KubeCon | CloudNativeCon

Europe 2022

WELCOME TO VALENCIA

KubeCon CloudNativeCon Europe 2022

KubeFlux: an HPC Scheduler Plugin for Kubernetes

Claudia Misale, IBM T.J. Watson Research Center, NY, USA
Daniel Milroy, Lawrence Livermore National Laboratory, CA, USA

Wednesday, May 18, 2022

KubeFlux: an HPC Scheduler Plugin for Kubernetes

KubeCon CloudNativeCon Europe 2022

IBM

Maurizio Drocco, Tonia Elengikal, Yoonho Park

Lawrence Livermore National Laboratory

Dong H. Ahn, Giorgis Georgakoudis, Tapasya Patki, Abhik Sarkar, Jae-Seung Yeom

Red Hat

Carlos Eduardo Arango Gutierrez

Daniel Milroy
Computer Scientist
Lawrence Livermore National Laboratory

Claudia Misale
Research Staff Member
IBM T.J. Watson Research Center

LLNL-PRES-834303

What is a supercomputer?

- High performance: LLNL El Capitan to achieve > 2 exaFLOPs (projected world's fastest in 2023)
- High power requirements: less than 40 MW
- High density, typically liquid cooled
- High utilization: typically 99%
- High bandwidth: 200Gb/s per link
- Low latency: sub-μs switching
- Heterogeneous co-processors

El Capitan will feature integration of cloud technology

Computing achieves High Performance (HPC) through efficiency, proximity, and shape.

KubeCon CloudNativeCon Europe 2022
PromCon North America 2021

- HPC applications strive for high parallel efficiency (scalability)
- Many HPC apps are tightly coupled
  - problem subdivided across processors
  - *neighbor* communication
- A major way to achieve HP is through placement:
  - topology (connectivity; shape): NUMA nodes, L1-3 cache, network
  - distance (link length; speed of light can be limiting factor)

Cloud is a dominant market force influencing HPC.

- Public cloud revenue forecast: $500B by 2022, $600B by 2023 (Gartner)
  - HPC spending 2022: $32B (Hyperion Research, 2020)
- KubeCon + Cloud Native Con is very popular
  - K8s has a huge user, open-source developer community
  - HPC shouldn't become a technology island
- Containers in HPC and CANOPIE (SC) workshops (among many others): increasing interest from HPC community
- *HPC scheduling in K8s is not fully realized*

PromCon
North America 2021

Strategically important LLNL workflows like AMPL exhibit the need for HPC+cloud converged computing.

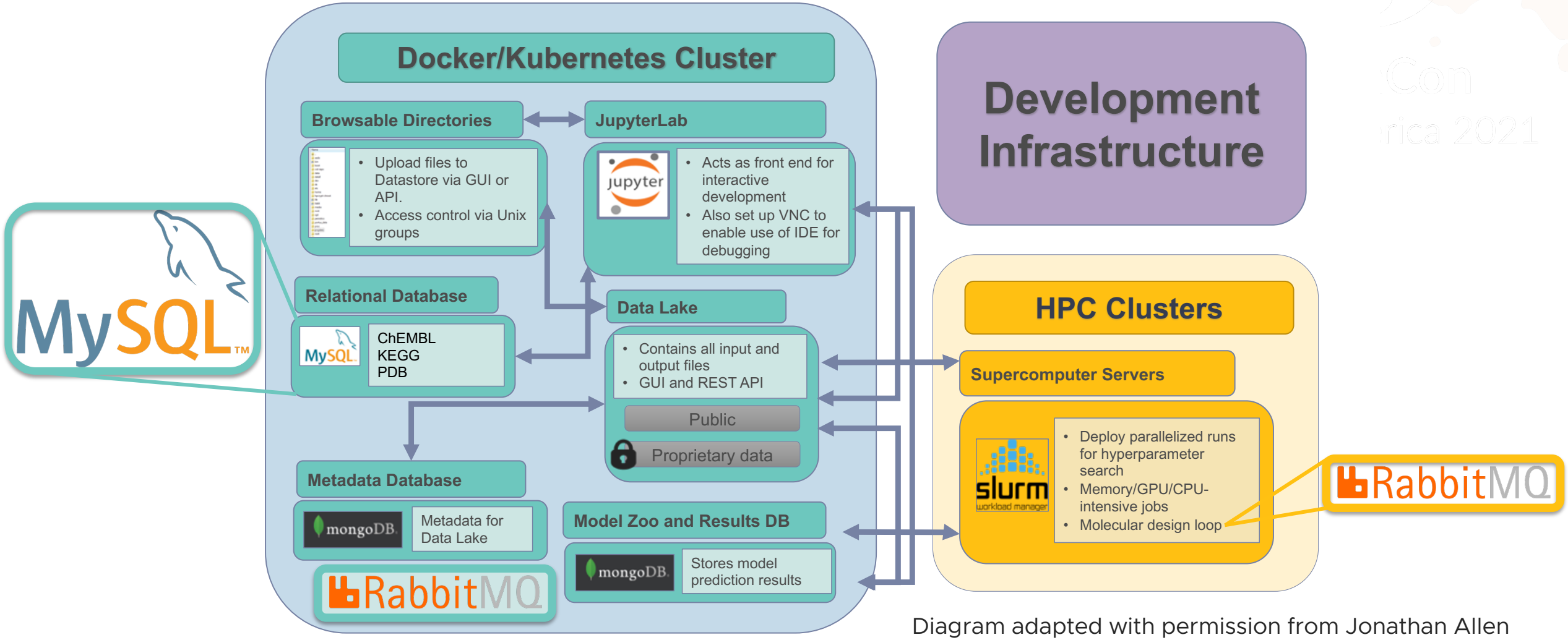


Diagram adapted with permission from Jonathan Allen

Other LLNL workflows are demanding cloud technologies in HPC, and the demand will increase.

