A data-centric view on workflows that couple HPC with large-scale models

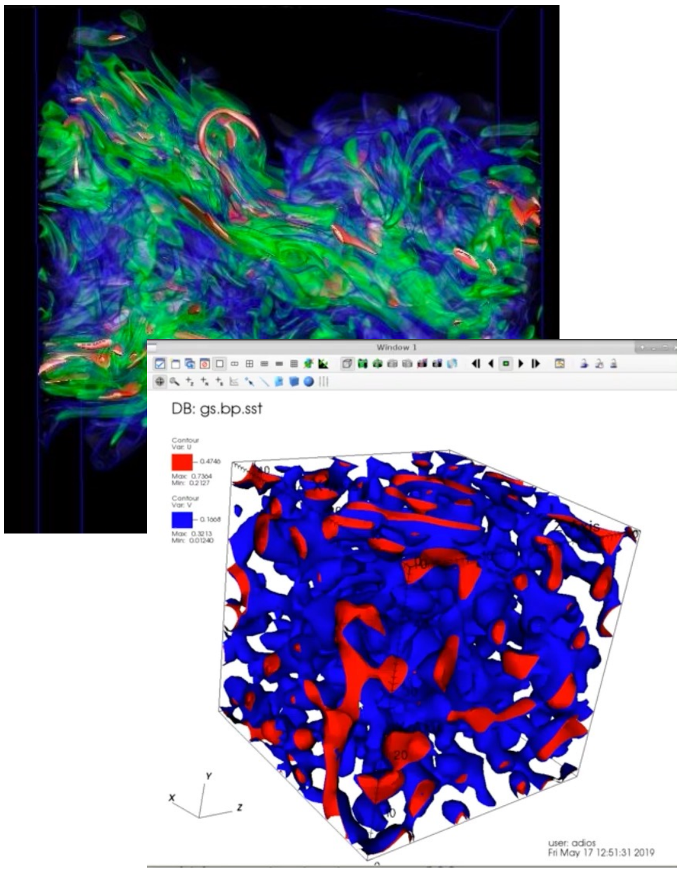
Ana Gainaru

Workshop on Advancing Neural Network Training
NeurIPS, Dec 16, 2023

What to expect for the next 25ish minutes

- I/O Profiles for HPC AI applications
  - **Bottlenecks when trying to run AI on HPC**
  - **How well does AI scale on HPC?**
- Large-scale workflows combining HPC and AI
  - **More bottlenecks**
- A data-centric approach to Neural Network Training
  - **How disruptive do we need to be?**
  - **Some results and recommendations**

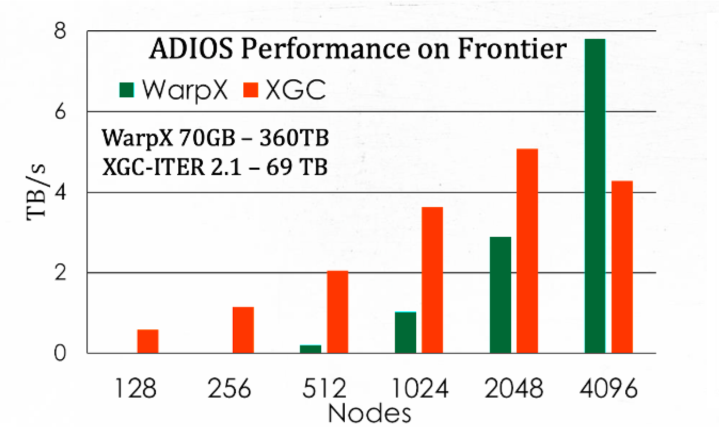
Traditional HPC

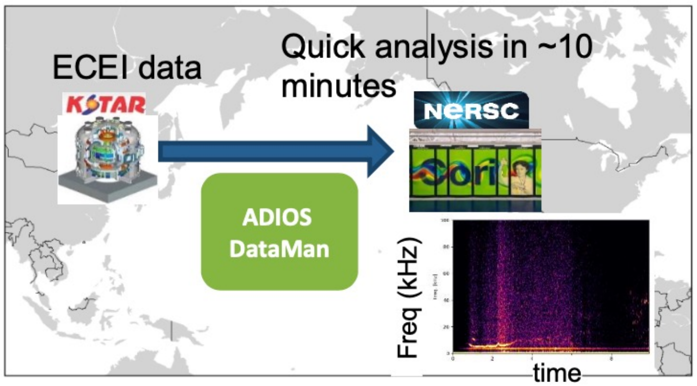


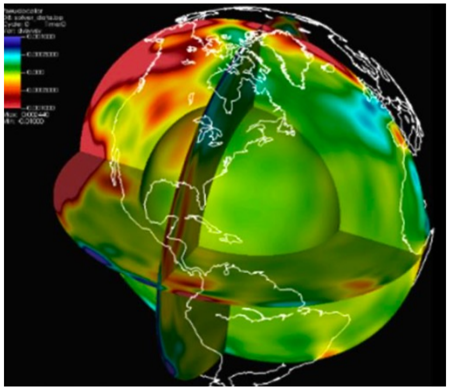
- Large monolithic codes
  - High fidelity simulations of physical phenomena
- Iterative in nature
  - Fairly predictable, roof model
- Write oriented (checkpoints, data)
  - Combined with visualization or in-situ analysis
- Workflow
  - Ensembles simulations
  - Analysis and viz

A few of our applications

- Wind Turbine (GE)
- Accelerator Physics (PIConGPU, WarpX)
- Fusion (GTC, XGC, GENE, KSTAR)
- Cancer research
- Combustion (S3D)
- Climate (E3SM)
- Radio astronomy (SKA)
- Seismic Tomography Workflow
- Molecular dynamic (DeepDriveMD)







Oak Ridge National Laboratory

Why use HPC for AI?

