HEP-CCE

HEP Center for Computational Excellence

Meifeng Lin
Brookhaven National Laboratory

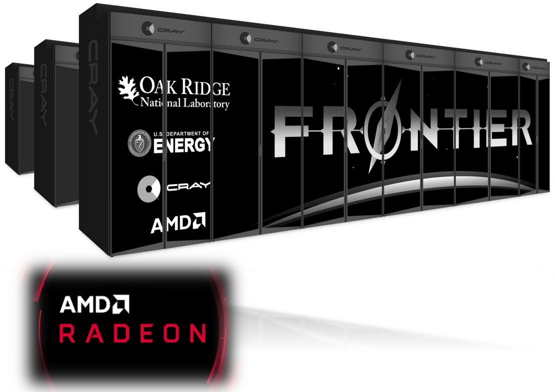
Snowmass Computational Frontier Workshop, August 10-11, 2020

The HEP-CCE Initiative

- The landscape of computing architectures is evolving rapidly.
- Traditional multi-/many-core CPU architecture is taking a back seat in HPC.
- Computational speed is increasingly driven by compute accelerators (NVIDIA, AMD, Intel GPUs).
- Support for dedicated HTC resources is shrinking.
- HEP experiments, simulations and data analysis pipelines that rely on traditional CPUs and HTC need to adapt.

- The **HEP-CCE initiative** aims to promote excellence in using HPC for data-intensive applications.
- Enhances connections between HEP and ASCR projects and researchers.