KubeCon CloudNativeCon Europe 2022
PromCon North America 2021

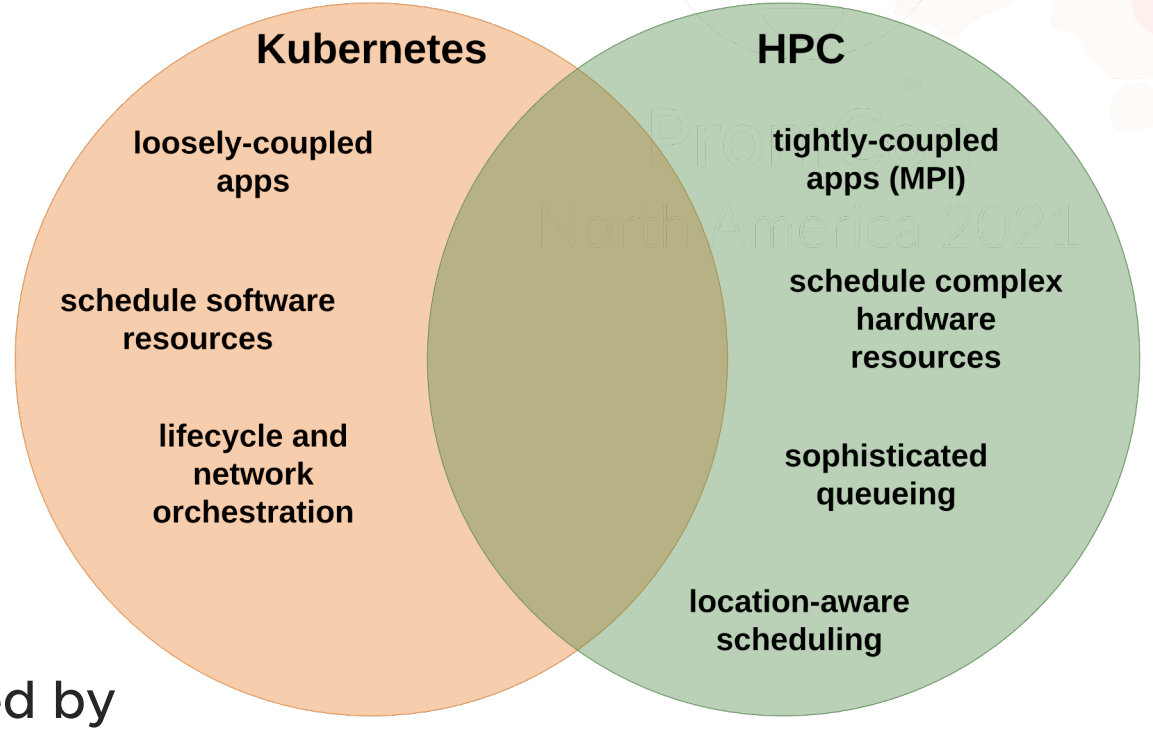
- American Heart Association Molecule Screening (AHA MoleS) workflow
- Rapid COVID-19 small molecule drug design workflow (based on 2020 ACM Gordon Bell Special Prize finalist)
- Autonomous MultiScale project
- Rutgers-LLNL: RADICAL-Pilot for COVID-19 research

The 2020 LLNL HPC Application Survey determined <10% of apps are currently using cloud… but 73% may adopt cloud in the future

