

# Slurm and/or/vs Kubernetes

Tim Wickberg  
*CTO*

# Tim Wickberg

tim@schedmd.com

Chief Technology Officer

SchedMD LLC

- M.S. Computer Science from RPI
  - Thesis on Slurm integration with burst buffers
    - Design underpins Slurm's current burst\_buffer plugin architecture
- "Recovering" HPC Systems Administrator
  - Previously at GW and RPI, responsible for Slurm configuration and management
    - Alongside 100 other hats
- Now responsible for development and future direction for the open-source Slurm workload manager

# Slurm and SchedMD



# Most people know Slurm

## **Allocates access**

to resources to users for some duration of time for a workload

---

## **Provides framework**

for starting, executing, and monitoring work on the set of allocated nodes

---

## **Arbitrates contention**

for resources by managing a queue of pending work

# Slurm on TOP500

60%

Slurm runs on  
6 of the top 10 systems  
on the TOP500

50%+

Slurm runs on more than  
50% of the TOP500

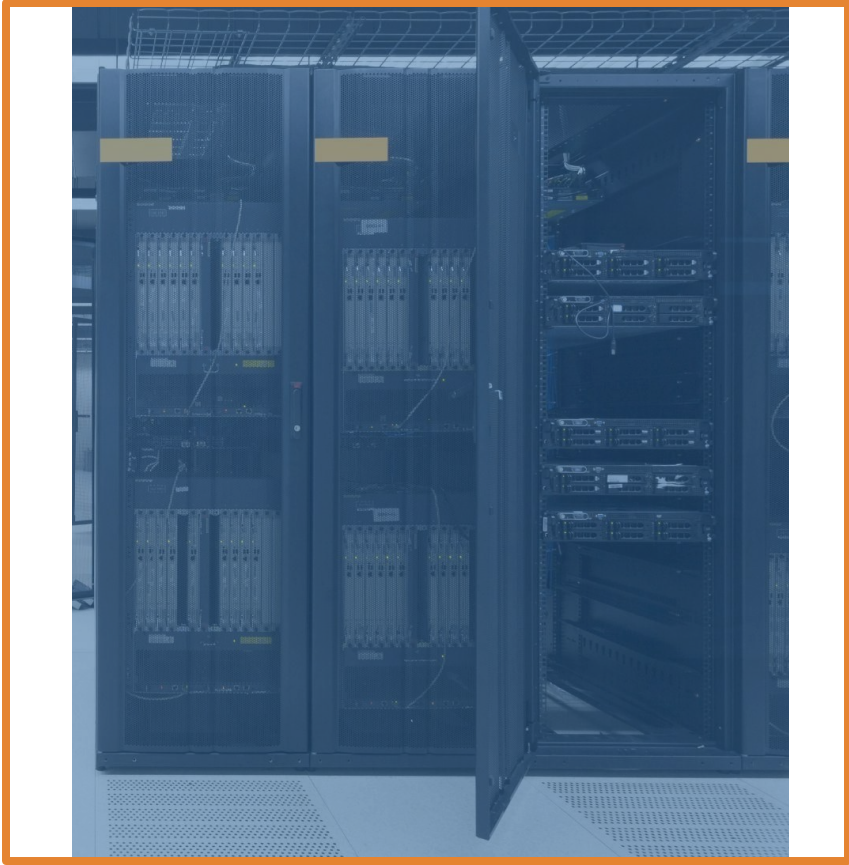
1. Frontier 3. LUMI 4. Leonardo 8. Perlmutter 9. Selene 10. Tianhe 2A

## But what is SchedMD?

Maintainers and supporters of Slurm

- Only organization providing Level-3 support
- Training
- Consultation
- Custom Development





# SchedMD + Dell

- Dell SKUs are available for:
  - SchedMD Support
  - Slurm Training
  - Consultation
- Slurm underpins the Dell APEX HPC offerings
- Slurm is at the heart of the Dell Omnia stack
- Contact your Dell sales rep or [jacob@schedmd.com](mailto:jacob@schedmd.com) for more details

