A Performance Deep Dive into HPC-AI Workflows with Digital Twins

Ana Gainaru, Norbert Podhorszki, Greg Eisenhauer, Fred Suter, Scott Klasky

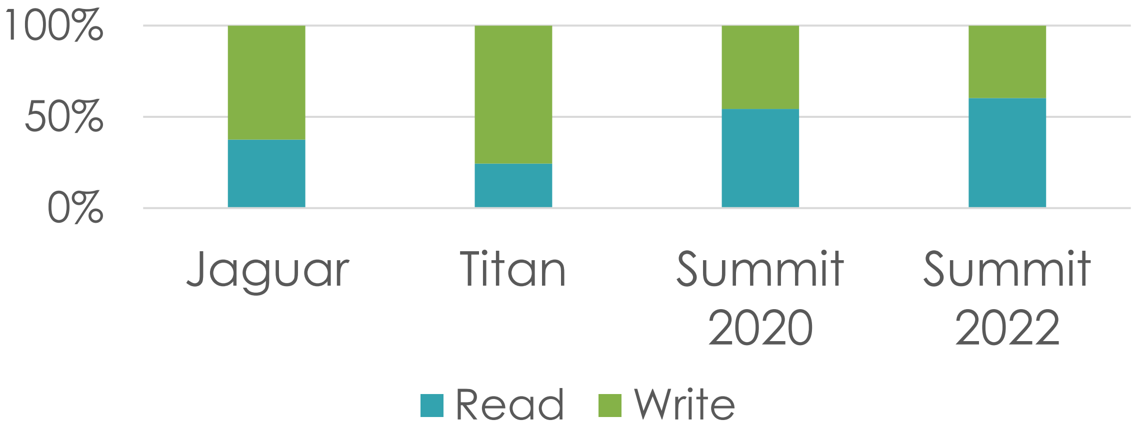
Cray User Group meeting, May 2024

Table of contents

- Evolution of HPC
  - **Evolution of HPC towards AI and digital twins (DT)**
  - **Complex workflows combining digital twins and HPC**
- Performance of AI/HPC workflows
  - **Computational and I/O patterns of DT**
  - **Models for predicting the performance**
- Offloading computation to I/O libraries
  - **Profiling workflows DT and traditional HPC**

Oak Ridge National Laboratory
gainarua@ornl.gov klasky@ornl.gov

Evolution of HPC



Frontier, 2024

Summit, 2022

Titan 2012-2019

Jaguar 2006-2012

- Shift towards read-intensive
  - From <30% reads for Titan
    - Less than Jaguar due to increase C/R routines
  - To ~60% reads for Summit 2022

Oak Ridge National Laboratory
gainarua@ornl.gov klasky@ornl.gov

AI usage on Summit

Learning to Scale the Summit: AI for Science on a Leadership Supercomputer
Wayne Joubert et al. at 2022 IEEE International Parallel and Distributed Processing Symposium Workshops

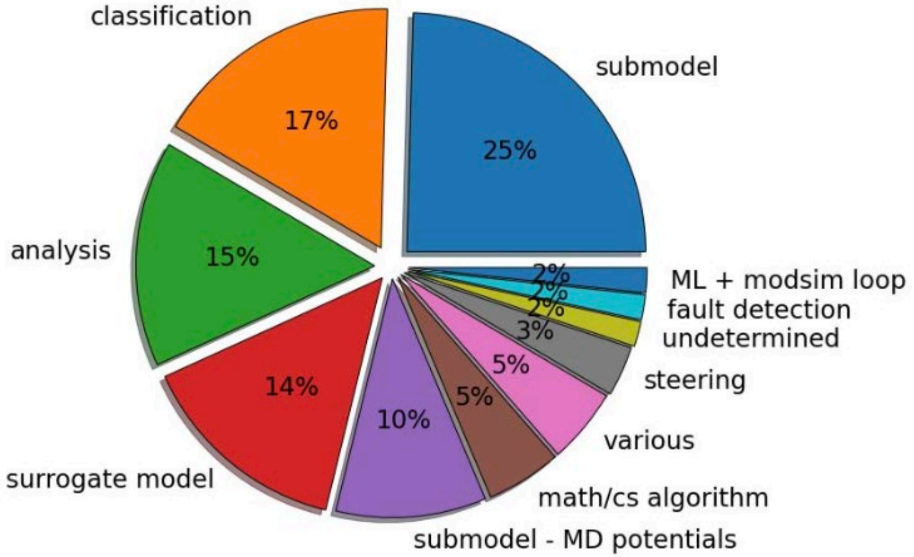
Average 2019-2022: 40% adoption of AI/ML out of which 58% DL/NN