- **Training** large AI models requires **large amounts of computing resources**
  - E.g. BERT model (3 years old) uses 110M parameters, Megatron-2 one trillion

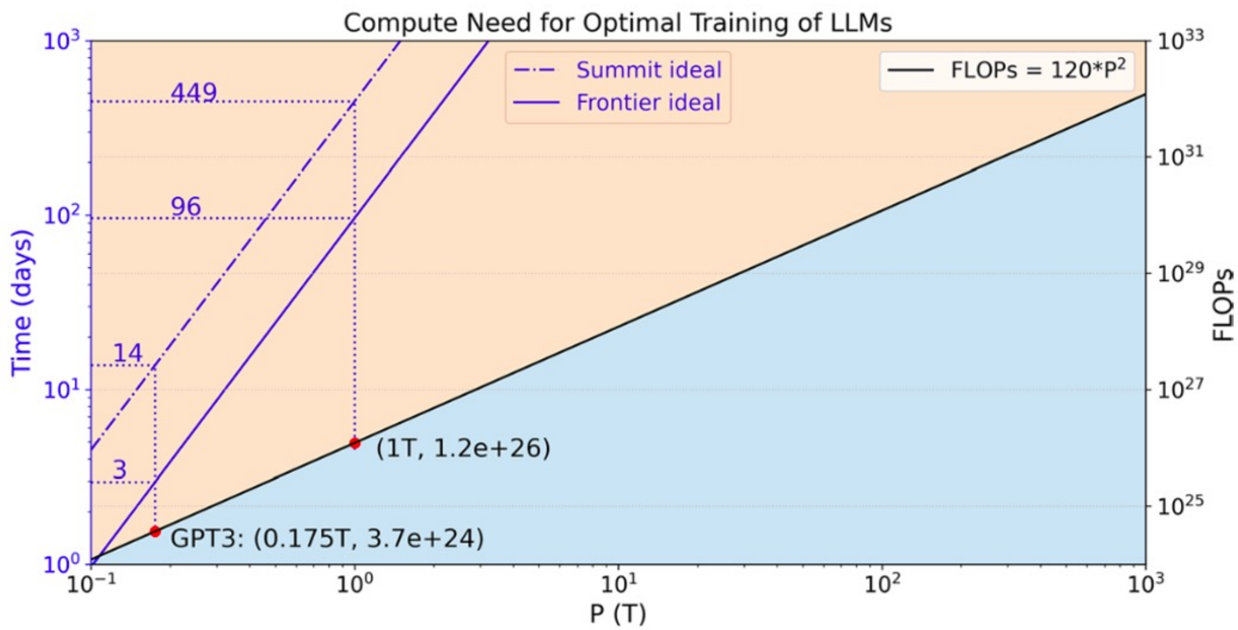
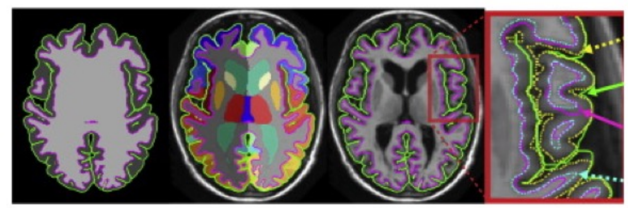


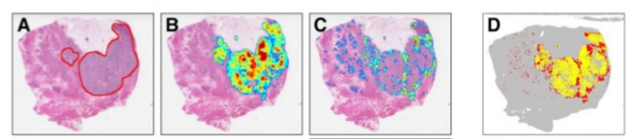
Figure from: Evaluation of pre-training large language models on leadership-class supercomputers
Junqi Yin, Sajal Dash, John Gounley, Feiyi Wang, Georgia Tourassi in The Journal of Supercomputing, June, 2023

Why use HPC for AI?

- **Inference** is usually done by parsing **large amounts of data**
  - Cancer research / neuroscience typically classify hundred of thousand WSI / MRIs in one study
    - Sometimes large images: e.g. a single whole slide image corresponding to a single prostate biopsy core can easily occupy 10 GB of space at 40x magnification

- Typical ways of training AI on HPC
  - **Data parallel**: all processes store the model: replicated or in shared memory; data is distributed
  - **Model parallel**: model is distributed; each process goes over the same dataset
  - **Pipeline parallelism:** combine the data and model parallel methods



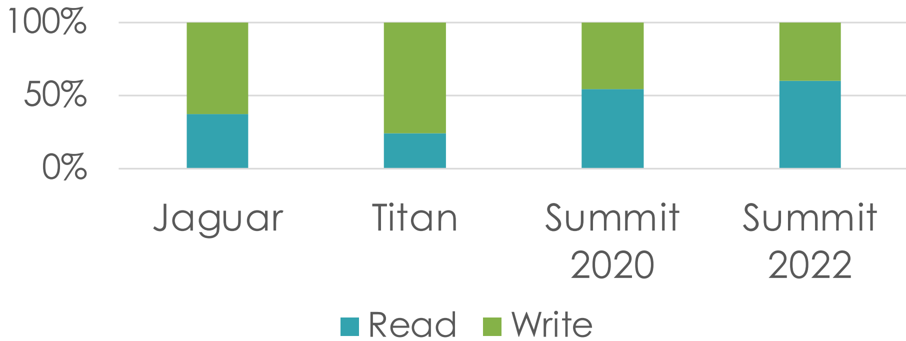


I/O patterns

- Three types of AI applications
  - **Inference**: dataset is distributed over processes
  - **Training data parallel**: dataset is distributed over processes
  - **Training model parallel**: all processes read the entire dataset

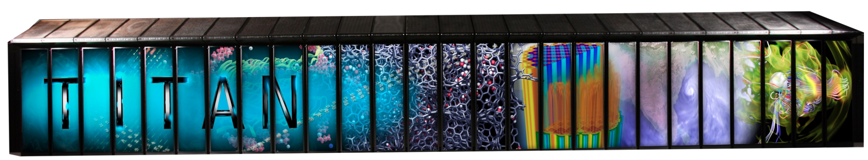
- Next few slides
  - I/O patterns in HPC before and after AI
  - Performance bottlenecks for the three types of AI

