HEP-CCE Portable Applications and Workflows

Charles Leggett
for HEP-CCE

ATLAS Computing and Software Activities at BNL Workshop
March 18 2024

Introduction

- HEP experiments run mission-critical workflows on owned and pledged resources (such as **OSG** and **WLCG**), but also need to leverage **HPC** and **commercial cloud** facilities to deliver timely physics output.
- The (anticipated) increase of data volume exacerbates the need to use HPC resources at the DOE leadership class facilities (LCFs).
- In Phase 1, **PPS** addresses **node-level** parallelization and portability issues of running HEP applications on LCFs, and we will continue to work with the experiments to address these issues and implement PPS solutions in production.
- In Phase 2, we will address the issues of running **complex workflows** on the LCFs, and develop a cross-cutting HEP workflow portability overlay to help HEP experiments build portable high-throughput workflows across different computing facilities.

HEP Experiment Workflows and HPC

- While HEP experiments can benefit greatly from the efficient use of the HPC systems, many challenges remain.
- HEP experiment workflows have unique characteristics and requirements that are not currently accommodated on the LCFs:
    - HEP workflows are highly **non-uniform:**
        - Different simulation and analysis steps have different potentials for HPC acceleration with varying computing resource requirements (some tasks take longer than others)

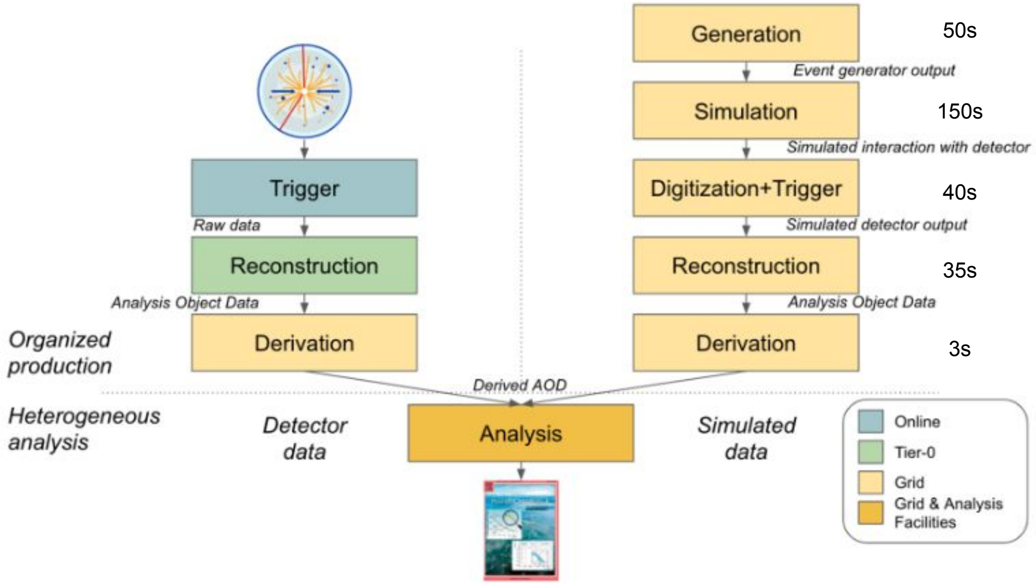
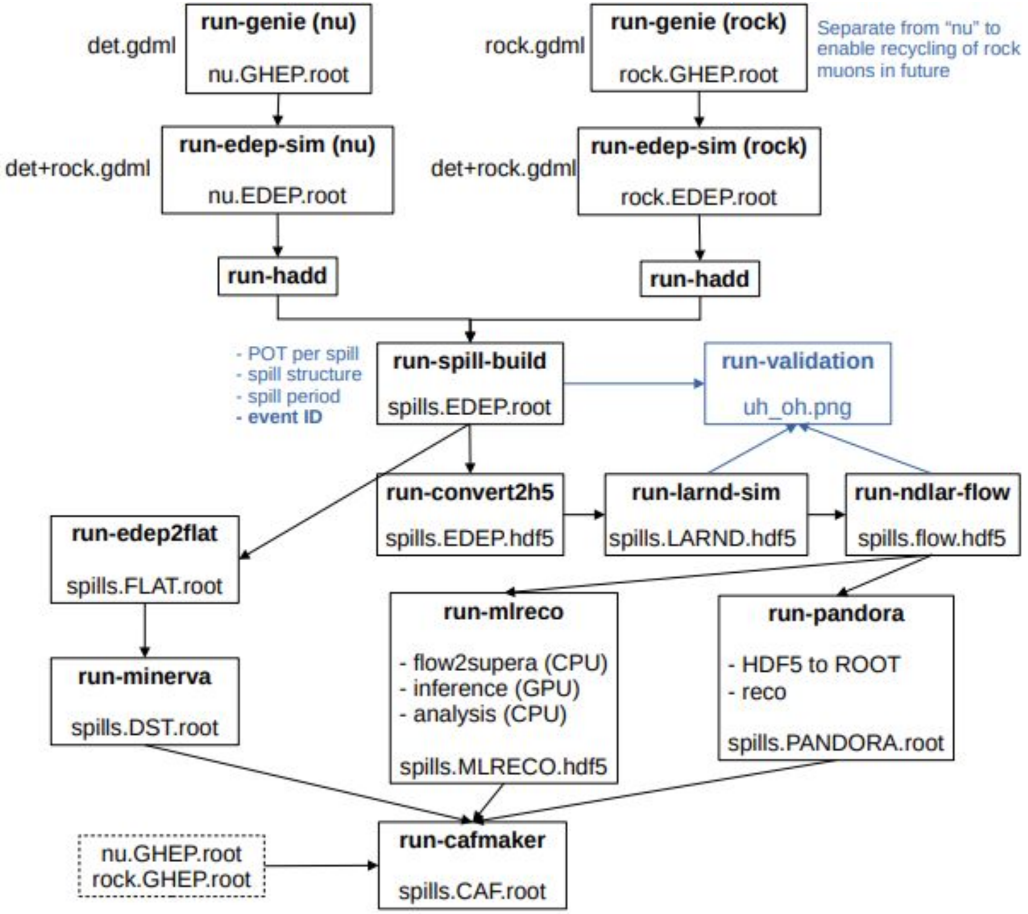


