ExaWorks: ECP Workflows Project

Dan Laney, Lawrence Livermore National Laboratory

Dong Ahn, Lawrence Livermore National Laboratory

Kyle Chard, Argonne National Laboratory

Shantenu Jha, Brookhaven National Laboratory

National Nuclear Security Administration

U.S. DEPARTMENT OF ENERGY | Office of Science

Exascale Computing Project

EXASCALE COMPUTING PROJECT

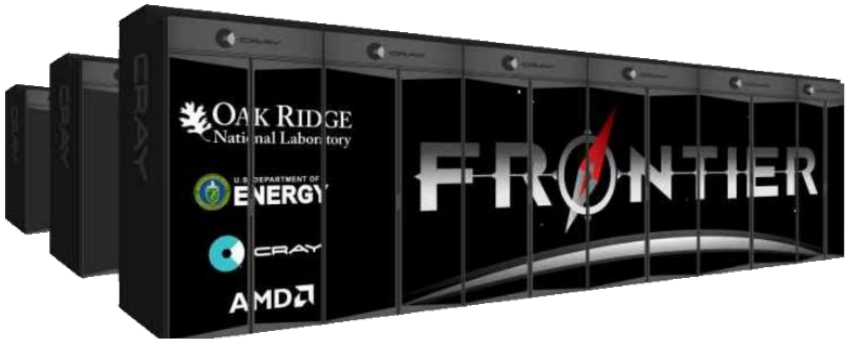
Seven-year, $1.8B project that aims to accelerate R&D, acquisition, and deployment of exascale computing capability to DOE

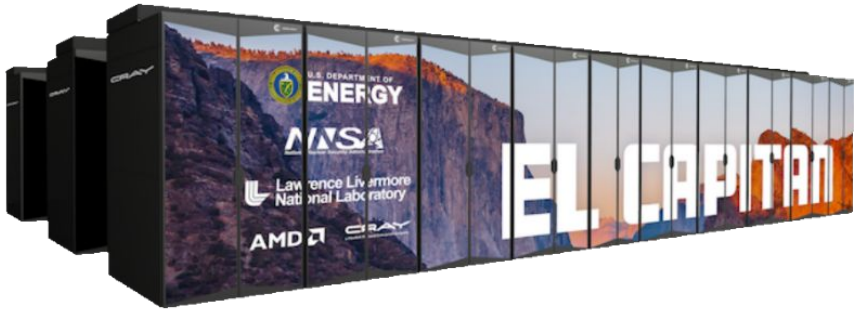
Six core national laboratories are focused on software, applications, hardware, system engineering and testbed platforms

Performant mission and science applications @ scale

Aggressive RD&D Project	Mission apps & integrated S/W stack	Deployment to DOE HPC Facilities	Hardware tech advances

Application Development (AD)	Software Technology (ST)	Hardware and Integration (HI)
Develop and enhance the predictive capability of applications critical to the DOE **24 applications** including national security, to energy, earth systems, economic security, materials, and data	Deliver expanded and vertically integrated software stack to achieve full potential of exascale computing **67 unique software products** spanning programming models and run times, math libraries, data and visualization	Integrated delivery of ECP products on targeted systems at leading DOE HPC facilities 6 US HPC vendors focused on exascale node and system design; application integration and software deployment to facilities





ExaWorks is an important and timely ECP project

- Workflows are crucial components of many ECP applications (e.g., CANDLE, ExaLearn, ExaSky)
- Workflows have traditionally been developed as monolithic, bespoke frameworks
  - Prone to scalability, portability, security, maintainability, extensibility, and fault tolerance issues
  - Development is inefficient (duplication across projects and facilities), siloed with little validation of performance/robustness, and significant sustainability impacts and delivery risks
- Exascale exacerbates these challenges:
  - Scale, throughput, workflow complexity, system heterogeneity, …