Summit Darshan logs

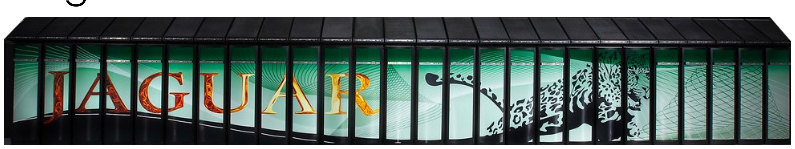


Summit 2018-still running

Titan 2012-2019



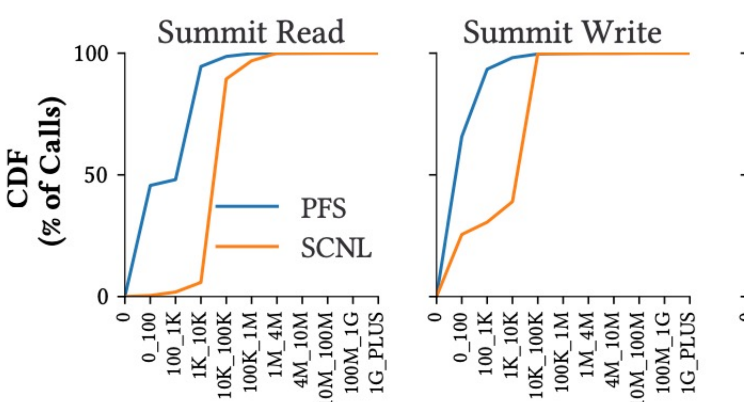
Jaguar 2006-2012



Comparative I/O Workload Characterization of Two Leadership Class Storage Clusters
Raghul Gunasekaran et al. at PDSW 2015

Summit Darshan logs

- High rank variance
- Mostly small size access
  - Many consecutive reads
  - Many open/close



Request size

- Read/write pattern
  - 32% write intensive
  - 44% read intensive
  - The rest balance between RW
- Metadata intensive **(41%)**
  - 22% write intensive
  - 52% read intensive

Access Patterns and Performance Behaviors of Multi-layer Supercomputer I/O Subsystems under Production Load
Jean Luca Bez et al. HPDC 2022

I/O patterns for AI applications

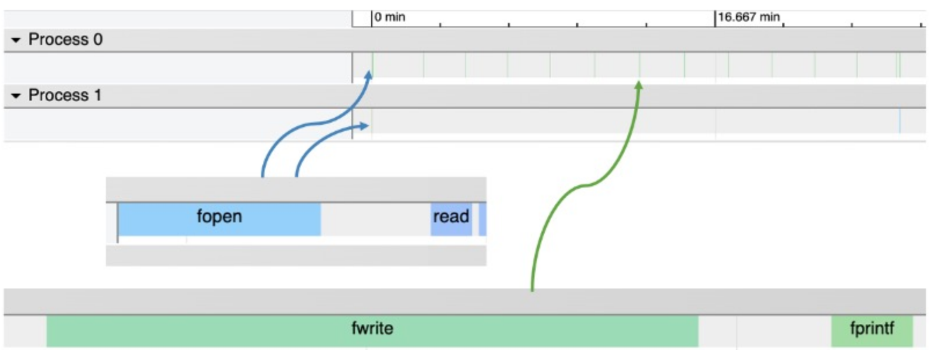
- There is a shift in the I/O patterns seen at the system level
  - Future I/O library design
  - Future system designers

Let's look at some application runs

Profiling typical HPC applications

- LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator)
  - 32000 atoms

Class method	Number of calls	Percentage Time
Pair_LJ_Charmm_Coul_Long::compute()	101	59.9
Neigh_half::half_bin_newton()	12	11.4
PPPM::fieldforce()	101	5.7
Neighbor::find_special()	144365706	5.4

