SAND2022-4641C

Proxy Apps for X (Co-design, Real life and Fun!)

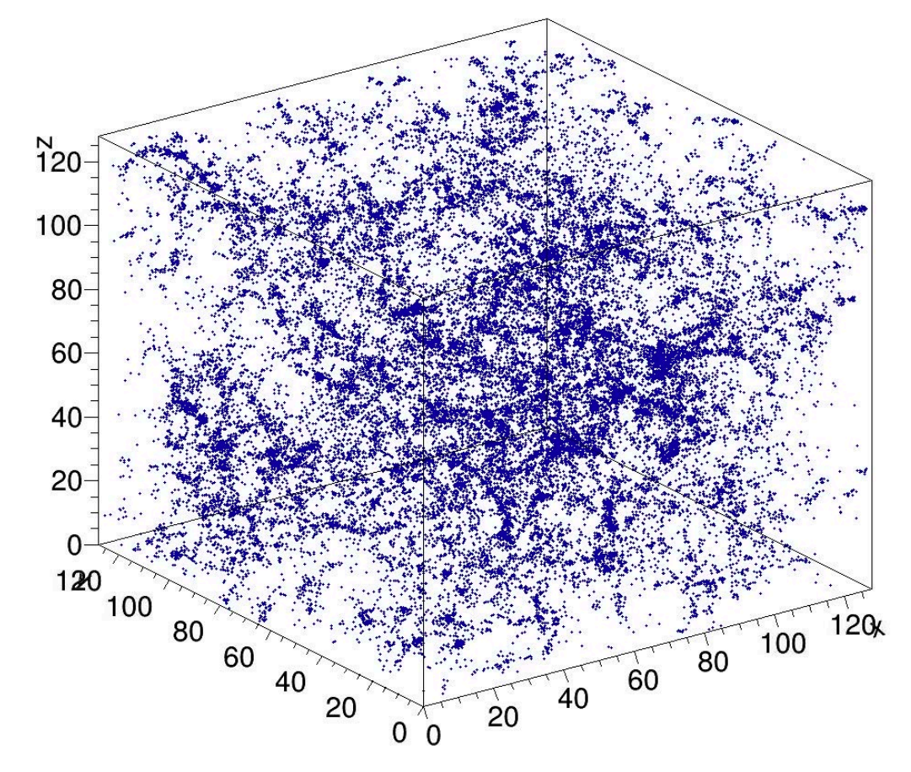
PI: Jeanine Cook (SNL)
Team: Omar Aaziz (SNL), Mayagoitia Alvaro (ANL), Lund Amanda (ANL), Eiffert Brett (ORNL), Zhang Chen (ORNL), Fletcher Graham (ANL), Watson Gregory (ORNL), Logan Jeremy (ORNL), Christoph Junghans (LANL), Peter McCorquodale (LBL), Robert Pavel (LANL), Fackler Philip (ORNL), Vinay Ramakrishnaiah (LANL), Balakrishnan Ramesh (ANL), Patel Saumil (ANL), Courtenay Vaughan (SNL), Godoy William (ORNL), Kim Youngdae (ANL), Alexeev Yuri (ANL)

