Current and Future Converged Cloud-HPC Workflows at LLNL

SC20 State of the Practice Session

Daniel Milroy, Stephen Herbein, Dong H. Ahn

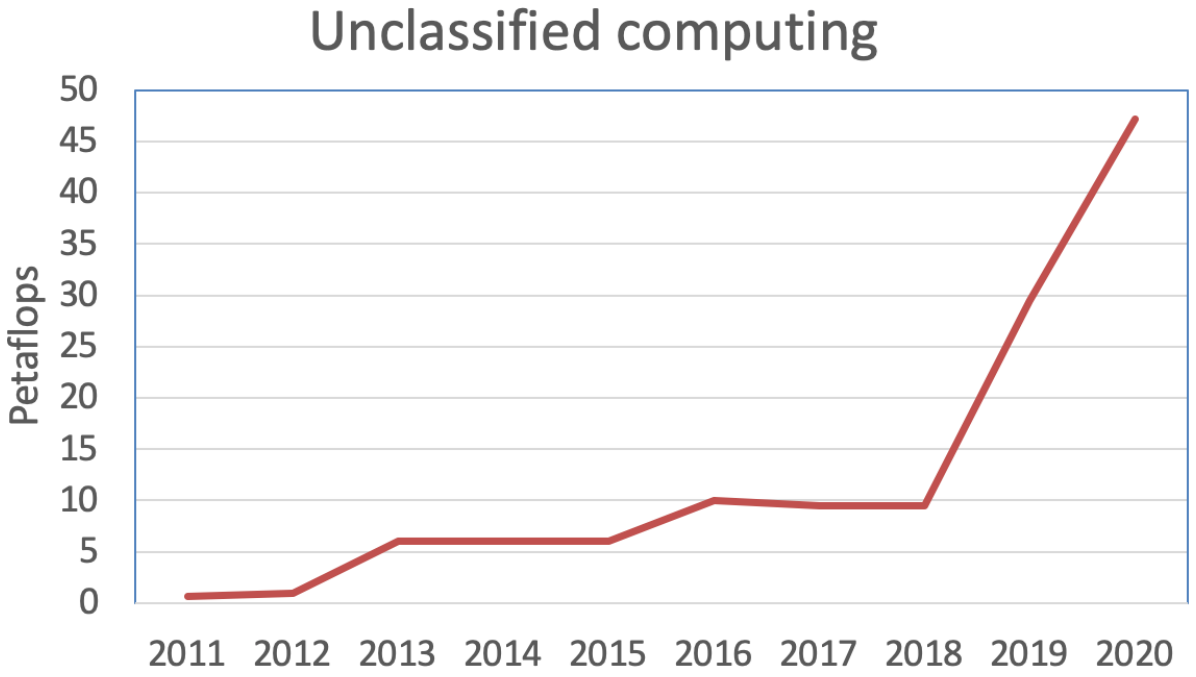
November 17, 2020

LLNL-PRES-815352

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

LLNL continues to be at the forefront of HPC



Sierra: 4320 compute nodes, 125 petaflop/s, #3 on TOP500 (June 2020)

Lassen: 793 compute nodes, 23 petaflop/s, #14 on TOP500 (June 2020)

El Capitan: expected to exceed 2 exaflop/s, deployment in 2023; will integrate cloud technologies

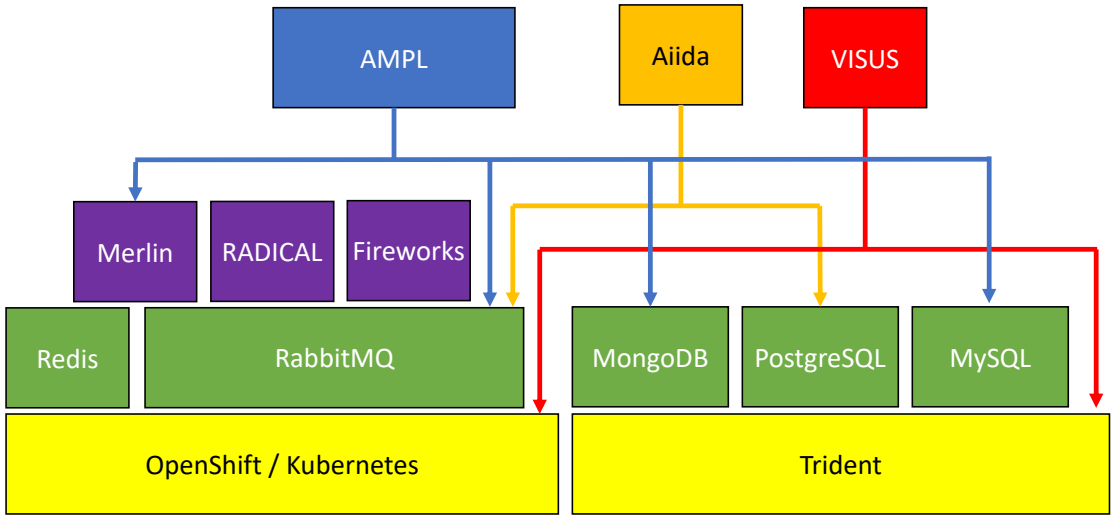
LLNL workflows are increasingly demanding cloud technologies within HPC

- AMPL: ATOM modeling pipeline ML-based COVID-19 drug design workflow
- Traumatic Brain Injury (TBI) research
- OpenViSUS data management and visualization framework
- Merlin workflow coordinator: EPICAST COVID-19 scenario modeling
- Rutgers-LLNL: RADICAL-Pilot for COVID-19 research

LLNL is providing institutional support for convergence efforts

- Workflow Enablement Group (WEG) deploying LLNL funded Persistent Data Services (PDS) platform
- Test applications include AMPL, ViSUS, AiiDA, COVID-19 workflows…
- Containerized DBs and RabbitMQ, Redis, Merlin, Fireworks
- A new project to study converged workflows and gaps
- More to come in the future…

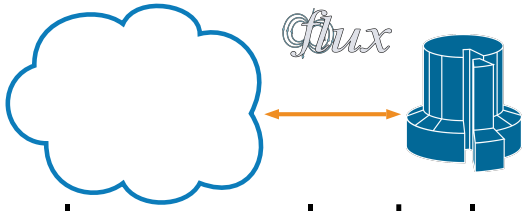
PDS cluster provides software and hardware infrastructure for convergence testing



