# Results from the High Energy Physics Center for Computational Excellence

Charles Leggett **for HEP-CCE**

***CHEP 2023***
May 9, 2023

# What is HEP-CCE?

## Three-year (2020-2023) pilot project

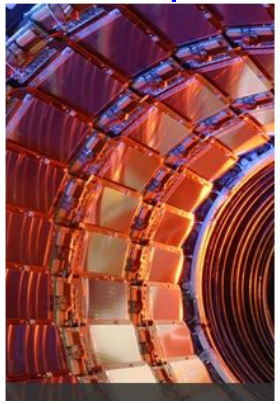
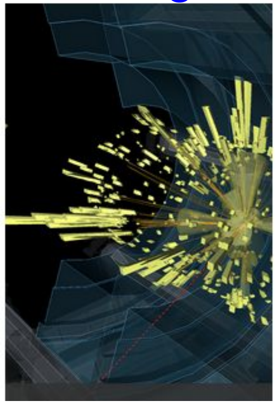
- Develop **practical** solutions to port hundreds of kernels to multiple platforms
- Collaborate with HPC & networking communities on **data-intensive** use cases

1. PPS: Portable parallelization strategies
   - exploit massive concurrency
   - portability requirements
2. IOS: HEP I/O and HPC storage issues
   - new data models (memcpy-able, SOA,...)
   - fine-grained I/O, workflow instrumentation
3. EG: Optimizing event generators
4. CW: Complex workflows on HPCs

## Open collaboration

https://indico.fnal.gov/category/1053/

https://www.anl.gov/hep-cce


Four US labs, six experiments, ~12 FTE over ~35 collaborators.
PI: Salman Habib (ANL), Co-PI: Paolo Calafiura (LBNL)


U.S. DEPARTMENT OF ENERGY Office of Science
Argonne NATIONAL LABORATORY
Brookhaven National Laboratory
Fermilab
BERKELEY LAB Bringing Science Solutions to the World

# Portable Parallelization Strategies (PPS)

| | CUDA | Kokkos | SYCL | HIP | OpenMP | alpaka | std::par |
|---|---|---|---|---|---|---|---|
| NVIDIA GPU | | | *intel/llvm compute-cpp* | *hipcc* | *nvc++ LLVM, Cray GCC, XL* | | *nvc++* |
| AMD GPU | | | *openSYCL intel/llvm* | *hipcc* | *AOMP LLVM Cray* | | |
| Intel GPU | | | *oneAPI intel/llvm* | CHIP-SPV: *early prototype* | *Intel OneAPI compiler* | *prototype* | *oneapi::dpl* |
| x86 CPU | | | *oneAPI intel/llvm computecpp* | *via HIP-CPU Runtime* | *nvc++ LLVM, CCE, GCC, XL* | | |
| FPGA | | | | *via Xilinx Runtime* | *prototype compilers (OpenArc, Intel, etc.)* | *protytype via SYCL* | |

# Portable Parallelization Strategies

Ported representative testbeds from ATLAS, CMS and DUNE to each portability layer.

| | Kokkos | SYCL | OpenMP | Alpaka | std::par |
|---|---|---|---|---|---|
| Patatrack | Done | Done* | WIP | Done* | Done compiler bugs |
| Wirecell | Done | Done | Done | no | Done |
| FastCaloSim | Done | Done | Done | Done | Done |
| P2R | done | Done | OpenACC | Done | Done |

Evaluated each porting experience according to a number of different objective and subjective metrics.

Related talks:

- p2r [Monday 11AM]
- Patatrack [Thursday 12PM]
- FastCaloSim: overview [Tuesday 3PM]
- FastCaloSim: OpenMP (poster)
- FastCaloSim: alpaka + std::par (poster)

# Portable Parallelization Strategies: Recommendations

## Software and hardware are still rapidly changing

- Lots of interactions with API developers in hackathons and to fix bugs
- Results remain preliminary

## API recommendations are very application dependent

- All perform approximately equally for simple kernels
- Complex algorithms and chained kernels bring out weaknesses of all APIs
  - interaction with external libraries adds extra complexities
  - even compilation can be an issue

## Learning curve / language complexity of APIs not all the same

- std::par → OMP → Kokkos / SYCL → alpaka → OMP
- subjective and dependent on code complexity and previous experience

## Porting from Serial CPU code → GPU concepts is the biggest hurdle

- starting with optimized code is extra challenging

## Very hard to extrapolate to next five years or beyond

- Vendors are pushing in different directions (but towards standards)
- Increasing proximity of CPU / GPU / memory will have significant impact

# IOS

# Input/Output and Storage: Activities

## Measuring performance of ROOT I/O in HEP workflows on HPC systems

- Darshan a scalable HPC I/O characterization tool has been enhanced (including fork safety) and used to monitor HEP production workflows.