HPC schedulers and K8s are complementary technologies

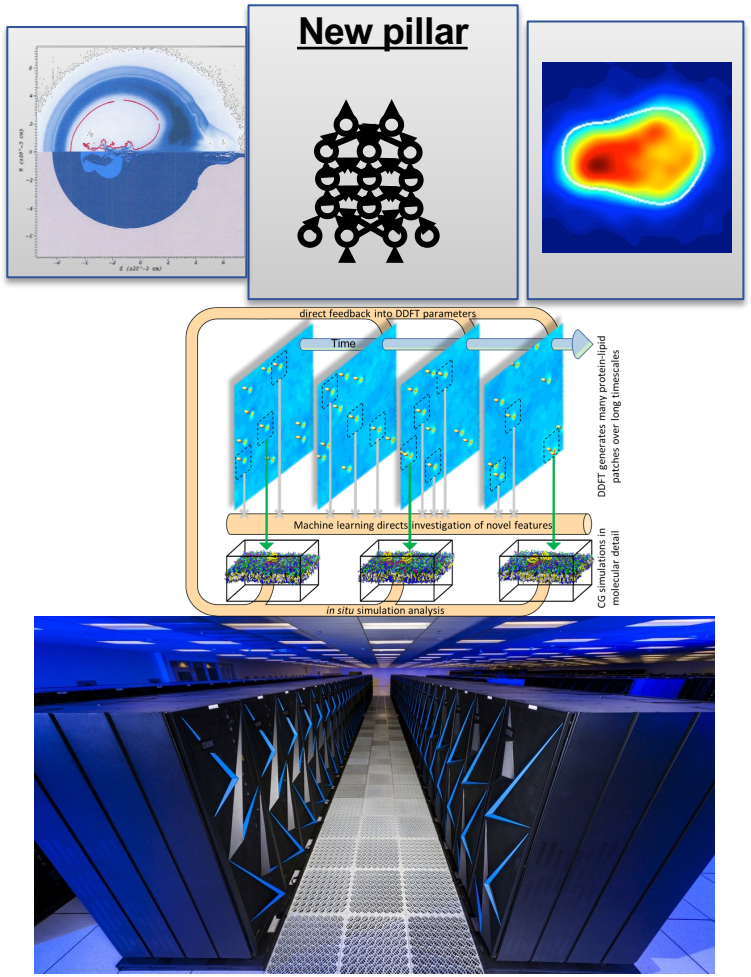
KubeCon CloudNativeCon Europe 2022

- HPC disadvantages:
  - can't orchestrate container lifecycle and networking
  - not designed for elasticity
- K8s disadvantages for converged computing:
  - designed for loosely coupled apps
  - kube-scheduler is limited
  - limited resource expression for HPC, mitigated by Node Feature Discovery (NFD)



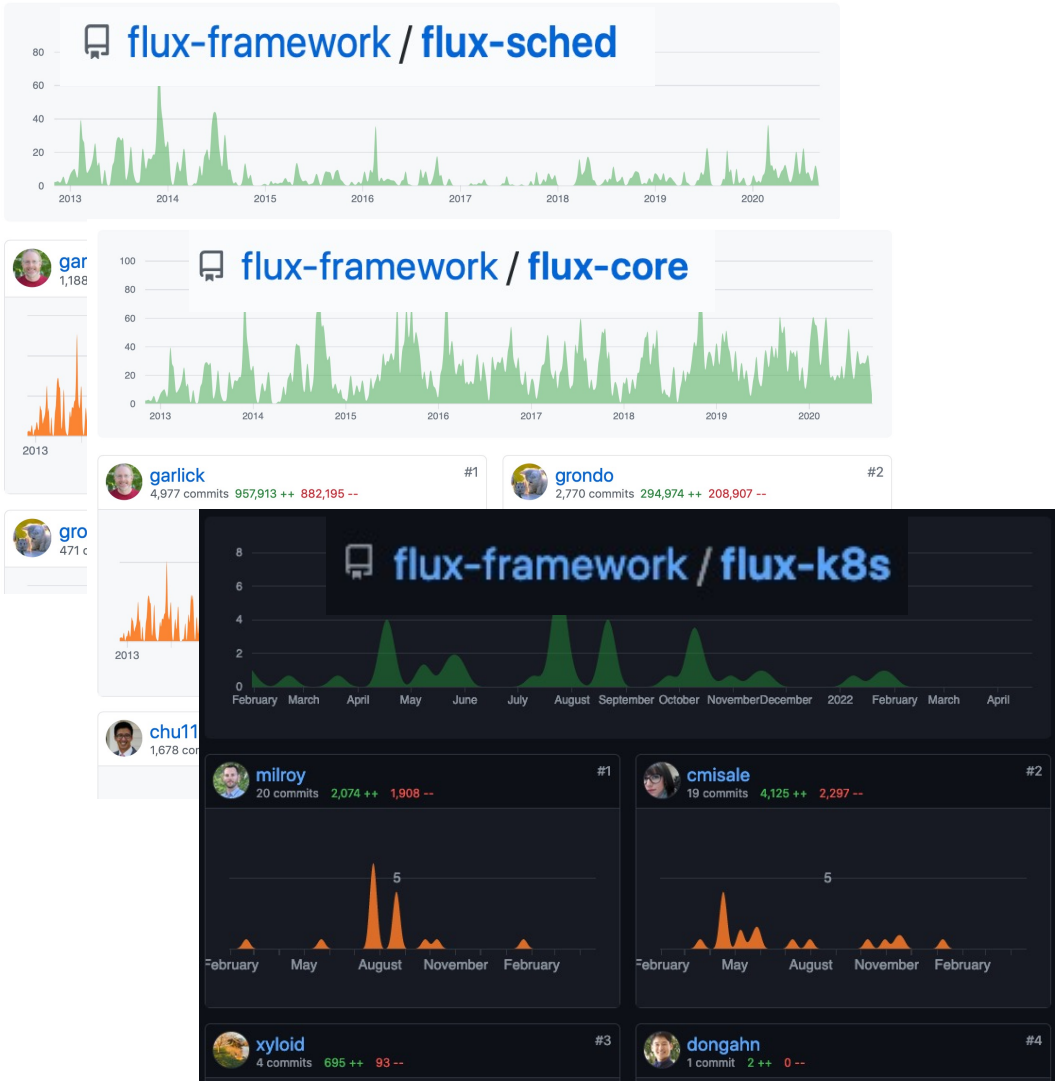
Fully-featured HPC scheduling in K8s has not been achieved.

Trends towards complex workflows, extreme resource heterogeneity, and converged computing render traditional workload managers increasingly ineffective.



- Co-scheduling
- Job throughput
- Job communication/coordination
- Portability
- Extremely heterogenous resources

Flux framework solves the key technical problems that emerge from these trends.