Figure 1: Workflow overview in the ATLAS and CMS experiments.

Figure from Megino, et al. "US ATLAS and US CMS HPC and Cloud Blueprint." *arXiv:2304.07376* (2023).

HEP Experiment Workflows and HPC

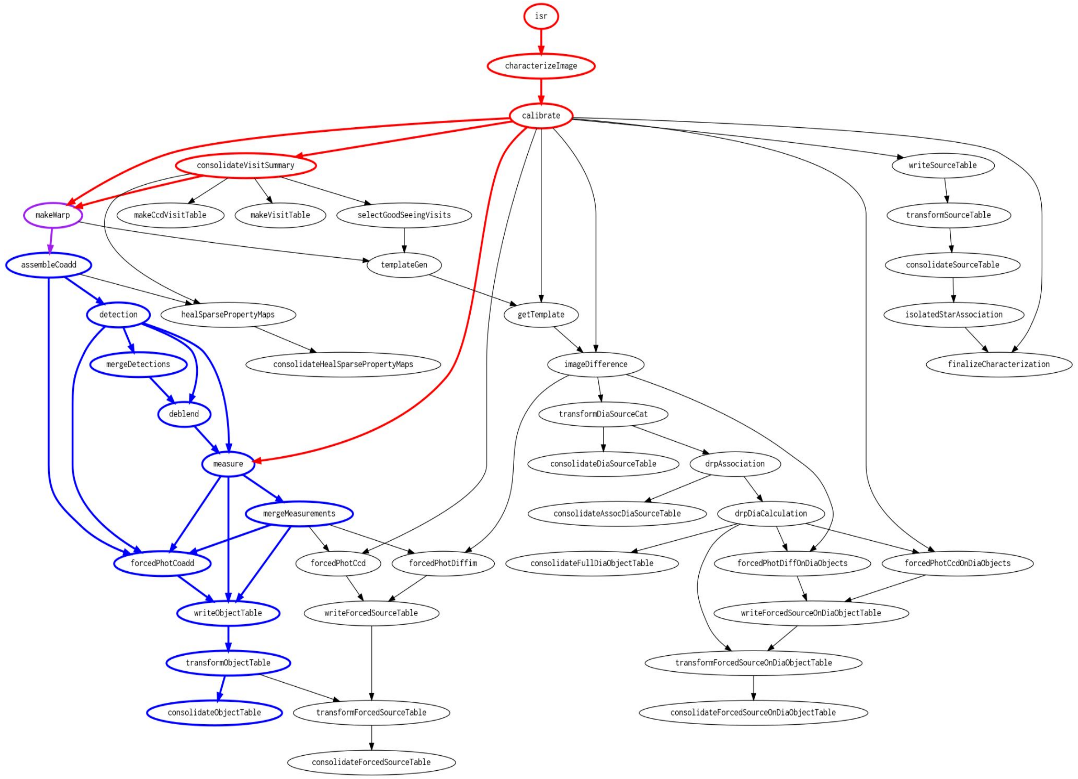
- While HEP experiments can benefit greatly from the efficient use of the HPC systems, many challenges remain.
- HEP experiment workflows have unique characteristics and requirements that are not currently accommodated on the LCFs:
  - HEP workflows are highly **non-uniform**
  - and increasingly **non-linear**



DUNE Near Detector 2x2 Simulation Workflow

HEP Experiment Workflows and HPC

- While HEP experiments can benefit greatly from the efficient use of the HPC systems, many challenges remain.
- HEP experiment workflows have unique characteristics and requirements that are not currently accommodated on the LCFs:
    - HEP workflows are highly **non-uniform** and increasingly **non-linear**: Different simulation and analysis steps have different potentials for HPC acceleration with varying computing resource requirements (some tasks take longer than others)
    - Need for real-time or on-demand access to resources



Rubin LSST Image Processing Pipeline

HEP Experiment Workflows and HPC

- While HEP experiments can benefit greatly from the efficient use of the HPC systems, many challenges remain.
- HEP experiment workflows have unique characteristics and requirements that are not currently accommodated on the LCFs:
  - HEP workflows are highly **non-uniform** and increasingly **non-linear**: Different simulation and analysis steps have different potentials for HPC acceleration with varying computing resource requirements (some tasks take longer than others)
  - Need for real-time or on-demand access to resources
  - Large data volumes make data delivery, cataloging, and storage challenging
  - Many HEP computational tasks are still CPU-based, with spare use of GPUs, while HPC systems are increasingly GPU-based.