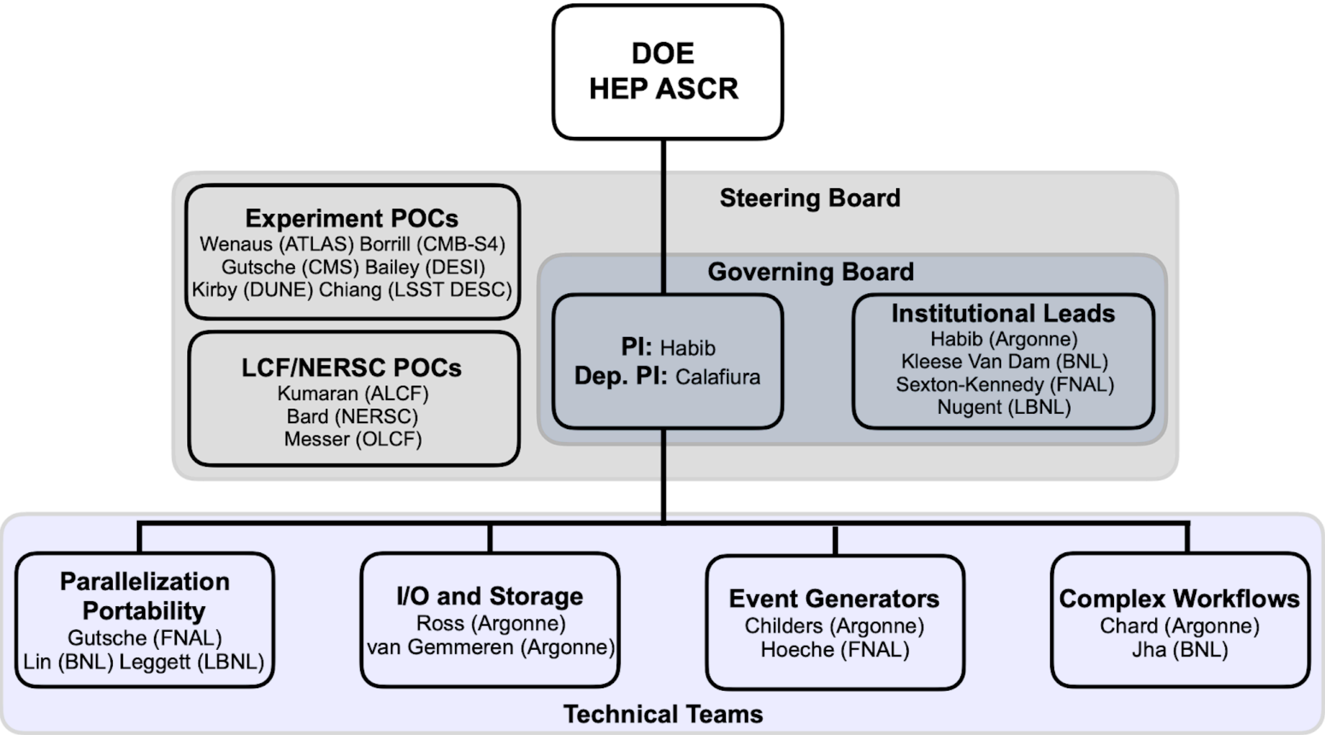
AMD CPU+NVIDIA GPU



The HEP-CCE Project

- The new **HEP-CCE project** is funded by **DOE HEP** to investigate the challenges and possibilities of using next-generation computing architectures, especially compute accelerators, for HEP data-intensive applications.
- With the goal to improve the utilization of large-scale HPC systems such as those in the DOE LCFs.
- Its scope involves Intensity Frontier, Cosmic Frontier and Energy Frontier.
- Four participating DOE labs: ANL, BNL, FNAL and LBNL.
  - Lead PI: Salman Habib (ANL)
  - Co-PI: Paolo Calafiura (LBNL)
- 3-year project, officially started in January 2020.
- ~10 FTEs, mix of physicists, computer scientists, software engineers and postdocs.
- Four thrusts
  - **Portable Parallelization Strategies (PPS)** - *Leads: Gutsche (FNAL), Leggett (LBNL), Lin (BNL)*
  - **I/O and Storage (IOS)** - *Leads: Ross (ANL), van Gemmeren (ANL)*
  - **Event Generators (EG)** - *Leads: Childers (ANL), Hoeche (FNAL)*
  - **Complex Workflows (CW)** - *Leads: Chard (ANL), Jha (BNL)*

HEP-CCE Management Structure



DOE
HEP ASCR

Steering Board

Experiment POCs
Wenaus (ATLAS) Borrill (CMB-S4)
Gutsche (CMS) Bailey (DESI)
Kirby (DUNE) Chiang (LSST DESC)

LCF/NERSC POCs
Kumaran (ALCF)
Bard (NERSC)
Messer (OLCF)

Governing Board

PI: Habib
Dep. PI: Calafiura

Institutional Leads
Habib (Argonne)
Kleese Van Dam (BNL)
Sexton-Kennedy (FNAL)
Nugent (LBNL)

Parallelization Portability
Gutsche (FNAL)
Lin (BNL) Leggett (LBNL)

I/O and Storage
Ross (Argonne)
van Gemmeren (Argonne)

Event Generators
Childers (Argonne)
Hoeche (FNAL)

Complex Workflows
Chard (Argonne)
Jha (BNL)

Technical Teams

- The experiment POCs help steer the directions of the project to meet the needs and priorities of the experiments.
- The LCF POCs help ensure our software and tools can run efficiently on current and future HPC systems.
- Institutional leads, experiment POCs and LCF POCs communicate to make sure the project stays on the right track.
- Technical leads drive the technical progress.

Portable Parallelization Strategies (PPS)

Technical Leads:

- *Oliver Gutsche (FNAL)*
- *Charles Leggett (LBNL)*
- *ML*

- HEP experiment software has long development cycles with stringent V&V requirements.
- Impractical to rewrite software for every new architecture.
- Best to have a portable implementation that runs *relatively* well on many architectures.
- Investigate a range of portable programming models on a few use cases
- Define a set of **metrics** to evaluate suitability of each programming model
  - **Productivity**: ease of porting/learning, code impact/change, build time, debugging, etc.
  - **Portability**: architectural portability, temporal portability (how sustainable the model is…),
  - **Performance**: absolute performance (sustained/peak) may not be highest priority
- Goal is to make recommendations to the broad HEP community through reports, publications and community outreach.

