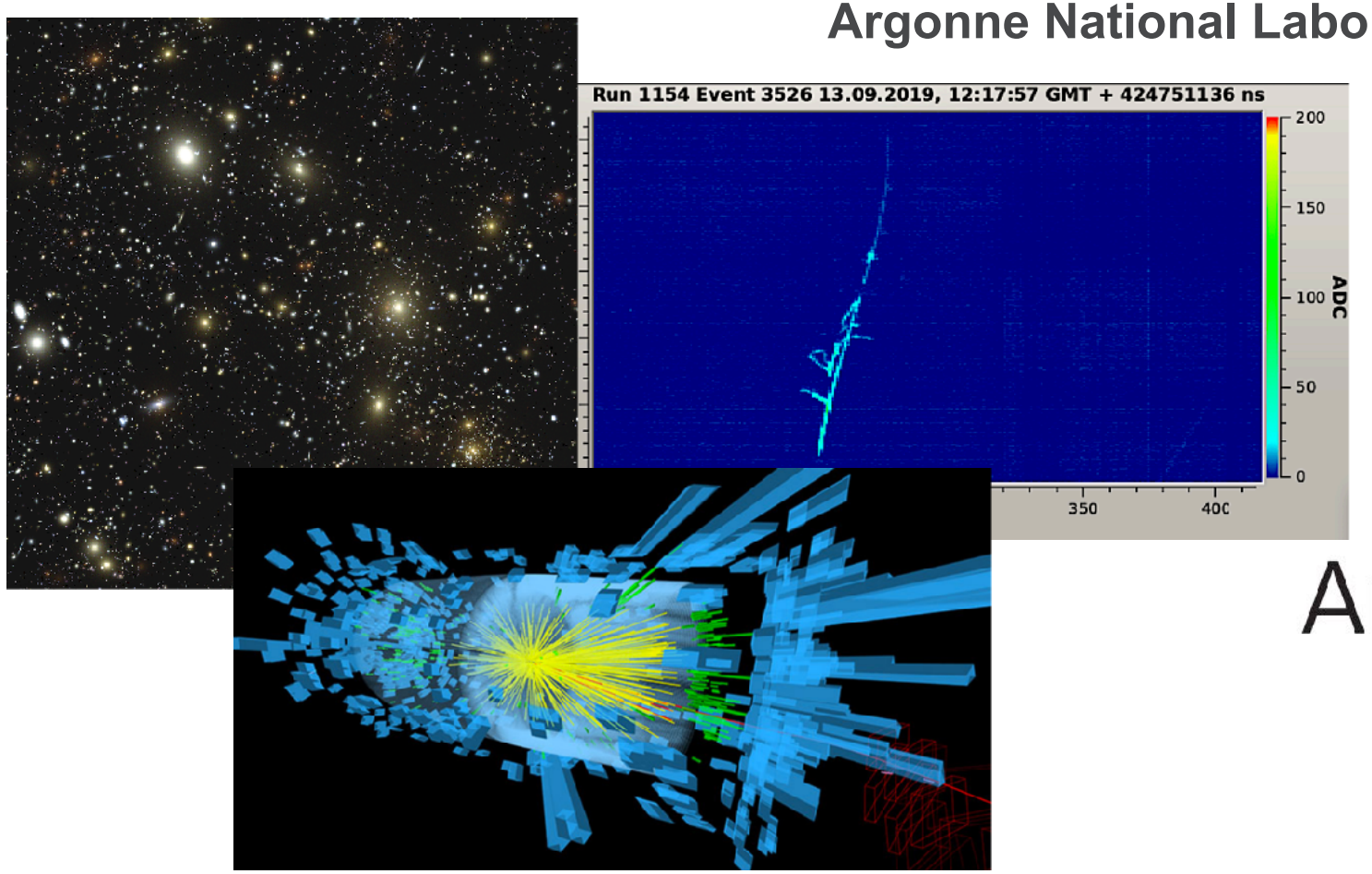
HEP-CCE (HEP Center for Computational Excellence)