## Investigate HDF5 as intermediate event storage for HPC processing

- Relying on ROOT to serialize complex Event Data Model used in Simulation/Reconstruction workflows
- Implementing Collective Writing to avoid potential merge step
- Mimicking framework for understanding scalability and performance of HEP output methods
    - Experiment agnostic tool allows scaling I/O beyond what is currently accessible by production and has uncovered/fixed bottlenecks in ROOT and frameworks.

## HPC friendly Data Model

- Together with PPS team started investigating efforts to make data model more suitable for offloading to accelerator and storage on HPC.

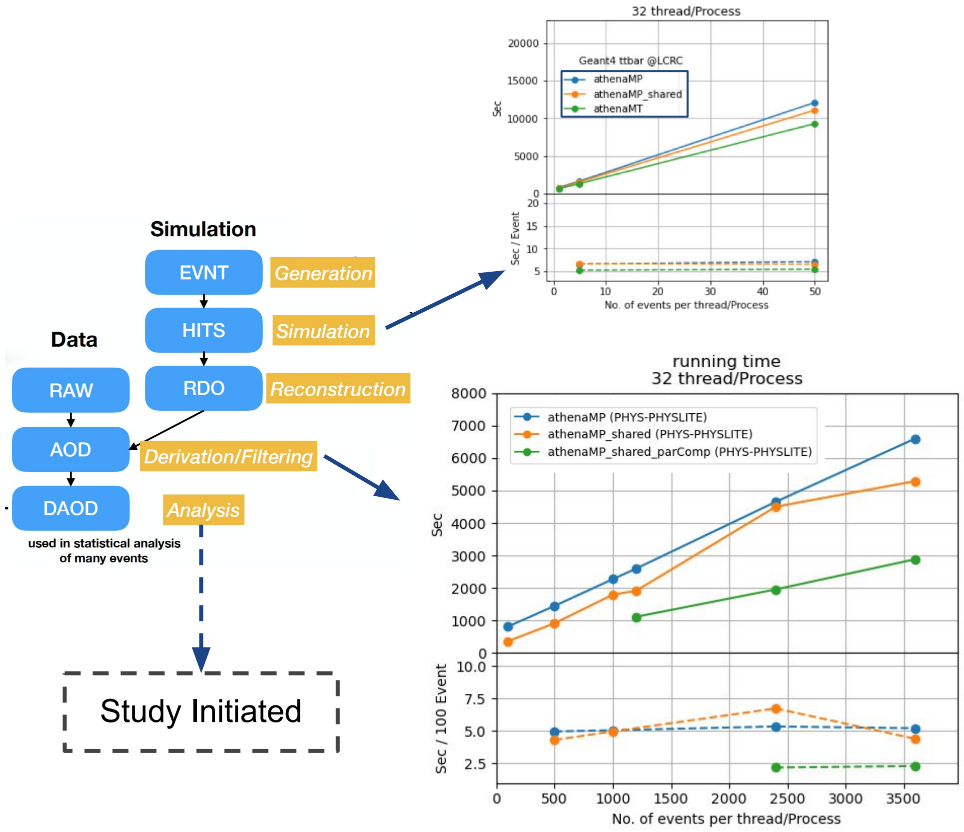
Darshan for HEP application
Thursday 12:30

Using Parallel I/O Libraries for Managing HEP Experimental Data
Tuesday 11:45

# I/O and Storage: Recommendations

## Work of the HEP-CCE/IOS team has resulted in

- Worthwhile insight to I/O behavior of HEP workflows
  - Including on HPC and for scales beyond current production.
- Fixes/enhancements to common software and experiments frameworks
  - Darshan included fork-safety and better filtering for I/O.
  - ROOT serialization bottleneck was fixed.
- Prototype development of new functionality in collaboration with experiments:
  - ATLAS developed functionality to store their production data in HDF5




