FCE – Starter projects

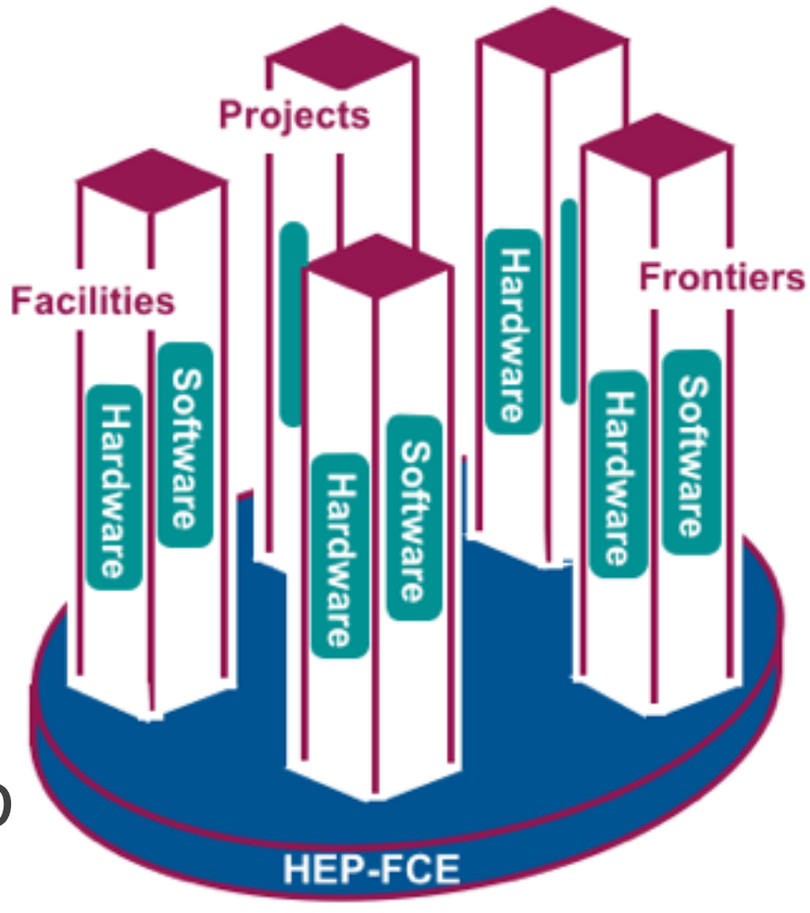
Rob and Salman

Kickoff Meeting

31 August, 2015

Logic for Starter Projects

- What is needed that is not being currently provided?
- Of these topics, which ones can be tackled in a small, cross-cutting team environment?
- What historical collaborations or activities or layers of expertise can be leveraged to achieve these aims?
- Will the projects provide broad benefits to multiple users/communities?
- Will the projects help to better connect HEP to itself and to other DOE offices?
- Can the projects be done so as to provide useful “success stories” in a short period of time?



Information Input

- A lot of studies already exist: HEP Topical Panel on Computing, Snowmass 2013, P5 2015, ASCR/HEP Exascale Requirements Review, most significantly the three HEP-FCE working group reports
- FCE reports: Applications Software, Software Libraries and Tools, Systems
- Reports are done, will be on the FCE website this week

HIGH ENERGY PHYSICS FORUM FOR COMPUTATIONAL PHYSICS:
WORKING GROUP REPORTS

I. APPLICATIONS SOFTWARE
II. SOFTWARE LIBRARIES AND TOOLS
III. SYSTEMS

Lead Editors: Salman Habib[1] and Robert Roser[2] (HEP-FCE Co-Directors)

Applications Software Leads: Tom LeCompte[1], Zach Marshall[3]
Software Libraries and Tools Leads: Anders Borgland[4], Brett Viren[5]
Systems Lead: Peter Nugent[3]

Applications Software Team:
Makoto Asai[4], Lothar Bauerdick[2], Hal Finkel[1], Steve Gottlieb[6], Stefan Hoeche[4], Tom LeCompte[1], Zach Marshall[3], Paul Sheldon[7], Jean-Luc Vay[3]