Salman Habib
CPS and HEP Divisions
Argonne National Laboratory



BROOKHAVEN NATIONAL LABORATORY

BERKELEY LAB

Argonne NATIONAL LABORATORY

Fermilab

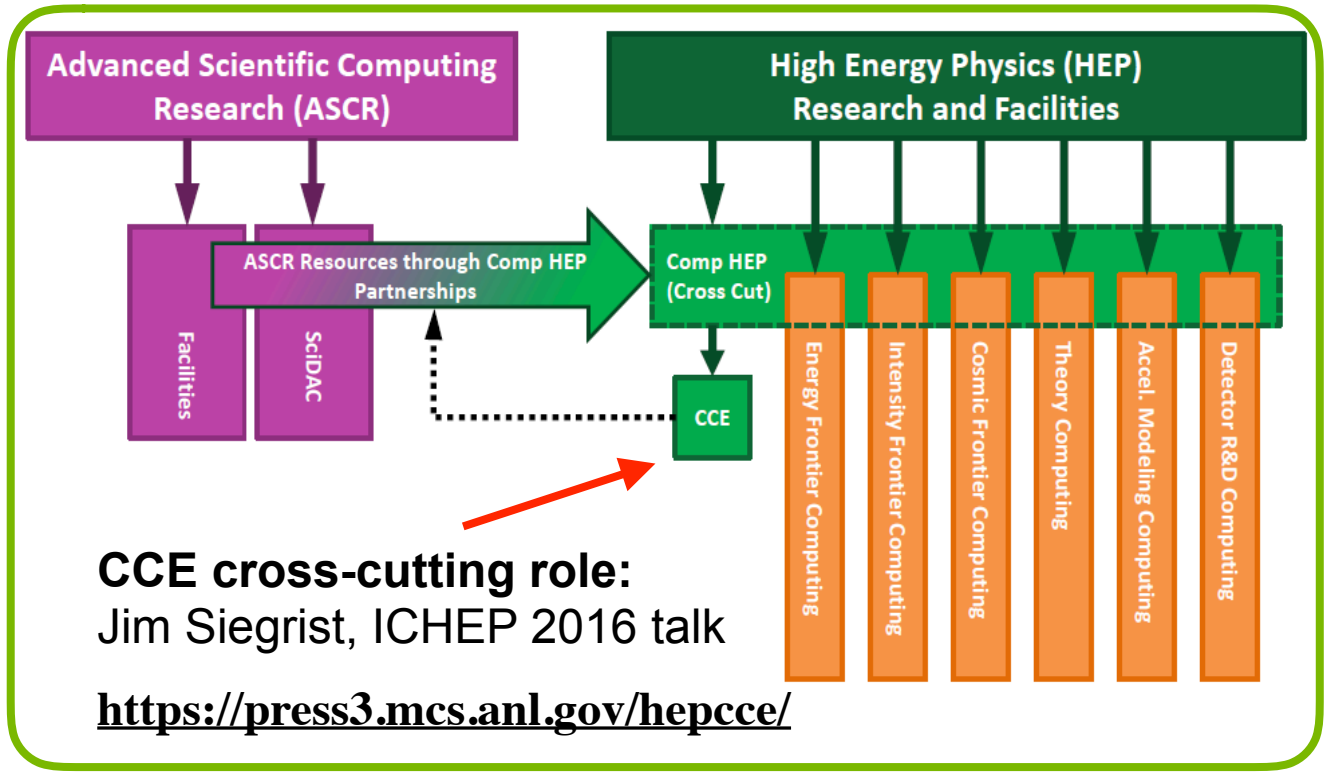
October 23, 2019
IRIS-HEP Meeting

U.S. DEPARTMENT OF ENERGY | Office of Science

HEP-CCE: A Mediated Link to ASCR Facilities/R&D

HEP-CCE:

- History: HEP and ASCR researchers at HEP laboratories (following 2013 panel recommendation)
- Bridge between the broader HEP community, HPC-oriented HEP research teams, and ASCR expertise and resources
- Develop and make available selected HPC tools/capabilities to aid HEP science in coordinated work with ASCR
- New “projectized” proposal aims to study use of ASCR HPC systems ***for all*** HEP frontiers, particularly experiments taking data starting from 2020 onwards (ATLAS, CMB-S4, CMS, DESI, DUNE, LSST DESC, —)



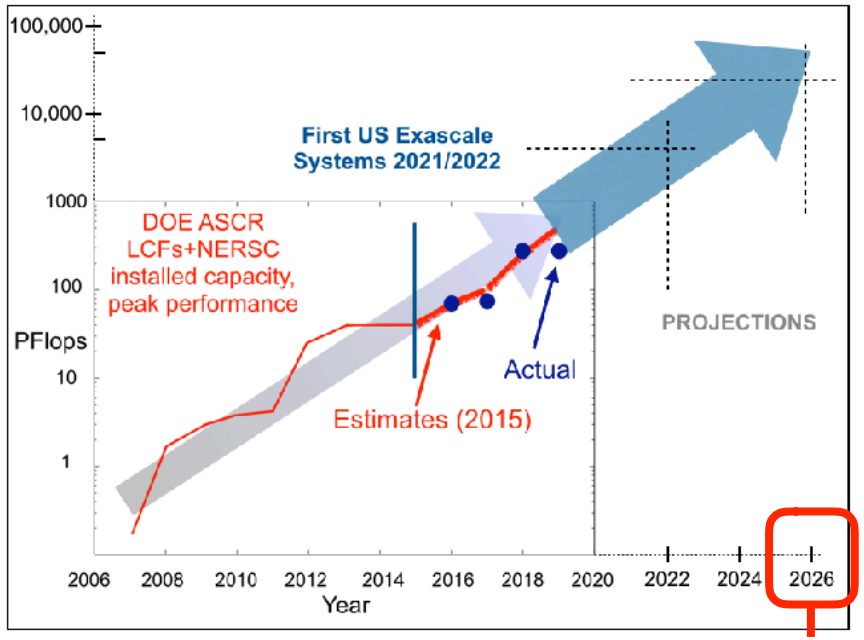
CCE cross-cutting role:
Jim Siegrist, ICHEP 2016 talk

