Enabling Emerging Scientific Workflows with Converged Cloud and HPC Environments

SOS26

March 13, 2024

Daniel Milroy

Computer Scientist

LLNL

LLNL-PRES-861497

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory

HPC and cloud computing are merging into a seamless, environment that can benefit scientific workflows.

Scheduling and resource management are key components to ensuring the environment is efficient and high-performance.

We are teaming up with experts at major cloud providers to make the environment a force multiplier for scientific workflows and the broader computing community.

Pre-exascale scientific workflows strain the capabilities of traditional HPC resource managers and schedulers.

Co-scheduling:

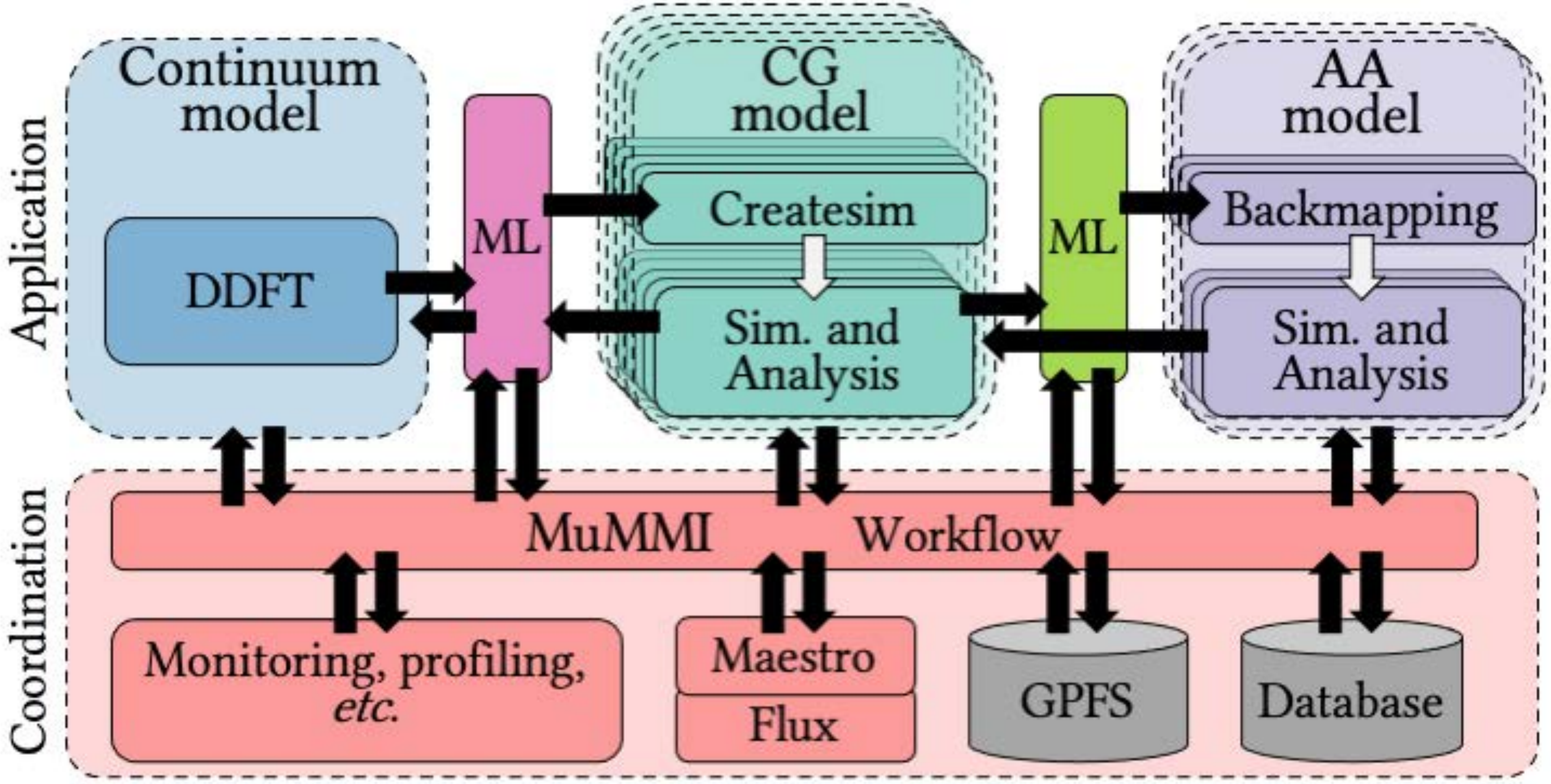
CG, analysis bound to cores nearest PCIe buses

Job comms/coordination:

36,000 concurrent tasks; 176,000 cores, 16,000 GPUs

Portability:

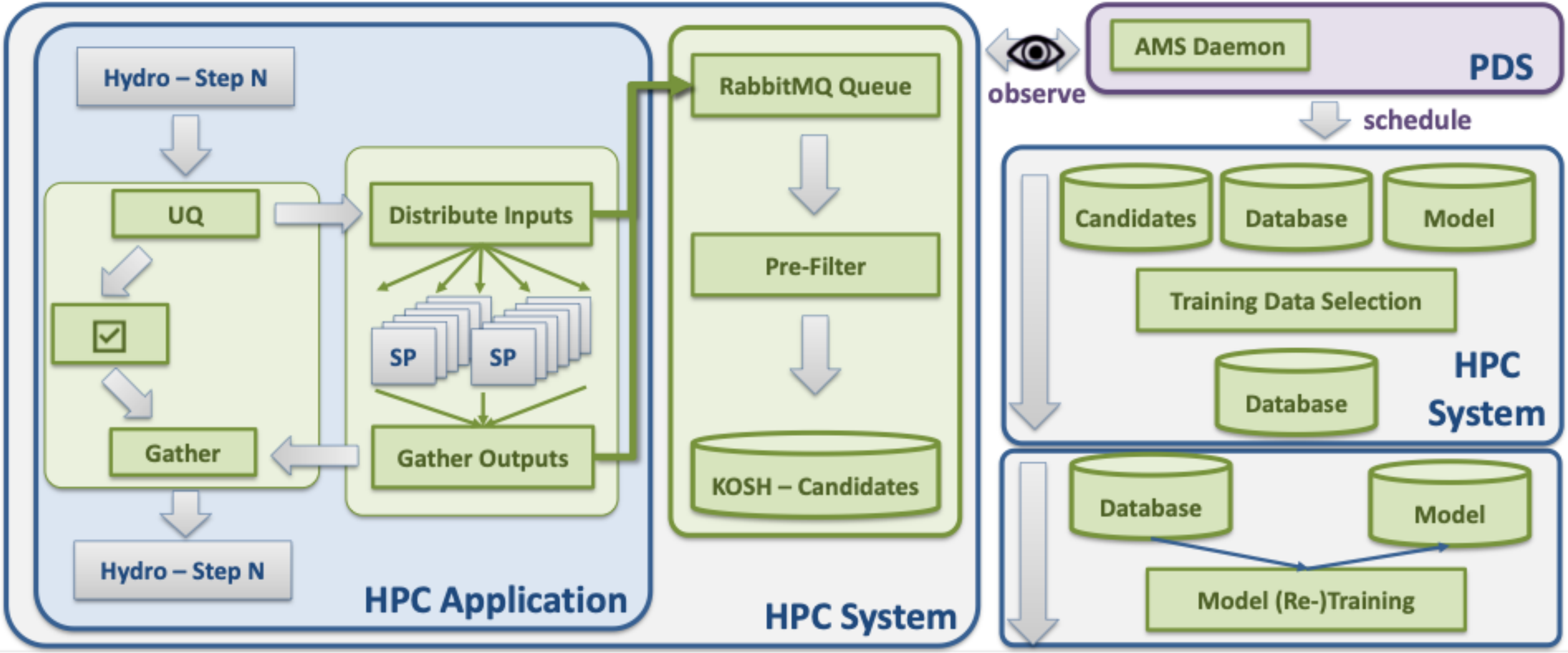
adapt tasks to different schedulers/managers