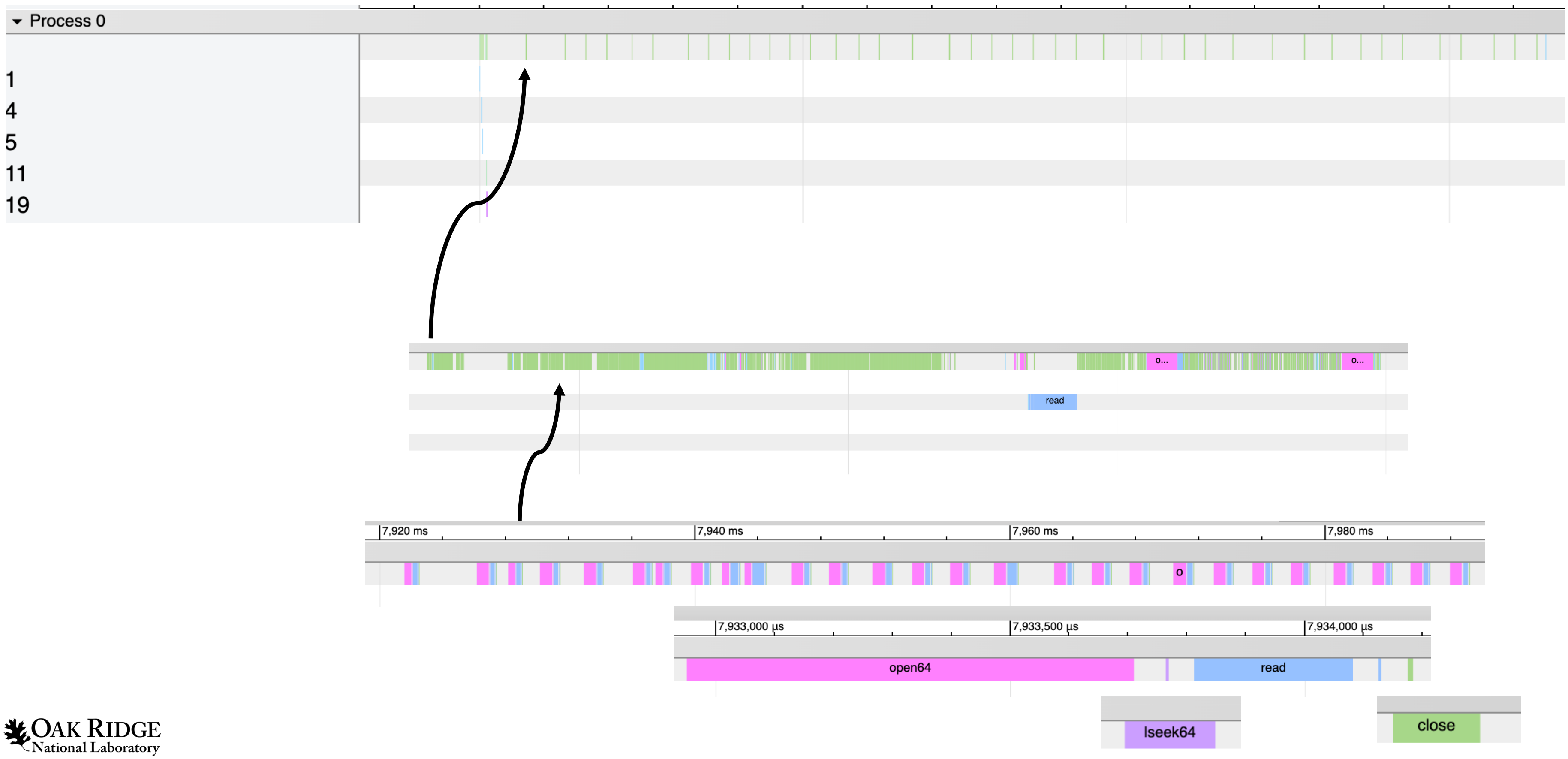


OAK RIDGE National Laboratory

I/O patterns for AI inference

TIL classification application

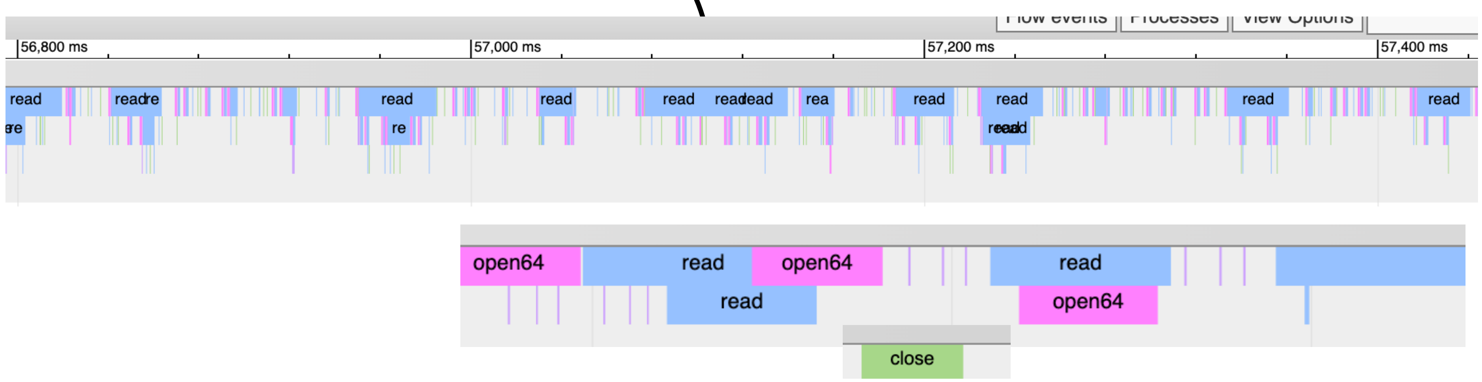
- Identify cancerous cells in WSI

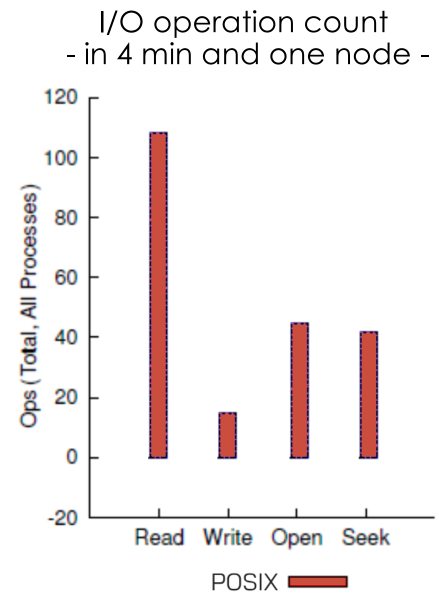


I/O patterns for AI training

- Multiple threads reading at the same time
- Multiple patterns of Open/Seek/Read/Close