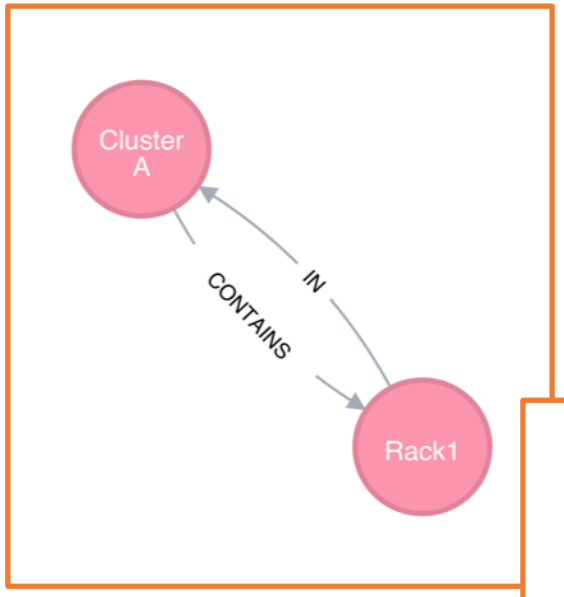
- Open-source, active flux-framework GitHub org.
  - flux-core, -sched, -security, ***-k8s***, etc.
  - over 15 contributors including some of the principal engineers behind SLURM
- Single-user and multi-user modes
  - single-user mode in production for ~4 years
  - multi-user mode debuts on LLNL Linux clusters
- Plan of record for El Capitan

KubeCon CloudNativeCon Europe 2022
PromCon North America 2021

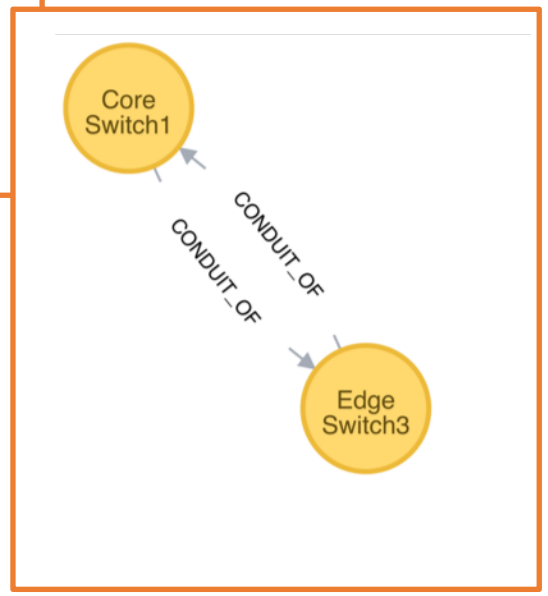
LLNL-PRES-834303

Flux pioneers graph-based scheduling to manage complex combinations of extremely heterogenous resources.

- Traditional resource data models are largely ineffective for resource heterogeneity
  - designed when systems were simpler
  - node-centric models are monogenous
- Elevate resource relationships (***edges***) to an equal footing with resources (***vertices***)
- Complex scheduling can be expressed without changing the scheduler code
- Rich and well-defined API for graph traversal and allocation