Mission

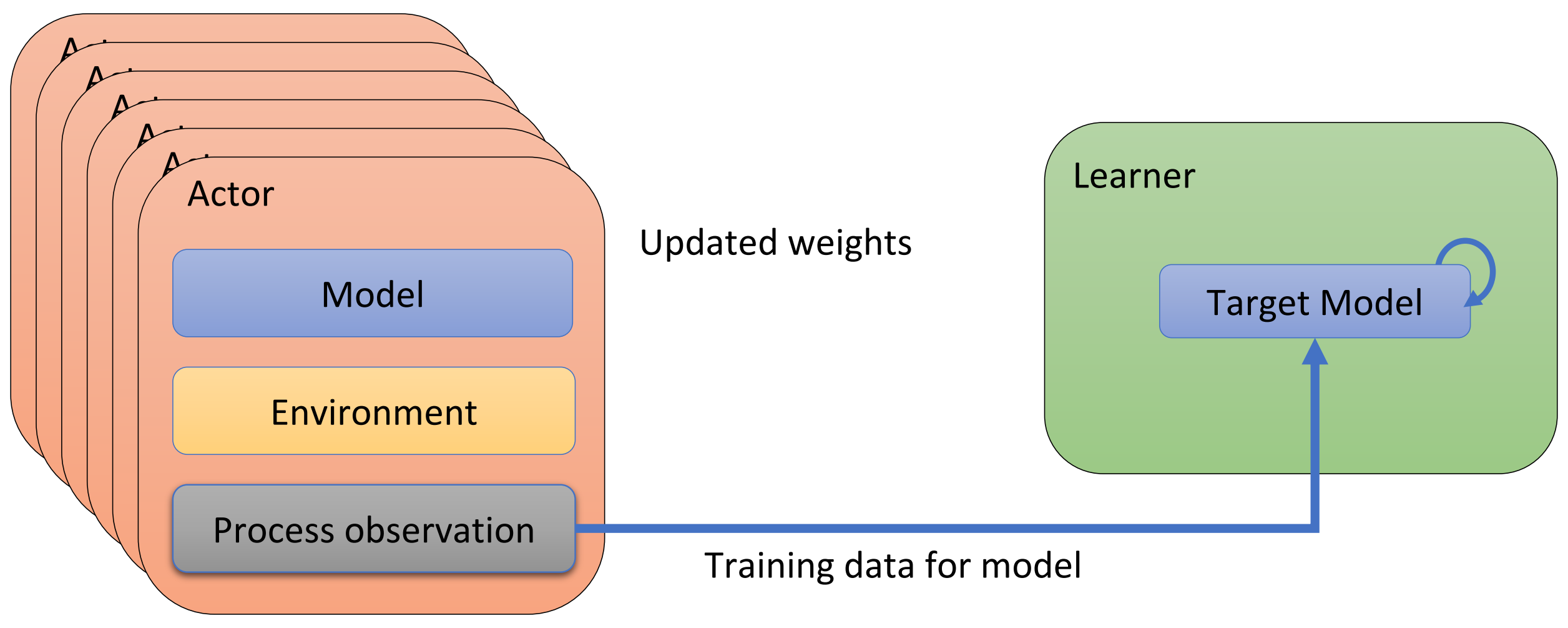
To curate a suite of proxy applications that are representative of the intended characteristics of their respective parent applications and are easy to obtain and use. Characteristics include hardware bottlenecks (e.g., memory, computation, communication) and programming models.

Machine Learning Proxy Suite

- Proxies for ML applications require a somewhat different approach
  - ML problems frequently present a data centric approach rather than an algorithm-centric one
  - May be necessary to simplify the dataset instead of the algorithm
- ML proxies are different from traditional proxies in other important ways
  - ML workloads often rely on a complex set of third-party dependencies such as PyTorch or TensorFlow
    - Contrary to the usual principles of simple-to-build proxies by avoiding dependencies
  - ML space is fortunate to have tools such as Spack to manage dependencies
  - Rapid pace of development requires ML proxies to be nimble enough to adapt to changes



Example simulation of dark matter in the universe, evolved over 3 billion years used as input to the Cosmoflow network [5].



miniRL architecture. Image courtesy: ExaLearn Control Team

Principles of Machine Learning Proxies

With the above-mentioned differences in mind, ML proxies are selected based on two sets of criteria:

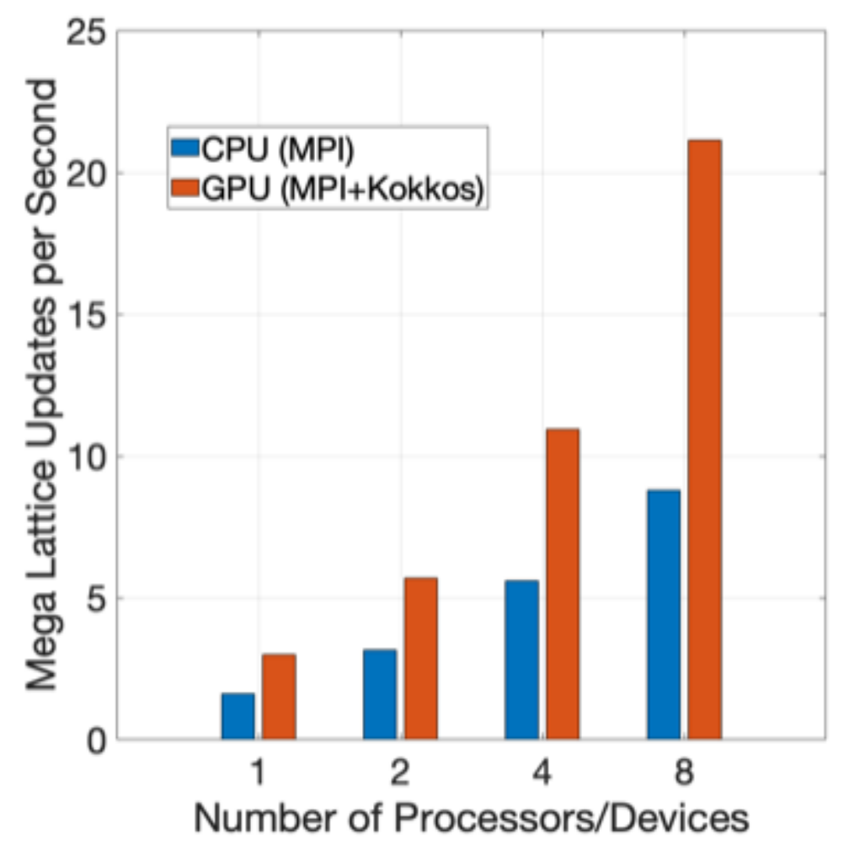
1. The suite must embody the core purposes and uses of proxy apps: Hardware and software co-design, programming model exploration and innovation, development of numerical methods and algorithms, optimization and benchmarking, and education
2. The suite must cover the breadth of the ML application space. The ML suite will encompass different ML proxy apps chosen to represent the basic learning paradigms associated with ML.

Machine Learning Proxies

The following proxies are included in the current version of the ML proxy suite:

- **miniGAN -** A Generative Adversarial Network (GAN) proxy for related ML applications in cosmology, such as CosmoFlow and ExaGAN, and in wind energy, such as ExaWind.
- **miniRL -** A reinforcement learning (RL) proxy application derived from the Easily eXtendable Architecture for Reinforcement Learning (EXARL) framework, used for control and optimization of applications or experiments.
- **CRADL -** Concurrent Relaxation through Accelerated Deep Learning (CRADL) performs inference, with a trained machine learning algorithm, on mesh geometry data from hydrodynamics simulations.
- **Cosmoflow-Benchmark -** Cosmoflow-Benchmark is an implementation of CosmoFlow 3D convolutional neural network for benchmarking.
- **Mlperf-DeepCam -** MLperf-DeepCam is a PyTorch implementation for the climate segmentation benchmark, based on the Exascale Deep Learning for Climate Analytics codebase.