Software Libraries and Tools Team:
Anders Borgland[4], Peter Elmer[8], Michael Kirby[2], Simon Patton[3], Maxim Potekhin[3], Brett Viren[3], Brian Yanny[2]

Systems Team:
Paolo Calafiura[3], Eli Dart[3], Oliver Gutsche[2], Taku Izubuchi[5], Adam Lyon[2], Peter Nugent[3], Don Petravick[9]

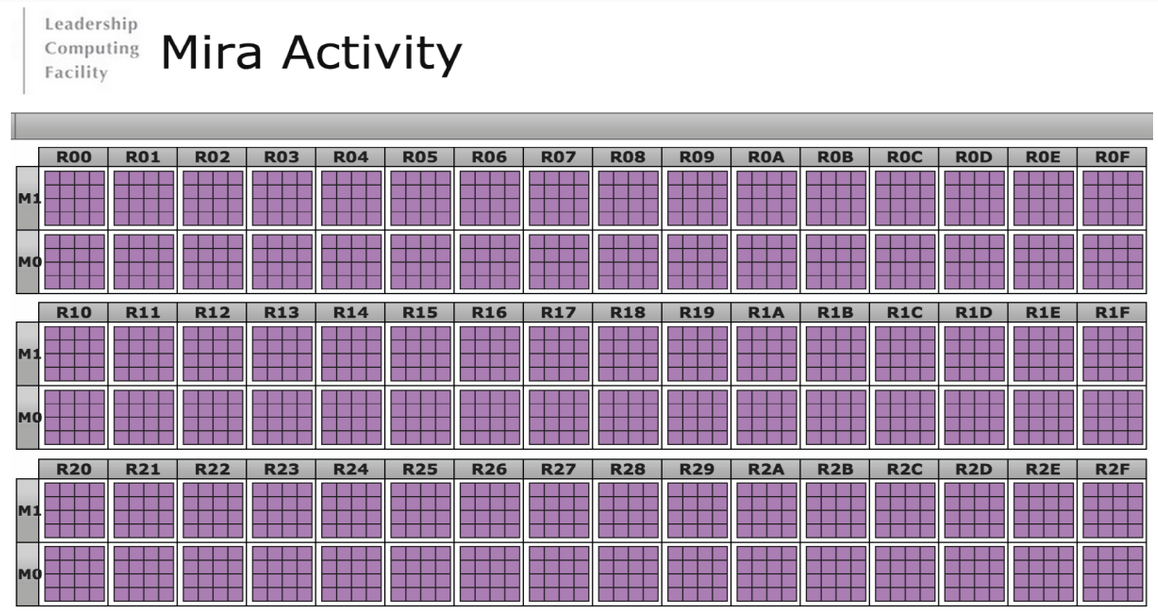
[1] *Argonne National Laboratory, 9700 S. Cass Ave., Lemont, IL 60439*
[2] *Fermi National Accelerator Laboratory, Batavia, IL 60510*
[3] *Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720*
[4] *SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, CA 94025*
[5] *Brookhaven National Laboratory, Upton, NY 11973*

Major Conclusions and “Hot Spots”

- Efficient execution of HEP workloads on next-generation architectures and computational resources, I/O bottlenecks
- Data access and storage technologies
- Shared libraries and tools — problems for small experiments, too many “local” solutions, lack of set of uniform best practices
- Only limited consolidation of applications software components possible
- Community generally supportive of HEP software developers (result of survey)
- Enhance visibility of HEP computational scientists within the community (major software projects <—> major detector R&D)
- Importance of improving the human element, training and other knowledge enhancing opportunities (cross-cutting projects)

HEP-FCE Initial Priority Areas (2 years)

- Next-generation architectures and supercomputer applications ("HPC for HEP")
- Data-intensive/Cloud computing (virtualization/ containers)
- High-speed networking (as a turnkey production resource)
- ASCR-HEP interactions (workshops, JLSE, —)
- Cross-cut software development
- HEP-FCE infrastructure support and community development