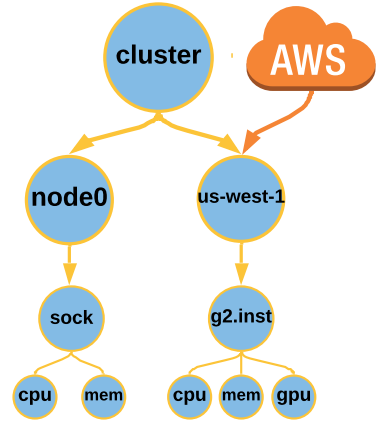
- Redhat OpenShift
- Quay and Clair
- Podman
- NetApp Trident

- TOSS management node, HA proxies for external network
- Three CoreOS K8s servers, five workers
- NetApp AFF A400, StorageGrid object store

Flux Resource and Job Management Software is the keystone of convergence efforts at LLNL



loosely coupled cloud and HPC (in use today)



hybrid/bursting to public cloud

HPC-in-cloud: HPC RJMS in K8s; workflows scale everywhere

cloud-in-HPC: K8s in RJMS

AMPL applied to COVID-19 drug design is an example of converged workflows running at extreme scale

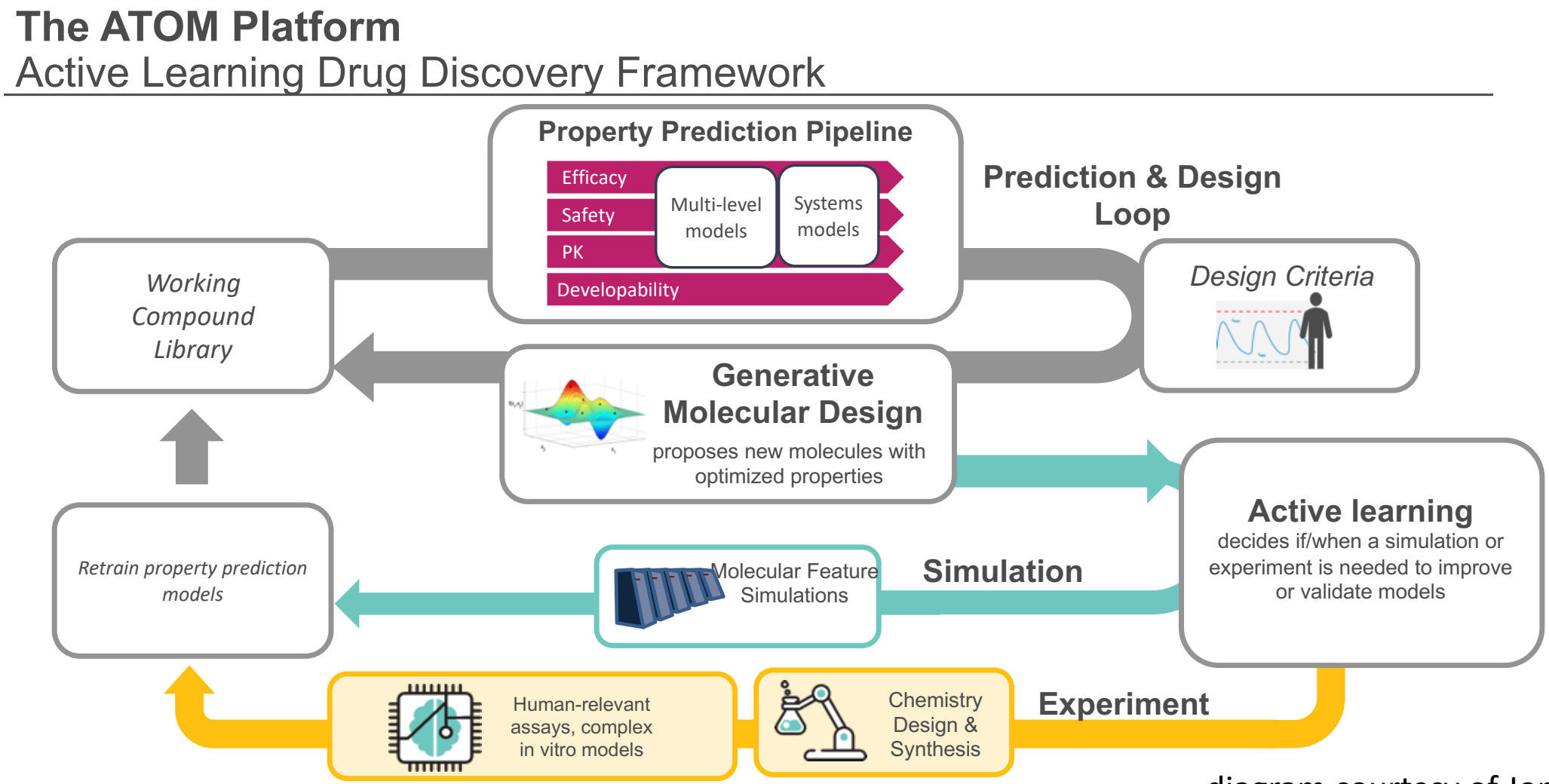


diagram courtesy of Jonathan Allen

AMPL architecture exhibits multiple levels of convergence

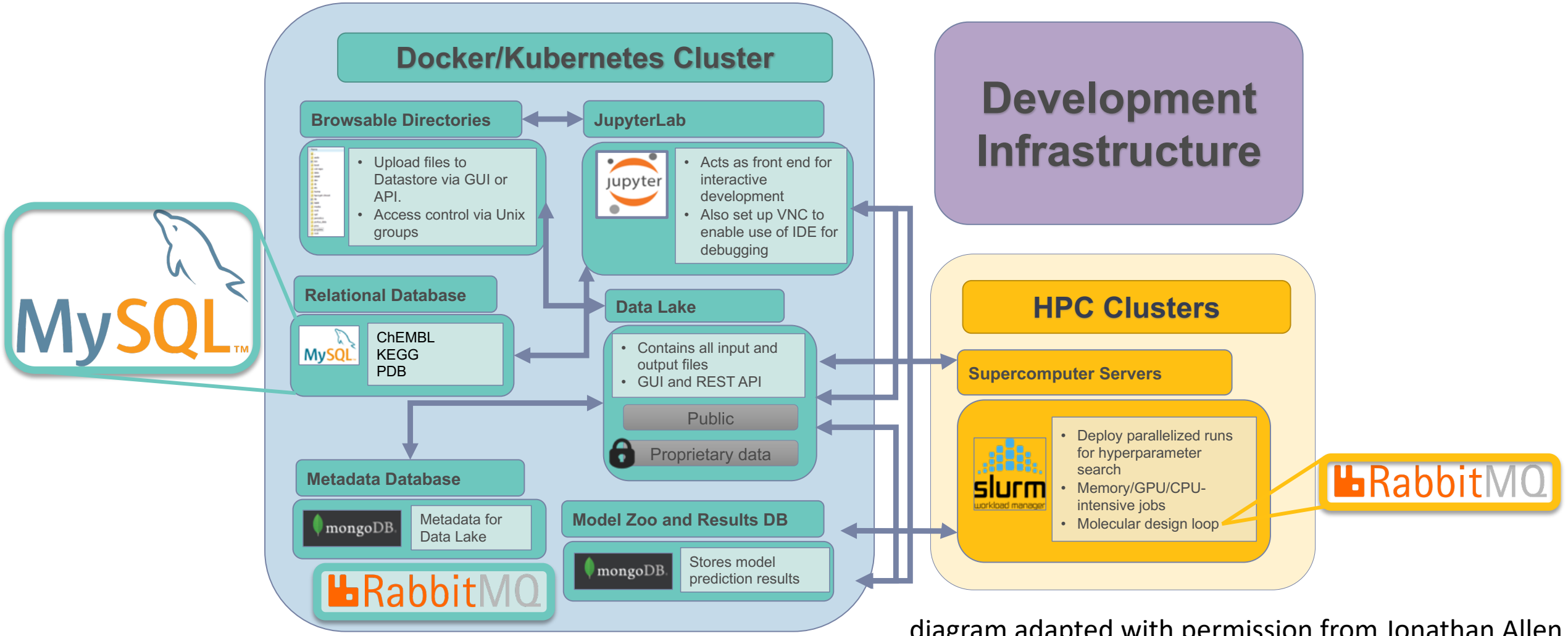


diagram adapted with permission from Jonathan Allen

ACTIV-TBI workflow has simplified and accelerated the connectome analysis pipeline through containerization