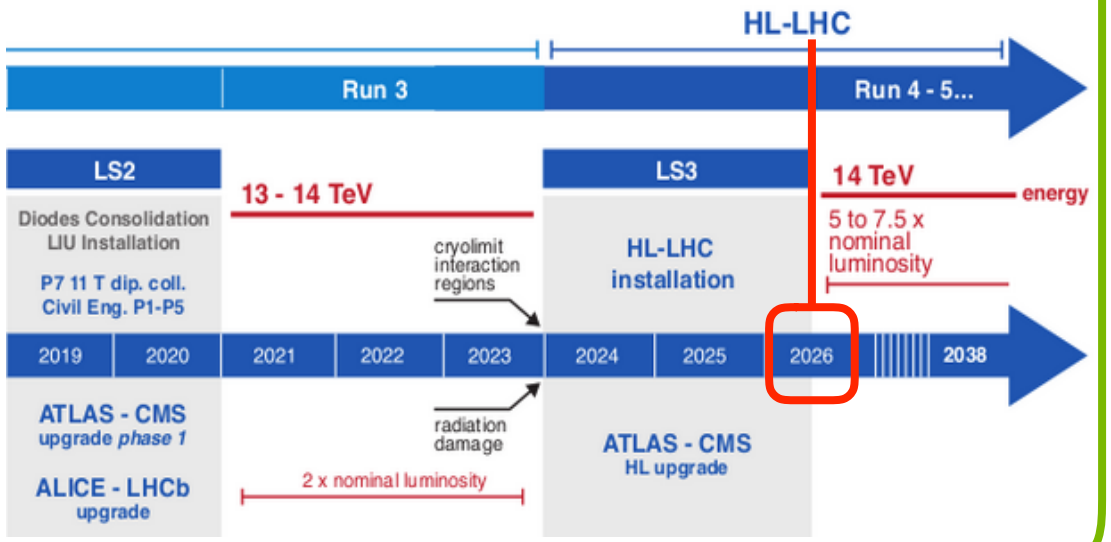
https://press3.mcs.anl.gov/hepcce/

ASCR HPC for HEP: Friend or Foe?

Basic HPC Arithmetic:

- Current HEP computing worldwide is ~1M cores (~few Pflops); US share is ~1Pflop
- May be a few PFlops by the mid-2020's; HL-LHC requirement is (very) roughly order of magnitude more (but could be less)
- By 2022, the LCFs and NERSC will have ~Eflops of compute power, by mid 2020's, tens of Eflops — ***4 orders of magnitude more than all dedicated US HEP computing combined***
- HEP usually gets ~10% of ASCR resources, assuming LHC experiments get 10% of this, still have 2 orders of magnitude leeway — ***10X the equivalent of the total compute requirement***
- However — ***obviously not as simple as that!***





U.S. DEPARTMENT OF ENERGY | Office of Science

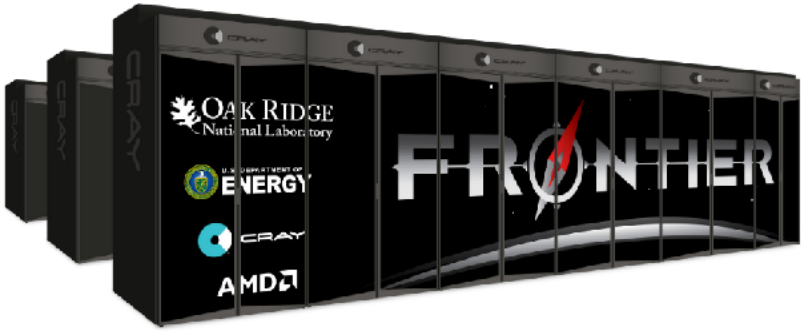
ASCR HPC for HEP: Friend or Foe?

Dose of Reality:

- Next-gen architecture characteristics
  - ***designed for massive (local) concurrency***
  - ***difficult to program***
  - ***future is very hard to predict***
- HEP experiment software not up to the task as yet (but some encouraging early results)
- ECP project making solid progress (but note these **are** HPC codes)
- HEP experiment software is in the millions of lines of code, but ***most of it probably does not need to be refactored***
- DOE/SC HPC codebase is probably O(100M) lines of code; include NNSA, etc., possibly a billion LOC — next-gen computing is a ***problem faced by everyone***

Public Specs on Aurora and Frontier:

- Performance: >1EF
- System interconnect: Cray Slingshot
- Compute node: CP+GPU
- System Memory: ~10PB
- Storage: >200PB at ~10TB/s
- # Cabinets: ~100



One cabinet is ~10PFlops!
Will be already obselete in the HL-LHC era

U.S. DEPARTMENT OF ENERGY | Office of Science

ASCR HPC for HEP: Friend or Foe?