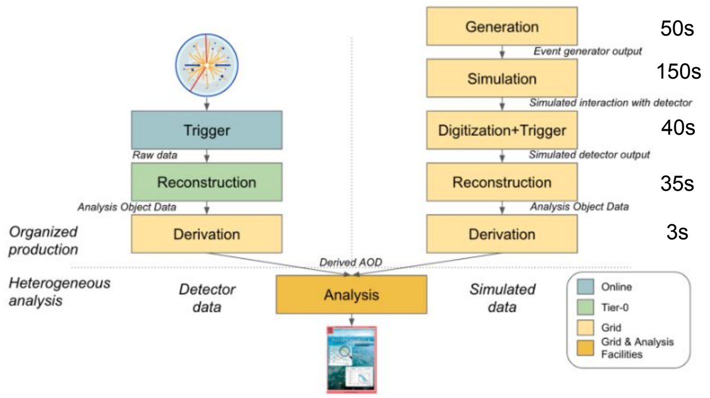
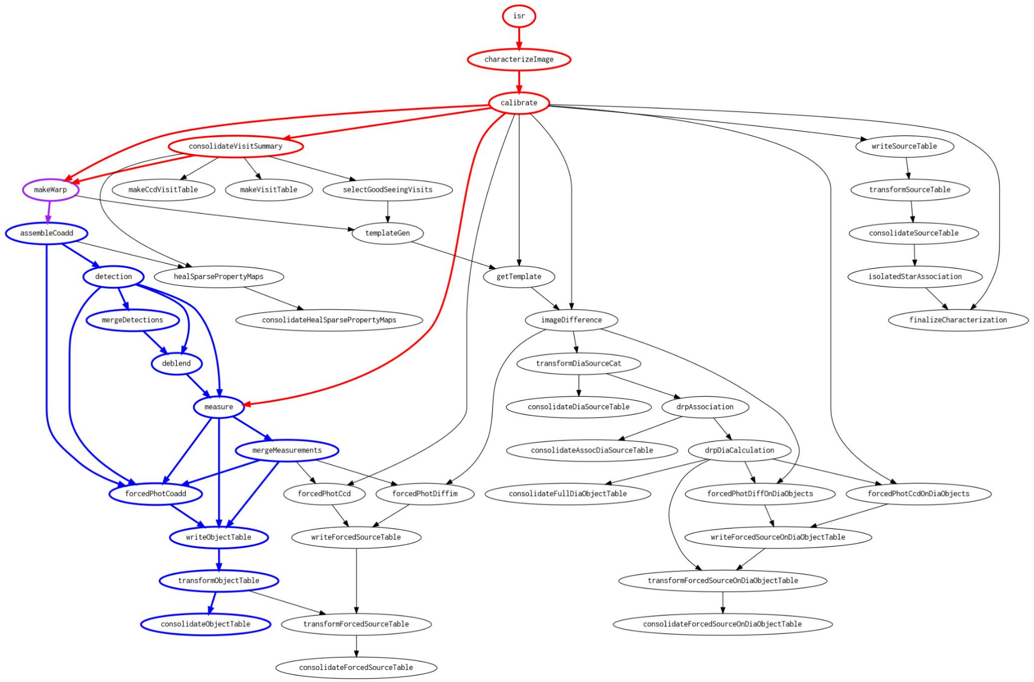


Figure 1: Workflow overview in the ATLAS and CMS experiments.

Figure from Megino, et al. "US ATLAS and US CMS HPC and Cloud Blueprint." *arXiv:2304.07376* (2023).



Rubin LSST Image Processing Pipeline

Challenges of HEP Workflows on HPC Systems - I

- **Resource Access Challenge**
  - HPC centers have decentralized identity management, unlike WLCG
  - HEP compute resource needs may be non-linear, sometimes requiring **burst or real-time access** for specific science needs, for example,
    - transient alerts from the Rubin Observatory
    - candidate supernova neutrino flashes detected in DUNE (data intensive, time critical)
  - "Small" experiments may have to rely on HPCs for real-time monitoring of detector performance or calibration quality, such as the LZ experiment

- **Data Challenge**
  - HEP experiments deal with Peta to ExaBytes of data. Data on HPC will be transient in nature.
    - Good data cataloging and delivery mechanism is needed.
  - Some experiments will have extremely **high data rates for relatively short periods** of time, such as during supernova neutrino burst events for DUNE.
  - Getting the data in and out of the HPC centers efficiently requires commonly supported high-throughput services.