Motif	Definition	Example
Fault Detection	Detect algorithmic or other failure in execution, send signal for automatic or manual remediation	Detect simulation defect caused by execution error
Math/CS algorithm	ML is used to enhance some mathematical computation	Solver's linear system dimension is reduced based on machine-learned parameter
Sub-model	A subset of a science computation is replaced by an ML model. molecular dynamics (MD) potentials as special case	Physics-based radiation model in a climate code replaced by ML model
Steering	Automatic steering of the direction of a computation for some internal process	ML method to guide Monte Carlo sampling to include undersampled regions
Surrogate Model	Full science model replaced by ML approximation that captures important aspects, used for speed or science understanding	Data from tokamak simulation runs used to train surrogate model
Analysis	Results from modeling and simulation (modsim) runs are analyzed by a human using ML methods	Use graph neural networks to analyze results of MD simulation
ML + ModSim loop	Both ML and traditional modsim, coupled	MD in loop used to refine deep learning model via active learning
Classification	"Pure" ML with little or no modsim used to classify some phenomenon; includes some other methods like reinforcement learning	Deep neural network inference to detect rare astro-physical event
Various	Umbrella project with multiple unrelated subprojects using possibly different kinds of AI/ML	CAAR/ESP/NESAP application readiness
Undetermined	Manner of AI/ML use is undetermined	Project is exploring AI/ML use but gives no details

- Using AI as a stand-alone code within a workflow to enhance some algorithm or logic (fault detection and Math/CS algorithm motifs)
- Using **AI to replace parts of a simulation** or parts of an algorithm (submodel)
- Using **AI to steer the simulation** either coupling the simulation with an analysis: **Steering, analysis, ML+Modsim**
- Running the AI application by itself (after it was trained to simulate some heavy weight application) either to do classification or prediction or act as a digital twin (classification, surrogate model motifs)

Oak Ridge National Laboratory
gainarua@ornl.gov klasky@ornl.gov

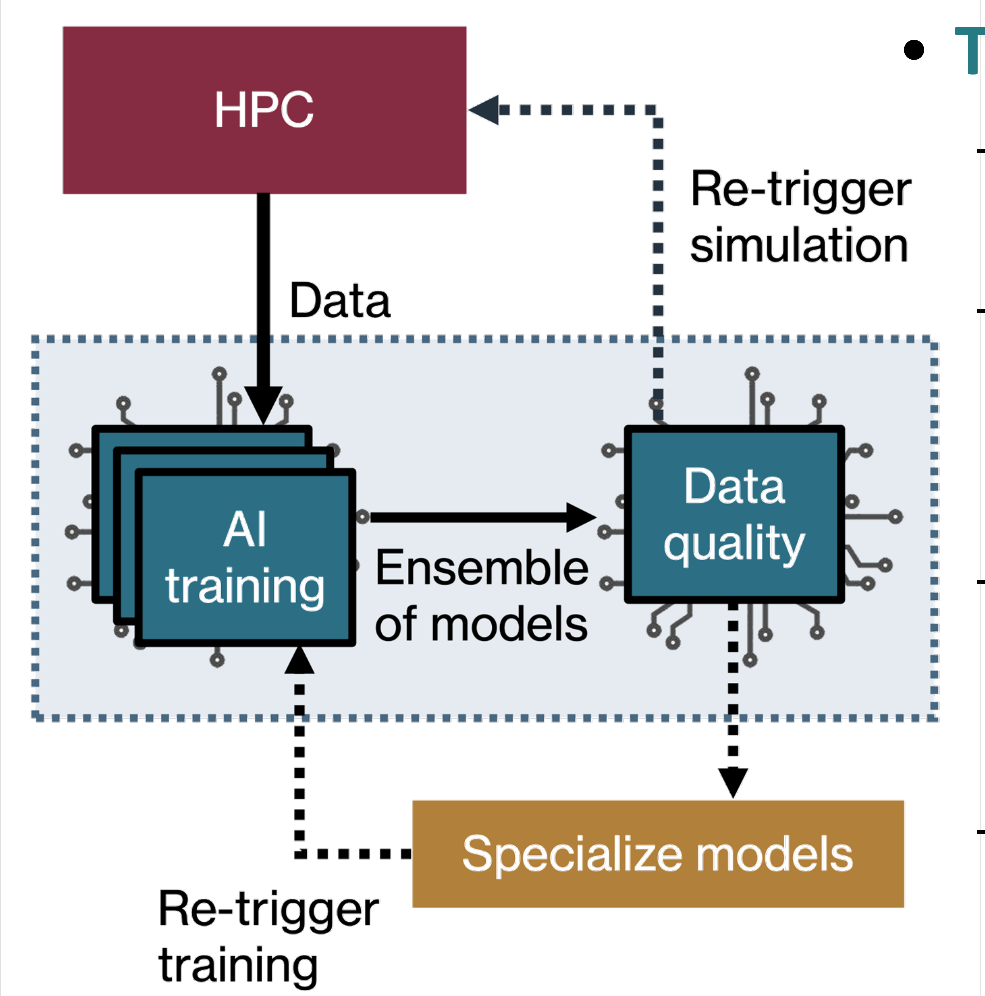
AI usage on Summit



- Parts of a code are replaced by ML: 18-20%
  - Fault tolerance, math/cs, submodel, steering
- Pure ML – 46%
  - for analysis, surrogate model, classification
- Most of the jobs are on 1 node, with just reads, so we don't see a problem, but when we run at scale, we have begun to see problems
- Coupled HPC/ML runs: 2-4%
- We are not seeing a huge problem running AI on HPC, because the AI codes run on a few nodes
- In the next slides we present two types of workflows that we expect will be executed on HPC systems in the near future that rely on digital twins