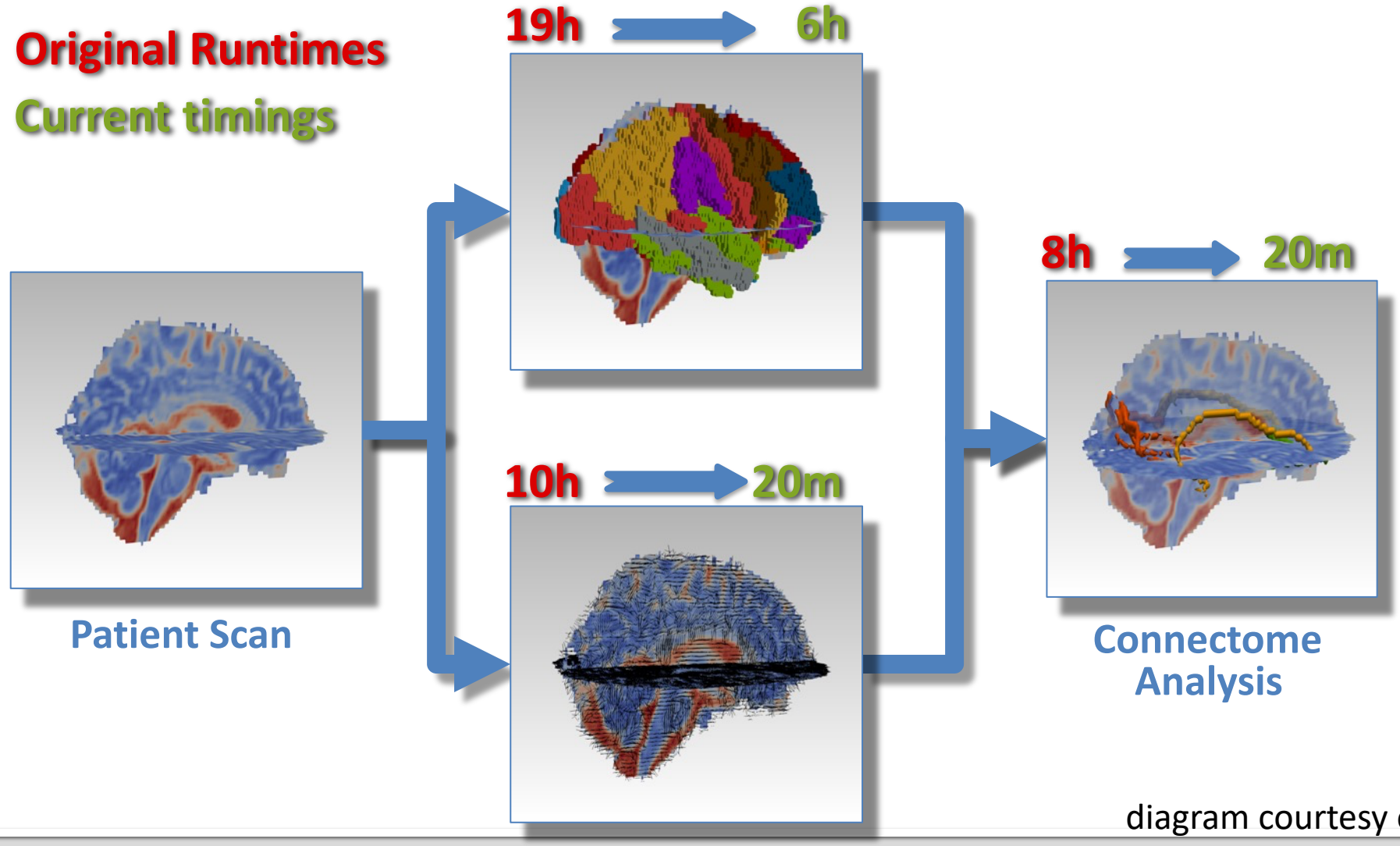


diagram courtesy of Timo Bremer

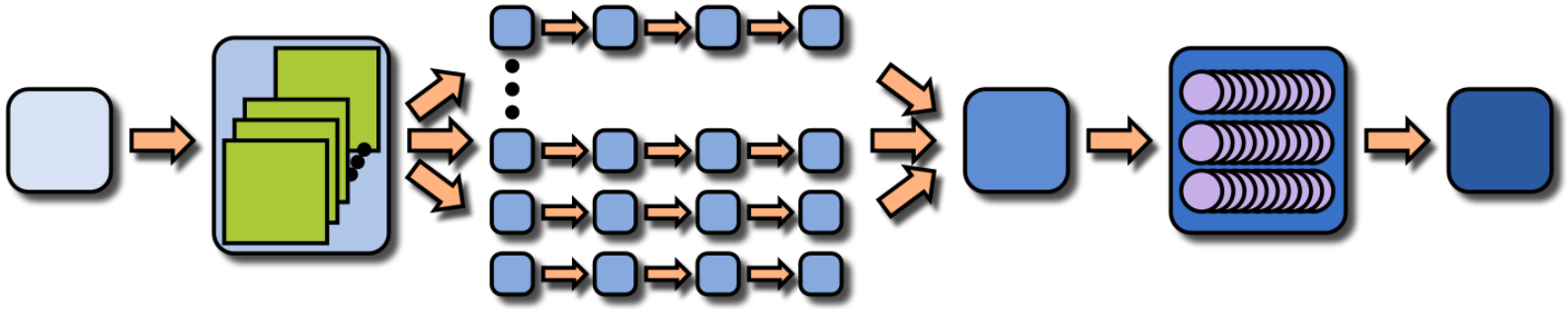
ACTIV-TBI needs containerized pipeline to enable research and deployment

- Easy to operate:
  - Users without extensive computing experience (medical researchers, MRI techs)
- Reliable:
  - Guard against both *hard* and *soft* failures without human in the loop
- Reproducible:
  - Functionally equivalent results at different institutions, hardware, etc.
- Portable:
  - No-dependency, no-hassle deployment at any site or hardware
- Scalable:
  - Ability to optimally exploit all available hardware including splitting workflow across machines and remote sites

slide courtesy of Timo Bremer

Scalability requires standardization and specialization

- Efficient workflow execution requires a standard description