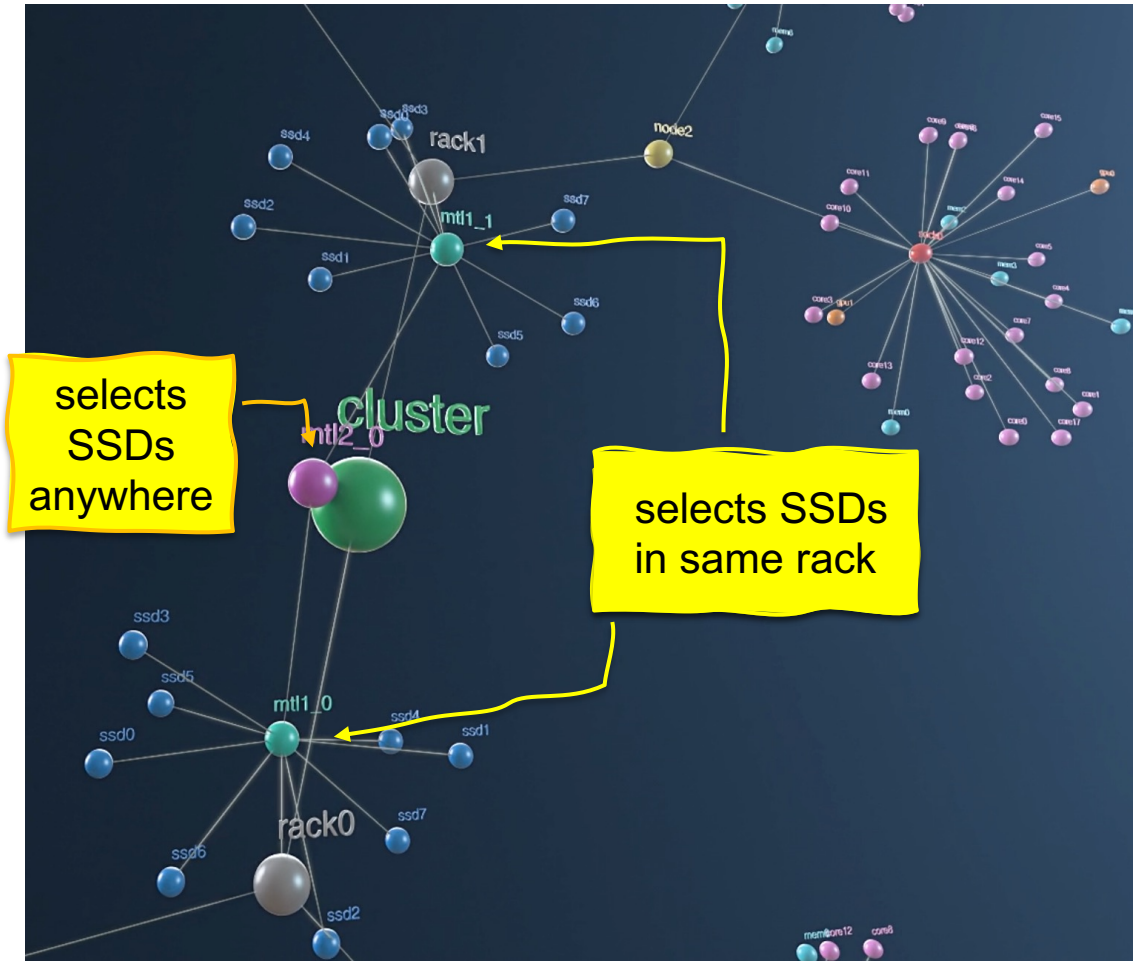


Containment subsystem



Network connectivity subsystem

Our graph approach solves the critical scheduling need for El Capitan's Rabbit multi-tired storage.



- Schedule SSDs in each rack
  - mount as node-local storage to compute in same rack
  - used to build ephemeral Lustre
  - deemed too difficult for traditional schedulers
- Easily enabled by Fluxion; no code change
- End-to-end job state transition
  - DataWarp/Rabbit mock service containers by HPE
  - Kubernetes CRD

KubeCon CloudNativeCon
Europe 2022

Requirements for HPC on K8s, KubeFlux Internals

Claudia Misale

Run HPC on Kubernetes

Requirements:

1. Batch/group scheduling
2. Topology awareness
3. Optimized placement for multiple workloads
4. Choose among placement algorithms

..possibly combine all of the above

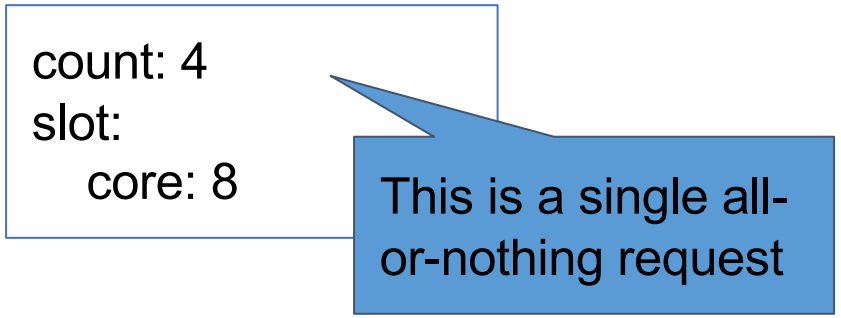
Batch/Group Scheduling

Upstream Kubernetes:
PodGroup CRD and Coscheduling[1] scheduler plugin implement group scheduling

KubeFlux uses PodGroup to create the groups

- Fluxion is a batch scheduler -> group placement done right and for free

Example: MPIJob, 4 worker nodes, 8 cores



[1]https://github.com/kubernetes-sigs/scheduler-plugins/blob/master/pkg/coscheduling/README.md

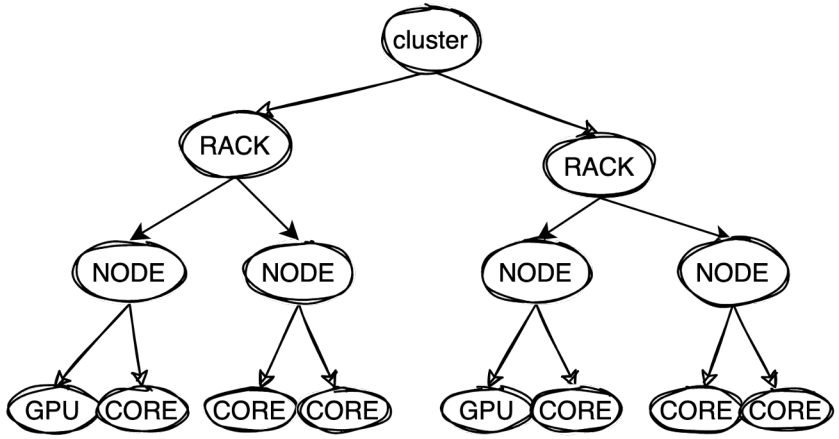
Topology Awareness

Upstream Kubernetes:

- NodeResourceTopology CRD and Topology[1] Aware scheduler plugin (NUMA)
  - Topology at node level
- Use of labels, (anti-)affinity, taint/toleration
  - Not scalable

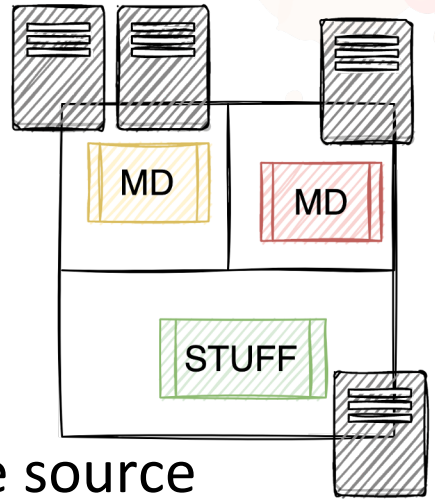
KubeFlux resource model is graph based

- Topology at cluster level
- Job spec converted to graph for matching



[1]https://github.com/kubernetes-sigs/scheduler-plugins/blob/master/pkg/noderesourcetopology/README.md