Oak Ridge National Laboratory
gainarua@ornl.gov klasky@ornl.gov

Complex workflows including digital twins



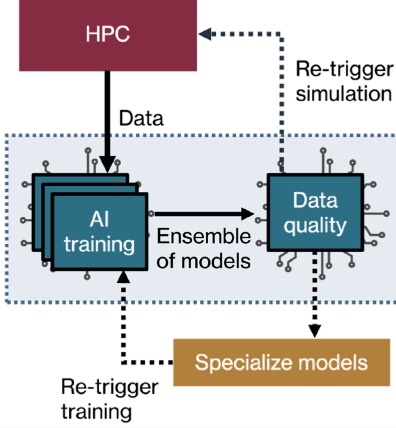
- **Training digital twins**
  - High-fidelity simulation or experiment running concurrently with the training
  - Based on model accuracy
    - Steer the simulation to produce data that is missing
    - Trigger new/Kill existing simulations in an ensemble
  - Data quality analysis might be needed
    - To decide when to trigger a training sequence for specializing a model
  - The AI training could be a hyperparameter search code

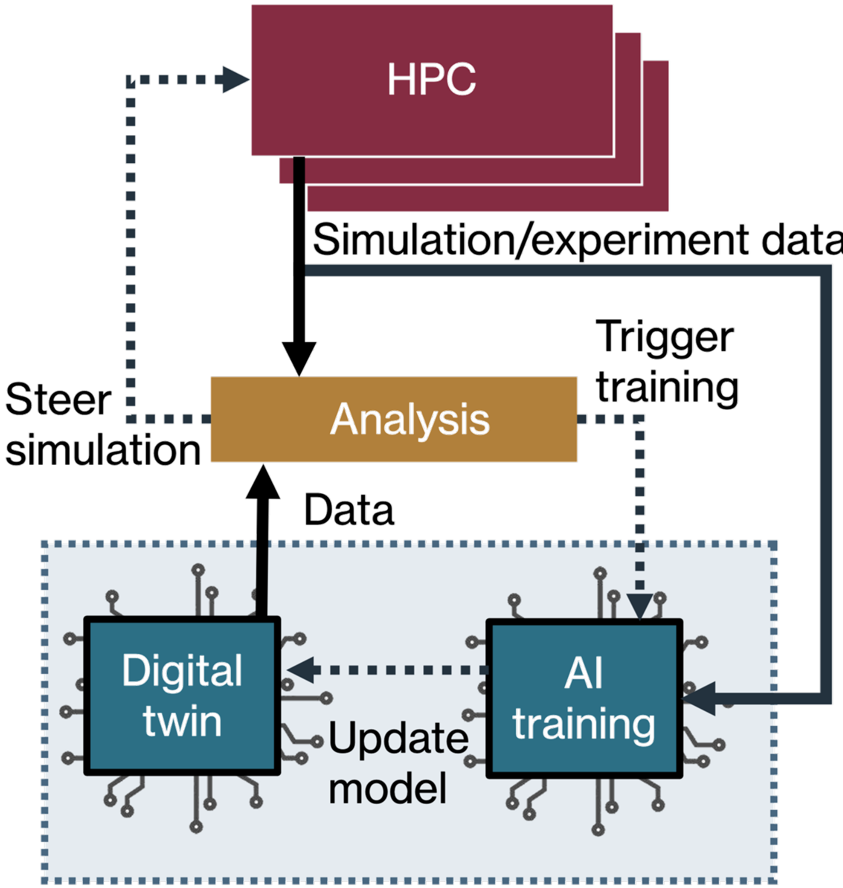
- If specialized models are needed, the workflow will used another code that analyzes the accuracy of models and the quality of the data and re-trigger a new training logic on a sub-set of the data or on models with more parameters

gainarua@ornl.gov klasky@ornl.gov

Complex workflows including digital twins

- **Using digital twins to steer HPC simulations**
  - High-fidelity simulation/ensemble simulations running concurrently with the digital twin
  - Analysis code receiving data from both
    - Steer the simulation based on predictions from the digital twin
    - Trigger training based on mismatch between the simulation and digital twin output
  - Training could be done offline or on the fly
- The main difference between this workflow is that the analysis looks for differences from the DT and simulation to either retrain or steer the sim





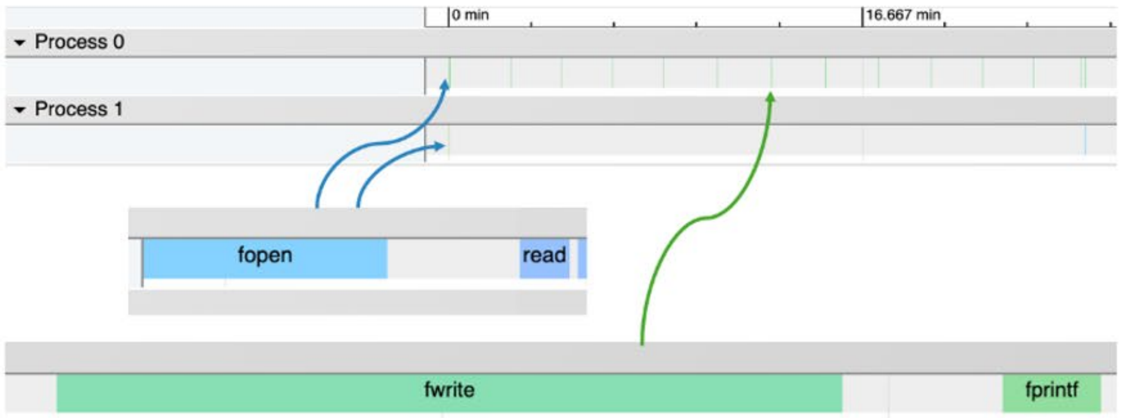
Oak Ridge National Laboratory
gainarua@ornl.gov klasky@ornl.gov