Darshan Monitoring of different ATLAS workflow steps

# PPS and IOS: Next Steps



Finalize and publish results and recommendations

Meet with stakeholder experiments to present conclusions

- General meetings, seminars and focussed workshops

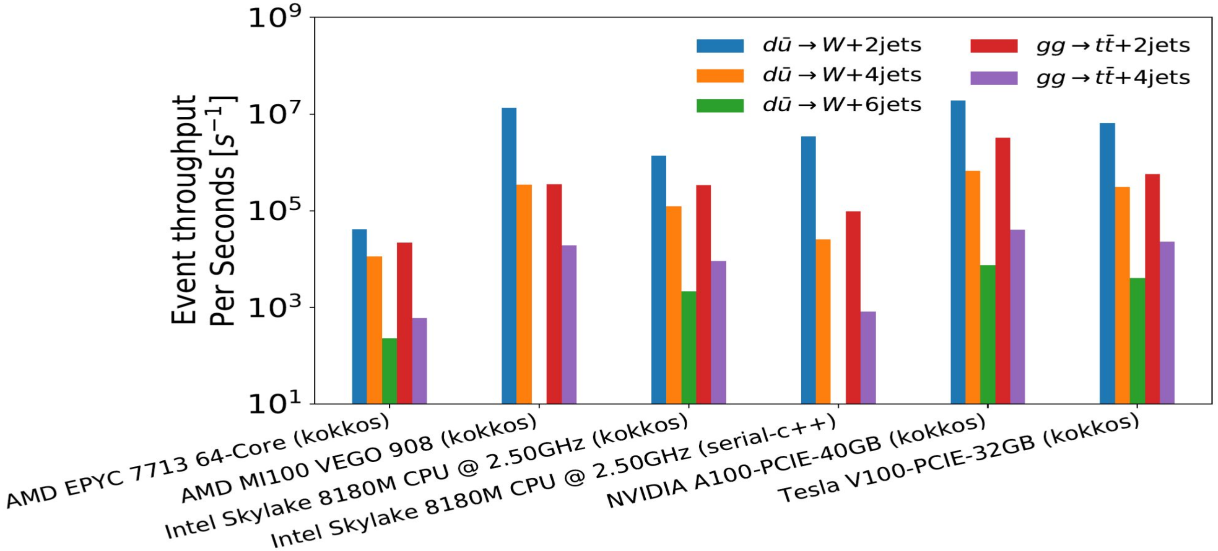
Present to HEP community via larger forums and external partners

