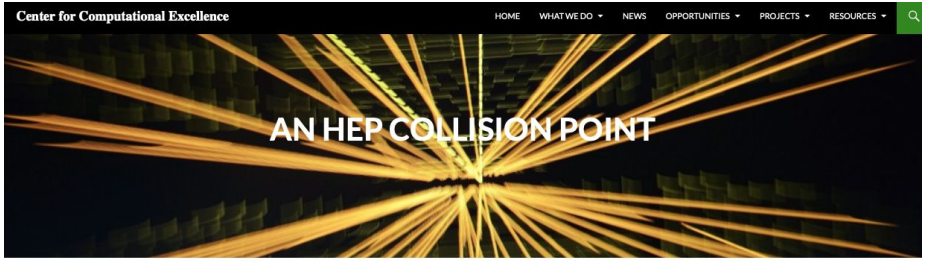


HEP Center for Computational Excellence (HEP-CCE) - Current Projects

CMS S&C week March 2020
Oliver Gutsche

HEP Center for Computational Excellence (HEP-CCE)

- Cross-cutting DOE initiative
  - To enable utilization of DOE's high performance computing (HPC) facilities
  - Through R&D activities with data-intensive applications, scientific simulations, and data movement and storage.
- CCE aims to develop common R&D projects
  - At the DOE laboratories, more broadly then just LHC:
    - U.S. ATLAS, U.S. CMS, DUNE, LSST and others.
  - CCE will also enhance connections with DOE's Advanced Scientific Computing Research (ASCR).



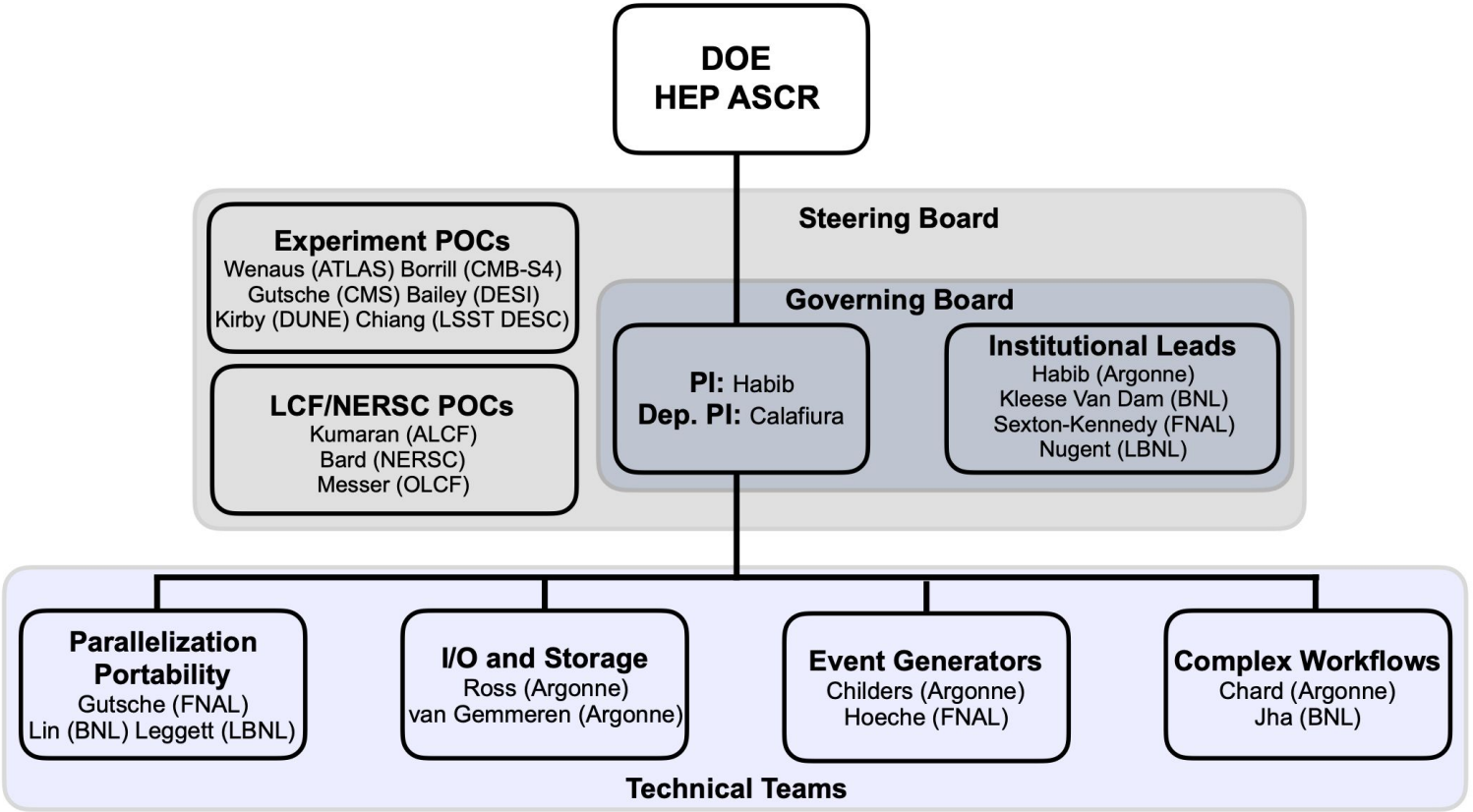
HEP-CCE: Promoting Computational Excellence