MuMMI: SC'19 best paper, SC'21 paper

MPI-based simulation with in-situ analysis plus AI/ML

Autonomous MultiScale (AMS) integrates ensemble techniques, ML surrogate models, services, and databases.



State-of-the-art, composite workflows run across clusters and integrate the cloud

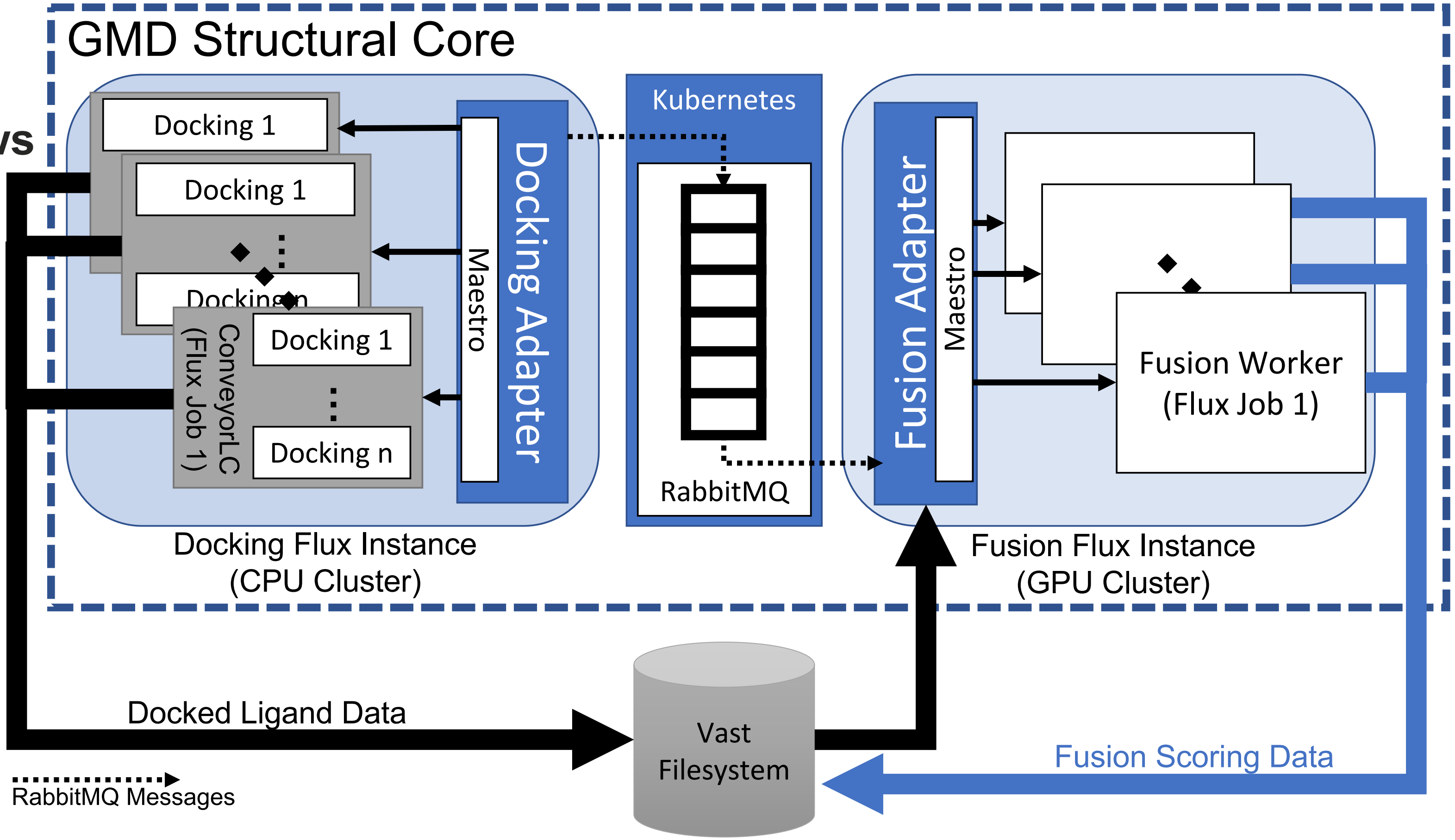
Single-cluster, composite scientific workflows like MuMMI challenge current computing environments

Multi-cluster, cloud enabled workflows emerging at LLNL and beyond

- Coupled MPI-based tasks and deep-learning models (**AHA MoleS**)
- HPC simulation with AI/ML surrogates (**AMS**)

2020 RADIUSS survey found 73% of LLNL workflows interested in cloud, <10% use it

Challenges: resource management, scheduling converged environment



The what and why of movement to the cloud; it's eating everyone's lunch.

The cloud is an environment (public, private, both) supporting:

- Portability, reproducibility
- Resiliency, efficiency, elasticity
- Reduced complexity via automation

Companies rent this environment; hugely profitable

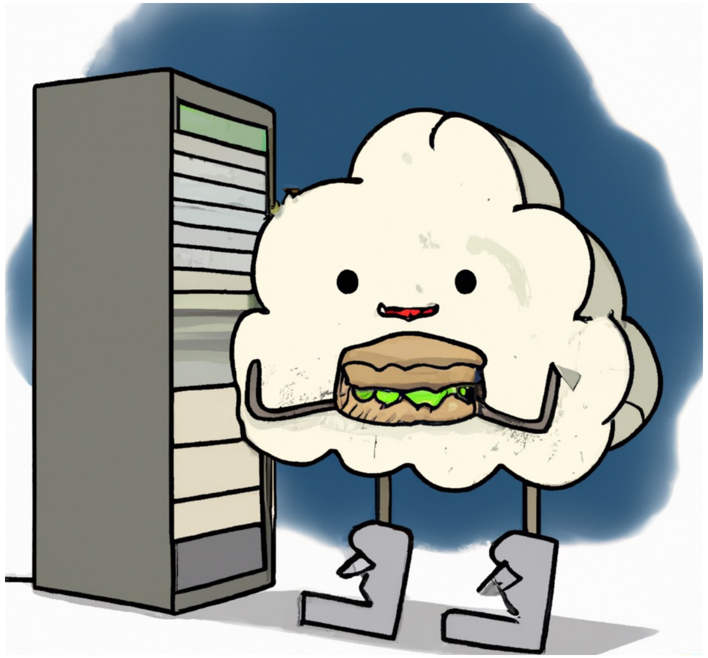
- projected **>$1.1T by 2027**, 20% CAGR (22-27)[1]
  - exceed traditional computing in 2025
- vs HPC: $40B by 2026, 21-26 CAGR 6%[2]