I/O patterns for workflows

- Likely I/O transfers through storage

Simulation side

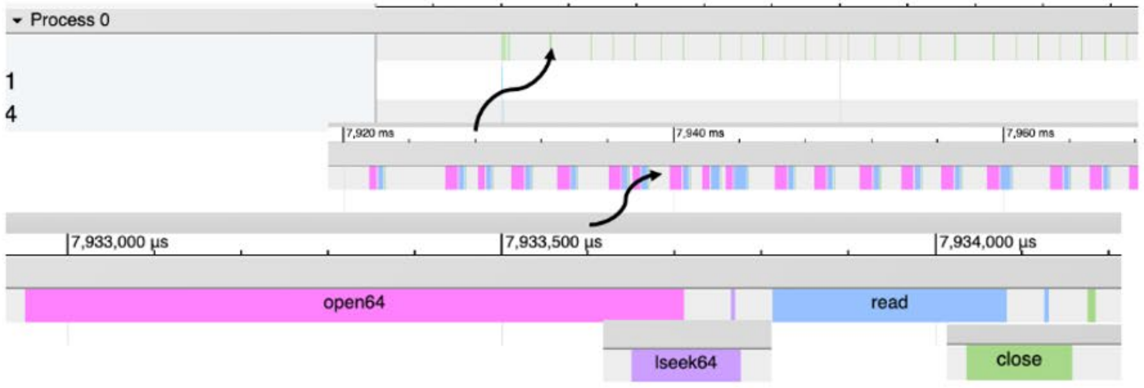
- Initial open and read ops
- Followed by periodic write ops to multiple files



Two processes of LAMMPS

AI side

- Multiple open/seek/read/close ops within a short timespan

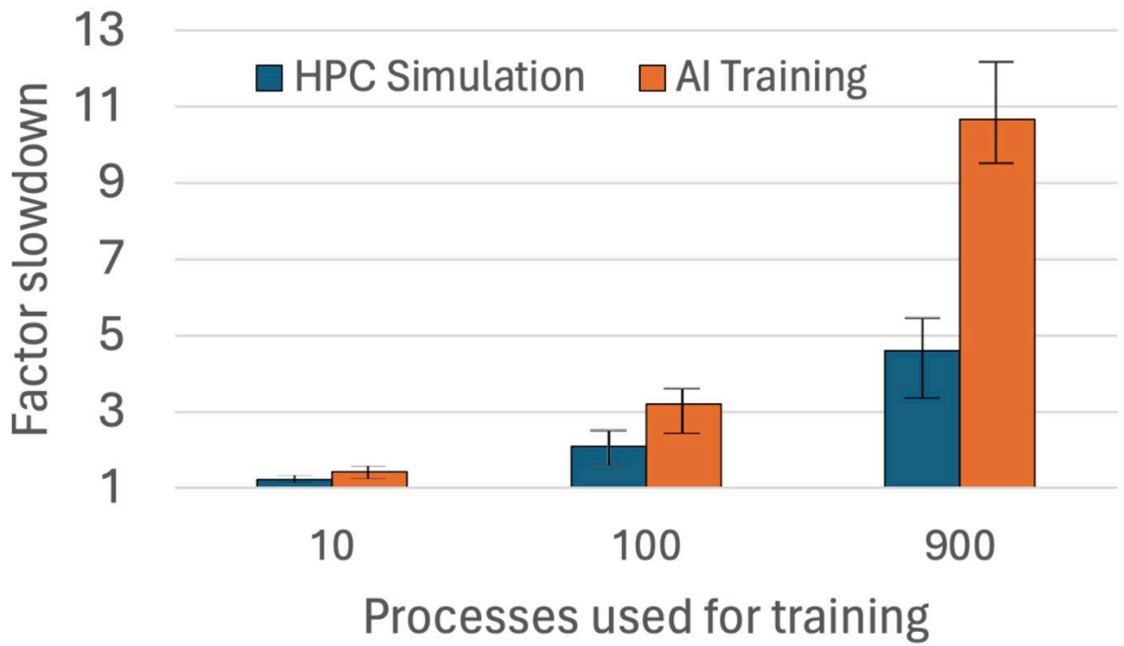


Training a 2-layer DNN on LAMMPS data
One process multiple threads

OAK RIDGE National Laboratory
gainarua@ornl.gov klasky@ornl.gov

I/O interference

- Interference on Frontier when writing/reading concurrently on the same dataset
  - Slowdown compared to running sequential
  - For 17 nodes (900 processes) there is a 10% slowdown on the training code and 5% on the simulation
  - Larger models/larger data will likely have less small read ops
  - Running at larger scale will likely have high interference
- There is a slowdown when we couple the codes because of the I/O interference



OAK RIDGE National Laboratory

Models

- Existing performance models
  - Many performance / roofline models for HPC simulations exist
  - Several performance models for training exist as well
  - Neither is considering interaction patterns
- Example performance model used for training
  - Predicting the training time for LLM models is based on the parameter size P(T) of the model and the FLOPS of the machine
  - Predictions for Summit and Frontier