Training ImageNet

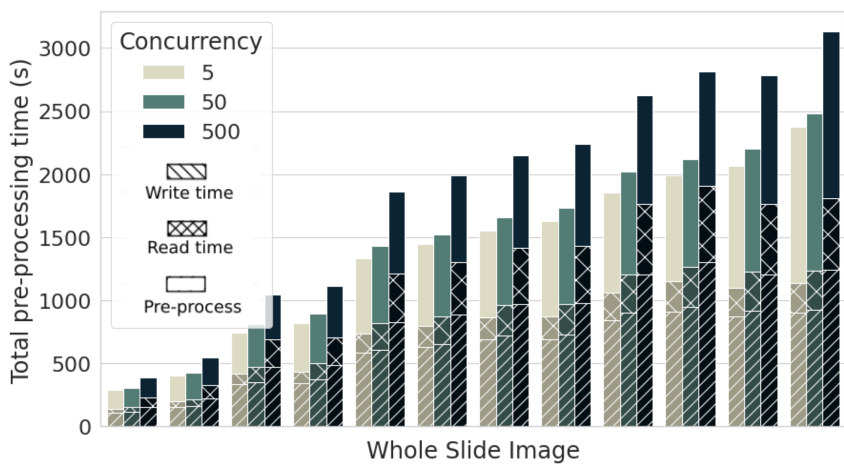




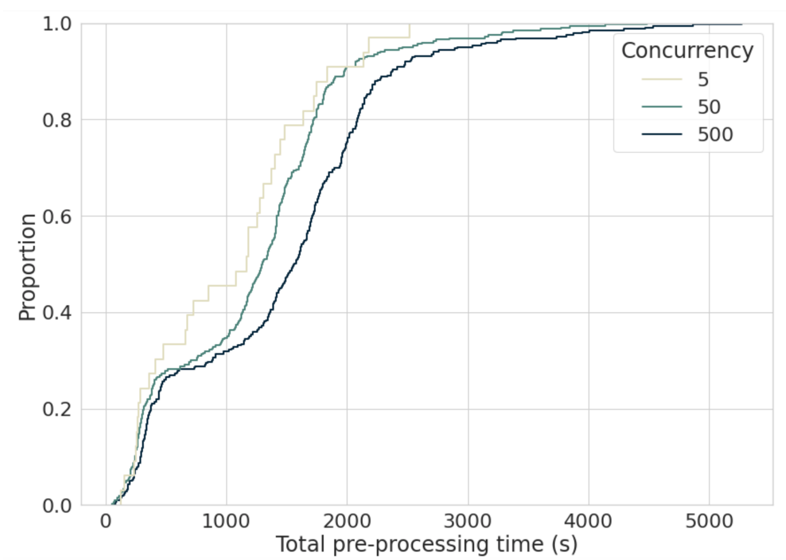
OAK RIDGE National Laboratory

Scaling

- Larger models
  - More time for training, I/O becomes less frequent
- Multiple processes
  - Less data per process



- At scale
  - Less frequent, less amount of I/O
  - However, very frequently the I/O is concurrent (e.g. input, model sync)

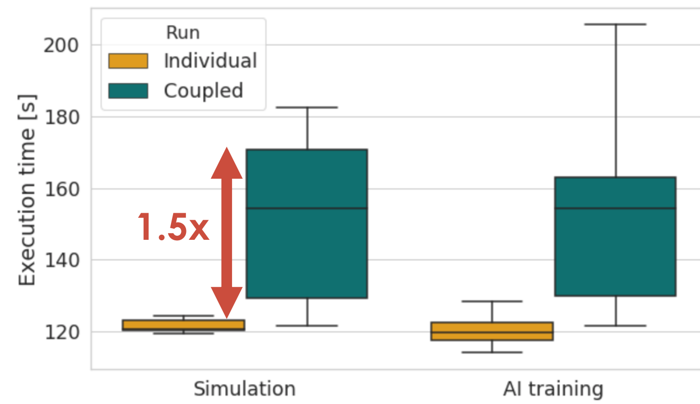


OAK RIDGE National Laboratory

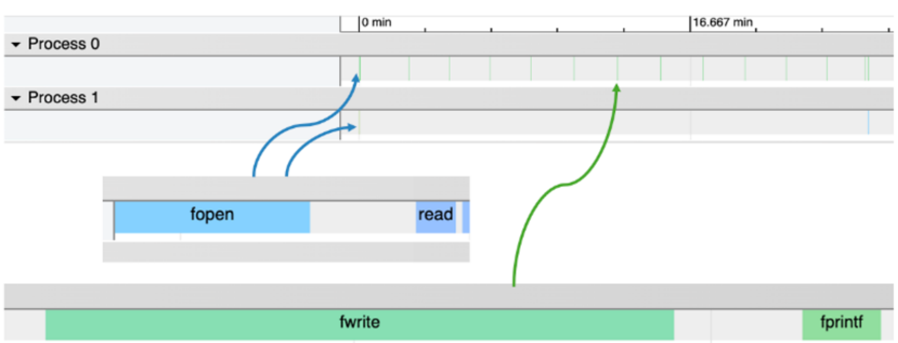
Can we do worse?

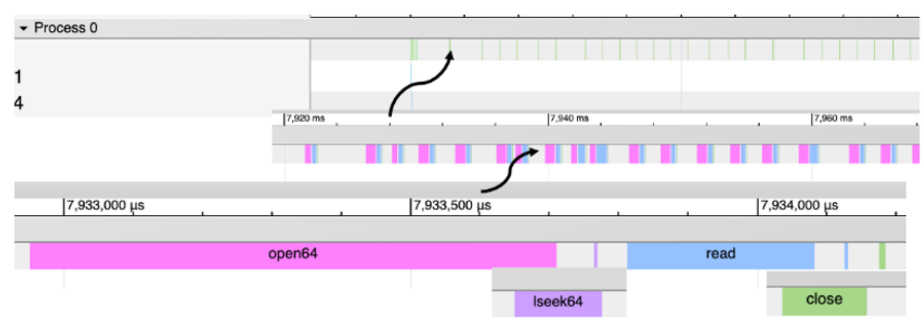
- Coupling AI with HPC
  - Simplified AI Steering HPC scenario
    - Running the Gray-Scott simulation
    - Running an AI training code to create a digital twin of the Gray-Scott simulation