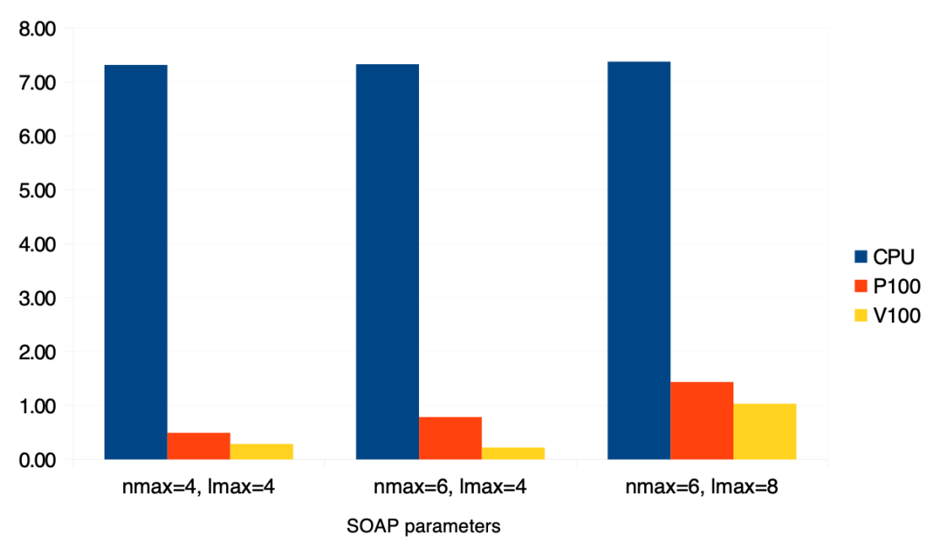
Argonne ProxyApps



IMEXLB: Lattice Boltzmann ProxyApp for multiphase flows

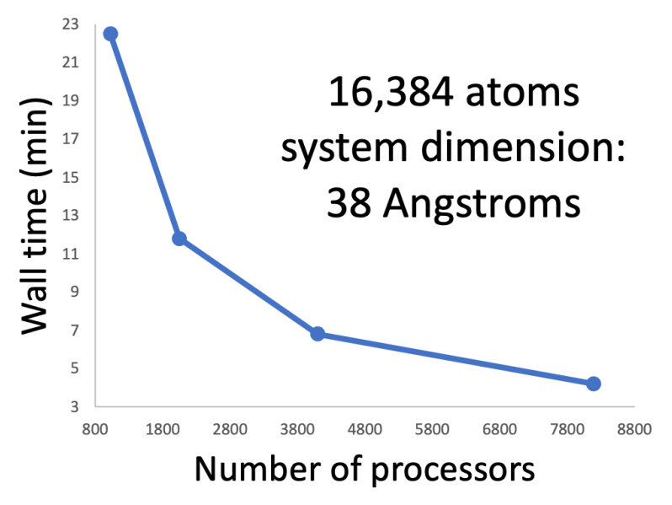
Intra-Node performance results for IMEXLB on ThetaGPU machine. The plot shows good strong scaling and exhibits speed-up (~2x) with the GPU

calculation over the CPU-only calculation.



MiniGap: Portable Machine Learning Proxy App for Molecular Properties

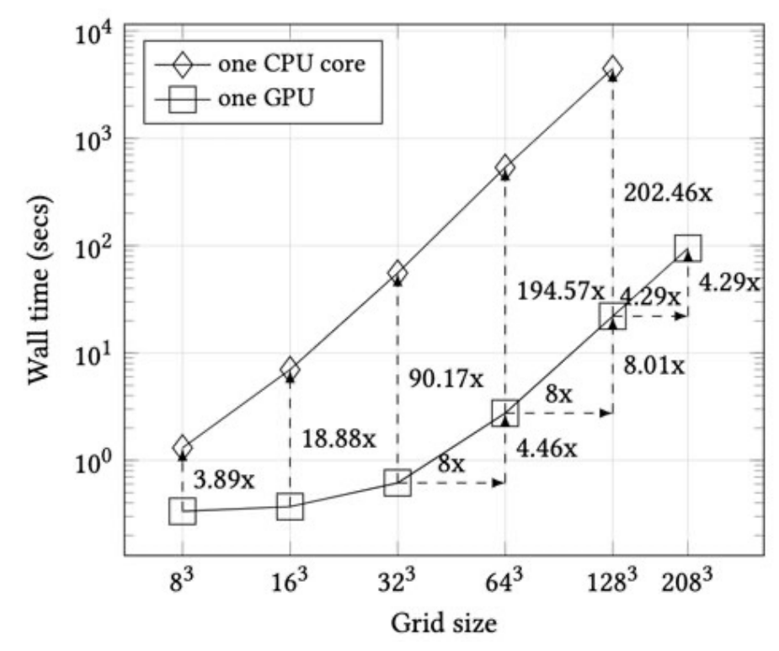
Development of a Gaussian Process Regression code and ML pipelines using using multiple programming models
Linear algebra and other ML core operations rely on elemental TensorFlow operations and other community-supported projects
This code could be used as a template for other projects with actual production codes
Code tested in CPU/GPU in AI test beds and TPU in Google cloud



Strong scaling of PGAS-FMO on Intel cluster.

PGAS-FMO: ProxyApp for Fragmented Molecular Orbital Calculations

- Parallel Global Address Space (PGAS) is a key paradigm in the design of massively parallel distributed data applications.
- Fragment Molecular Orbital (FMO) is an *ab initio N*-body method targeting systems with thousands of atoms.
- Proxy features a PGAS interface with GET, PUT capability (see, https://github.com/gdfletcher/pgas-fmo).