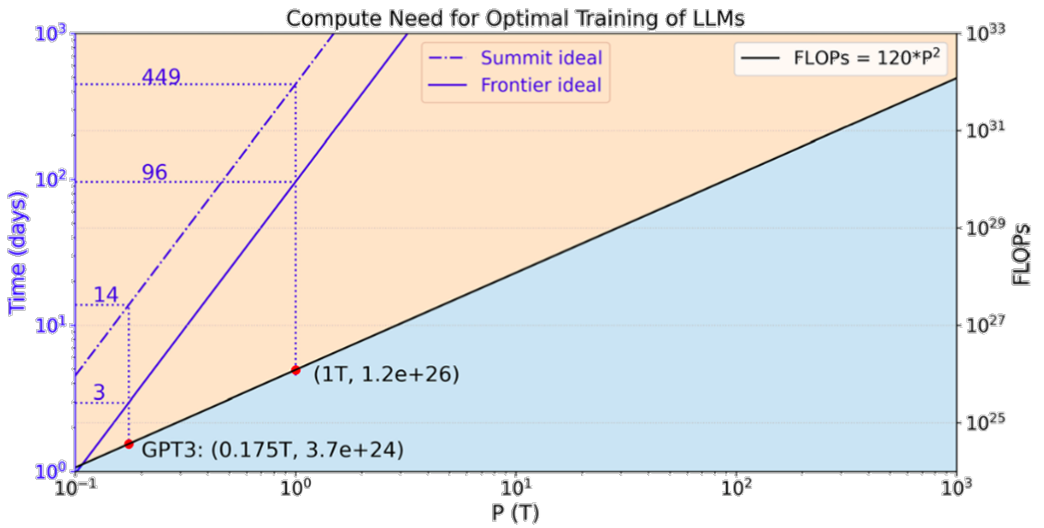
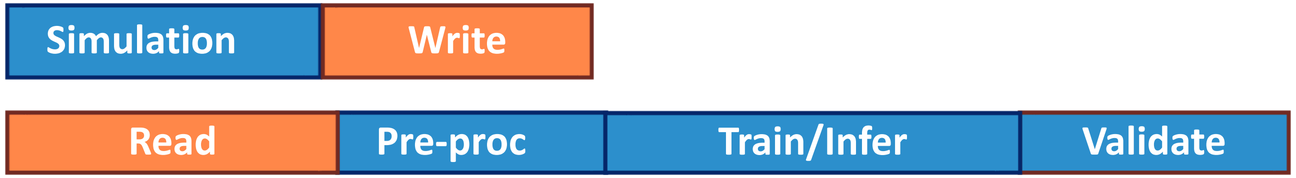


Figure from: Evaluation of pre-training large language models on leadership-class supercomputers
Junqi Yin, Sajal Dash, John Gounley, Feiyi Wang, Georgia Tourassi in The Journal of Supercomputing, June, 2023

gainarua@ornl.gov klasky@ornl.gov

Workflow performance model

- Model
  - Extend AI and HPC existing models, include terms for interaction (orange boxes)



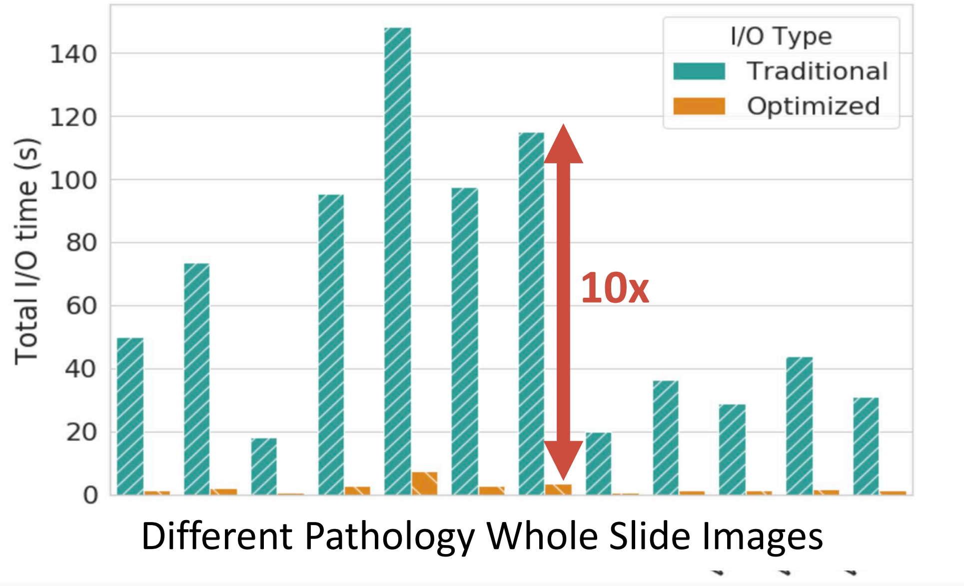
- Offloading the data management to a common I/O layer
  - Allows the I/O layer to choose the best algorithm
- Taking it further
  - Offload pre-processing
  - Offload caching decision
  - Offload steering decision
- If we could breakdown the simulation and AI into basic building blocks, we could make the I/O layer aware of them and offload a lot of the logic to the I/O layer, this optimizes the execution