ATLAS event simulation on Mira: While this job was running, Mira was producing the equivalent computing as 5-6 ATLAS Grids. "On our best days, we provide the equivalent computing capacity of the whole ATLAS Grid." — Tom LeCompte

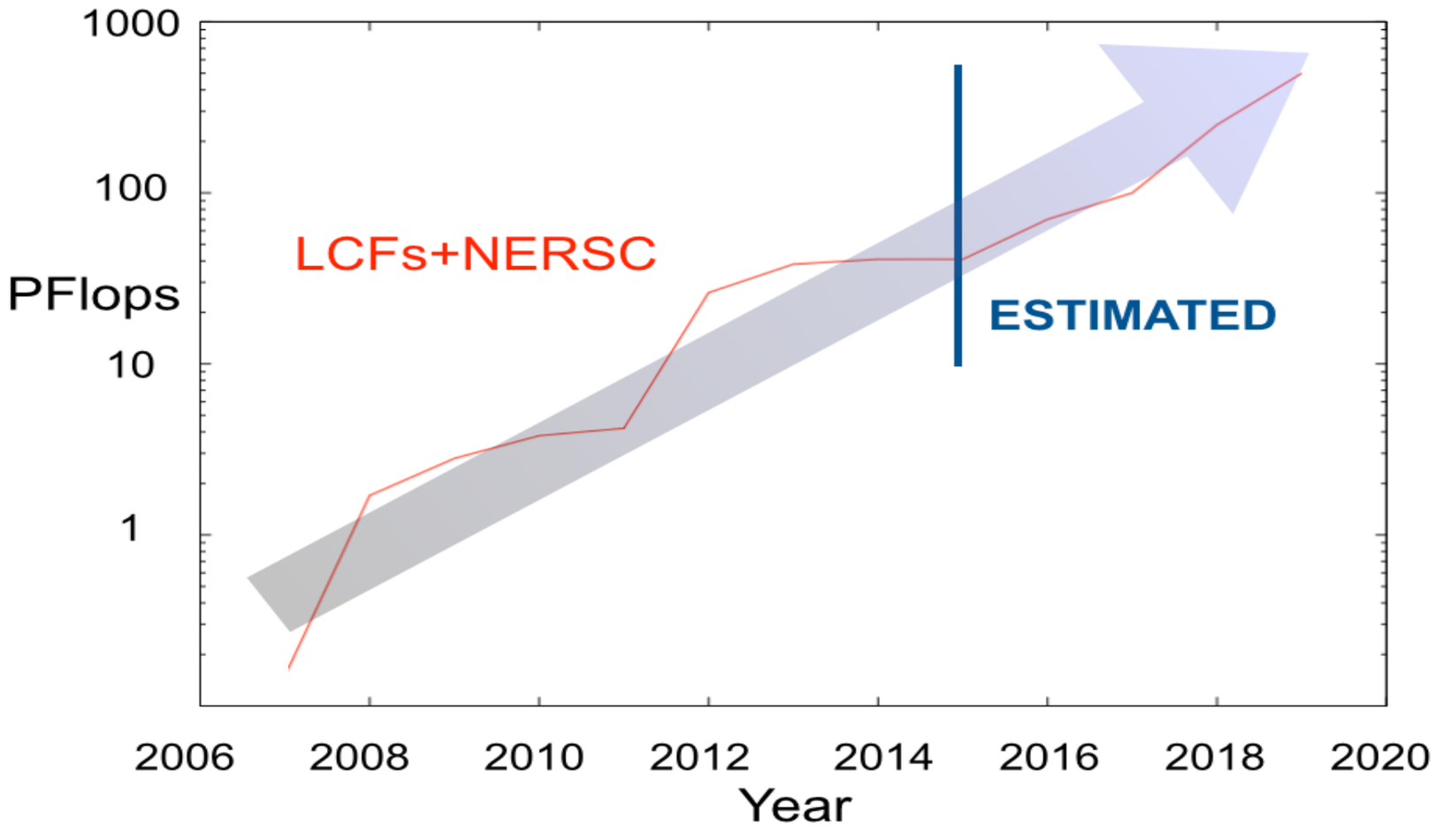
DECam Processing at NERSC

- **Project began, from scratch at NCSA, on June 2nd**
  - NCSA was able to move their code onto a set of docker containers in 2 weeks and run it locally and determine it produced the same results
    - Biggest issue was getting HTCondor to play nicely through docker with their server at NCSA.
    - Tested bandwidth for image processing and determined that a raw image (1GB) and processed image and catalogs (4+GB) could be shipped via ESnet from NCSA to NERSC and back in 1/50th the time it took to process (negligible I/O)