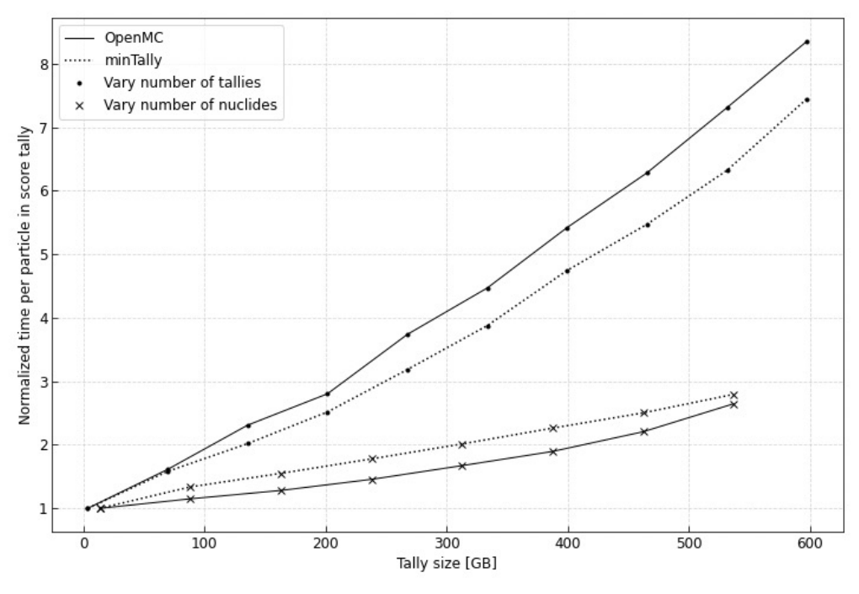
HyPar-GPU: Compressible Finite Volume/Finite Difference ProxyApp

- HyPar-GPU, that use the fifth-order WENO and explicit time integration schemes to achieve more than 200x acceleration on Nvidia's V100 GPUs.
- Lexicographic thread configuration provides no idle threads. Compared to the case where we use a tiling thread configuration, we achieved more than 1.3x acceleration.
- GPU only memory access pipeline enables us to fully utilize GPUs by removing the need for expensive data transfers between CPUs and GPUs.
- Coalesced memory access significantly reduces memory transactions by greater than 3.6x.

Backend Name	Device	Time (seconds)	Speedup
NumPy_Merged	CPU	383	0.64×
NumPy (baseline)	CPU	246	1.00×
CuPy	GPU	6.7	36.7×
CuPy_Merged	GPU	5.6	43.9×
NumPy + CuPy	Mixed	2.1	117×
NumPy_Merged + CuPy_Merged	Mixed	1.4	**176**×

Qtensor: ProxyApp for Tensor Network Quantum Circuit Simulator

- Developed a proxy app to study how to contract efficiently tensors on GPUs and CPUs using various backends to simulate efficiently quantum circuits using Argonne-developed tensor network simulator QTensor.
- Developed backends and dynamic load balancing technique to contract tensors on CPU and GPU devices to simulate quantum circuits with 176x speed up
- This work is accepted for publication in SC21 workshop proceedings: Lykov, D., Chen, A., Chen, H., Keipert, K., Zhang, Z., Gibbs, T. and Alexeev, Y., Proceedings of SC21 IEEE/ACM Second International Workshop on Quantum Computing Software (QCS) (pp. 27-34)