OAK RIDGE National Laboratory

gainarua@ornl.gov klasky@ornl.gov

Inference on a large dataset

- Results for offloading data management to the I/O layer
  - Allows the I/O library to choose the format of pre-processed 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 Whole Slide Images (WSI)
  - VGG16 network



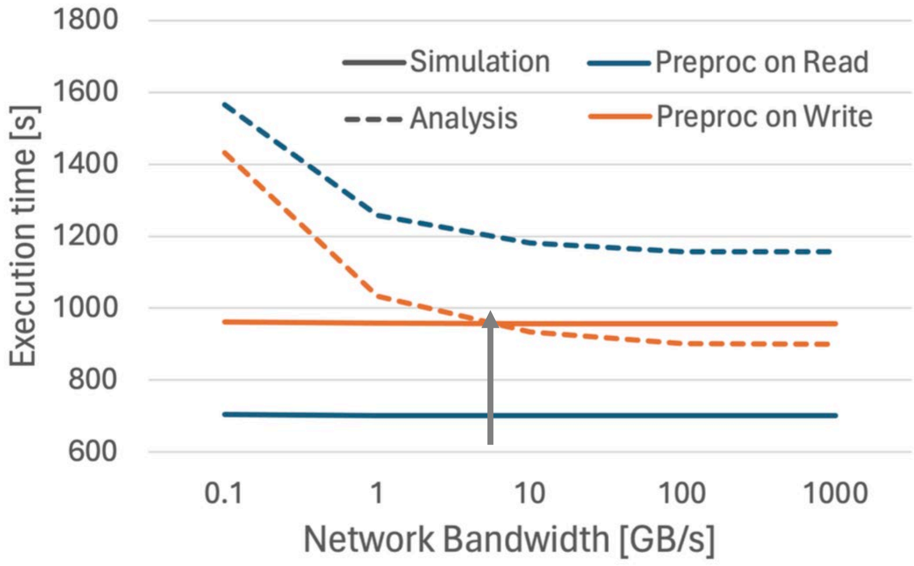
- Separating the process and streaming
  - **Speed-up** of 10x

OAK RIDGE National Laboratory
gainarua@ornl.gov klasky@ornl.gov

Offload pre-processing

- Example: strong coupling of LAMMPS and AI application; training a DT for LAMMPS
  - In general the analysis can be training or classification/prediction, and AI pre-processing can be offloaded
  - Use performance model to decide when to pre-process the generated data (read or write time)

- In our example, its better to preproc on the write



Pre-process on the reader

Pre-process on the writer

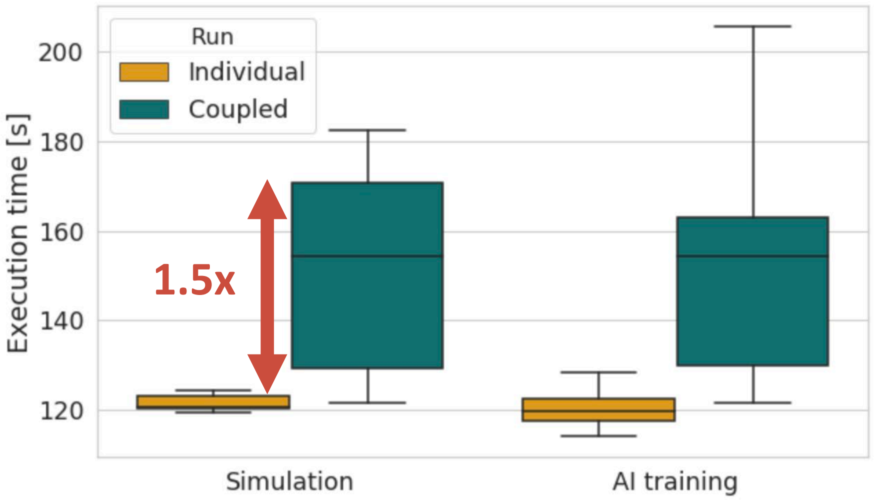
Best so that no one writes but need. We need to pre-process on the write, If we can get at least 1 GB/s

Oak Ridge National Laboratory
gainarua@ornl.gov klasky@ornl.gov

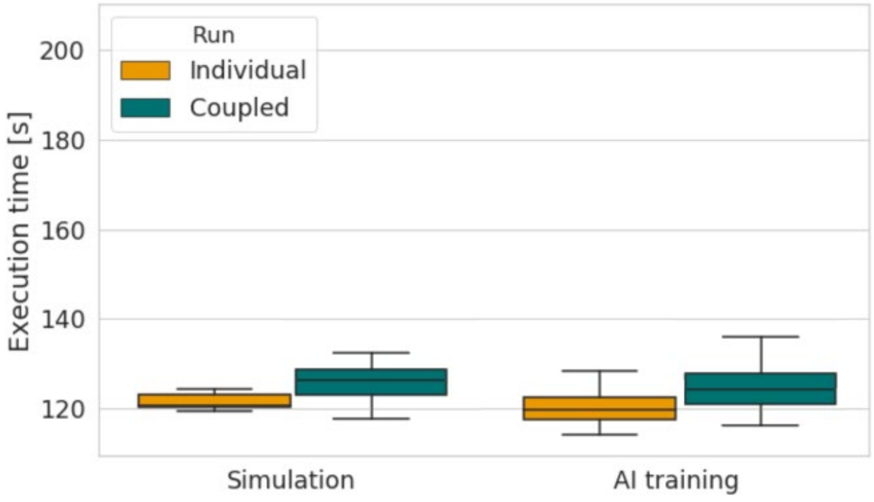
Digital twin training

- Ex. Running LAMMPS and DT concurrently
- 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
- Its better to avoid the file system, and use the pre-processor on the write,
- We use ADIOS with SST to stream data, which speeds up the overall execution of the coupled codes