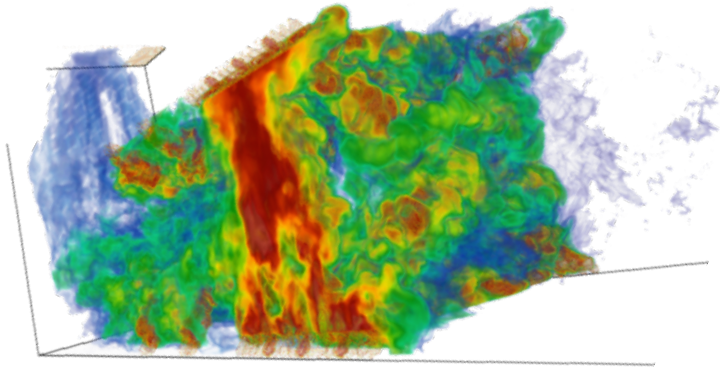


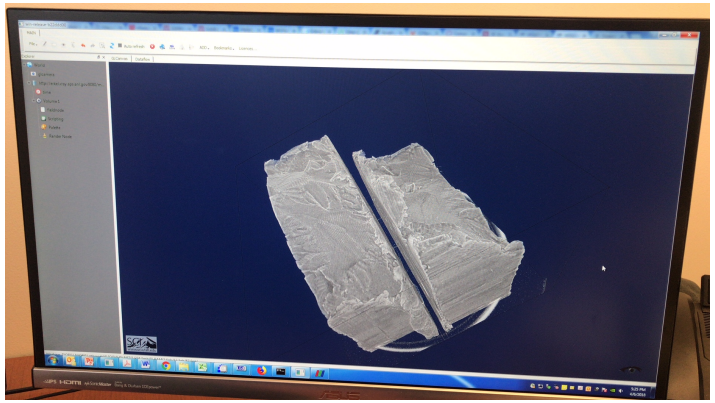
- Parsl – Parallel scripting library (ANL) provides a language to express complex workloads
- Parsl decouples workflow specification from execution environment
- Parsl maps workflows across computing resources, across sites
- Enable task spawning with Flux in the future

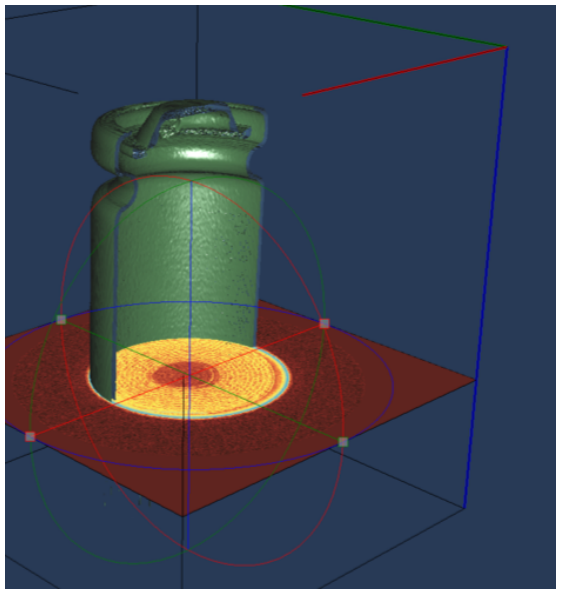
slide adapted with permission from Timo Bremer

ViSUS has a wide variety of requirements and uses

- PCMDI at LLNL for remote visualization of climate data
- ANL live streams data from tomographic and high energy X-ray diffraction microscopy experiments
- Advanced Characterization for LLNL AML for large scale CT and neutron scan visualization
- Lattice Light Sheet Microscope: time dependent imaging of live cells