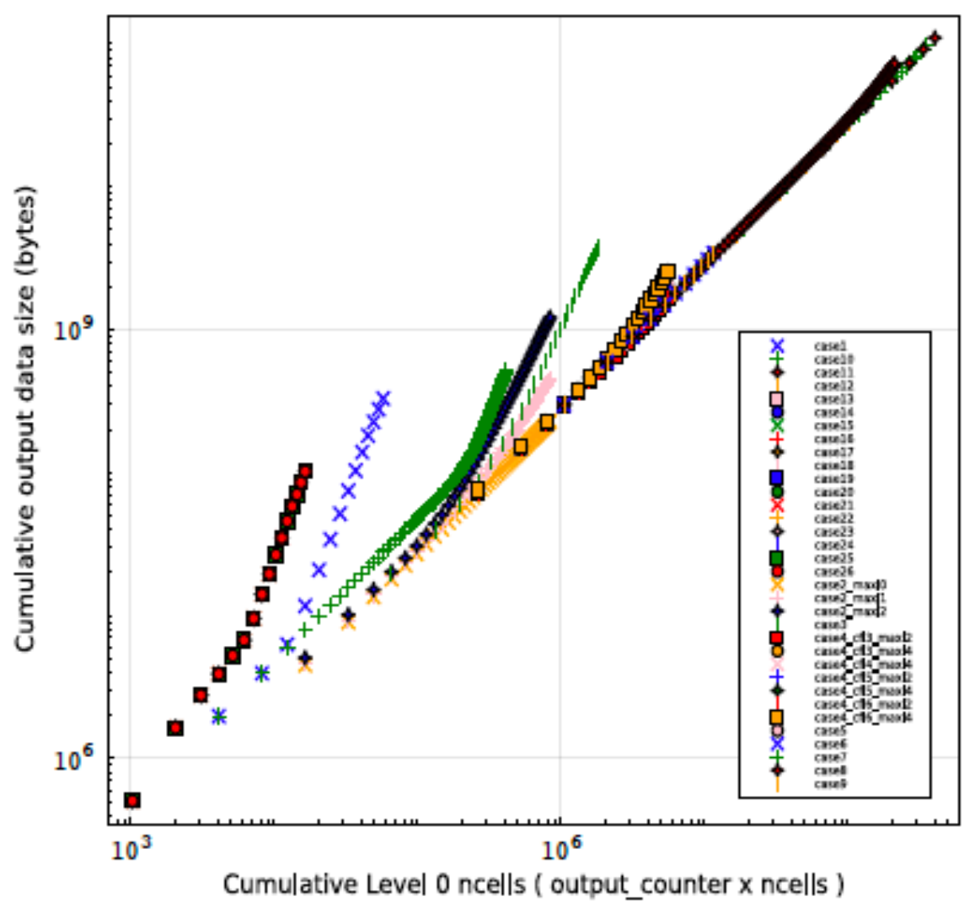


MinTally: Proxy App for Reactor Physics

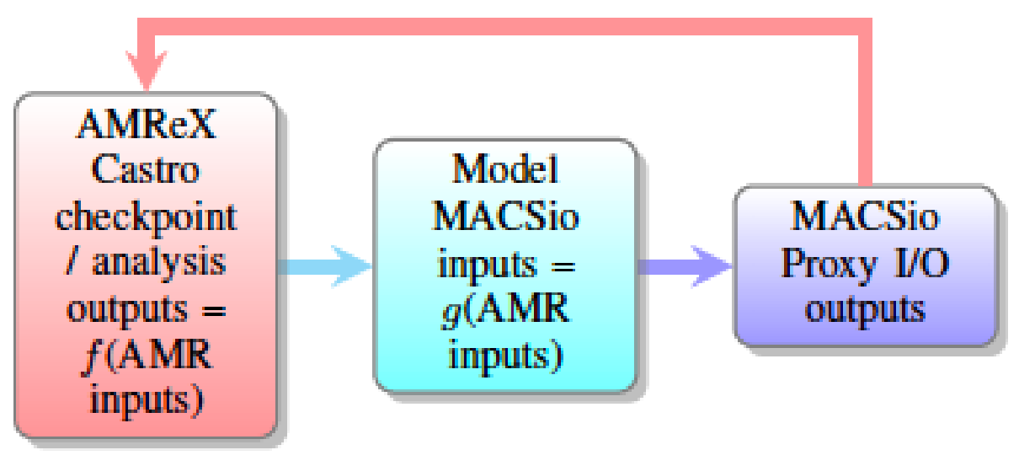
- Use reactor physics Kord Smith Challenge as exemplar use case for large tally data set. Explore use of NVRAM on node.
- Develop and verify greatly simplified proxy app suitable for testing on large HPC systems.
- MinTally abstracts key performance-critical components of full tally system while greatly simplifying use and deployment.
- Results show a superlinear increase in execution time with tally size for the full application, and a qualitatively very similar behavior for the proxy application for most tally sizes. This is true when both increasing tally size using additional geometric regions (upper two curves) or nuclide inventory (lower two curves).

Assessing AMReX I/O Behavior

Goal: provide understanding to AMReX Castro [1,2] I/O workloads via modeling using the MACSio [3] proxy app



