming?
    - NVIDIA’s ‘gpu direct storage’: https://developer.nvidia.com/gpudirect-storage
  - DAOS:
    - Novel storage architecture, showing up on Aurora

BIG PICTURE SUMMARY

- I/O subsystems complex with lots of layers
- Initial experiences not likely to be ideal
- Use libraries and frameworks (where available)
  - Portability across file systems, machines, storage technologies
- Darshan helps Scientists and I/O folks meet on common ground
- Consultants at your site (e.g. ALCF, OLCF, NERSC) love solving problems

ACKNOWLEDGEMENTS

- Some material drawn from *Parallel I/O In Practice,* our full day SC tutorial. Thanks to Brent Welch, Rob Ross, Glenn Lockwood, Katie Antypas, Marc Unangst, Rajeev Thakur, and Bill Loewe.
- Some material drawn from ATPESC “Data and I/O Day”. Thanks to Phil Carns and Shane Snyder
- This research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.
- This work was supported by the Better Scientific Software Fellowship Program, funded by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy (DOE) Office of Science and the National Nuclear Security Administration; and by the National Science Foundation (NSF) under Grant No. 2154495. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the DOE or NSF.