- **Slowdown** of 1.5x due to congestion



Simulation and analysis execution time if ran separately or coupled



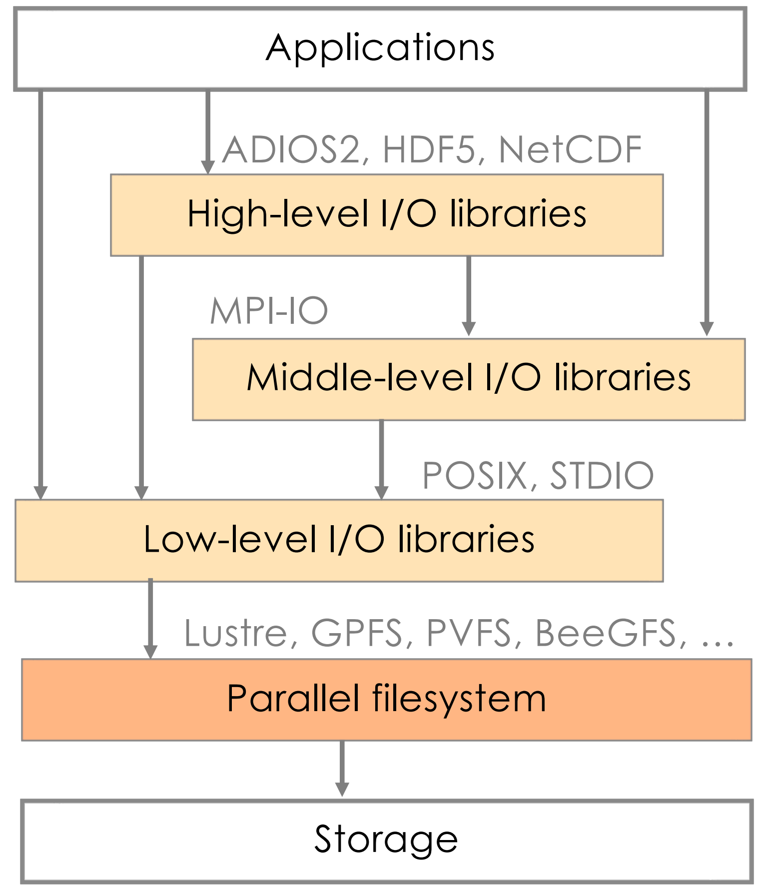


OAK RIDGE National Laboratory

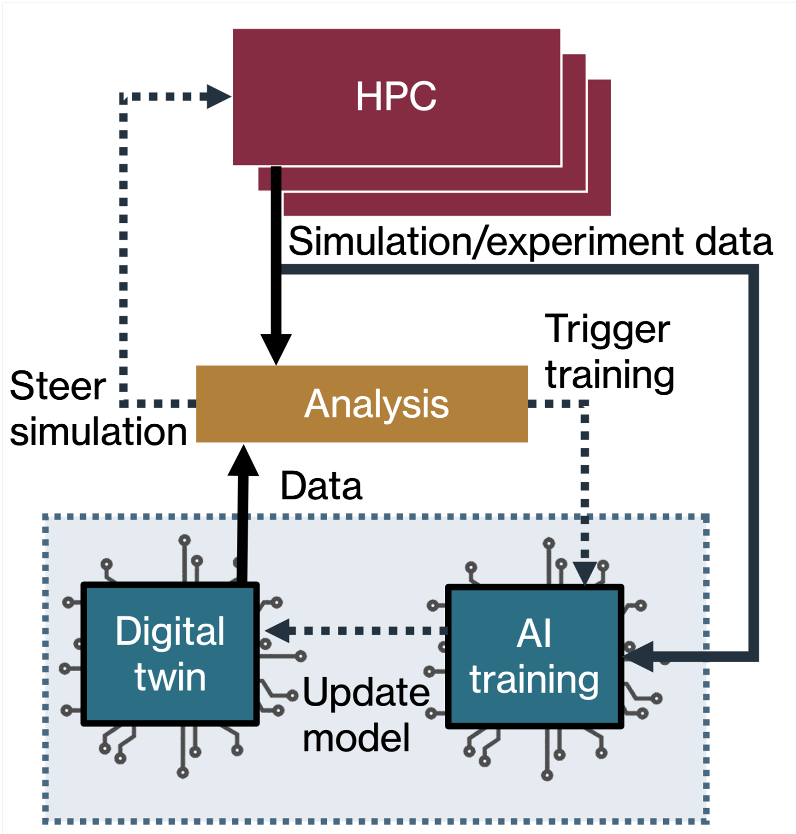
Complex I/O stack

- Filesystems have multiple software layers
  - With inter-dependencies
- Each layer has tunable parameters
- Understanding performance is tricky
  - Especially when the stack is misused

Can we avoid the storage altogether?



Large-scale workflows



Data centric approach to neural networks

- **Split** the applications into units
  - Based on their I/O needs
- **Stream** data directly to everywhere that is needed

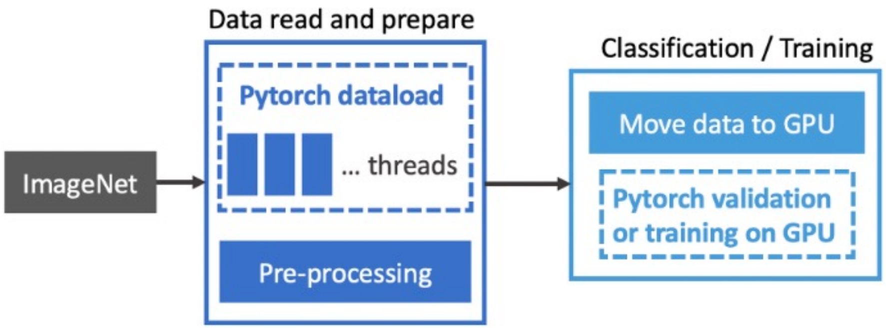
- Example
  - For training on a dataset from the PFS
    - One application reads the dataset from PFS and streams each individual data
    - The second trains the model
  - For workflows the applications are probably already split