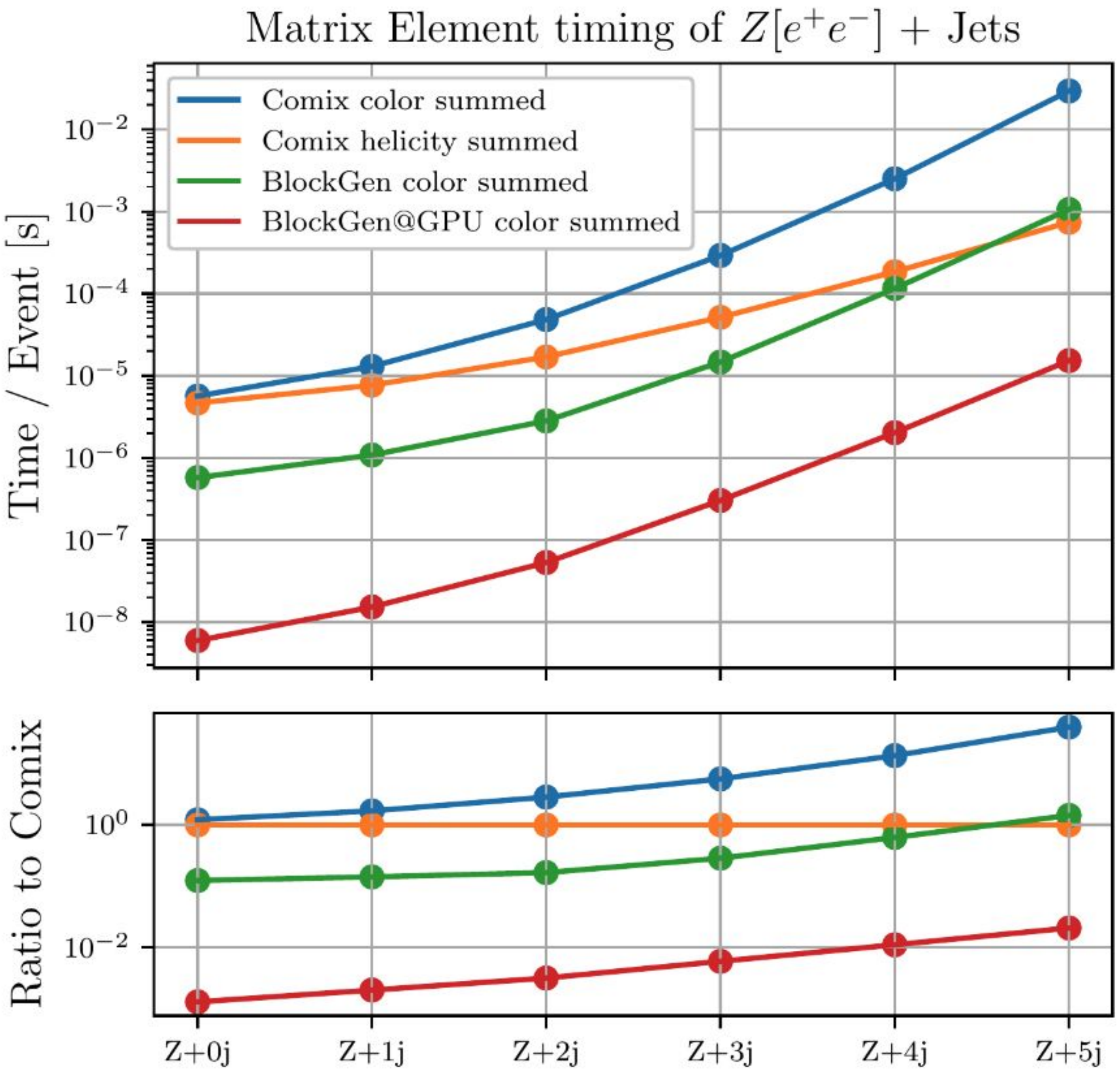
- HSF
- WLCG
- IRIS-HEP
- OpenLab

Outreach to other experiments to transfer knowledge and experiences

# Event Generators

# Pepper: A New Leading-Order Matrix Element Integrator



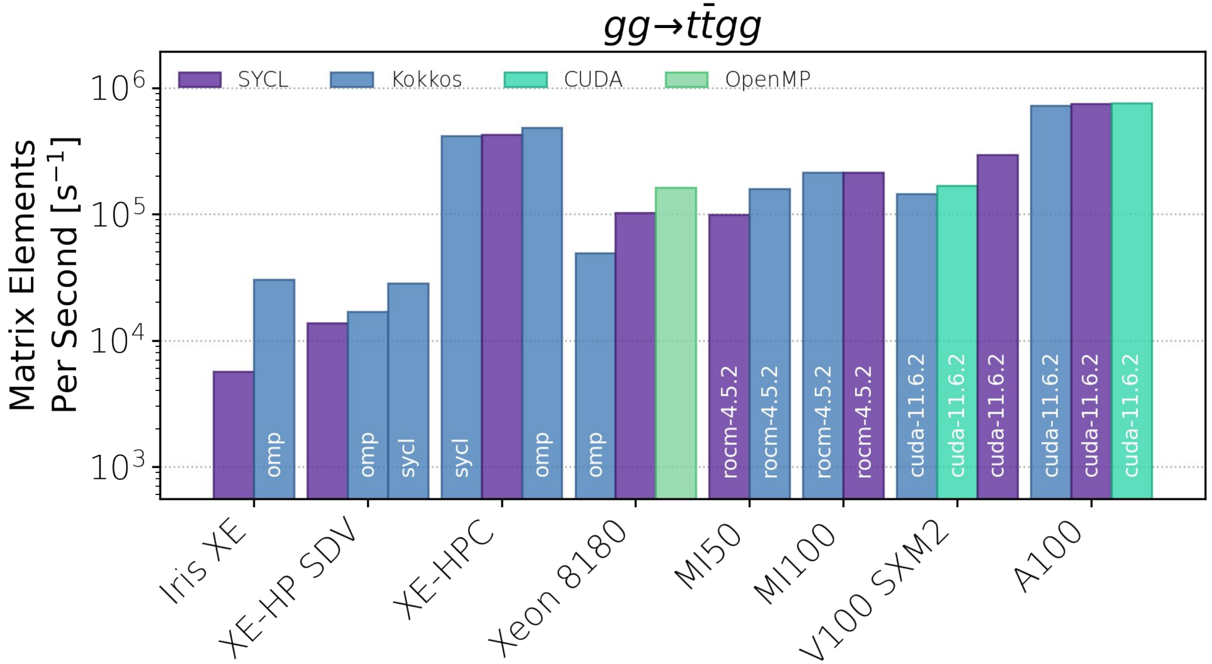

New LO generator developed from the ground up for parallel architectures [alg-paper]