Dose of Reality (contd.):

- Many other ***data-intensive use cases*** exist on the HL-LHC timescale (light sources, genomics, distributed sensors, —)
- Main issues:
  - exploit ***concurrency*** (SIMD/SIMT etc.)
  - be ***portable;*** able to run on all systems and get the same answer (nontrivial)
- Secondary issues are technical (I/O, storage, edge services, temperamental HPC systems, queue policies) and political, but potentially surmountable

HEP-CCE Plan

- HEP and ASCR researchers at HEP Labs working together on:
  - **pilot projects on concurrency/offloading (leveraging ongoing work by experiments)**
  - **data model/structure issues and IO/storage**
  - event generation
  - complex distributed workflows on HPC systems

BROOKHAVEN NATIONAL LABORATORY

Fermilab

BERKELEY LAB

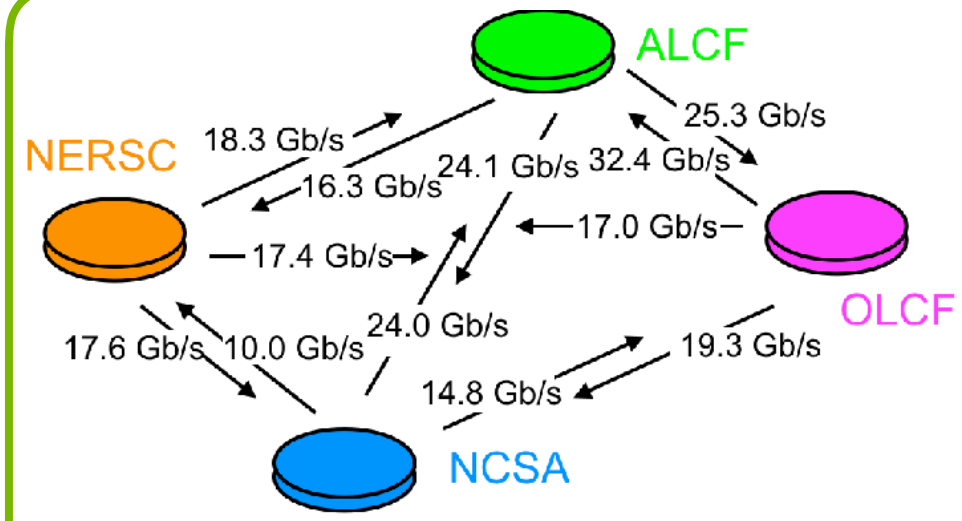
Argonne NATIONAL LABORATORY

U.S. DEPARTMENT OF ENERGY | Office of Science

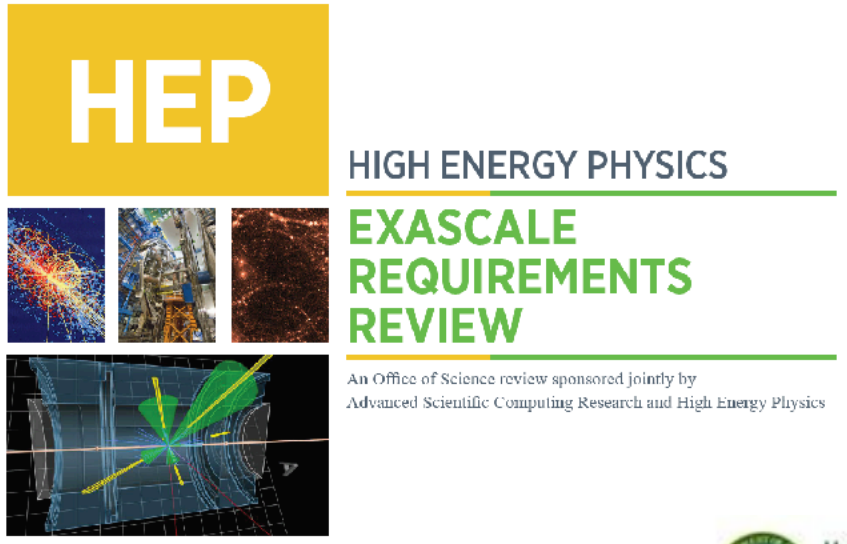
HEP-CCE and Other ASCR/HEP Programs (ECP and SciDAC-4)