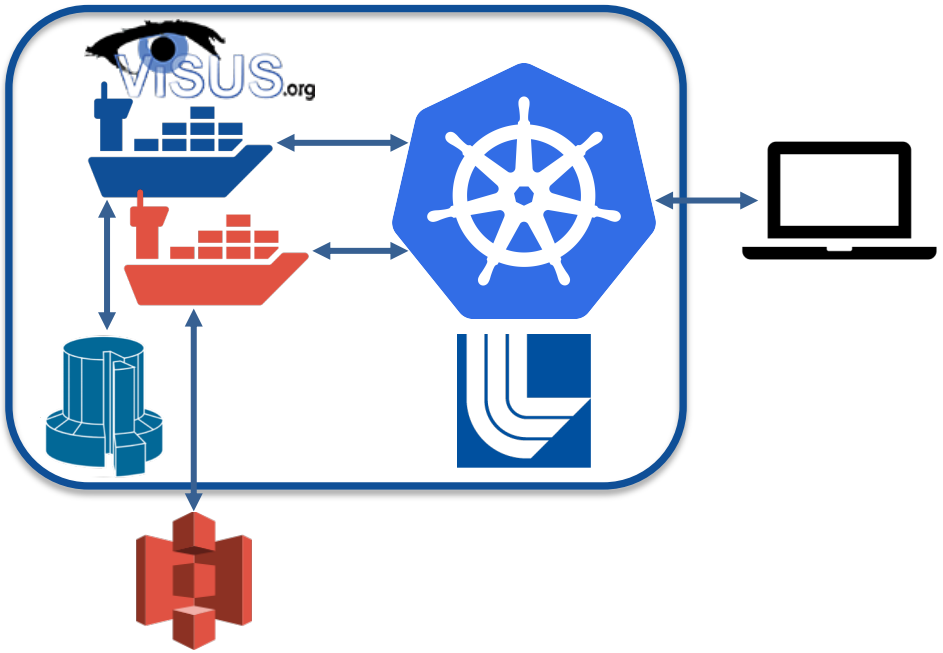






slide adapted with permission from Timo Bremer

ViSUS is built on a scalable, efficient server which enables remote access to federated data archives



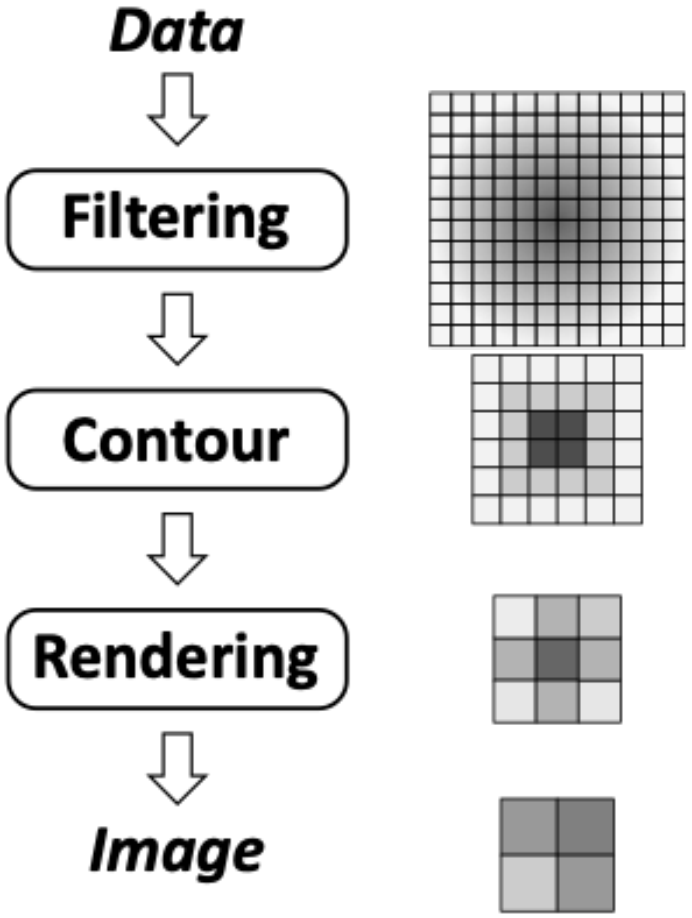
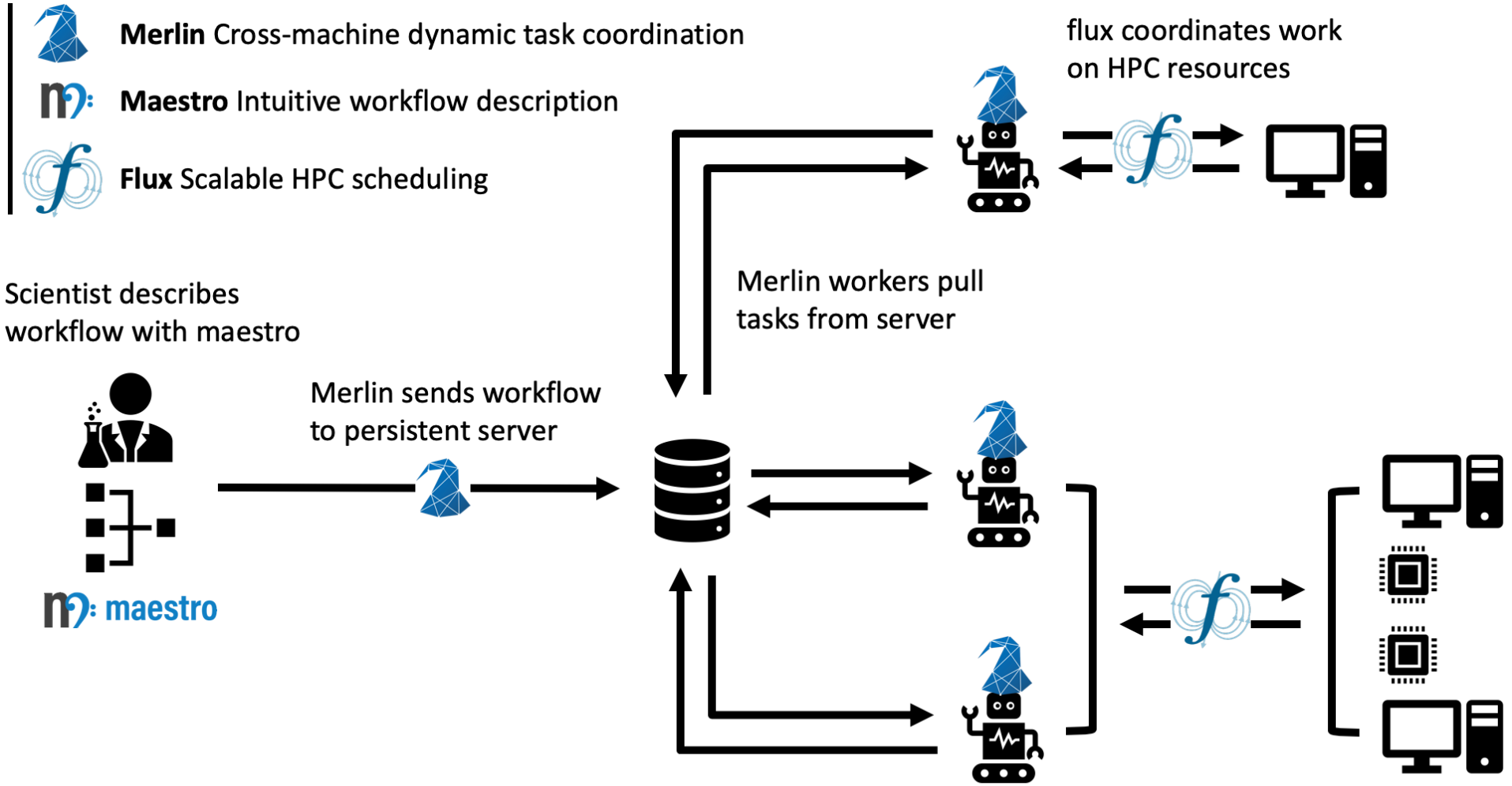