Optimized Placement for Multiple Workloads



Upstream Kubernetes:

- Use labels, taints/tolerations, (anti-) affinity, not scalable solution
- Trimaran load-aware scheduler plugin
    - Learns from running workloads, does not solve the problem at the source

KubeFlux provides exclusive usage of resources at any level of the topology

- Avoid noisy neighbors
- Oversubscription possible but not preferable for HPC

[1]https://github.com/kubernetes-sigs/scheduler-plugins/blob/master/pkg/trimaran/README.md

Different Placement Algorithms

Upstream Kubernetes:

- Add multiple profiles to the scheduler. Facilitated by Scheduler Framework
- Can force placement with affinity/taint up to a certain level
- Cannot spread with anti-affinity

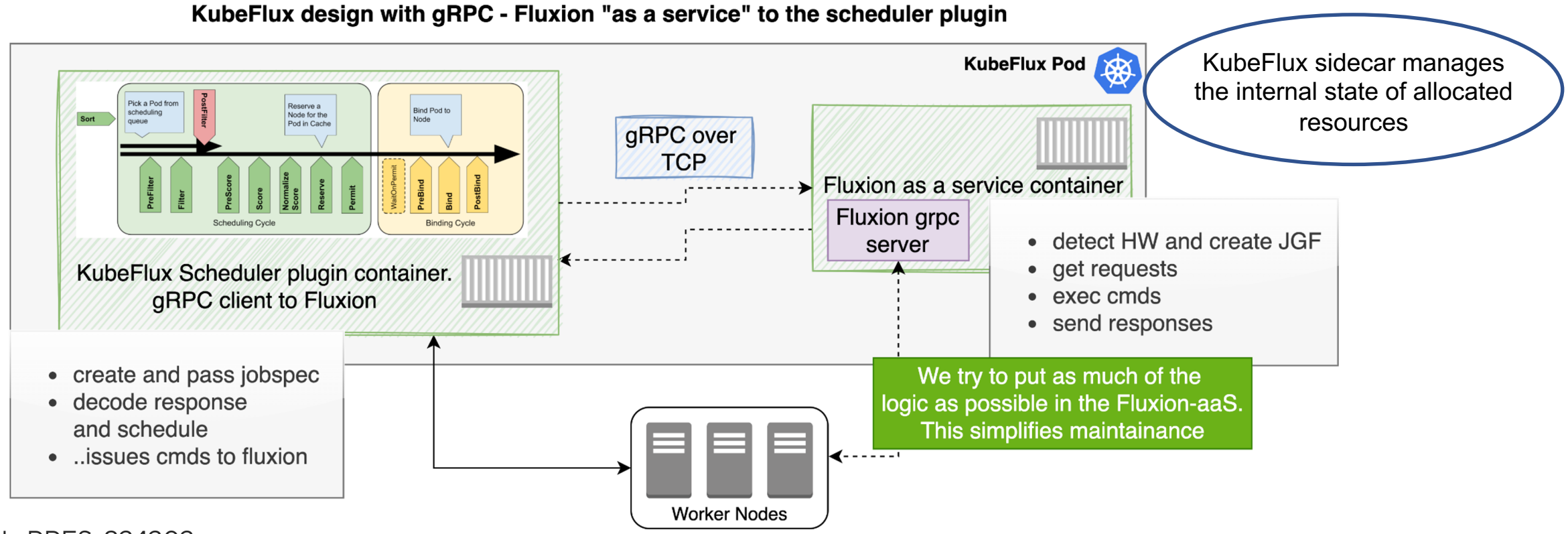
KubeFlux:

- Can decide between different flavors of packing and spreading
    - compute intensive vs network intensive workloads
- High- and low-level constraints captured in a jobspec also create different placement (i.e., shared vs exclusive use)