PPS Progress to Date

- Started with three self-contained use cases:
  - FastCaloSim (ATLAS) – fast calorimeter MC simulations
  - Wire-Cell Toolkit (DUNE) – detector response simulations
  - Patatrack (CMS) – track reconstruction
- All three have preliminary **CUDA implementations** to test the feasibility of GPU acceleration.
- Represent different algorithms and code complexities in the HEP software ecosystem.
- Plan is to investigate different portable programming models for each use case
  - **Kokkos/RAJA**: C++ abstraction layer built on top of various backends (CUDA, OpenMP, HIP, …);
  - **SyCL/DPC++**: C++ abstraction layer on top of the OpenCL backend;
  - **OpenMP/OpenACC:** compiler directives that can be added to the source code;
- *Challenges:* different programming models reach different levels of maturity. Still evolving.
- **Perhaps also an opportunity to influence/request feature development.**
- Currently working on Kokkos implementations of the use cases:
  - Working prototype of FCS on both NVIDIA and AMD GPUs.

Fine-Grained I/O and Storage (IOS)

- In the HPC environment, data-intensive applications have large I/O overhead compared to the cost of computation and communication.
- **Goals of IOS:**
  - Explore options for improving I/O performance by exploring parallelism during workflows.
    - Alternatives to use of files
    - More PPS-friendly data organization
  - Understand implications of different event data models and representations
    - Allow for events to be segmented into smaller regions to speed up processing
    - Mapping back to traditional file-based representations at end of workflow
  - Demonstrate promising options in real-world HEP workflows
- **Current Activities:**
  - Investigate performance issues of ROOT I/O in HEP workflows on HPC
  - Investigate HDF5 as intermediate event storage for HPC processing

Technical Leads:
- *Peter van Gemmeren (ANL)*
- *Rob Ross (ANL)*

Event Generators (EG)

Technical Leads:
- *Taylor Childers (ANL)*
- *Stefan Hoche (FNAL)*

- Event generation might take up more computing resources at HL-LHC than currently extrapolated.
- Need efficient EG code that scales well on HPC
- Previous Improvements:
  - Parallelized Pythia particle-level event generation using DIY
  - Improved performance of Sherpa, particularly I/O with HDF5
  - Novel integrators using Neural Networks and Normalizing Flows
- **Goals of EG:**
  - Improve performance of EG for both CPUs and GPUs
  - Exploration of different frameworks (low-level programming models vs high-level such as Kokkos)

Complex Workflows (CW)

Technical Leads:
- *Kyle Chard (ANL)*
- *Shantenu Jha (BNL)*

- Workflows enable representation and execution of analyses composed of heterogeneous components:
  - Single node codes, multi-node MPI applications, analysis scripts, binaries, etc.
- Heterogeneity of the HPC systems make workflow execution even more challenging.
  - How to efficiently map millions of heterogeneous tasks (different requirements for cores/duration/CPU or GPU) to the available resources?
- **Goals of CW:**
  - Support development and use of workflows within target domains
  - Apply a modular, interoperable and inclusive approach
  - Enable use of accelerators, heterogeneous hardware
  - R&D related to portability, scalability, performance and reproducibility

Summary

- HEP-CCE project is well underway. Focus is on portable parallelization strategies (PPS), complemented by I/O and workflow optimizations.
- PPS is challenging due to large phase space to explore and HEP code complexities.
- Engagement with experiments: regular status reports at collaboration meetings, S&C weeks, etc.
- Close collaboration with ASCR researchers to employ state-of-the-art tools and best practices.
- **3-year project will only produce prototypes. Production-level implementations will require a lot more resources and time.**