HEP-CCE

- HEP-CCE activity examples:
  - Containers on HPC systems (NERSC, ALCF)
  - Joint work with ESnet on Petascale Transfer project across 4 HPC facilities
  - Summer student programs
  - HEP code ports on ALCF systems and at NERSC, FastCaloSim on GPUs
  - Edge services and data portals
  - Assessment studies (GalSim, Sherpa, WCT, —), community reviews/papers
  - Hackathons (KNLs, GPUs, —)
  - HPC-oriented workshops (event generators, I/O)



Petascale data transfer project: "Fire and forget" PBs in days under real-life conditions



HEP-CCE organized ASCR/HEP requirements review (2017)

U.S. DEPARTMENT OF ENERGY | Office of Science

HEP-CCE and Other ASCR/HEP Programs (ECP and SciDAC-4)

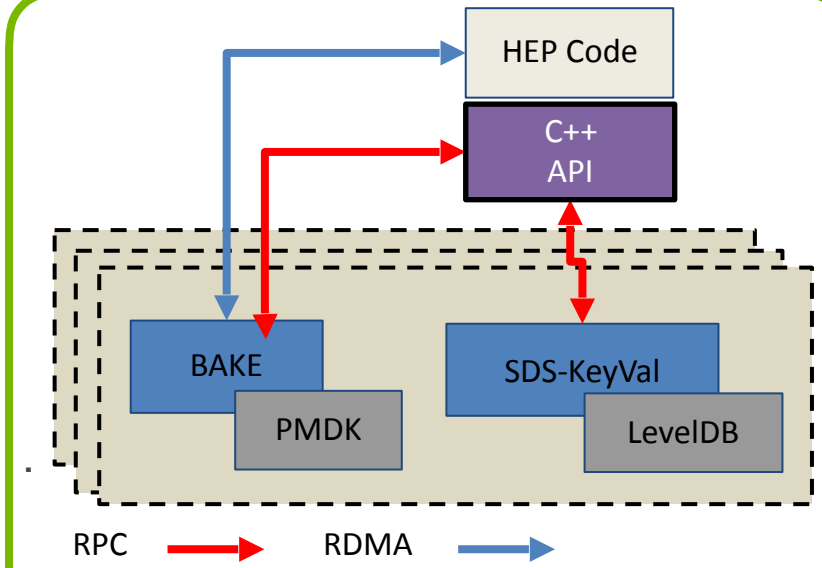
ECP, SciDAC, others:

- ECP: ExaSky, ExaLearn, LatticeQCD, WarpX, AMReX/ CoPA/ST efforts (e.g., Kokkos, SOLLVE, —)
- Multiple ASCR/HEP SciDAC-4 projects; example: HEPnOS, fast event store for HEP (also RAPIDS institute)
- Geant and GPUs — pilot project with ECP: Stand-alone application, minimal external dependencies, minimum set of physics and software framework processes to describe a HEP simulation problem, understand trade-offs in CPU/ GPU space

https://www.exascaleproject.org

HEP ECP project example:

- ExaSky — extreme-scale cosmological simulations, ~0.5M lines of code, two main codes (HACC and Nyx), one large-scale analysis framework
- HACC has about 300K lines of code, 95% are machine-independent C++/MPI (C++/MPI + 'X' paradigm)
- Typical HACC performance is around ~50+% of peak on all DOE systems, ***independent of architecture***
- Note: "HPC application and software development is a ***contact sport***" — Doug Kothe



The HEPnOS event store is built using Mochi, a framework for developing specialized HPC data services. Mochi allows use of state-of-the-art libraries (e.g., LevelDB for key/value storage) while providing convenient APIs.