ExaWorks goals

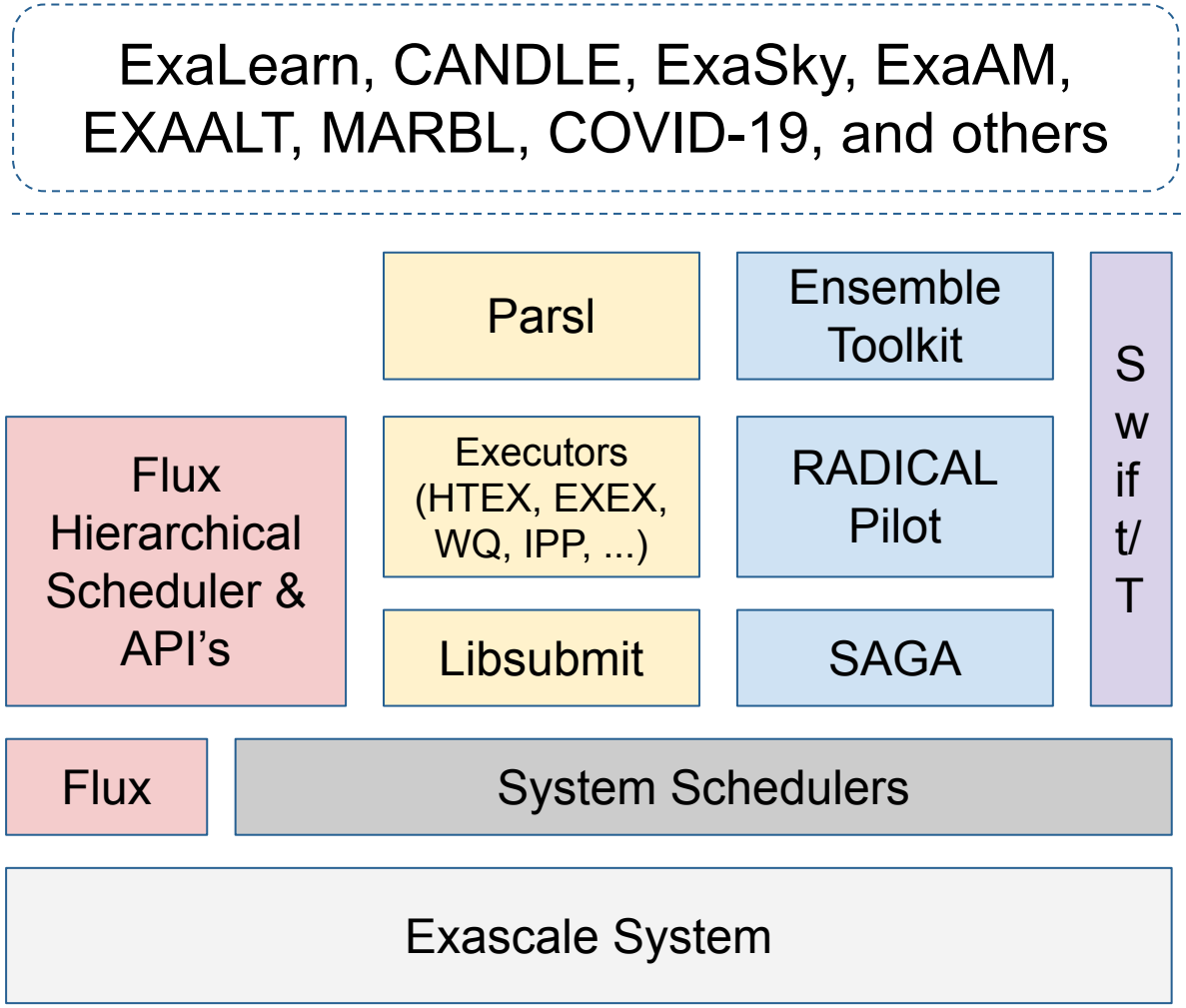
1. Build a robust, performant, extensible toolkit based on functional capabilities found in many resource and workflow management systems
2. Directly impact ECP and other applications
3. Deploy and test tools at DOE facilities; support future software stacks
4. Engage the broader workflow tools and users communities to create a sustainable, component-based workflows ecosystem

ExaWorks is an innovative approach to the workflow ecosystem

We focus on the modularity and flexibility that is sorely needed as the science community moves toward complex, heterogeneous workflows