Small test

- Imagenet Training

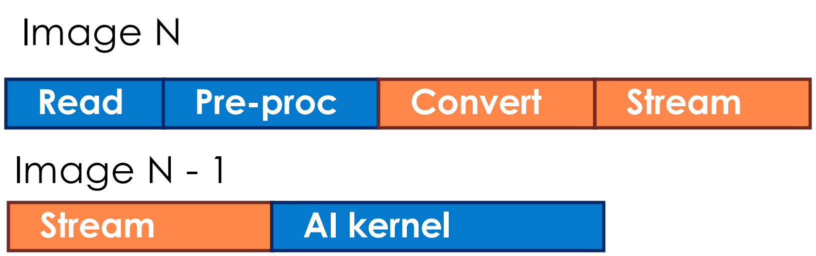
Image N

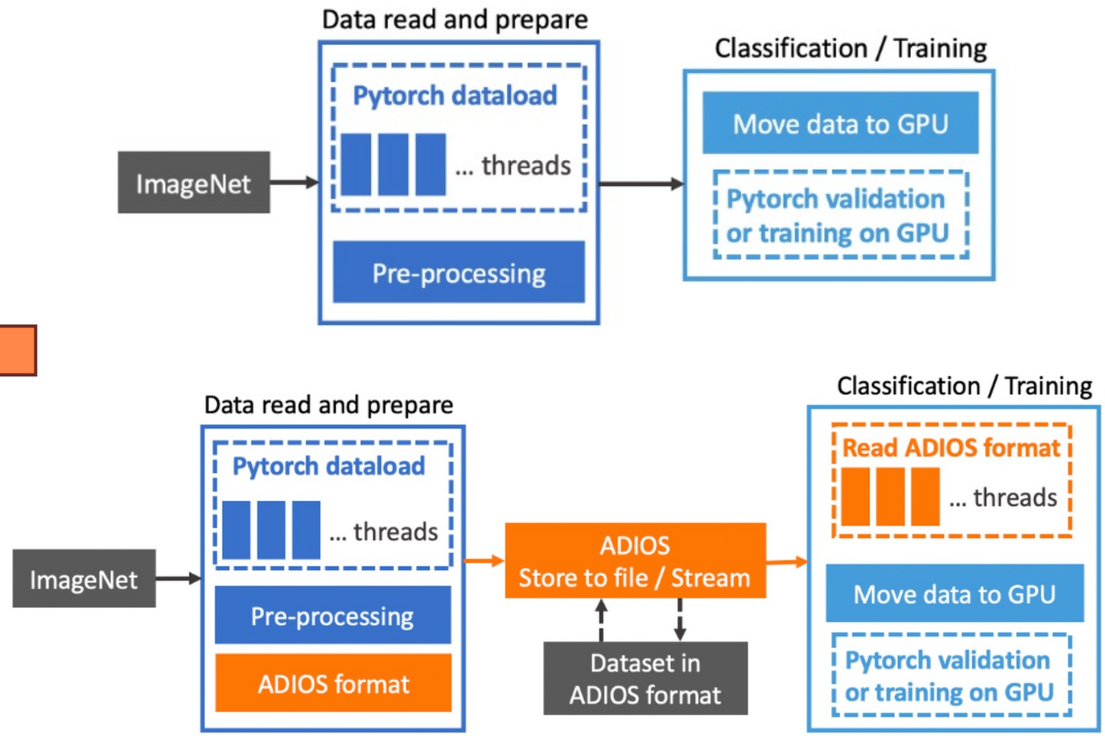
Read	Pre-proc	AI kernel



Small test

- Imagenet Training



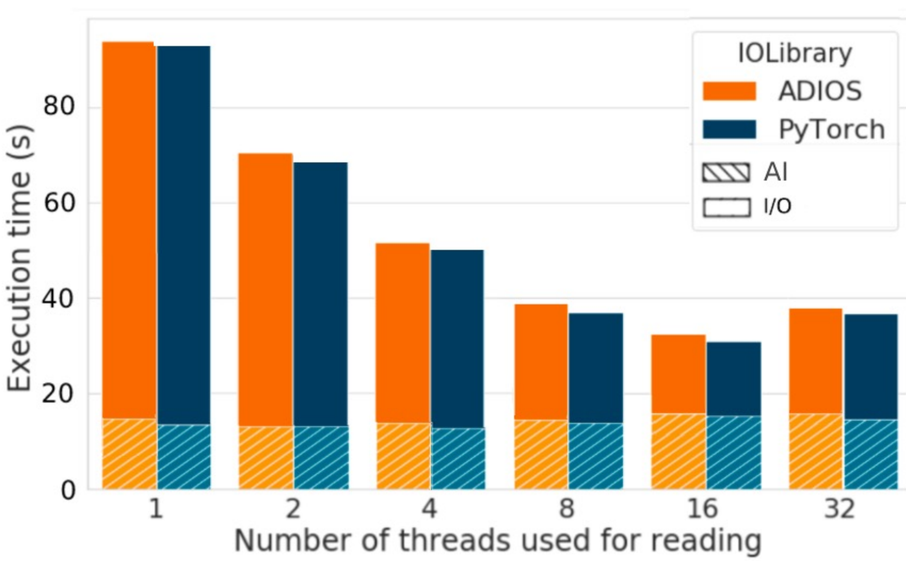


Same workflow but using two separate processes

Streaming ImageNet

- Performance of streaming
  - Less than 5% overhead
  - Using twice more resources
    - Unless we use in-line
  - For 16 threads
    - I/O time = AI kernel time
    - Initial version and streaming have the same cost

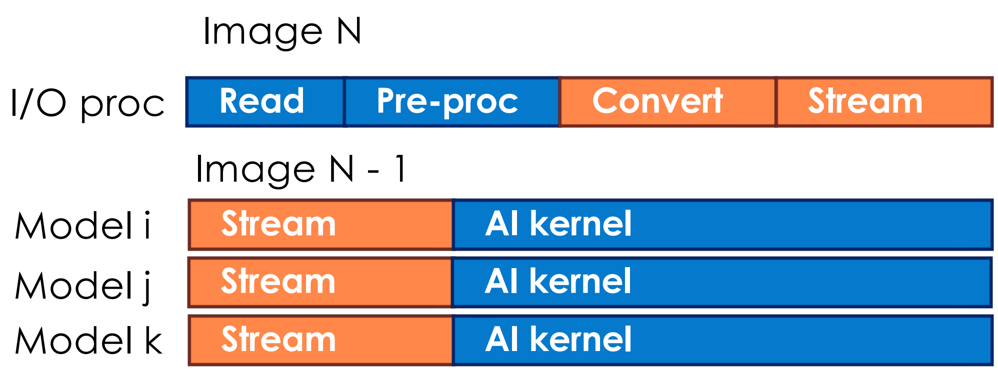
Total execution time of training one model using the initial code and the one through ADIOS