Thompson, Spanuth, 2021[3]: Computing areas become more distinct, provide fewer benefits to others. **Areas that get left behind:**

- **See little performance benefit**
- **Market too small to justify NRE costs**
- **Cannot coordinate demand**

Reed, Gannon, Dongarra, 2023[4]

- **Few HPC vendors build large systems**
- **HPC endothermic, cloud is exothermic**
- **$500M system/5 yrs little incentive for hyperscalers**



[1]Gartner 2023, [2]Hyperion 2023, [3]*The Decline of Computers as a General Purpose Technology*, CACM March 2021 [4]HPC Forecast: Cloudy and Uncertain, CACM February 2023

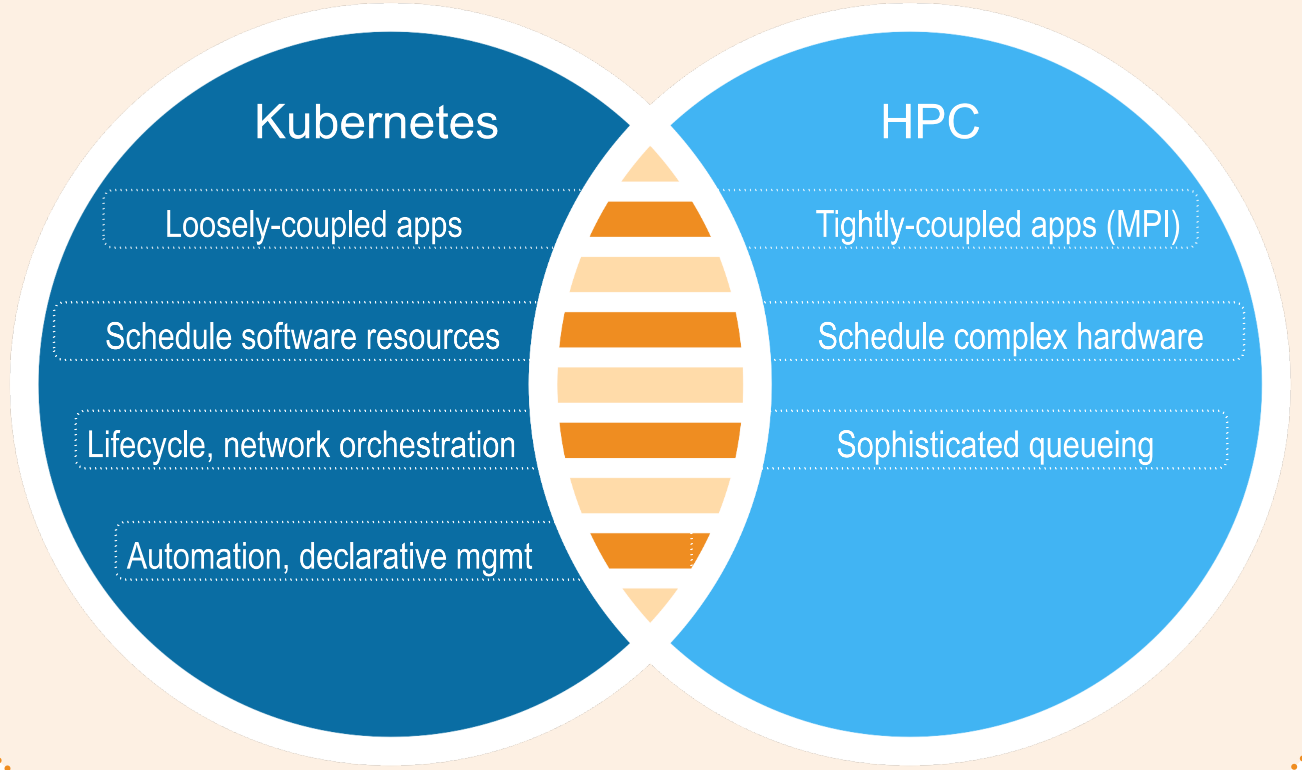
A key to best-of-both-worlds converged computing is combining HPC with Kubernetes.

Kubernetes (K8s):

- cloud “OS” with 77K contributors (second largest OSS project ever)
- designed for loosely coupled apps
- not focused on performance (scheduling limitations, throughput…)

Converged Computing project:

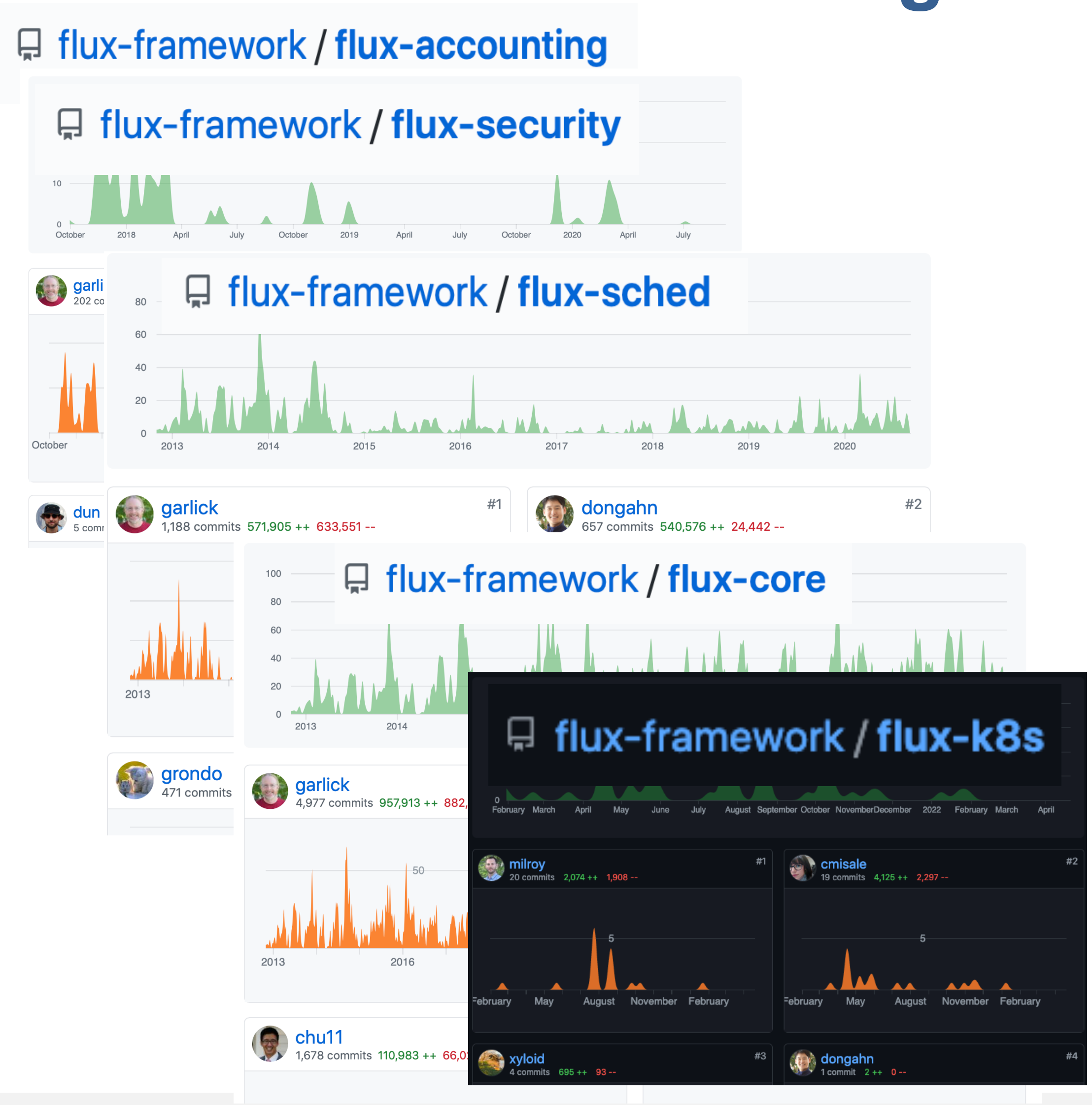
how to create a converged environment composed of the best of both worlds?