    - Pushed the container to docker.io so it could be pulled to NERSC

“HPC for HEP”: Resources

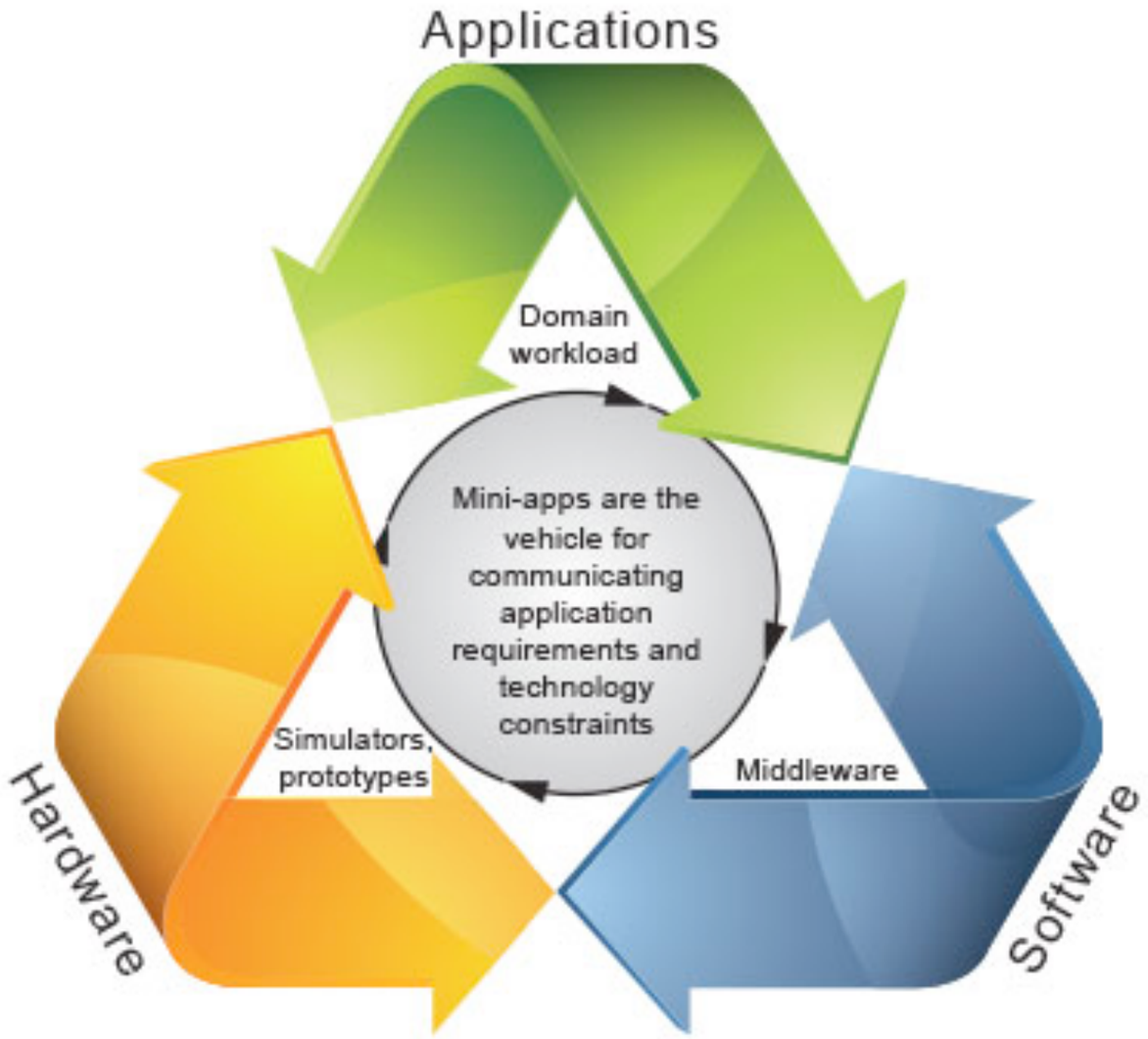
LCFs	*NERSC*	*ESnet*	*HEP Facilities*	*Local Resources*
• Expert users • Very large HPC calculations • Timely allocations • Data simulation • Software suite optimized for LCF use	• Wide range of users • Mission-lifetime allocations • Data archives • Data simulation and analysis • General purpose software available	• Network management solutions and optimization • LCF/NERSC high-speed data transfers • HEP services	• HEP users • Mission-specific services • Data archives • Data simulation and analysis • Software suite optimized for HEP tasks	• Local users and allocations • Specific calculations • Data analysis (partial datasets) • Locally specific software suite