KubeFlux Plugin Description

Based on **Fluxion**, the scheduler component in **Flux Framework**

- Used in KubeFlux as sidecar container

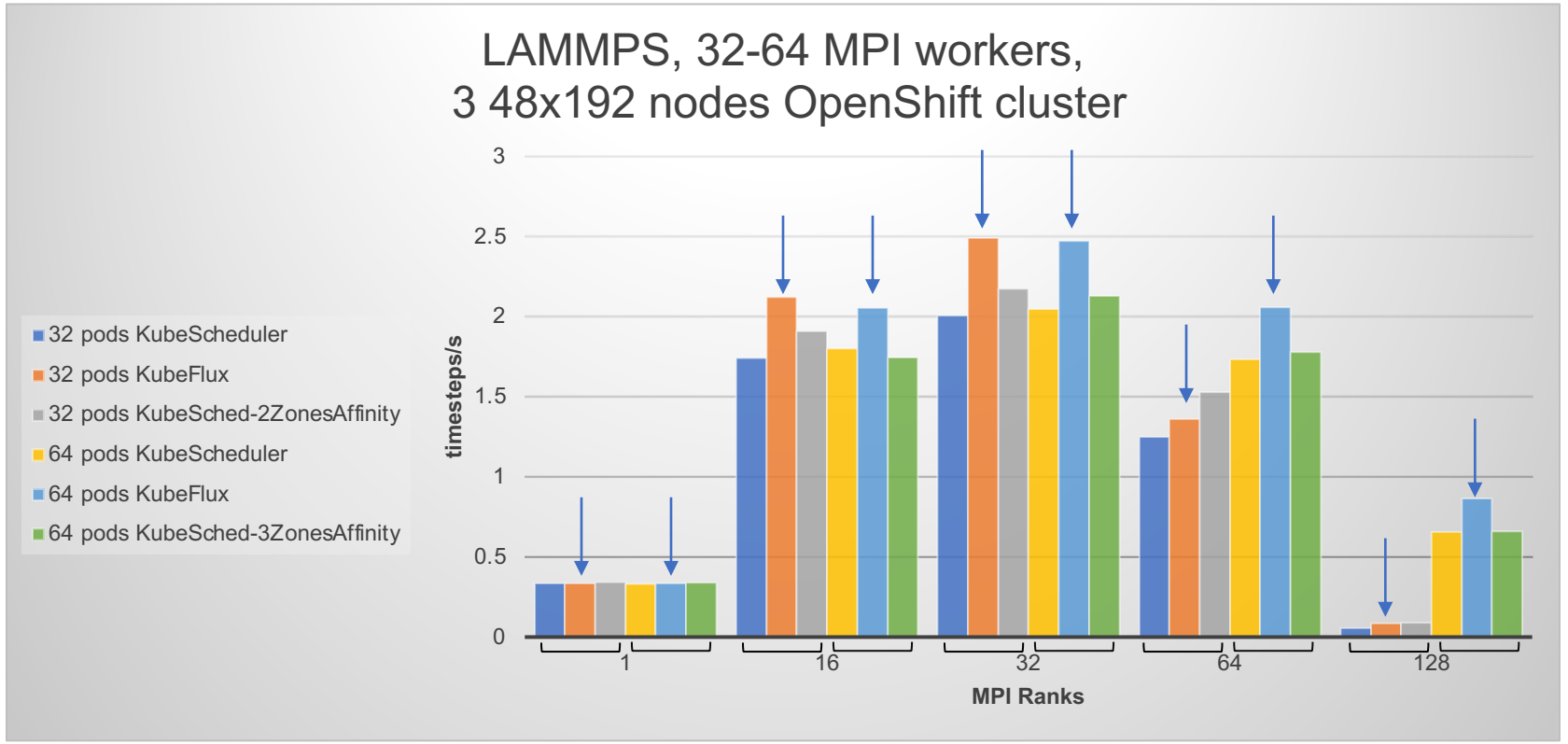


Use Case: Traditional HPC (LAMMPS)

MPI-based molecular dynamics application that models ensembles of particles in a liquid, solid, or gaseous state