HEP-CCE Coordinators: Salman Habib (Argonne), Kerstin Kleese Van Dam (Brookhaven), Liz Sexton-Kennedy (Fermilab), Peter Nugent (Lawrence Berkeley), Richard Dubois (SLAC)

The HEP-CCE is a cross-cutting initiative to promote excellence in high performance computing (HPC) including data-intensive applications, scientific simulations, and data movement and storage. Enhancing connections with DOE's Advanced Scientific Computing Research (ASCR) program is an important part of the Center's activities. This includes promoting future-looking R&D initiatives in exascale architectures and systems, intelligent networking, and new data management and data analysis tools. Although the HEP-CCE is not a service-oriented entity, limited resources are available to support collaborative computing efforts for the HEP community, including a common GitHub repository for open source codes, a website for aggregating useful information, and expertise within and without HEP for solving computational problems via the Expert Forum. The HEP-CCE also sponsors topical workshops and student training programs.

- Accelerated architectures are rapidly changing and new HPC systems are commissioned approx. every 5 years
  - Investigate solutions and perform R&D to enable effortless migration of experiment's workflows to new and newer HPC platforms

CCE management structure



HEP-CCE: current projects

- 3-year funding period
  - Started FY20 (October 2019) - Ends FY22 (October 2022)
  - Approx. 10 FTE
  - Will recruit from national laboratories staff (both science and computing) and will try to hire postdocs (shared with science research, example: half of the postdoc's time on CCE project, half on CMS research)
- Projects:
  - Portable Parallelization Strategies (PPS)
  - Fine-Grained I/O and Storage (IOS)
  - Event Generators (EG)
  - Complex Workflows (CW)

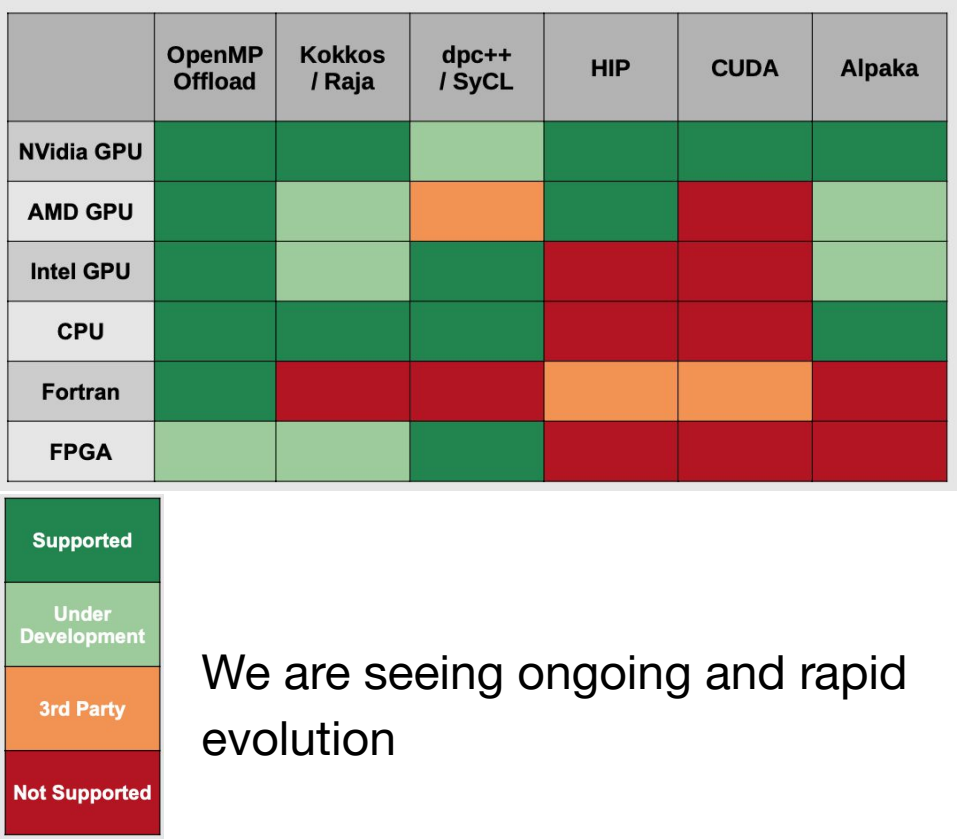
PIs:
Salman Habib (ANL)
Paolo Calafiura (LBNL)