- ASCR HPC resources overwhelm HEP compute (X 1000)
- But need to understand I/O nature of HEP tasks
- Also code refactoring is a major problem, does this need to be done anyway?



“HPC for HEP”: Next-Generation Architectures

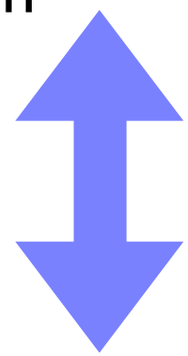
- Exposure to architecture “swim-lanes” — CPU/GPU, manycore, nontrivial memory hierarchy, special architectures (e.g., IBM’s TrueNorth)
- I/O bound applications — NVRAM exploits
- Edge servers/job automation
- Portability/Performance issues, useful metrics, overall optimization
- “Mini-Apps” for help with code (re-)design and performance modeling



Co-design loop using mini-apps to approximate the workload of the parent application (LLNL STR 2013)

Data-intensive Computing

- High Performance Computing
  - Parallel systems with a fast network
  - Designed to run tightly coupled jobs
  - High performance parallel file system
  - Batch processing
- Data-Intensive Computing
  - Parallel systems with balanced I/O
  - Designed for data analytics
  - System level storage model
  - Interactive processing
- High Throughput Computing
  - Distributed systems with ‘slow’ networks
  - Designed to run loosely coupled jobs
  - System level/Distributed data model
  - Batch processing