KubeFlux:
- pack match policy

KubeScheduler:
- Default
- Zone affinity to force a limited packing

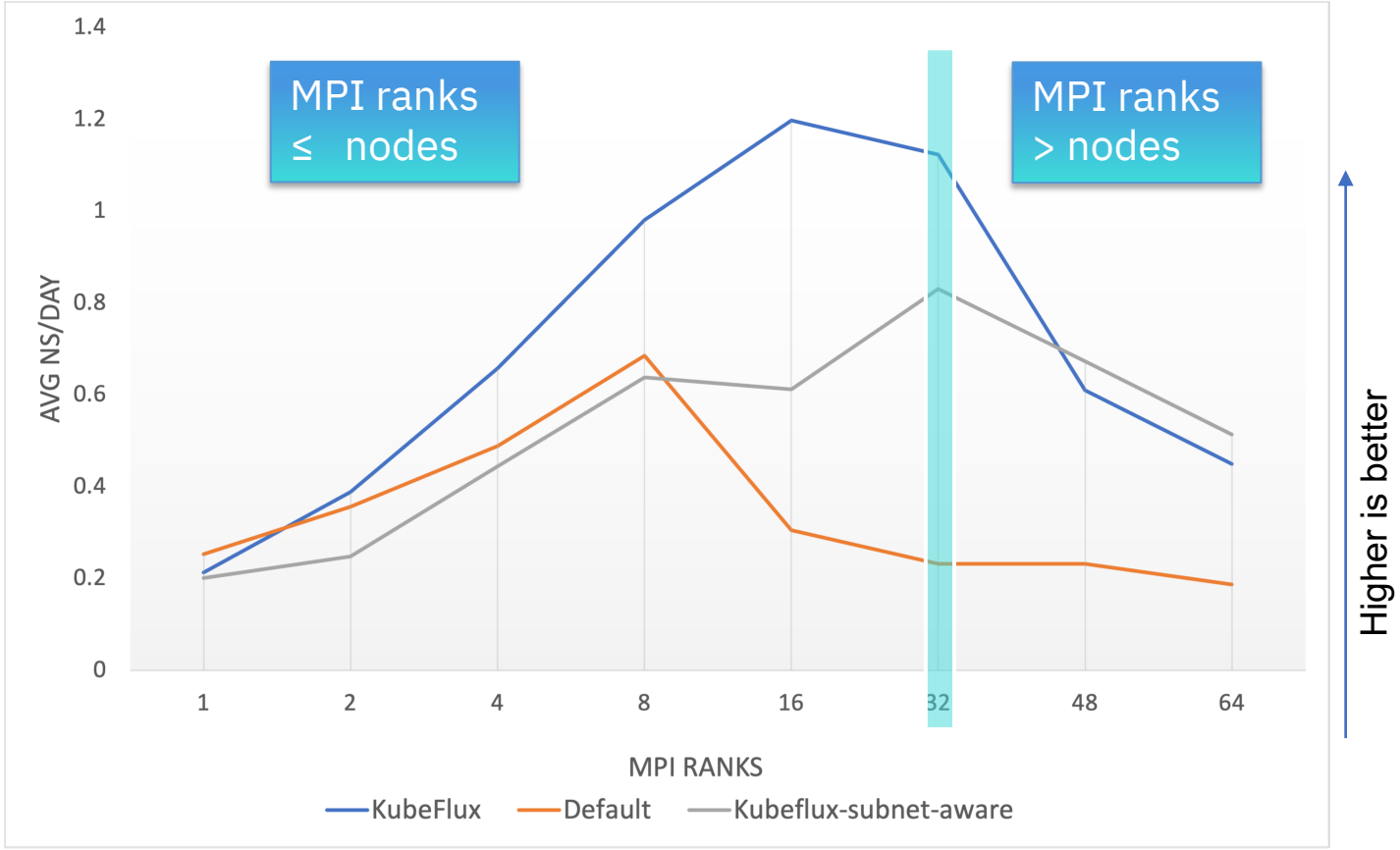


Use Case: Traditional HPC (GROMACS)

MPI-based large-scale simulations of molecular dynamics – benchmark: water molecules

KubeFlux:
- Vanilla spread match policy
- AZ aware spread match policy

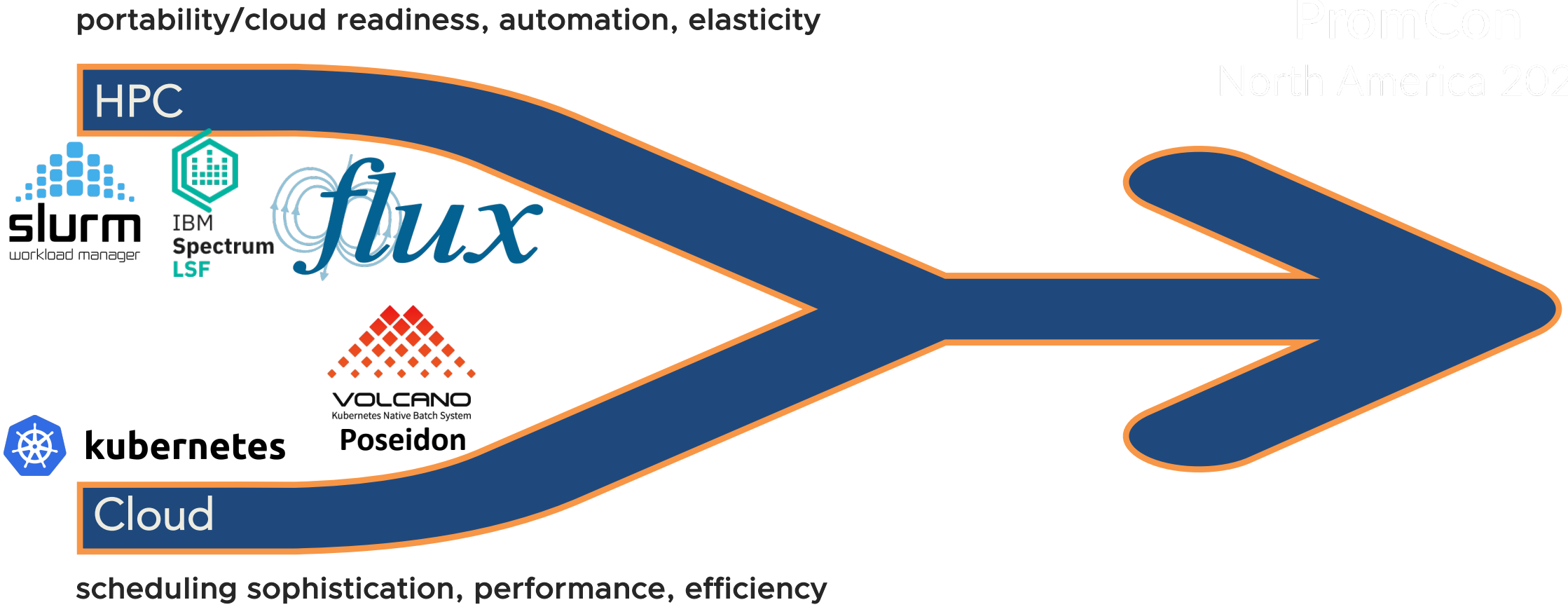
KubeScheduler:
- Default



Published at SC'21 CANOPIE Workshop (to appear)
It's a Scheduling Affair: GROMACS in the Cloud with the KubeFlux Scheduler

There is a significant gap between K8s and HPC RJMS. KubeFlux is the first step to closing it.

KubeCon CloudNativeCon Europe 2022
PromCon North America 2021



Thanks!

milroy1@llnl.gov
c.misale@ibm.com

git@github.com:cmisale/scheduler-plugins.git

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

LLNL-PRES-834303