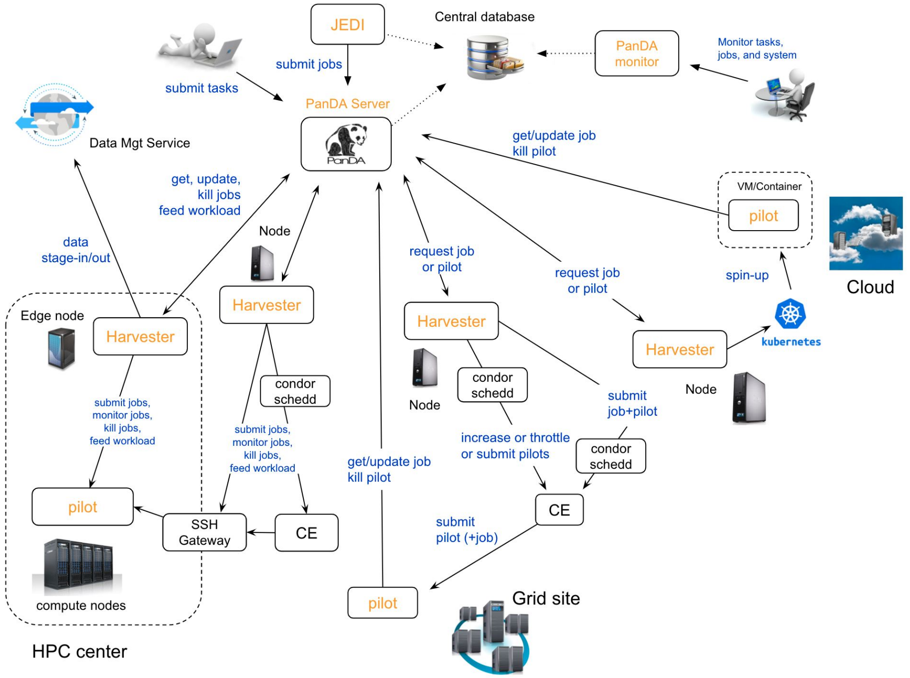
Challenges of HEP Workflows on HPC Systems - II

- **Software Environment Challenge**
  - HEP software support for HPC architectures varies
    - Different CPUs and GPUs (AMD, Intel and NVIDIA)
    - Partially addressed by **CCE Phase 1**
  - HPC center software environments differ
    - Different OS, compilers, batch systems (PBS, Slurm, …)
    - Even the supported container technologies may be different
  - Integration of HEP software frameworks with HPC services is non-trivial
    - cvmfs
    - Each experiment has tended to develop its own middleware tools

- **Performance, Reliability and Reproducibility Considerations**
  - With the diversity of architectures and software environments, how to guarantee reproducibility of the results becomes a challenge.
    - Need to have careful data cataloging and documentation
  - Heterogeneous and hybrid tasks in a HEP workflow may perform best on different hardware architectures (some work better on CPUs while others on GPUs)
    - Need to maximize performance with careful allocation and mapping of resources
  - Can we resume critical workflows elsewhere if the current system fails?
    - Need to look into resubmission/restart mechanism

Leveraging Current Workflow Technologies

- Both the **HEP** and **ASCR** communities have recognized these challenges and started developing tools and services to address them.
- **ATLAS** has developed a distributed workflow system that can interact with HPC, Cloud and Grid.
    - HEP-developed tools such as **Harvester, PanDA** may be leveraged for other workflows.
- **CMS** has successfully integrated their workflows with user-facility-type HPC centers through the **HEPCloud** portal.
    - Running on LCFs remains challenging
- **DUNE** offline computing CDR explicitly targets HPCs
    - Current plan to use a combination of **JobSub, GlideinWMS,** and **HEPCloud** for both Grid and HPC sites



ATLAS distributed workflow management system

HEP-CCE Phase 2 Plan

- In HEP-CCE Phase 2, our goal is to provide the experiments with both a validated, ready-to-use portability solution and a suite of portability tools that can be integrated into their production systems.
  - To reconcile different services and tools provided by HEP and ASCR.
  - To reduce the operation and maintenance overhead of deploying HEP workflows on HPC systems