HPC:

- performance is in the name
- very difficult to manage modern workflows
- not designed for dynamism/elasticity

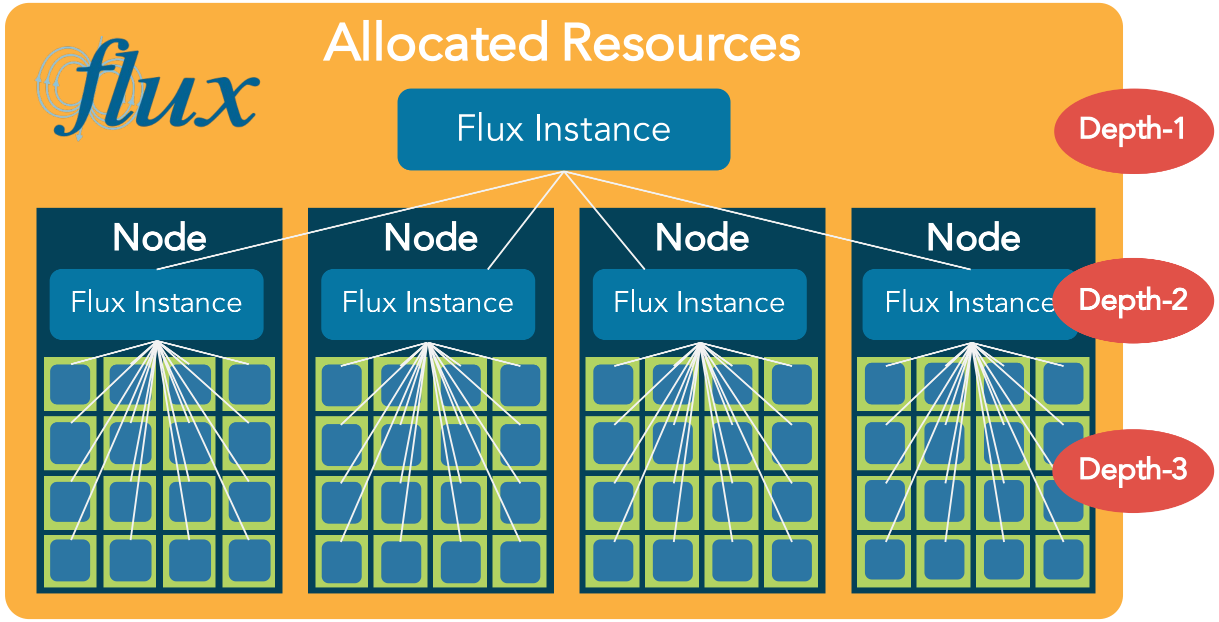
Flux addresses key technical problems that emerge from exascale and converged computing.

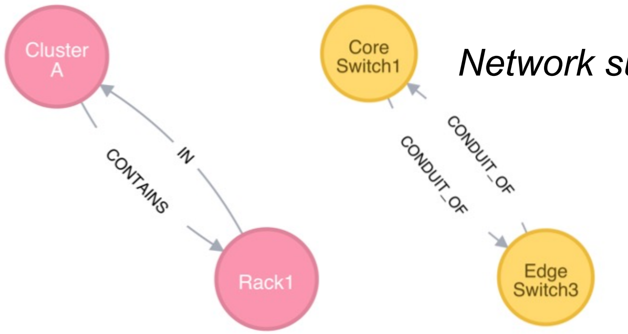


- Open-source project in active development at flux-framework GitHub organization
  - Multiple projects: flux-core, -sched, -security, -accounting, -k8s, -operator etc.
  - Over 15 contributors, including some principal engineers behind Slurm
- Single-user and System instance modes
  - Single-user mode in production for about 5 years
  - System instance on several LLNL Linux clusters
- Deploying on LLNL El Capitan exascale system

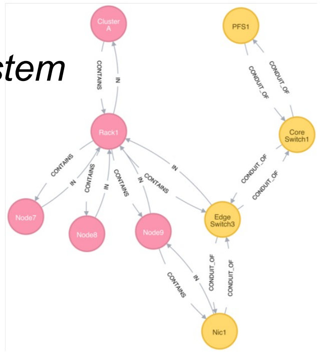
2021 R&D 100 WINNER

Flux hierarchical management and graph-based scheduling address converged computing challenges.