# Slurm and/or vs Kubernetes



# Background

- This talk is meant to discuss the interplay between traditional HPC workload managers - Slurm - and cloud native orchestrators - Kubernetes
- "and/or/vs"... why not just pick a single conjunction?
  - Well... it's not that simple
  - Depending on your site, users, and systems, either Slurm or Kubernetes, or Slurm and Kubernetes combined, may be appropriate stacks

# Warning!

- This is meant as a high-level, somewhat simplified, view of two complex products
  - Slurm and Kubernetes are both open-source
    - There are patches, plugins, and configurations that look radically different than what I've described
    - Both systems continue to evolve well beyond their original designs

# Perspectives - Kubernetes

- Kubernetes was built to manage long-running processes
  - Designed to **orchestrate** multiple microservices
    - Usually in support of one or more web services
  - Core architecture permits scaling cluster size according to external demand
    - And managing availability and redundancy for the constituent services
- Cloud-native systems assume "**infinite**" resources are available
  - And the workload is **finite**
    - Albeit, with fluctuations in instantaneous demand
- Prioritization not a central aspect of cloud orchestration
  - All workload is expected to run concurrently by default

# Perspectives - Kubernetes

- Kubernetes approaches scheduling at a different level - node centric
  - Scheduling API granularity is fixed at the node level
    - Extensions such as NVIDIA's DRA allows for GPU management
  - No model for CPU core affinity
    - Can't - centrally - ensure a pod won't share a core with other workloads
  - Scheduling semantics reflecting cloud workload demands, rather than HPC
    - E.g, **Affinity** and **Anti-Affinity** scheduling policies
      - Anti-Affinity is used to ensure pod instances don't share a node
        - Critical for architecting redundant systems
        - But doesn't translate into traditional HPC batch scheduling
- Services are containerized by default
- System use is generally programmatic, through tools like Terraform

# Perspectives - HPC Batch Scheduling

- HPC systems assume system size is **fixed**
  - And the workload is **infinite**
  - Queue prioritization is thus critical
- "Slurm is a policy engine" - quote stolen from a colleague
- Slurm manages a number of intertwined HPC system management tasks
  - Job queuing and prioritization - **scheduling**
  - Job accounting
  - Control user access to compute resources (cgroups, pam\_slurm\_adopt)
  - Enable large-scale concurrent job launch (MPI, PMIx, nss\_slurm, sbcast)
- Jobs assume access to a usable, fully-featured, default Linux environment
  - Containerization - including Slurm's built-in container support - is optional
- Jobs are usually ad-hoc scripts, submitted through the command line
  - Newer features such as Slurm's RESTful API can support more programmatic interaction, but are not yet as widely adopted

# Current Kubernetes Batch Support

- Kubernetes has limited support for batch workflows
  - Modeled as either individual "pods", or as "jobs"
  - Most workflows use "pods" due to issues around the "jobs" model
- Prioritization models are limited
  - FIFO is most common

# Current Kubernetes Batch Support

- MPI-style workload support is weak
  - Concurrent pod scheduling is not guaranteed by default Kubernetes components
    - Default behavior for HPC batch schedulers
- "MPI Operator" is the most commonly used component to ensure pods launch roughly simultaneously
  - But does not scale - struggles to launch above more than 80 ranks
    - Citation - <https://doi.org/10.1109/CANOPIE-HPC56864.2022.00011>

# Convergence of HPC and Cloud-Native

- So... why am I talking about this?
- There's an opportunity to bridge the gap between HPC and Cloud-Native workloads
  - Find a way to bring familiar commands, tooling, prioritization models into newer architectures
  - Clusters will continue to evolve - users are interested in access to new tools and technologies
  - Both ecosystems stand to benefit from each other
    - Kubernetes from increased throughput, different approaches to job scheduling and prioritization
    - Slurm from newer cloud native technologies and tools, and increased focus on flexibility in support of new user workflows