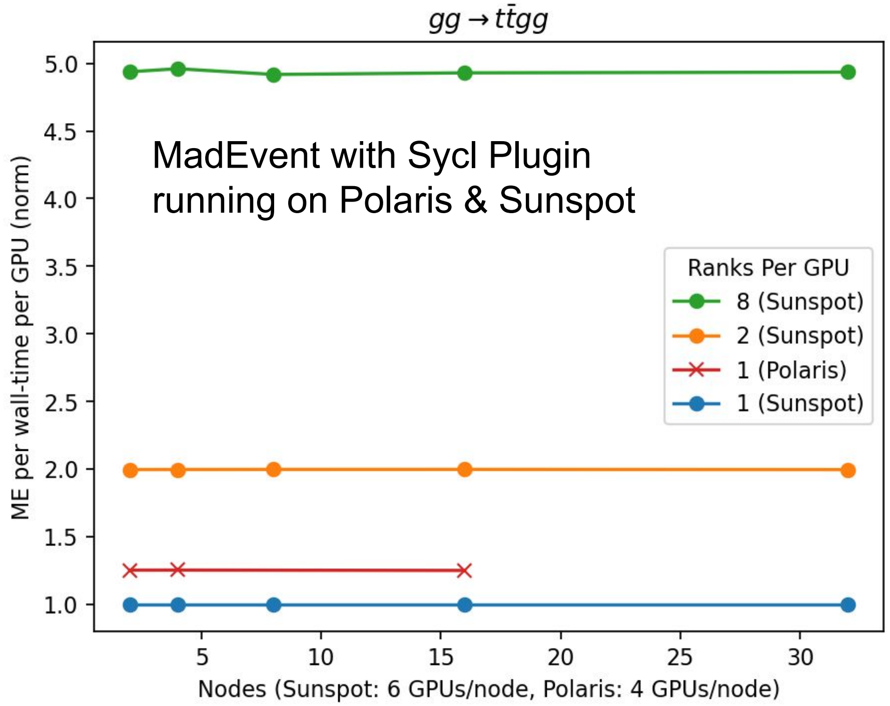
- Strong performance for CUDA over a wide range of processes

HEP-CCE facilitating portability via Kokkos port of the algorithm with excellent results on multiple modern hardware

HEP-CCE

# MadGraph: Portable MadEvent




See Parallel Talk [Monday 2:30PM]