Network subsystem



Containment subsystem

Modular, hierarchical design

- Hierarchical resource management and scheduling (separate modules)
- Sub-manager with specialized scheduler
- Schedules cloud resources

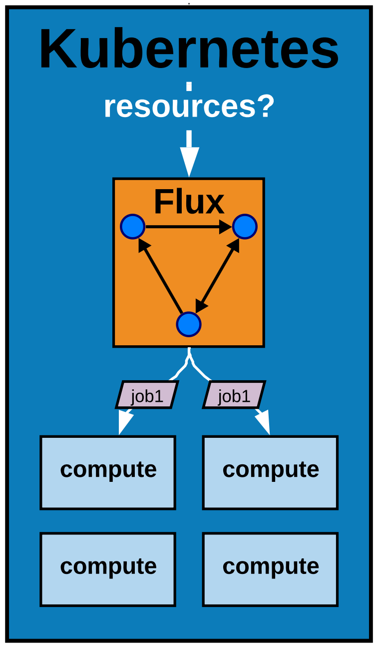
Manages resources nearly anywhere

- Bare metal resources, virtual machines in the cloud, HPC resources in another workload manager, pods in Kubernetes
- Workflows only need to program to Flux
- Directed graph resource model expresses complex, dynamic resources

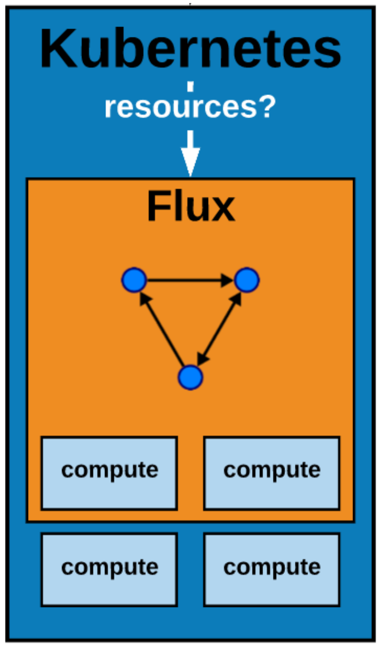
Rich, well-defined interfaces

- Facilitate communications and coordination among tasks within a workflow
- CLI, Python, C, C++, Rust, Go, etc.

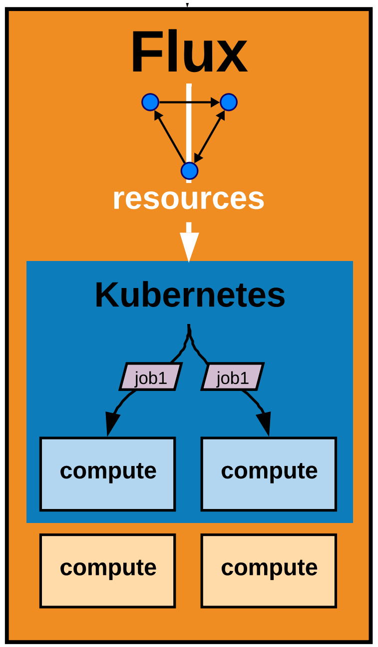
The converged computing project advances convergence with representative, Flux-based models.



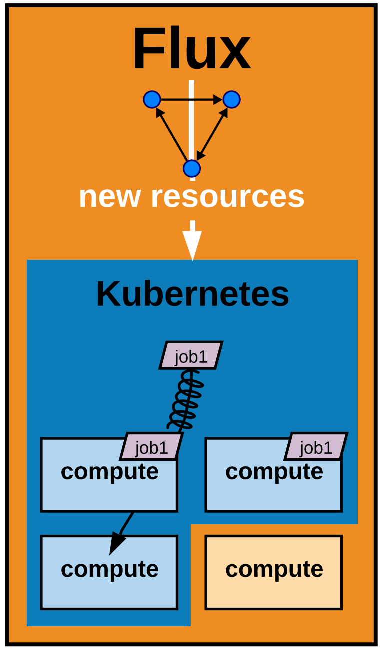
L1: Fluence project enables portable converged workflows



L2: Flux Operator. Enables portable converged workflows



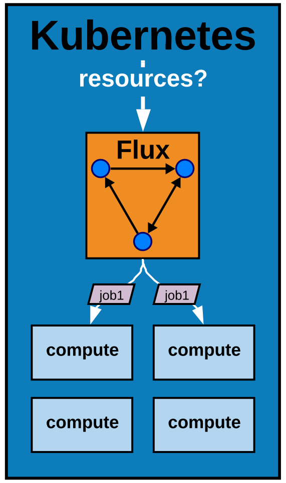
L3: Flux + Usernetes reduces software complexity, increases automation, performance

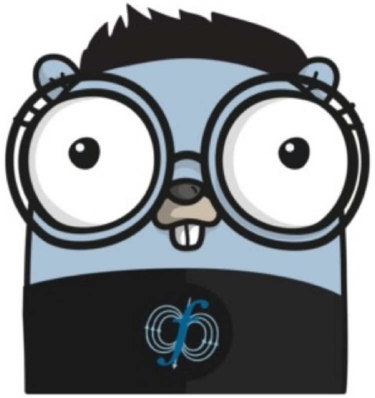


L4: Elastic cloud nesting model enables autoscaling and dynamism

Portability is a component of performance and means of cost control

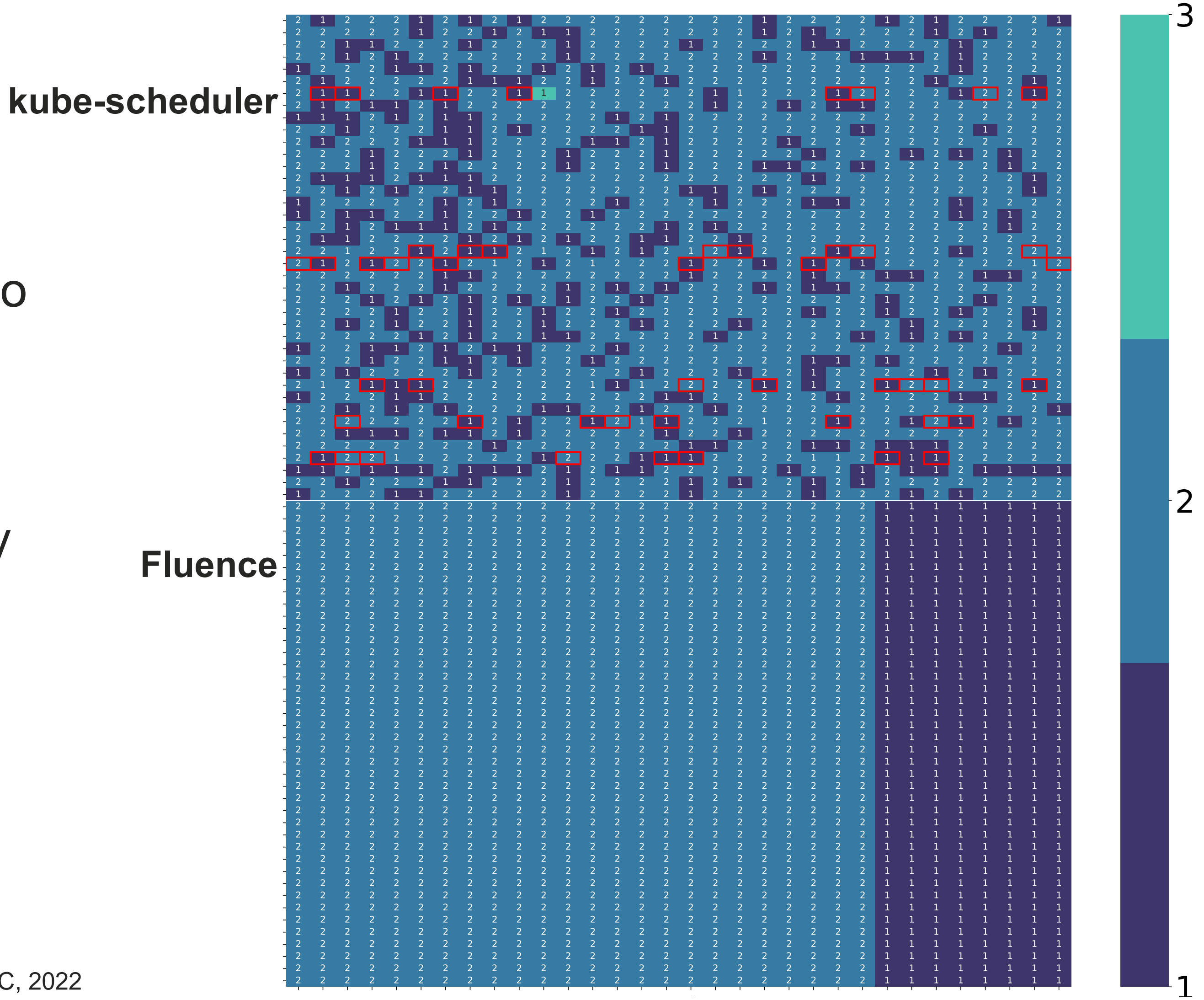
The scheduler is a big contributor to converged workload performance variability.