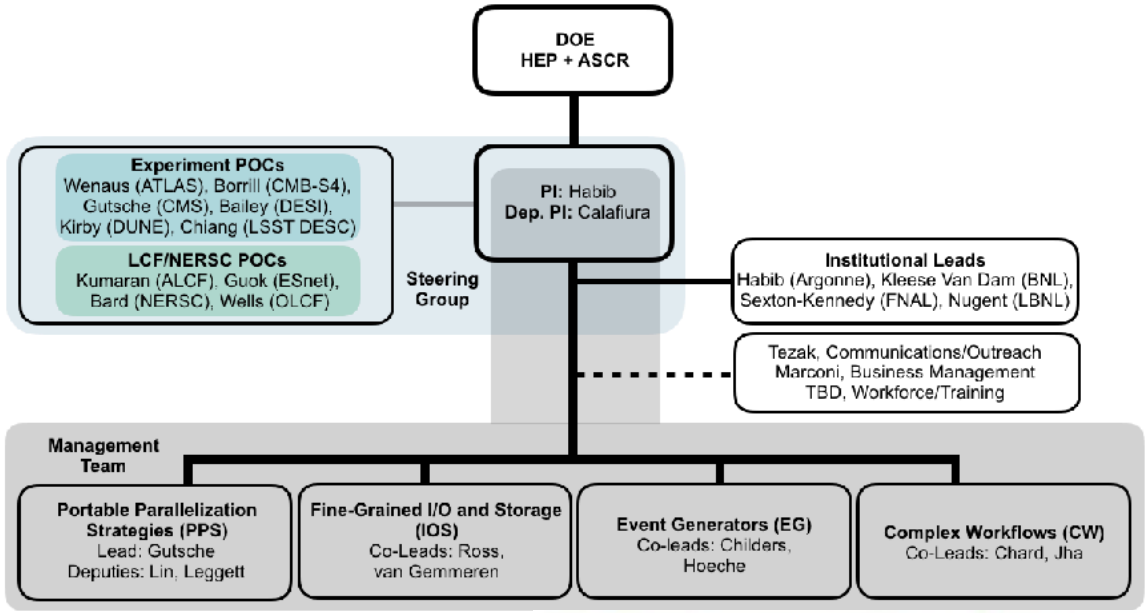
HEP-CCE Future Program (Proposed)

(Proposed) Program:

- **Portable Parallelization Strategies:** Strategies to explore ***evolutionary*** approach to HEP software on HPC systems:
  - Concurrency exploits for next-gen systems
  - Portability requirements
- **Fine-Grained I/O and Storage:** Strategies for dealing with data handling (e.g., file I/O issues):
  - New data models
  - Event subdivision
- **Event Generators:** Accelerator-oriented developments for HL-LHC
- **Complex Workflows:** Dynamic workflows with GPU exploits

Programming Models:

- **Kokkos:** C++ performance portability abstraction layer, multiple backends, native C++ features
- **RAJA:** Similar aim as Kokkos (LLNL applications)
- **OpenMP:** Multi-threading programming model on multi/many-core CPUs, accelerator offloading support
- **SYCL:** C++ abstraction layer on top of OpenCL, allowing for heterogeneous device support (precursor to DPC++); similarities with Kokkos
- **Numba:** Python JIT compiler support, accelerators



Back-Up Info

PPS

- Data structures that support SIMD/SIMT parallelization targeting CPU/vector units and GPUs
- Efficient communication between CPUs and GPUs
- Memory layout optimization to enable bacth operations spanning multiple events
- Potential use cases: FastJet, Track seeding, LAr TPC data preparation
- Investigate Kokkos, RAJA, OpenMP, DPC++/SYCL, etc. as performance portability models (also Python/Numba)

Timescales

- FY20/Q1 — US ATLAS/CMS ops program HL-LHC R&D strategic plan
- FY20/Q1 — DUNE software and computing CDR
- FY20/Q2-Q3 — ATLAS/CMS interim R&D statements
- FY21/Q3 — CMS High Level Trigger TDR
- FY22/Q1 — ProtoDUNE-II beam operations, DUNE Software and Computing TDR
- FY22/Q3 — WLCG and ATLAS/CMS Software and Computing TDRs