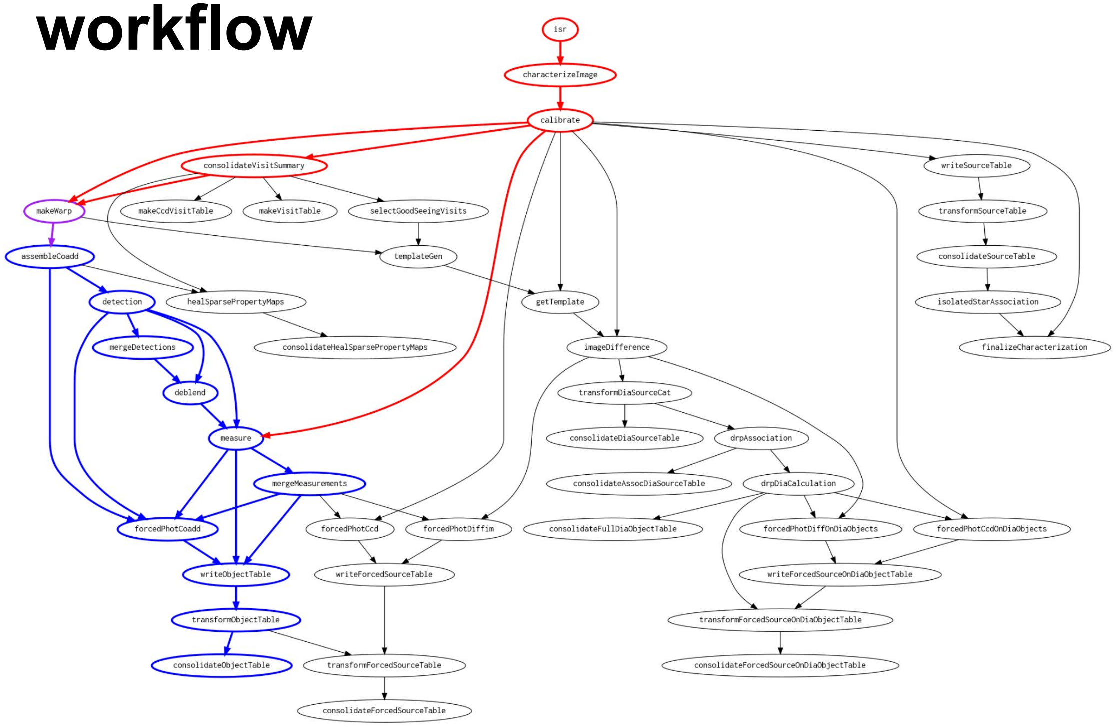
- Working with a team of developers from CERN and UCLouvain on a GPU version of the LO MadEvent generator with Matrix Element (ME) calculations ported to Kokkos/SYCL
- HEP-CCE developed Kokkos version and took over the Sycl version
- Kokkos and SYCL have enabled running on a broad variety of architectures including multicore CPUs, AMD and NVIDIA GPUs, and new Intel GPUs on Sunspot (Aurora) including oversubscription models, with good performance characteristics (now)

# Complex Workflows

# HEP Experiments require HPC workflows

- HEP has many workflow technologies, serving specific use-cases, that are monolithic, and difficult to extend
- Performance on leadership platforms is complex even for simple workflows; HPC workflows will become more important, but increasingly harder
- Opportunity to harmonize use of experiment-agnostic components, integrable solutions for extensibility and modularity on leadership platforms.
- HEP-CCE workflows group (including workflows and experiments) identified common challenges and prototyped cross-workflow approaches
    - Interoperability between workflow systems
    - Common task graph representation
    - Streamlined remote execution
    - Diverse monitoring information

## Rubin LSST image processing workflow




Graph showing dependencies between task types for Rubin image processing. red operate on CCD-visits, blue tasks on patches, and purple on both.

# Summary

Landscape of hardware and software continues to evolve rapidly

- HPC systems moving towards more specialized nodal architectures, with few exceptions
- Challenging for experiments to capitalize on emerging systems
- Matching HEP workloads to different architectures and managing multiple timelines is difficult

HEP experiments are facing significant obstacles in data management and compute tasks

- HPC systems offer enormous possibilities in computational power and size, but come with their own challenges in portability, IO, and workflows, including integration within the wider workload management infrastructure

HEP software needs to be adapted to utilize the diverse architectures present in the current and upcoming HPC computers.

- Portability is a major consideration for such software adaptation.
- Overhead in implementing portability layers may be worth the effort
- Different solutions have own pros and cons. Best is use case dependent


U.S. DEPARTMENT OF ENERGY | Office of Science
Argonne NATIONAL LABORATORY
Brookhaven National Laboratory
Fermilab
BERKELEY LAB Bringing Science Solutions to the World

# Future Directions



## Exploiting power of HPCs may help address storage shortage

- Domain-aware compression, processing on demand

## Increase the efficiency and usability of HPCs for HEP workflows

- Software deployment, edge services for data and workflow management, resilience

## Advocate unique HEP computing needs

- Package representative “mini-apps” and datasets as benchmarks for future HPC architectures.
- The 5As: Allocation, availability, authentication, authorization, accounting
- Developing working relationship with HPC facilities for improved and new edge services and access mechanisms

# fin

This work was supported by the U.S. Department of Energy, Office of Science, Office of High Energy Physics, High Energy Physics Center for Computational Excellence (HEP-CCE). This research used resources at Argonne Leadership Computing Facility, NERSC and BNL Scientific Data and Computing Center.