Fluence

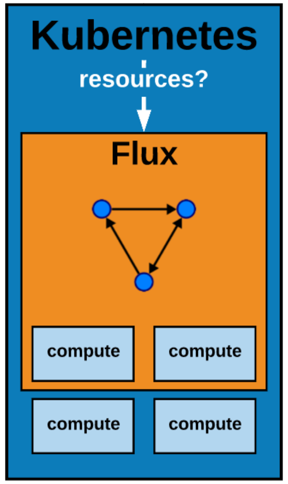
- Plug Fluxion into K8s Scheduling Framework
- Enable cloud-native MPI to scale **three orders of magnitude** higher
- Compare CORAL-2 benchmarks scheduled by Fluence, kube-scheduler: **up to 3x** shorter workflow runtimes, much less variability, deterministic placement[1]



https://github.com/flux-framework/flux-k8s

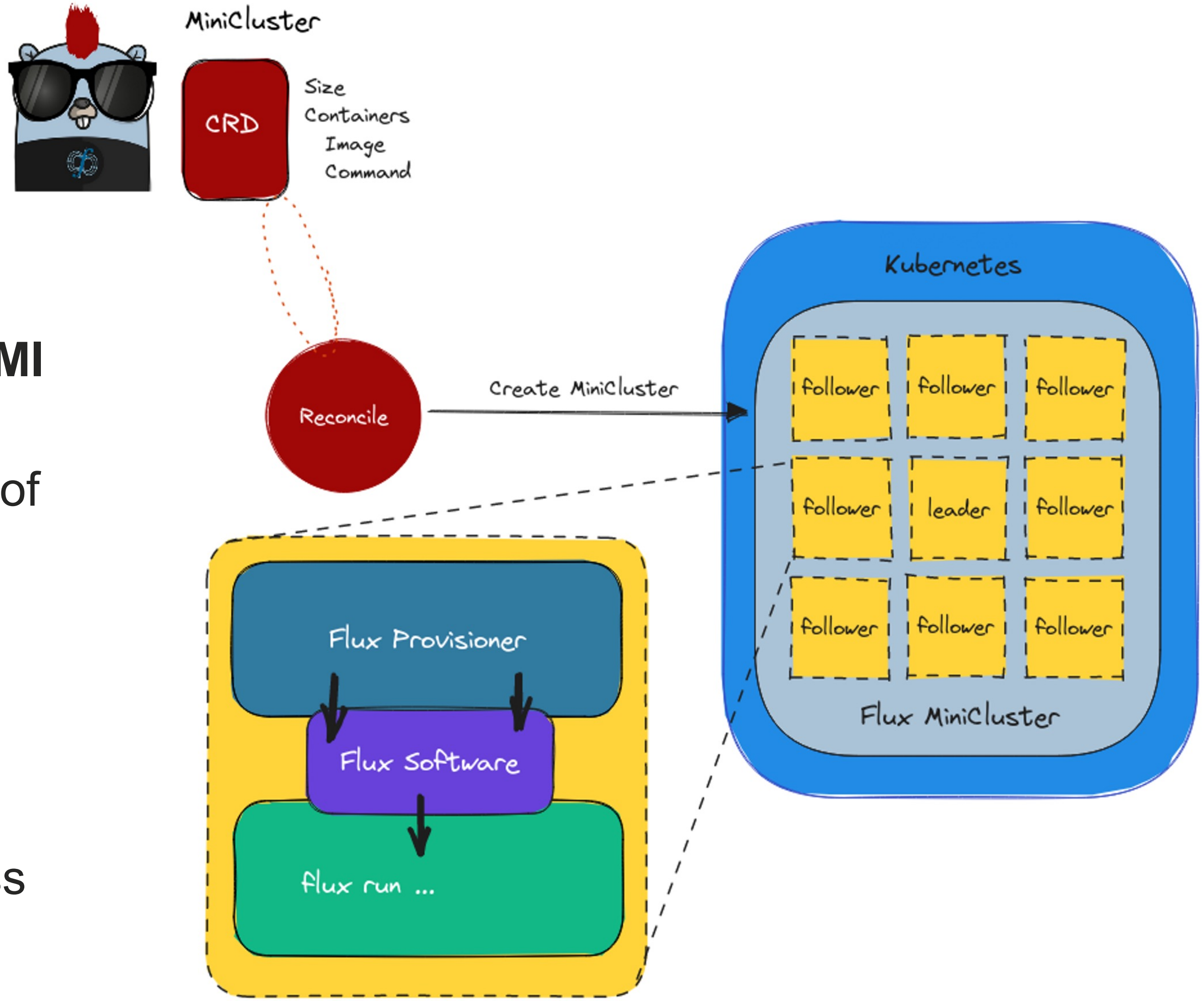
[1]One Step Closer to Converged Computing: Achieving Scalability with Cloud-Native HPC, 2022

We are improving converged workflow portability and efficiency.