- Building on the experience of **PPS** and **CW** groups in HEP-CCE Phase I, we will have two main tasks in Phase 2:
  - **Task 1:** apply lessons learned in PPS to help HEP experiments develop portability solutions in their applications
  - **Task 2:** develop portable, experiment-agnostic, workflow overlays to interface existing HEP workflows with HPC centers

Task 1: Applying Lessons Learned to HEP Experiments

- The goal of this task is two-fold:
  - capitalize on the Phase 1 PPS findings to help experiments develop portable solutions on more components of their workflows for HPC,
  - help HPC centers understand and consider HEP requirements for future software and hardware
- Phase 2 activities include:
  - Work with experiments to develop tailored application portability recommendations depending on the experiment size, codebase, data, and timescale.
  - Turn Phase 1 PPS test beds into representative HEP mini-apps to share with ASCR facilities to help define requirements and KPPs for facility infrastructure.
  - Develop experiment-independent algorithmic examples/benchmarks that could be used for training and form the basis of a portable parallelization "cookbook."
  - Some of the benchmarks can be contributed to community standard benchmarks such as SPEChpc, HEPScore, etc. to ensure HEP requirements are well supported by future HPC software and hardware

Task 2: Develop Workflow Portability Overlay

- The overarching goal of this task is to enable diverse HEP experiment workflows to run efficiently on LCFs and other HPC centers with little overhead.

- This will be done through the development of a portability overlay that would include a set of tools and services to seamlessly integrate HEP workflows with HPC, such as
  - Software delivery and container management
  - Scalable, distributed execution engines
  - Application services including Function as a Service (FaaS) microservices like funcX
  - Accelerator/Inference as a Service (AaaS) microservices like NVIDIA Triton, DLHub, etc.
  - Identity management (following rules of engagement as set by the facilities)
  - Computing and storage resource brokering with a focus on resource availability and overall throughput.
  - Edge services, including pilot management (Harvester, HEPCloud), remote logging and reporting, and database access
  - Data cataloging, delivery, and access, leveraging XRootD, Globus, Rucio

Year 1 Plan

Task 1 - Application Portability:

- Develop a **cookbook** for portability layers based on Phase 1 findings
- Outreach to experiments for portable solution implementation (**workshops/hackathons**, followed by regular office hours)
  - Understand the experiments' timescales for portable accelerator uses
- Create **mini-apps** based on two of the Phase I PPS testbeds that can be executed at NERSC, OLCF and ALCF, preferably with the same software environment (FCS, p2r)
- Use mini-apps to extract **figures of merit** for ASCR facilities and LCFs to use as baselines

Task 2 - Workflow Portability:

- Complete **survey** of existing HEP experiment workflow technologies on HPC; also look into workflow technologies used by **other experiment facilities such as light sources**.
  - Find commonalities between experiment workflow systems
- Explore the needs of HEP in terms of **ML workflows/pipelines and microservices** (synergistic with the distributed ML activity)
- Investigate common layers and interfaces (batch scheduler, policies, pilots, …) to facilitate portability and interoperability across ASCR facilities in collaboration with **IRI testbeds**
- Create **2 representative HEP experiment workflows** to run two different HPC systems. Candidates include: LSST/DESC, LZ, **DUNE**, LHC Experiments (**ATLAS**/CMS).

fin

ASCR-Supported Tools

- DOE LCFs (NERSC, ALCF, and OLCF) are also developing tools to support experiment workflows. For example,
  - **Slate** at OLCF: container orchestration service for running user-managed persistent application services
  - **Spin** at NERSC: container-based platform to support user-defined services, workflows, databases and API endpoints.
  - The LCFs are also working on technologies to support cross-facility workflows. (Prelude to IRI, perhaps?)
- The US Exascale Computing Project (ECP) has generated a rich exascale-ready software ecosystem.
  - In particular, ExaWorks has developed a workflow SDK that can be adopted for HEP.