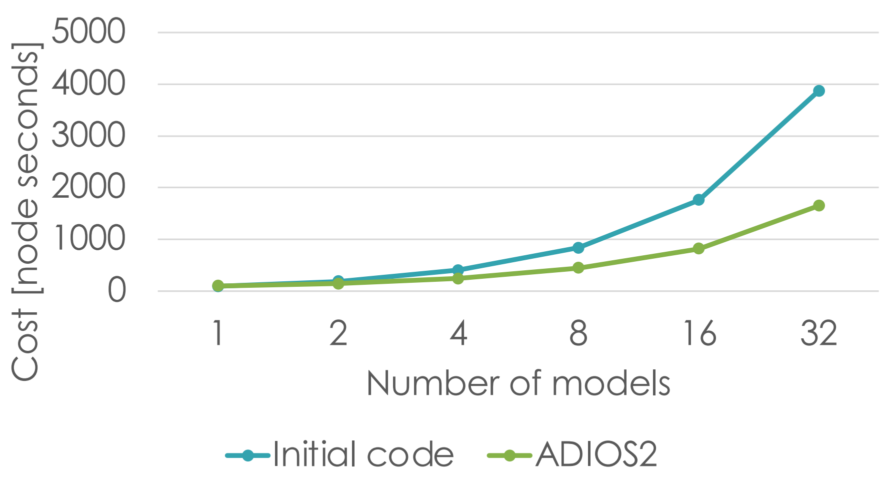


OAK RIDGE National Laboratory

Streaming ImageNet

- Training multiple models at the same time

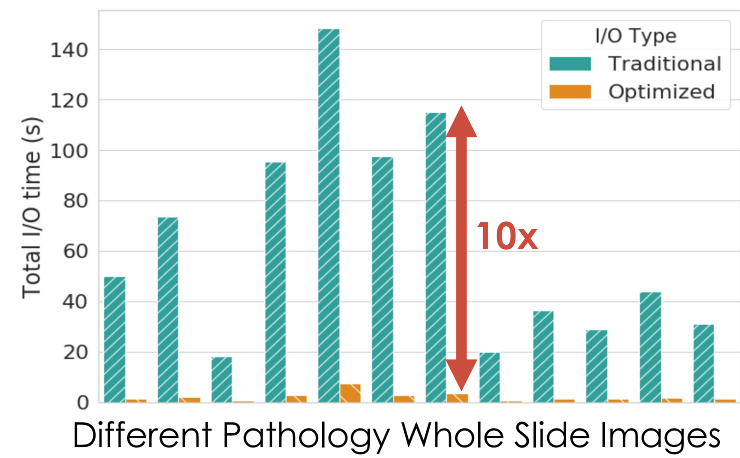




Great, if all models train on the same datasets

Moving past ImageNet: Inference on a large dataset

- Everyone that subscribe to a stream gets all the data
  - Modified the I/O library to support multiple streaming formats
    - Round Robin, On Demand
    - Future: Random shuffle

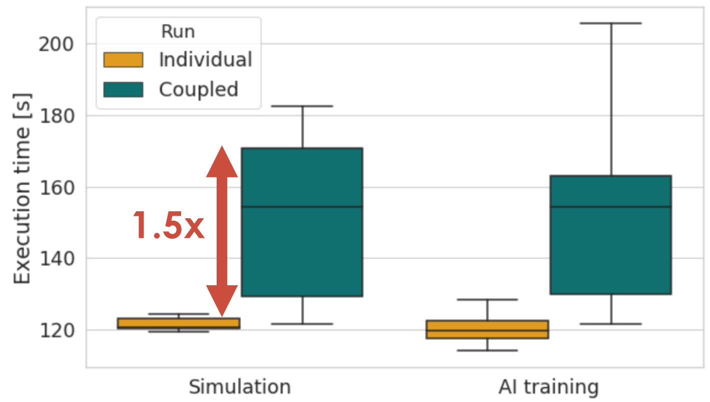


- Cancer research application
  - Classifying cancerous cells in WSI
  - VGG16 network

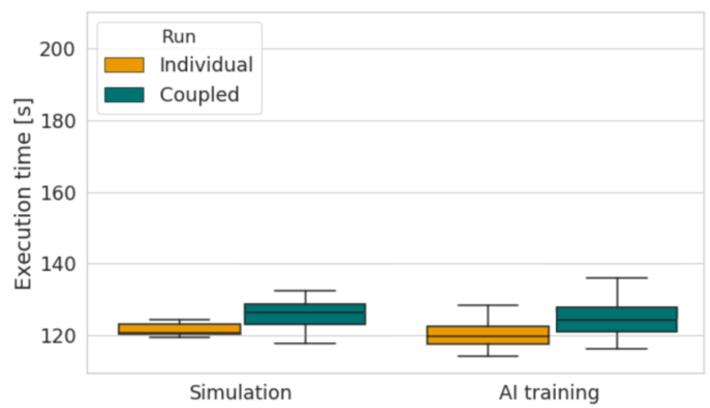
- Separating the process and streaming
  - **Speed-up** of 10x

Digital twin training

- Separate runs
  - Less than 3% performance degradation compared to separate runs
  - Less variation
  - If more models are needed
    - Overhead stays below 5% for 3 models
    - Variation increases with the number of nodes
- Throughput of 40 TFlops/node
  - On Frontier



Simulation and analysis execution time if ran separately or coupled



Execution time when streaming between coupled codes

Oak Ridge National Laboratory

Conclusions

- Many DOE proposals will develop AI / HPC workflows
  - HPC systems are not prepared for the I/O patterns of AI workflows
  - HPC I/O libraries and AI data loaders have individual views
    - Often contradicting optimizations

- Until something better occurs
  - It's better to avoid the filesystem
  - Separate workflow into units of work
    - Offload data transfer to streaming libraries

Next: run scale runs training LLMs on Frontier

Thank you

Ana Gainaru

gainarua@ornl.gov

Relevant publications

Junqi Yin et al. **Evaluation of pre-training large language models on leadership-class supercomputers**
The Journal of Supercomputing, June, 2023

Gainaru et al. **Understanding the Impact of Data Staging for Coupled Scientific Workflows**
IEEE Transactions on Parallel and Distributed Systems, 2022

Gainaru et al. **Framework for Automating the I/O of Deep Learning Methods**
In revision, Transactions on Computational Biology and Bioinformatics, 2022

Suchyta et al. **Hybrid Analysis of Fusion Data for Online Understanding of Complex Science on Extreme Scale Computers**, Cluster, 2022

Jean Luca Bez et al. **Access Patterns and Performance Behaviors of Multi-layer Supercomputer I/O Subsystems under Production Load**, HPDC 2022

Wang et al. **Improving I/O Performance for Exascale Applications through Online Data Layout Reorganization**,
IEEE Transactions on Parallel and Distributed Systems, 2021

Gainaru et al. **Profiles of upcoming HPC Applications and their Impact on Reservation Strategies**,
IEEE Transactions on Parallel and Distributed Systems, 2020

Gainaru et al. **Speculative scheduling for stochastic HPC applications**,
Proceedings of the 48th International Conference on Parallel Processing, 2019

Raghul Gunasekaran et al. **Comparative I/O Workload Characterization of Two Leadership Class Storage Clusters**, PDSW 2015