Flux Operator[1]

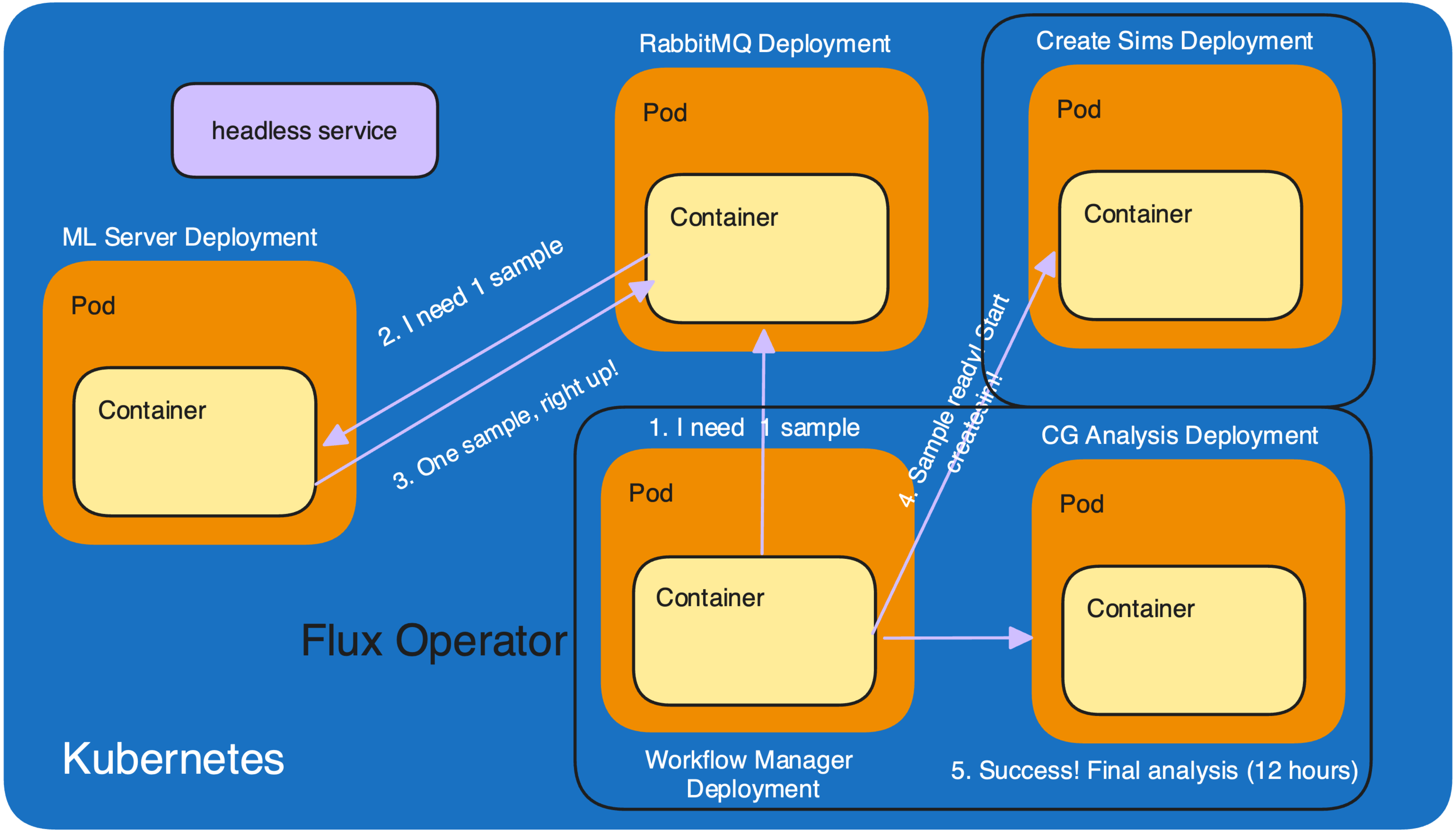
- Bootstrap Flux in K8s; hierarchically manage, schedule pods
- Ported 15 proxy apps; **MuMMI in progress**
- Addresses major limitations of K8s: throughput, multiple users, fairness, allocation usage
- Scalable MPI bootstrap
- Autoscaling in progress
- RESTful interface in progress



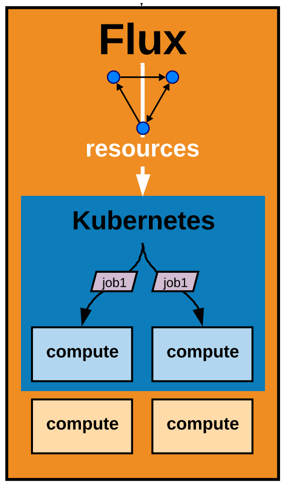
https://flux-framework.org/flux-operator/

[1]The Flux Operator, F1000Research, in press 2024

We are porting updated MuMMI to Kubernetes.