Throughput of 40 TFlops/node on Frontier



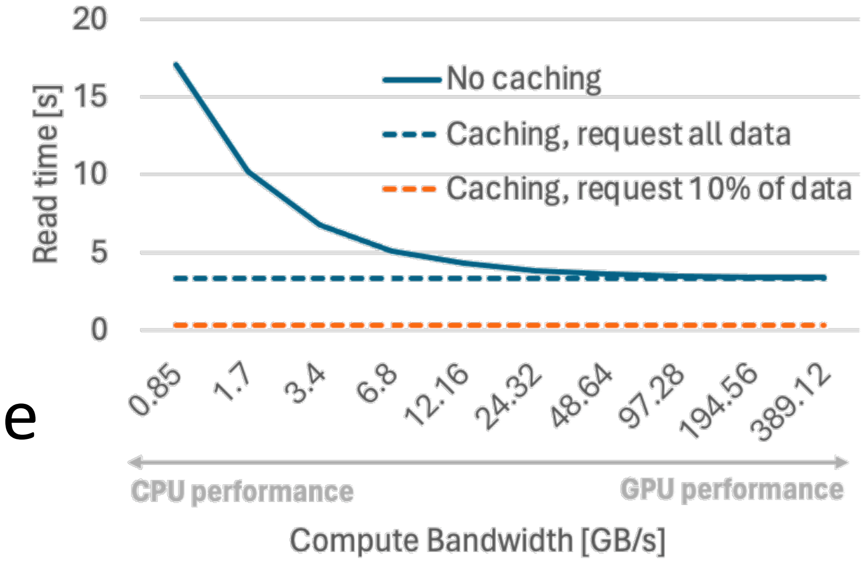
Simulation and analysis execution time if ran separately or coupled



Execution time when streaming between coupled codes

OAK RIDGE National Laboratory
gainarua@ornl.gov klasky@ornl.gov

Offloading caching



- In addition to deciding where to compute the pre-process and what algorithm to use for data transfer, the next question is **whether** to cache the intermediate data or recompute
- The pre-processed data **could be saved** on the writer and each reader will bring its piece of data or recompute for every read request
- When recomputing is expensive (CPU performance) caching pre-processed data decreases the read time up to 6X as shown on the lhs of the graph
- When recomputing is cheap (GPU) caching is beneficial if only a % of data is read

Oak Ridge National Laboratory
gainarua@ornl.gov klasky@ornl.gov

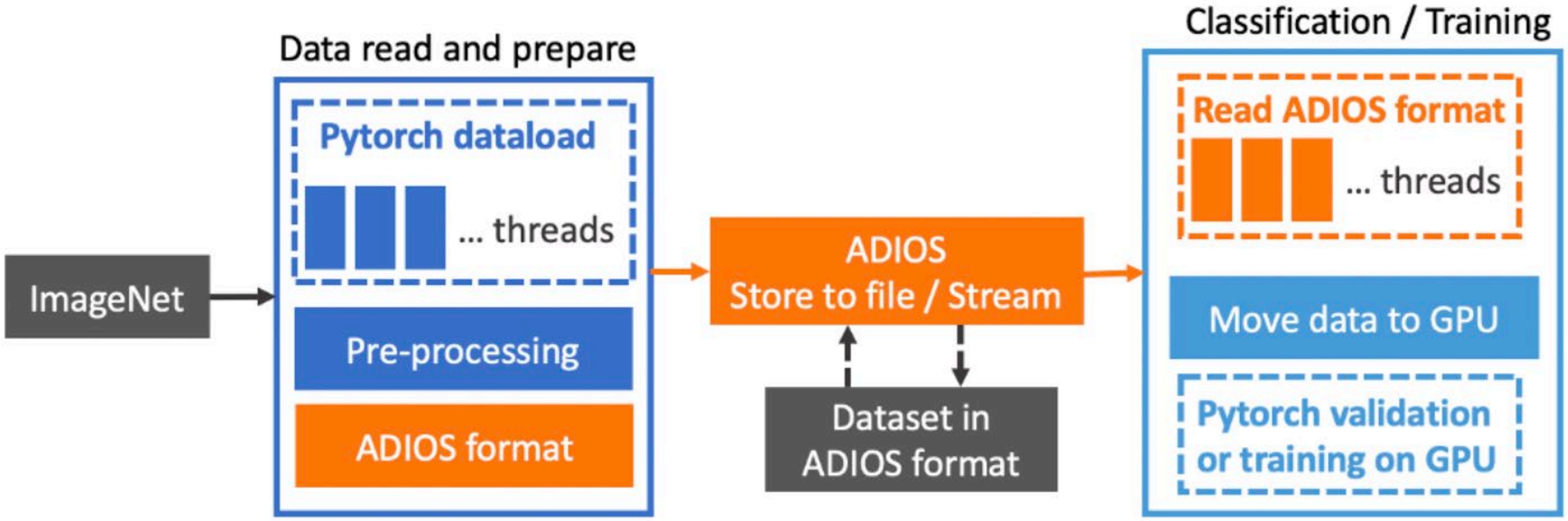
Data centric approach to neural networks