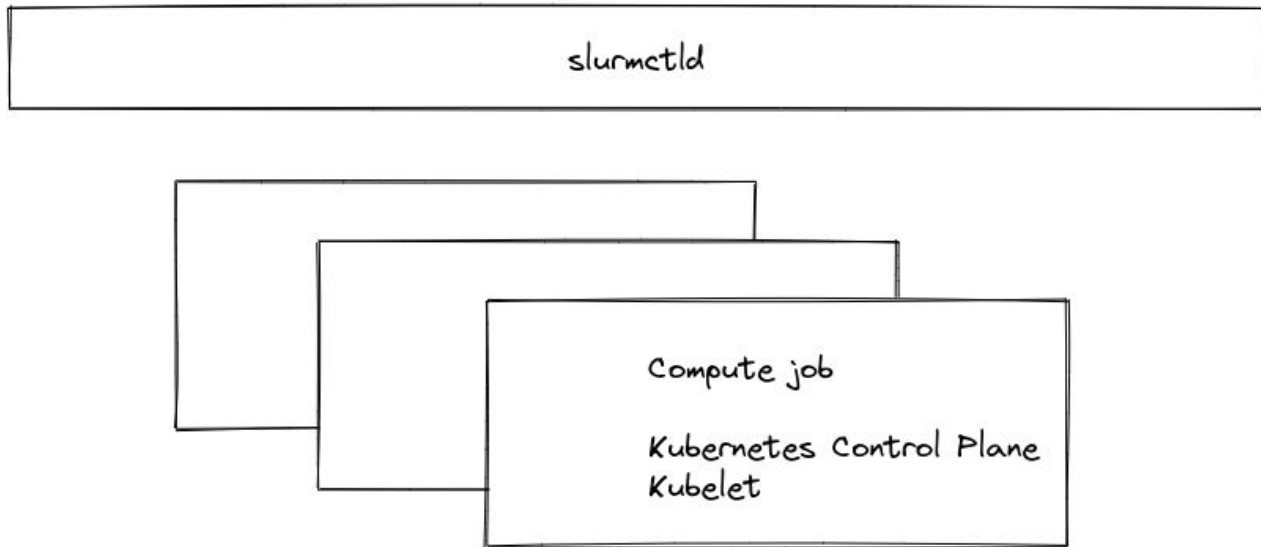


# Converged Environments

# Models of Converged Environments

- Four high-level models for a converged Slurm + Kubernetes environment:
  - Over
  - Distant
  - Adjacent
  - Under
- These are from Slurm's perspective... flip the Over/Under terms for Kubernetes' viewpoint

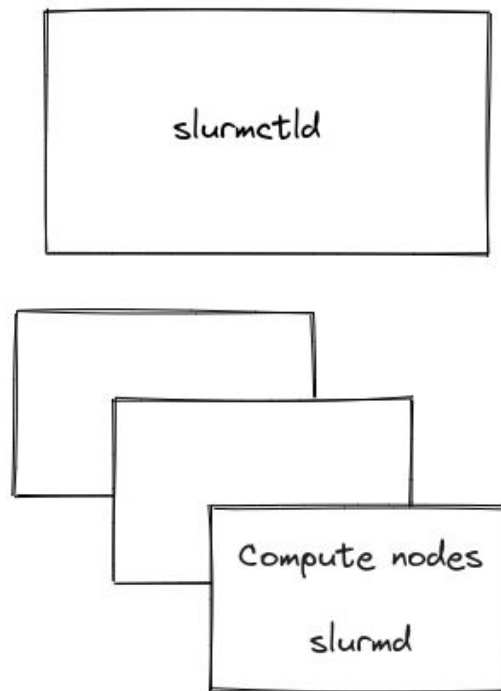
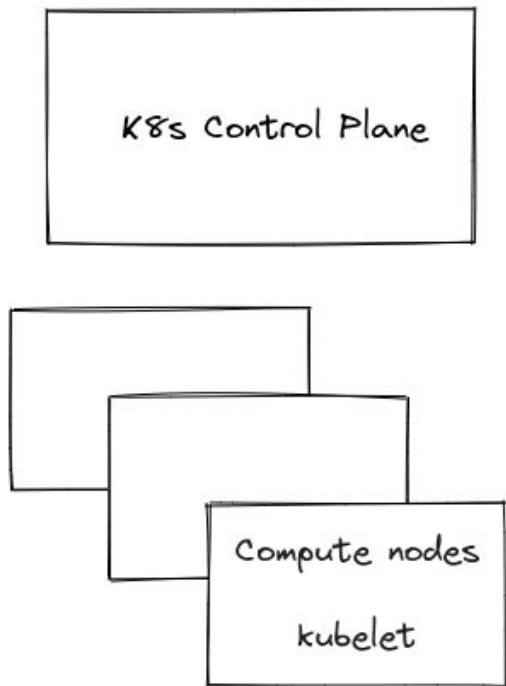
# Over