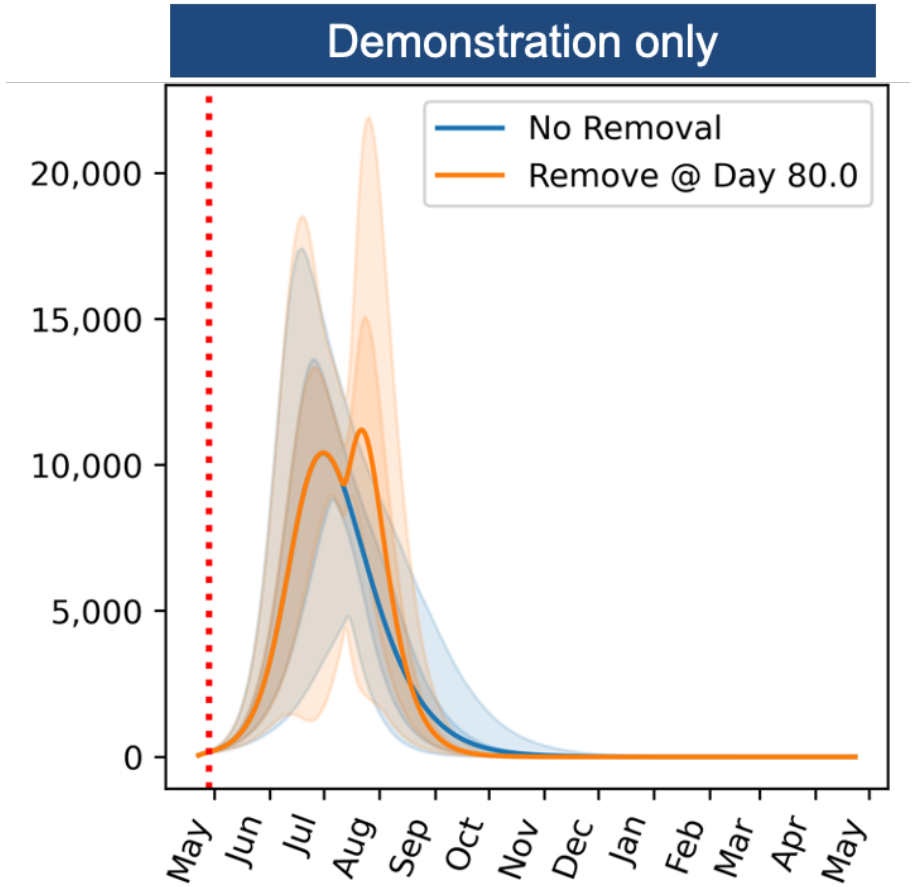
diagram courtesy of Timo Bremer

Merlin workflow coordinator is part of a plug-and-play workflow ecosystem



https://github.com/LLNL/merlin

Flux has been providing Merlin-based workflow aid for COVID-19 scenario modeling workflow



- Collaboration among LLNL, LANL and NERSC
- A large ensemble of EPICAST simulations are employed to model COVID-19 spread patterns
- Inform agencies like FEMA of the prediction
- If school is open only 2 days a week how differently will COVID-19 spread?
- Port Flux to ORNL Summit and NERSC CORI
- *"With flux, we can model one scenario with UQ for the entire country in ~5 minutes on a few [lassen] nodes: near real-time feedback."* – Luc Peterson