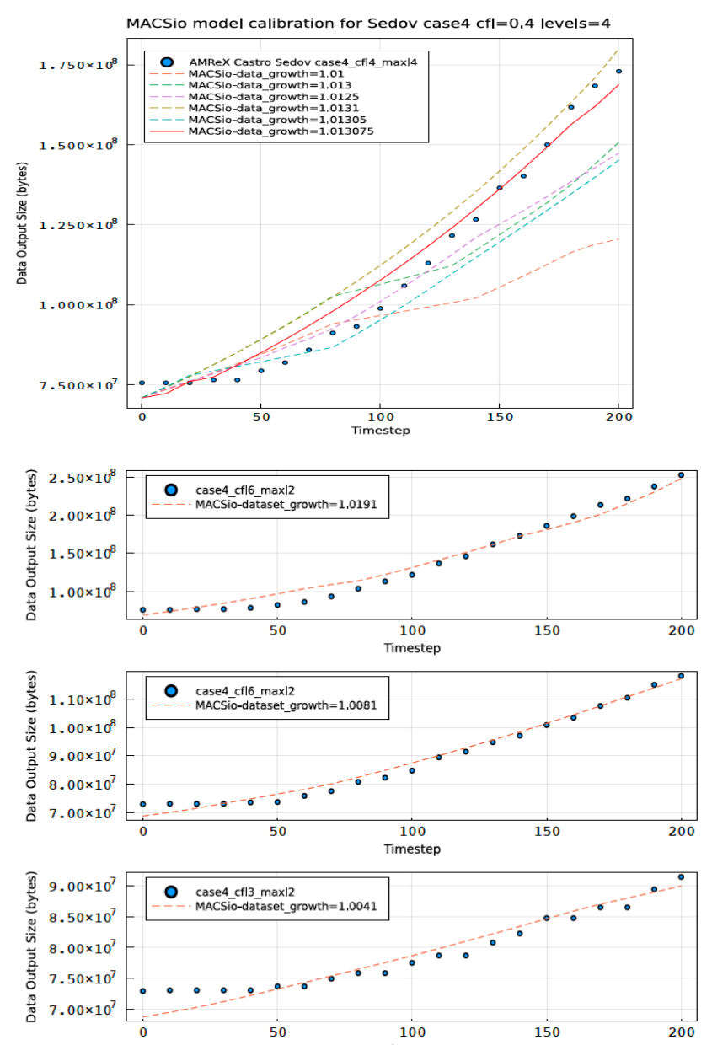
AMReX Castro non-linear Outputs


AMReX Castro checkpoint / analysis outputs = f(AMR inputs)
Model MACSio inputs = g(AMR inputs)
MACSio Proxy I/O outputs

Listing 1: Proxy app model formulation for mapping MACSio executable to AMReX Castro inputs on Summit.

```
jsrun -n nproc
macsio
 --interface miftmpl
 --parallel_file_mode MIF nproc
 --num_dumps amr.max_steps/amr.plot_int
 --part_size f(amr.n_cell)
 --avg_num_parts 1
 --vars_per_part 1
 --compute_time f(platform,all_inputs)
 --meta_size f(all_inputs)
 --dataset_growth f(amr.n_cell,castro.cfl,amr.maxlevel,...)
```

Model Formulation



MACSio fitting and calibration

MACSio provides <step, rank> and non-linear growth capabilities to model N-to-N data. AMReX N-to-N outputs at every <step,level,rank>. Our model is from a parameterized and calibration study identifying the problems size, number of levels, CFL condition at a "step" level from 40 Summit runs of the Sedov hydrodynamic problem. Work accepted at the iWAPT workshop, IPDPS 2022 [4].