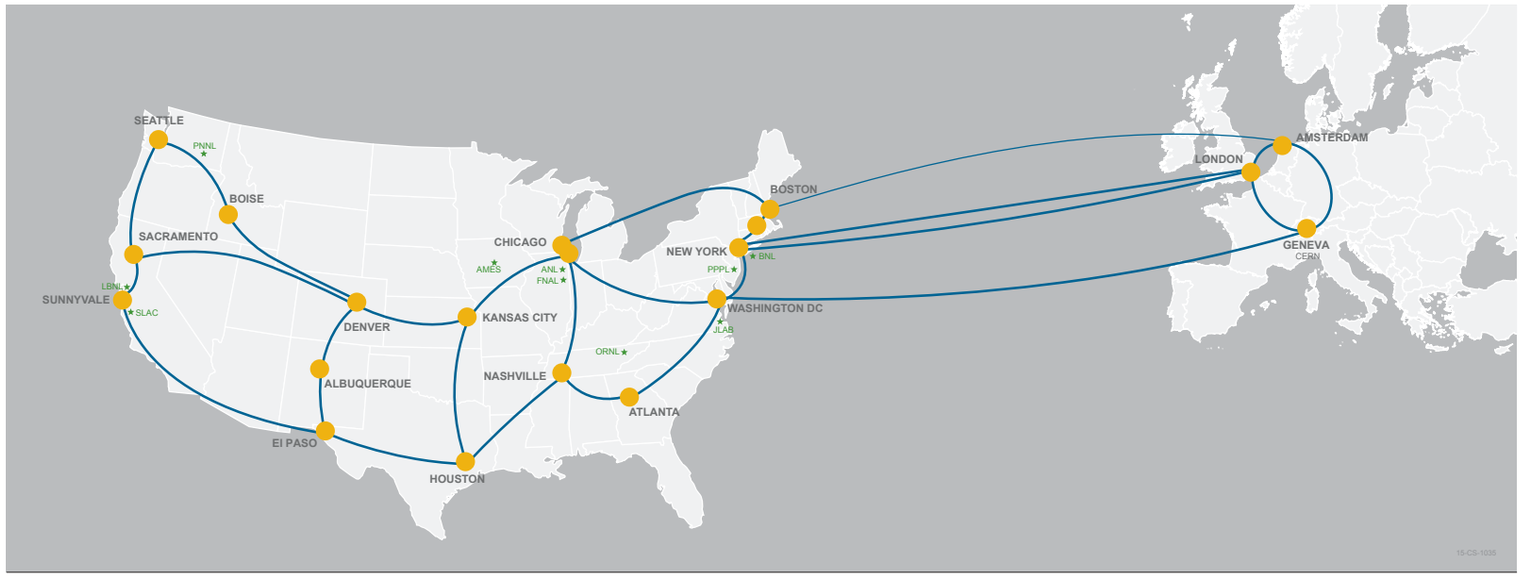


- Ability to run multiple complex software stacks on the same parallel platform (virtualization and containers)
- Resource elasticity to allow “interactive” access to the extent possible
- R&D work in understanding I/O patterns and their optimization
- Optimization of multiple compute resource use including available storage and networking (learn from new SciDAC project)
- Leverage ASCR/HEP Cosmic Frontier project (EASy-HEP)
- Potential “instant” customers include DES, DESI, LSST, and SPT

High-Speed Networking

- Many HEP data sources exist with high throughput demands on WANs
- Cosmic Frontier simulations set desired data transfer rate of 1Pbyte/week (file sizes from KB to ~100GB) as a robust production requirement, possible in principle on ESnet
- Many “mundane” bottlenecks are in the way
- Joint ESnet/HEP-FCE/DTWG project on solving this problem is underway (desired tools: Globus and HPSS, want FS to FS and HPSS to HPSS, etc.)
- Initial characterization and testing between ASCR HPC Centers (ALCF, NERSC, OLCF), add NCSA next, finally BNL and FNAL
- Need to deal with software “handshake” problems when crossing ASCR and HEP facility boundaries (NCSA may be less of a problem but not tested yet)

- Computational cosmology data transfer “data transfer test package” in the works, other use cases will follow
- Characterize pathways, DTNs, firewalls, identity management, chokepoints, failure modes, etc. and document them
- Implement dashboard for tracking transfers



More Open Areas —

- **ASCR/HEP Interactions:**
  Exascale requirements review just completed; report almost ready — many avenues for improved interaction with ASCR — 1) Need to avoid fragmented interactions (but this may be unavoidable given the nature of FOAs), 2) HPC, data-intensive computing, and networks are natural conduits, but already identified, 3) Other areas — software engineering? Uncertainty quantification?
- **Cross-Cut Software Development:**
  How can HEP-FCE contribute to this? 1) Big experiments can take care of their own problems, but their solutions often don't scale down, 2) Don't wish to interfere in ongoing software projects ("if it works —"), 3) Too many entrenched opinions in some areas, 4) Need to find good targets (best situation — they should self-identify)
- **HEP-FCE Infrastructure/Community Development:**
  How to create an environment for community engagement and support? 1) No reason to reproduce Hepforge, Stack Overfow, GitHub, Google — 2) No desire to be the "police", enforce standards, best practice notions, or get in the way of people trying to do work, 3) What would be helpful and does not currently exist?