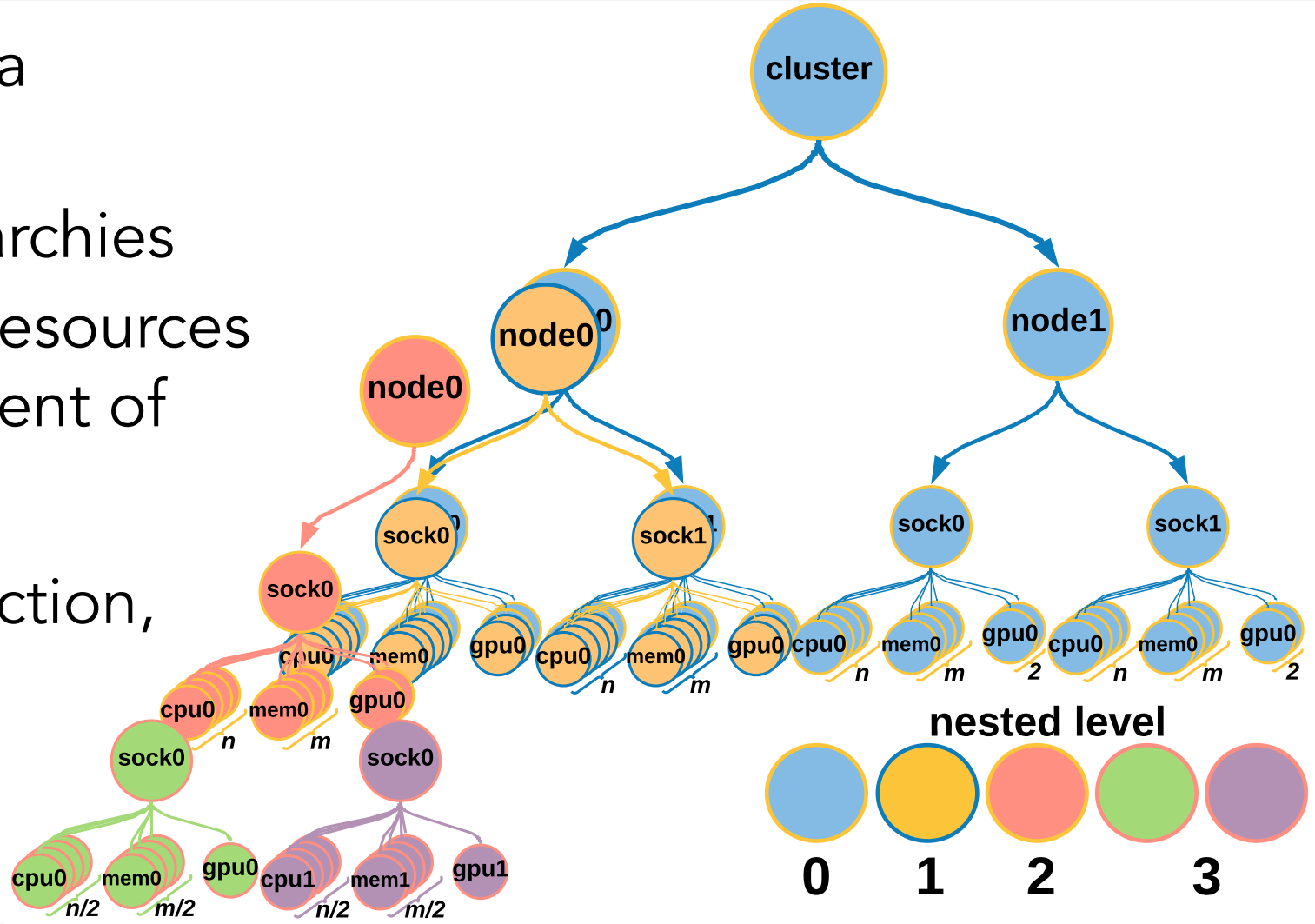
A Demo COVID-19 Scenario Modeling (Credited to Luc Peterson)

Current converged workflows have common challenges

- Autoscaling requires RJMS elasticity
  - Orchestrated apps that scale beyond a dedicated K8s cluster need to be run on HPC
    - How to coordinate authority between K8s and RJMS?
  - What about containerized DBs? Data location, etc.?
  - HPC-in-cloud needs autoscaling, too
- Need a plug-and-play, component-based workflow ecosystem
  - Decoupling workflow specification from environment
  - Maintain separation of concerns
- How do we address security?
  - ViSUS: authentication for federated archives
  - How to prevent CVEs? Quay and Claire?

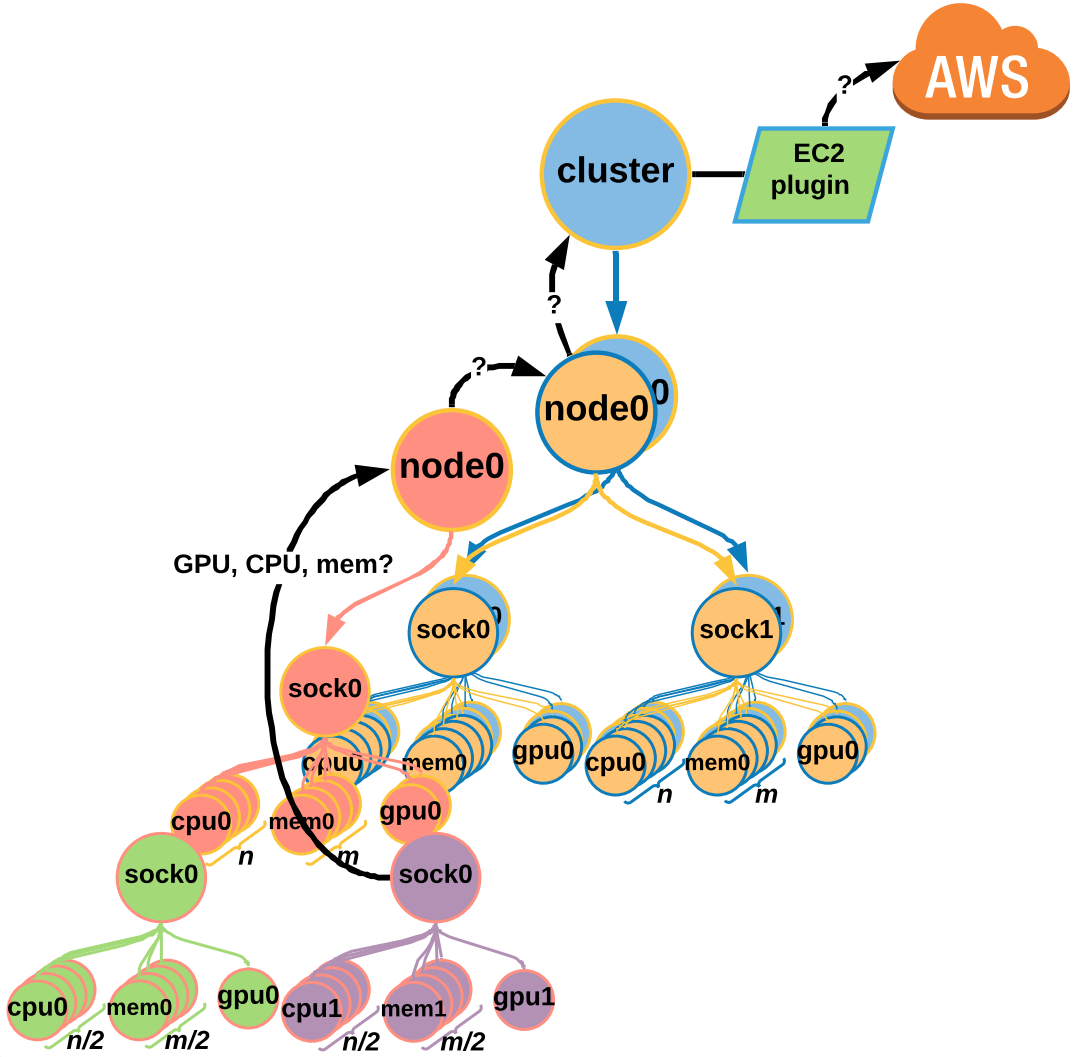
Enabling elasticity in Flux-framework via nested schedulers and directed graphs