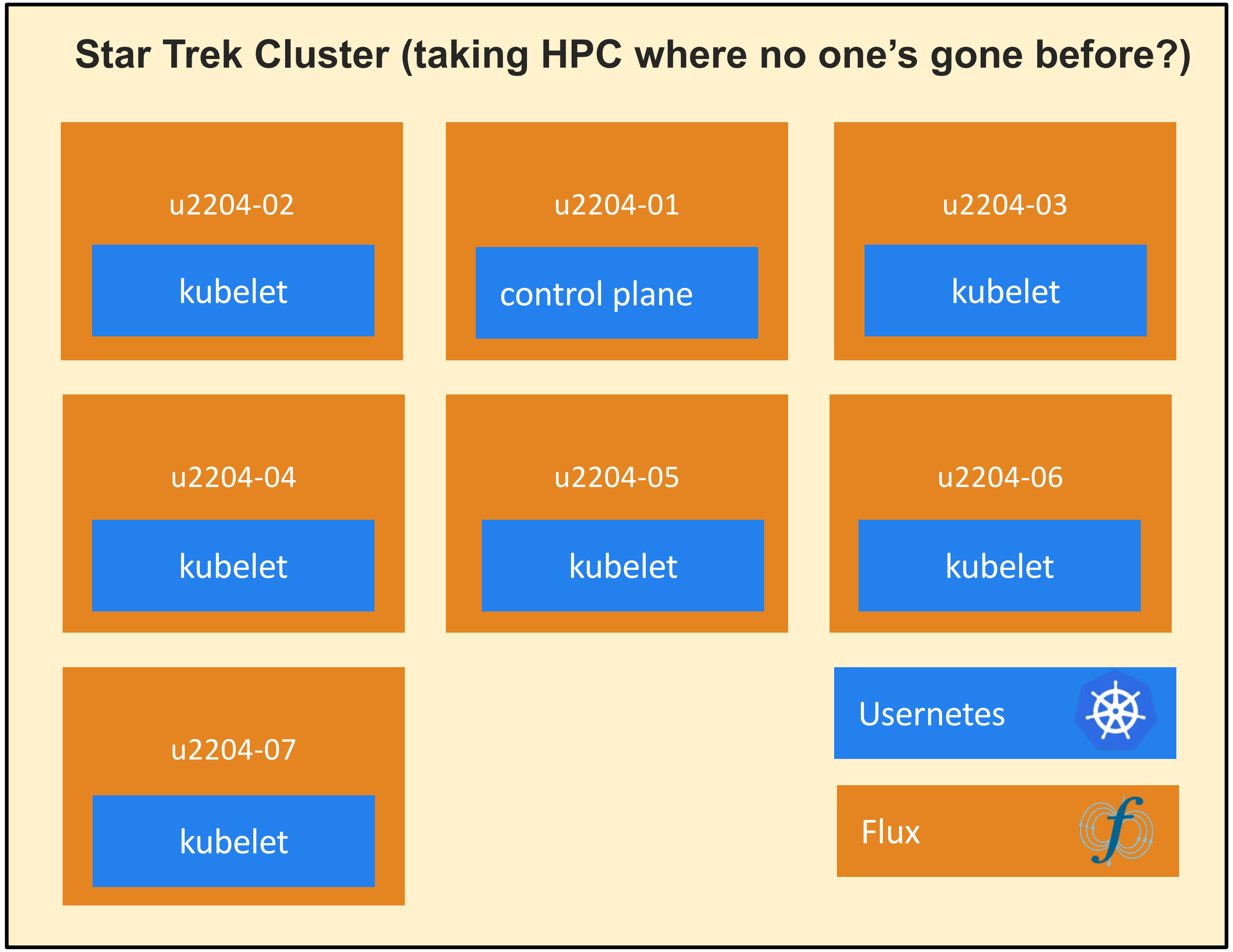


Testing Flux + Usernetes enables cloud on HPC and further enhances portability.



Flux + Usernetes

- Bootstrap K8s in Flux
- K8s is restricted to user scope
- Accounts for usage/fairshare
- Solves resource conflict for multiple managers/schedulers
- No measured performance impact (e.g., noisy neighbor)
- Networking bottleneck: order of magnitude lower bandwidth
- Tested simulation + ML model



Adapted from V. Sochat, Kubernetes and HPC: Bare-metal bros, FOSDEM '24: https://fosdem.org/2024/schedule/event/fosdem-2024-2590-kubernetes-and-hpc-bare-metal-bros/
https://github.com/converged-computing/flux-usernetes

We are collaborating with industry to tackle converged computing, contribute to community, and advocate for HPC.

IBM T.J. Watson Research Center:

- Expertise in K8s Scheduling

Red Hat:

- Developers of OpenShift K8s platform

AWS:

- Largest cloud platform, deep HPC expertise

Starting collaboration with Google!

Starting collaboration with Microsoft!

Meet the converged computing team!

- Daniel Milroy | scheduling, cloud systems
- Giorgis Georgakoudis | HPC runtimes, network-aware scheduling
- Aniruddha Marathe | instance scheduling, application performance
- Zeke Morton | software development
- Tapasya Patki | graph-based scheduling
- Abhik Sarkar | co-management challenges, performance profiling
- Vanessa Sochat | containers (a creator of Singularity), runtimes, cloud systems, software
- Jae-Seung Yeom | scheduling cloud resources

Doctoral student:

- Md Rajib Hossen | Flux-Kubernetes co-management performance

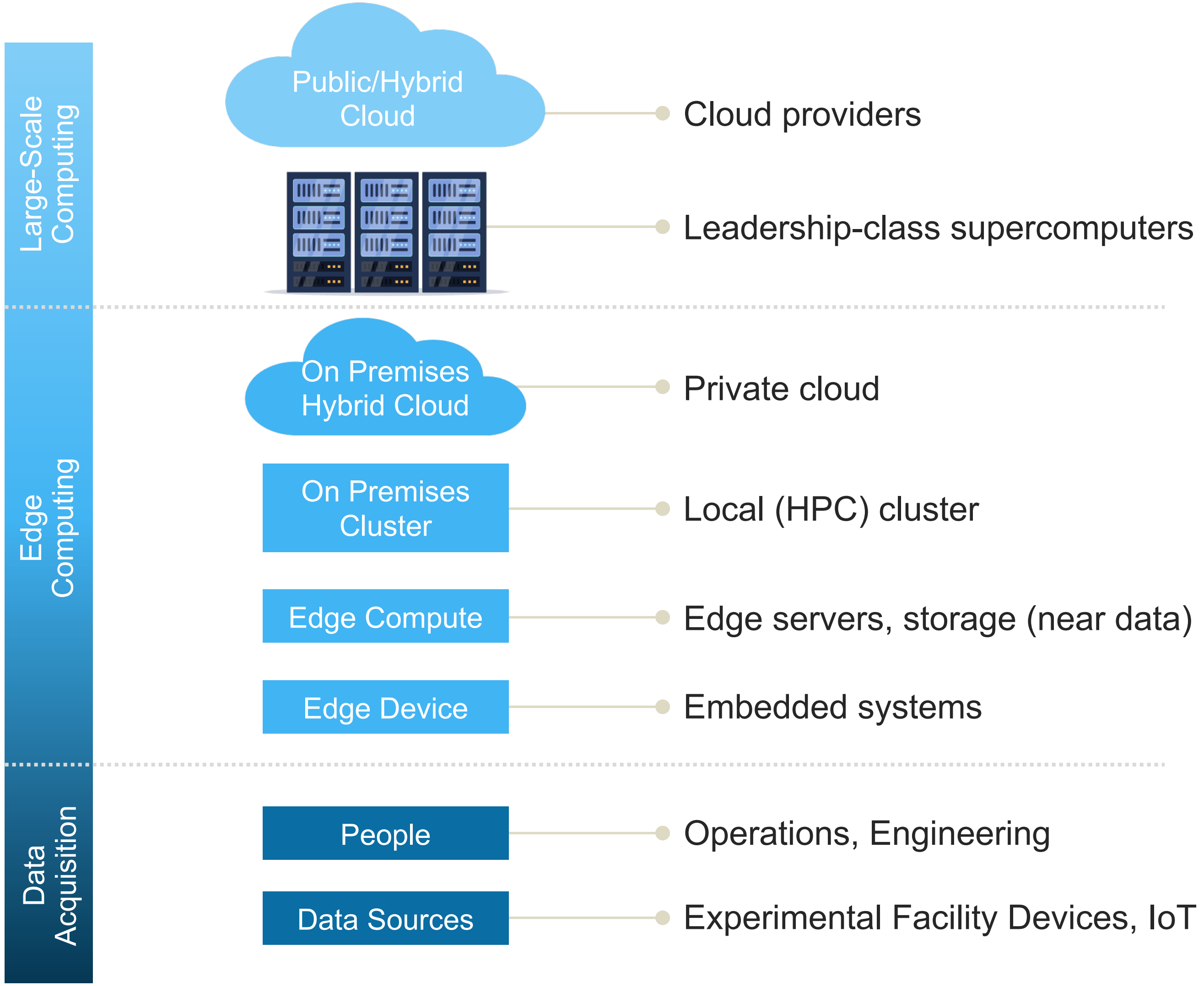
Next-generation workflows will run on a continuum from sensors to converged clusters.

Full computing continuum already used by industry, e.g,. autonomous vehicle control and fleet management

Scientific workflows on the continuum: from sensors on a source beamline to converged compute clusters in another region

Major challenges:

- Resilience of distributed system
- Resource dynamism and elasticity
- Distributed resource management and scheduling



Thank you!
Questions?