- **Split** the applications into units, based on their I/O needs
- **Offload** pre-processing and caching to the I/O layer
- **Stream** data directly to everywhere that is needed
- Example
  - For training on a dataset from the parallel file system (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

Oak Ridge National Laboratory
gainarua@ornl.gov klasky@ornl.gov

Overhead of splitting the training for Whole Slide Imaging

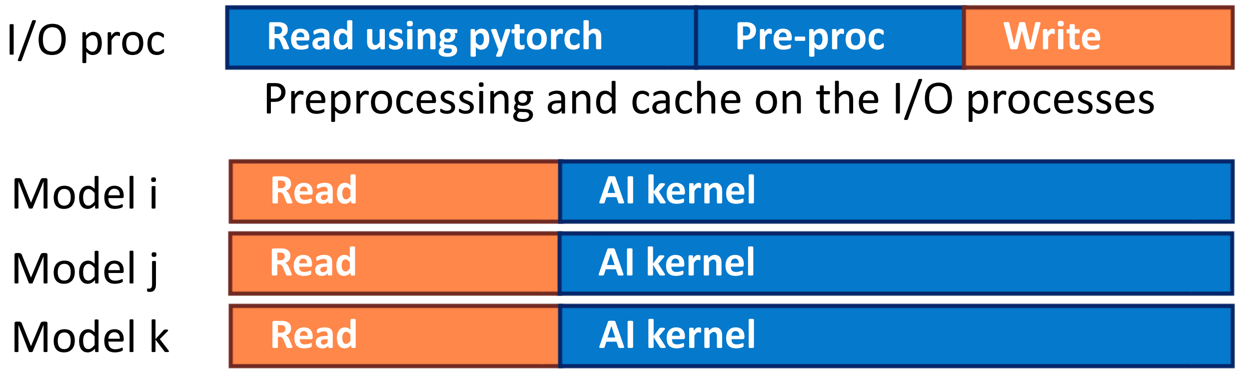
- Performance of streaming
  - Less than 5% overhead
  - Using the same resources
  - Pre-processing and training are split into 2 separate applications
    - Training uses GPU / pre-process CPU

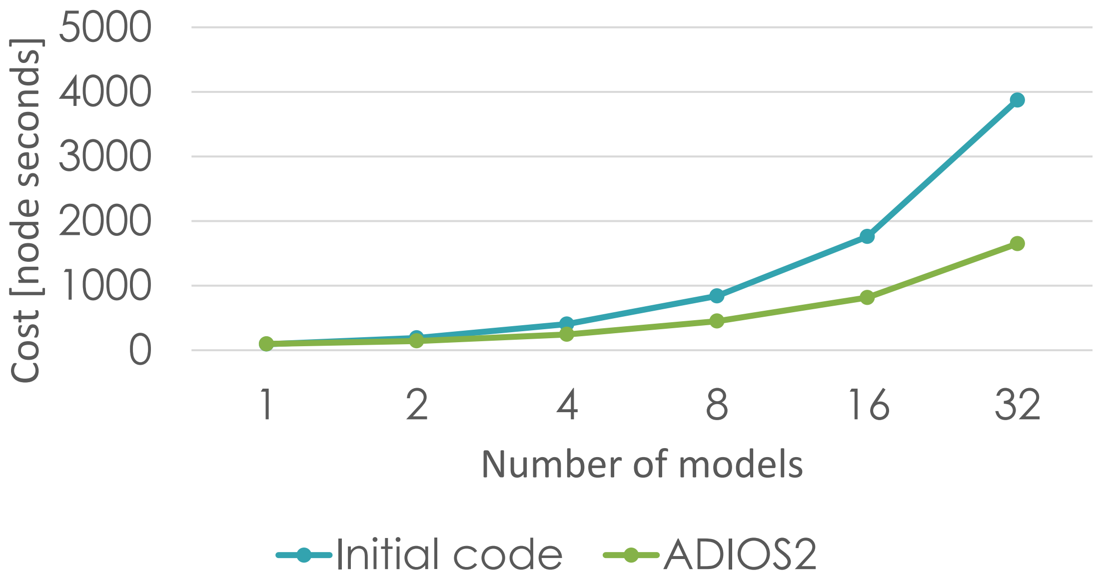


gainarua@ornl.gov klasky@ornl.gov

Hyperparameter search on Frontier for LAMMPS

- Training multiple models at the same time
- Summary: we can use the techniques from the previous slides to train multiple models, and by using ADIOS + SST we can get a 2X speedup for 32 models





gainarua@ornl.gov klasky@ornl.gov

Conclusions

- AI / HPC workflows are the future aps on leadership computing facilities
  - HPC I/O libraries and AI data loaders have individual views
    - Often contradicting optimizations
  - It’s better to avoid the filesystem
  - Separate workflow into units of work

- Offload data transfer to good I/O libraries (ADIOS 2)
  - I/O layer can decide when and where to compute intermediate representation of data and where to cache it
  - Models need to consider interaction patterns

gainarua@ornl.gov klasky@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

gainarua@ornl.gov klasky@ornl.gov

Thank you

gainarua@ornl.gov klasky@ornl.gov