- Fluxion resource model is a directed graph
  - —naturally expresses hierarchies
  - —child instance requests resources from parent, notifies parent of relinquished resources
- Mutation is a recursive function, pairwise operation

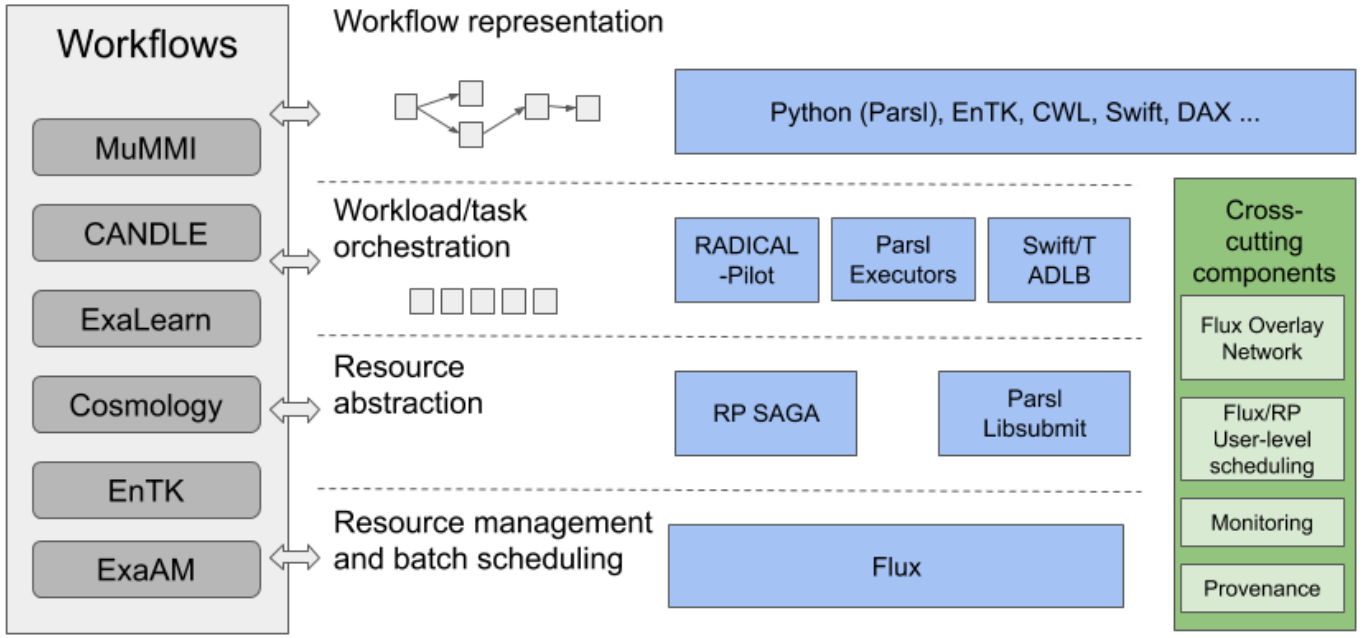


Generalized Multi-Level Scheduling works for bursting: cloud is just another level

- traditional schedulers not designed for dynamic resources
  - hard to add new relationships without scheduler modification/restart
- elastic resources as a graph operation



Flux is a part of ECP ExaWorks to define APIs and methodology to create a plug-and-play component-based workflow ecosystem

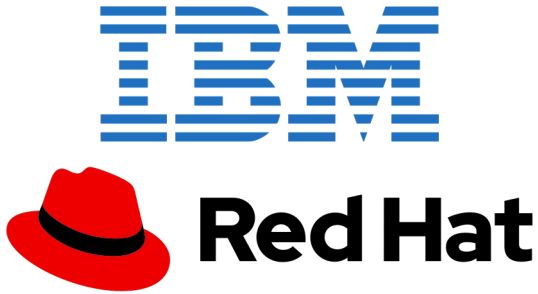


Components of ECP ExaWorks

- Three representative workflow and resource management software systems.
  - Flux (LLNL), Parsl (ANL) and RADICAL-Pilot (BNL)
- Co-design requisite interfaces (vertical and horizontal) while engaging with a wider range of HPC community
- Harden our reference implementation (i.e., ExaWorks toolbox) written on top of these interfaces and deploy it on exascale systems

Building industry collaborations for cloud-in-HPC, HPC-in-cloud, hybrid/bursting

- Explore best ways to express converged resources
- Enable Fluxion to schedule pod binding in OpenShift
- Develop tenancy model for cloud-in-HPC
- Flux in the cloud, bursting to



Cloud and HPC converge at LLNL with the help of Flux

- Cloud-dependent workflows are increasingly common at LLNL
- LLNL is providing institutional support for convergence efforts
- LLNL supports key workflows: AMPL, ACTIV-TBI, Parsl, Merlin, RADICAL-Pilot
- State of the art at LLNL: four convergence paradigms connected by Flux
- Interlab collaborations and industry partnerships will allow LLNL to take leadership role in convergence