Portable Parallelization Strategies (PPS)

- Investigate a range of portability solutions such as
  - Kokkos / Raja
  - SyCL
  - Alpaka
  - OpenMP / OpenACC
- Port a small number of HEP testbeds to each language
  - Patatrack pixel tracking (CMS)
  - WireCell Toolkit (DUNE)
  - FastCaloSim (ATLAS)
- Define a set of metrics to evaluate the ports, and apply them
  - ease of porting, performance, code impact, relevance, etc
  - Google doc
- Make recommendations to the experiments
  - must address needs of both LHC style workflows with many modules and many developers, and smaller/simpler workflows

Technical leads:
Oliver Gutsche (FNAL)
Meifeng Lin (BNL)
Charles Leggett (LBNL)

Portable Parallelization Strategies (PPS)



We are seeing ongoing and rapid evolution

Timeline

- Phase 1: Preparation
  - Deliverable: matrix of benchmarks of the unaltered use cases
- Phase 2: First Implementation
  - Deliverable: choose one technology, implement all use cases and benchmark them ➜ then move on to the next technology
- Phase 3: Consolidate benchmarking results
  - Deliverable: write-up summarizing the benchmarking results of the use cases and recommending parallelization strategies for the HEP community.
- Phase 4: Fully-functional prototypes
  - Provide recommendations on portability strategies to experiments
  - Deliverable: fully-functional prototypes made available to the experiments.

Fine-Grained I/O and Storage (IOS)

- In HPC:
  - File access overheads are high(er) relative to cost of computation and communication.
  - Larger files tend to help amortize costs, but force reorganization of data
  - Lots of ongoing work in alternatives to files for passing data within workflows
- What are we trying to accomplish?
  - Explore options for improving absolute performance and parallelism of I/O during workflows
    - Alternatives to use of files
    - Connection with parallelization strategies work – data organization should have good “impedance match” with what is needed for computation
  - Understand implications of options for event data models and representations
    - Allow for events to be segmented into smaller regions to speed up processing that can scale poorly with high multiplicity
    - Mapping back to traditional file-based representations at end of workflow
  - Demonstrate promising options in real-world HEP workflows

Technical leads:
Peter van Gemmeren (ANL)
Rob Ross (ANL)

Fine-Grained I/O and Storage (IOS) - Plan of Work

- Phase I: Preparation
  - Document existing implementations for participating experiments
  - Define a set of real use cases
  - Discuss viability of alternatives for HPC workflows
- Phase II: Prototyping
  - Develop proof-of-concept prototype(s)
  - Work with PPS team to ensure efficient mapping to memory constructs
- Phase III: Benchmarking and reporting
  - Run experiments using synthetic benchmarks on relevant platforms, refine prototypes
  - Develop recommendations for experiments and engage in dialog on outcomes

Event Generators (EG)

- Why improve event generators?
  - Event generation might consume more resources at HL-LHC that we currently extrapolate
  - Generators code generally not well written ➜ scaling issues even on HTC resources, not to even speak of HPC
  - Coordinated with dedicated effort in HSF
- Building on previous improvements
  - Parallelized Pythia particle-level event generation using DIY
  - Improved performance of Sherpa, particularly I/O (HDF5)
  - Novel integrator using Neural Networks and Normalizing Flows
- CCE plans
  - Complete rewrite of matrix-element generator for CPUs & GPUs
  - Paradigm shift from “best of scaling” to “best for computation”
  - Exploration of different frameworks (low-level, Kokkos, …)
  - Alternatively try to revive existing implementations (HELAS/MG)
  - Coordinate with HSF and worldwide efforts

Technical leads:
Taylor Childers (ANL)
Stefan Höche (FNAL)

Complex Workflows

- Why complex workflows?
  - Workflows enable representation and execution of analyses composed of heterogeneous components
    - Single node codes, multi-node MPI applications, scripts, binaries, glue code, etc.
  - HPC systems and environments are not designed for such workloads
    - Millions of tasks, diverse and varying requirements (cores, duration, CPU/GPU), co-scheduling and dynamic sizing
- CCE plans
  - Support development and use of workflows within target domains
  - Apply a modular, interoperable, and inclusive approach
    - Leverage/combine components from existing workflow systems
  - Democratize access to advanced workflow capabilities and reduce overheads on small collaborations
    - Enable use of accelerators, heterogeneous hardware, without knowledge of low-level programming
  - R&D related to portability, scalability, performance, and reproducibility

Technical leads:
Kyle Chard (ANL)
Shantenu Jha (BNL)