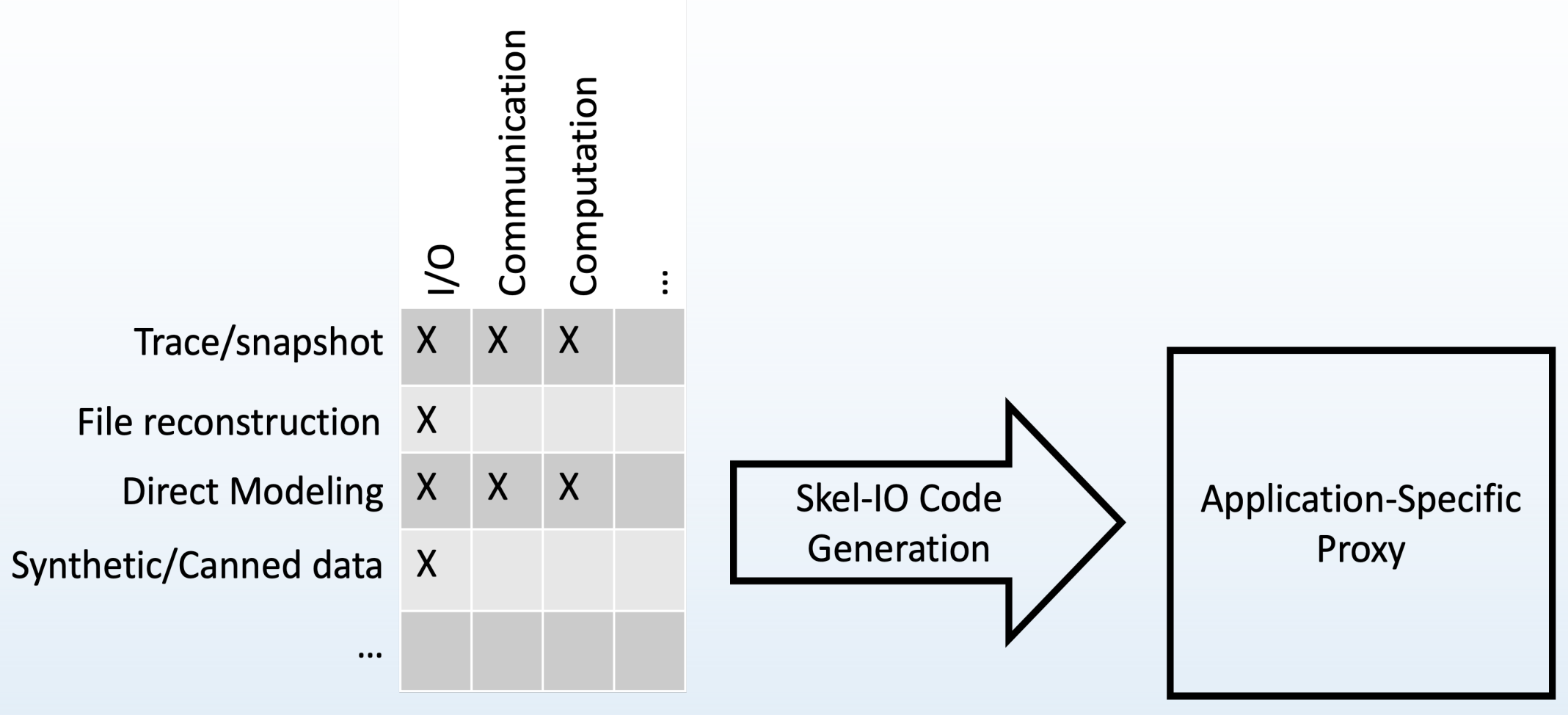
Proxy Apps in FFTX

ECP-funded FFTX is an exascale follow-on to the 1990s FFTW package for executing the Fast Fourier Transform, as well as compositions of FFTs with linear operations. FFTX optimizes algorithms over high-performance kernels for specific environments, and generates code for CPU & GPU platforms. Four proxy applications representative of HPC applications space using FFTs:

1. Hockney free-space convolution
2. WarpX Pseudo-Spectral Maxwell solver
3. NWChemEx planewave solver
4. Poisson solver

FFTX and CUDA library cuFFT on cori @ NERSC.
FFTX and HIP library rocFFT on spock and crusher @ OLCF.

Generating I/O Proxy Apps



	I/O	Communication	Computation	…
Trace/snapshot	X	X	X	
File reconstruction	X			
Direct Modeling	X	X	X	
Synthetic/Canned data	X			
…				

- Skel-IO allows application behavior to be modeled, and supports generation of benchmark codes based on those models
- Current work involves extending Skel-IO to support combining multi-faceted models into unified, application-specific proxies

1. W. Zhang, et al., "AMReX: a framework for block-structured adaptive mesh refinement," JOSS, https://doi.org/10.21105/joss.01370
2. A. S. Almgren, et al., "CASTRO: A New Compressible Astrophysical Solver. I. Hydrodynamics and Self-gravity," The Astrophysical Journal, vol. 715, pp. 1221–1238, Jun. 2010
3. M. Miller, "Design & Implementation of MACSio," LLNL Tech. Rep., 2015.
4. Godoy, DeLozier, Watson "Modeling pre-Exascale AMR Parallel I/O Workloads via Proxy Applications" iWAPT workshop, IPDPS 2022
5. Image Ref: Mathuriya, Amrita, et al. "CosmoFlow: Using deep learning to learn the universe at scale." *SC18: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2018.