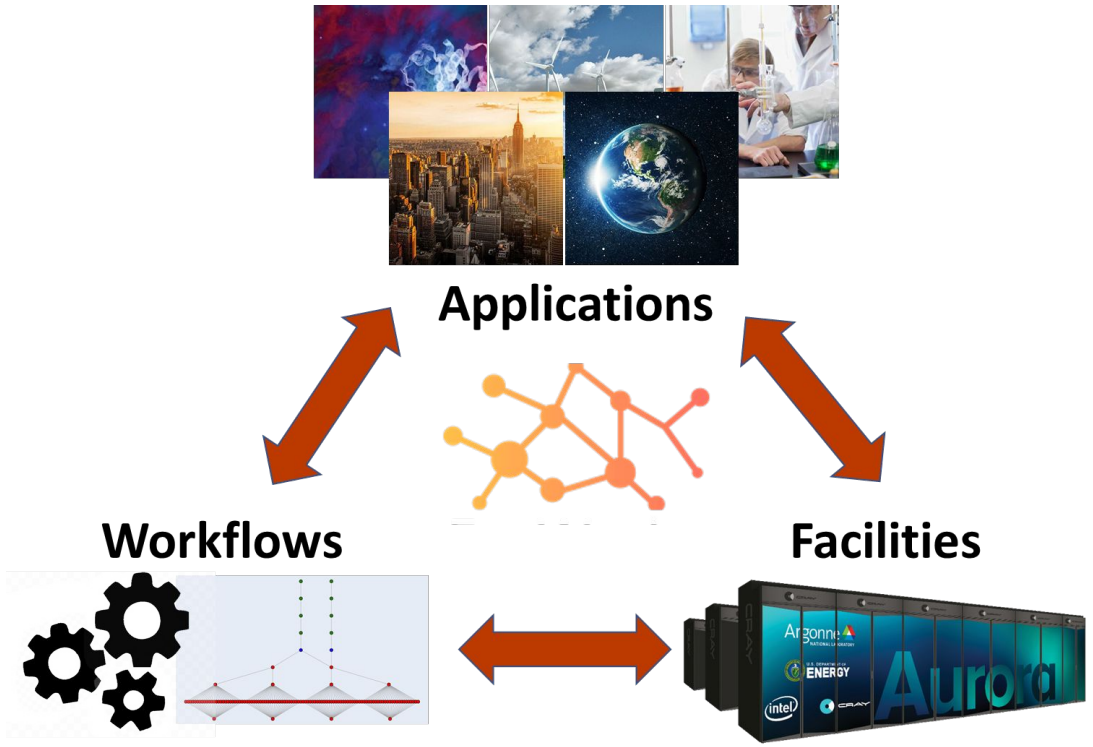
Seeding the toolkit with hardened components from robust workflow tools that are widely deployed, span the workflow stack, and are used on DOE systems (Parsl, Flux, RADICAL)

We aim to make an ***immediate impact*** **on ECP applications** to address exascale challenges



ExaWorks is a community focused project

- We are not attempting to replace or supplant existing approaches and standards with the ‘one true way’ to do workflow
- We will actively engage the applications, facilities, and workflows communities (including other WMSs)
- Community-based approach in which we work together to integrate diverse approaches and to collaboratively develop shared API’s



Addressing Exascale workflow challenges

- Emerging exascale workflows pose significant challenges to the creation of portable, repeatable, and performant WMSs.
- Scheduling and heterogeneity
  - Execution of heterogeneous tasks with complex interdependencies on heterogeneous resources (e.g., CPU/GPU, single and multi-node tasks, ms-hours)
- Scale, performance, and throughput
  - Systems will have thousands of nodes, millions of cores, and diverse accelerators == millions of independent computing elements to be scheduled
- Coordination and communication
  - Coordination the workflow and tasks, between tasks, and with external services
- Fault tolerance and robustness
  - Faults are inevitable in large-scale systems and when running enormous numbers of concurrent tasks
- Portability
  - Systems have different schedulers, launchers, container technology