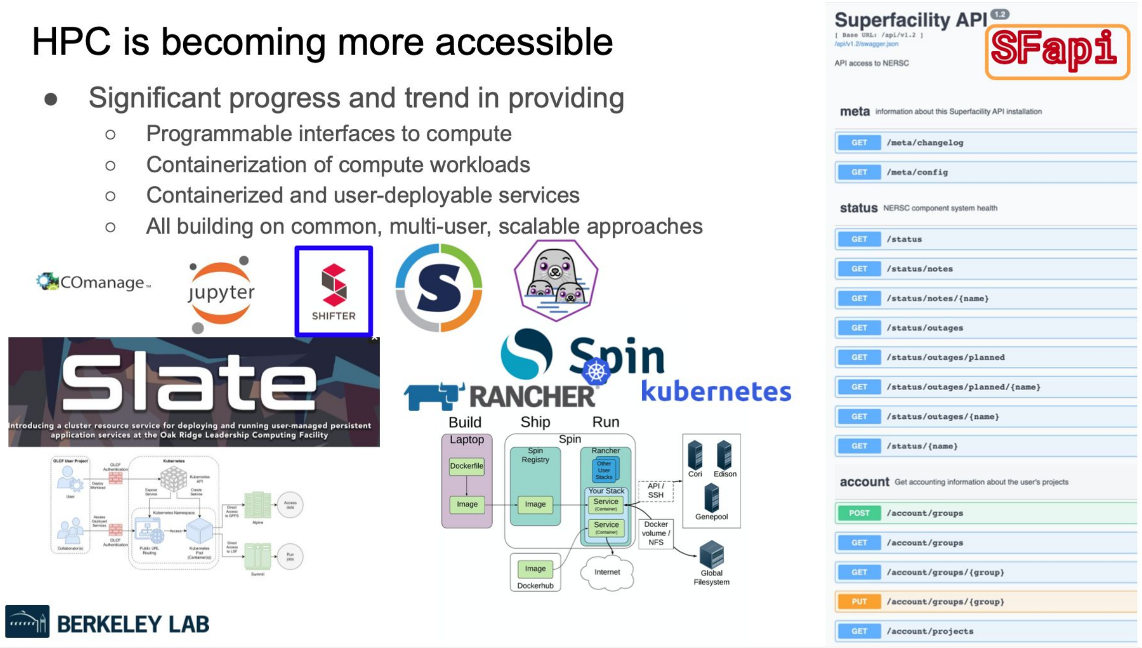
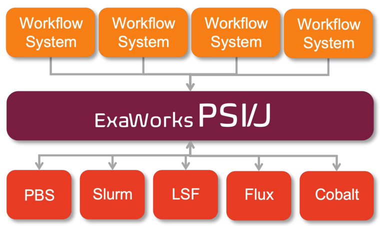


Figure 3: DOE LCFs are developing new tools to make HPC more accessible. Image taken from Wahid Bhimji (NERSC) presentation at Snowmass CompF4 topical workshop: https://indico.fnal.gov/event/53251/



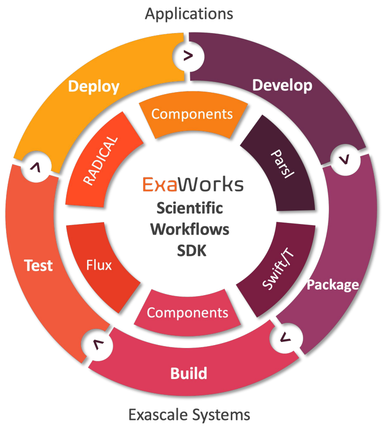


Figure 4: ECP project ExaWorks is also developing tools to support complex workflows on HPC systems. https://exaworks.org/