# Over

- Slurm manages all cluster resources
- Kubernetes clusters are created ephemeraly within Slurm batch jobs
- Kubernetes control plane unavailable until job launches...
  - Or needs to be hosted outside of the traditional cluster
- Not especially useful beyond test / development environments IMNSHO

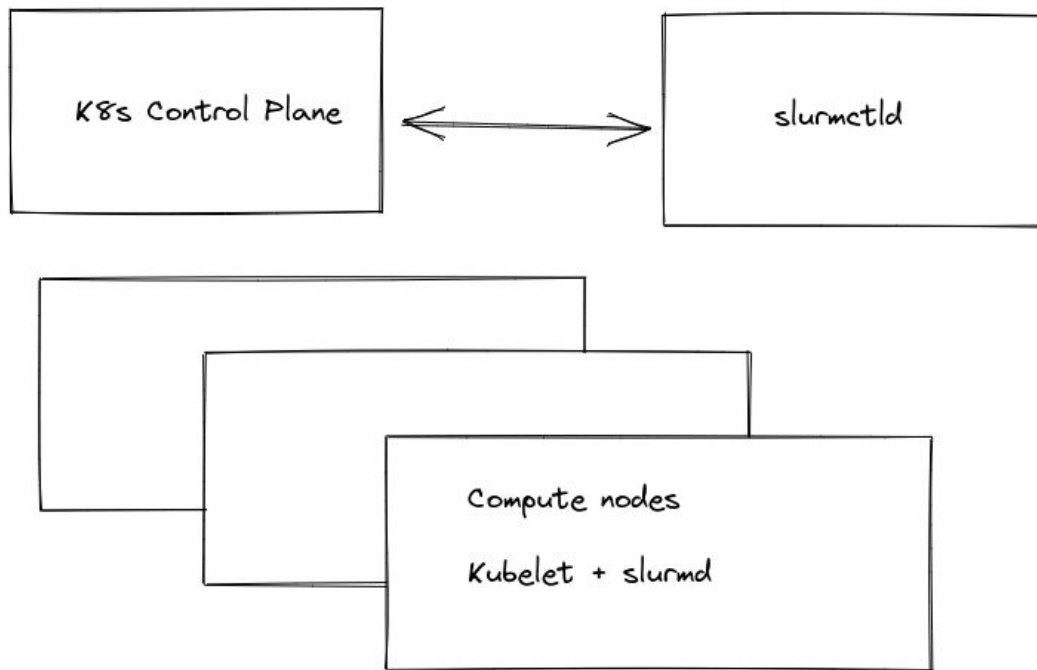
# Distant



# Distant

- Run both Slurm and Kubernetes within the cluster environment
- Potential to enlist an additional management tool to shift nodes between the two sides
- Neither Slurm nor Kubernetes are aware of the current resources and demand for the other environment
  - Management tool needs to handle assignment of resources between environments
- Approach taken today by tools such as Dell's Omnia toolkit

# Adjacent



# Adjacent

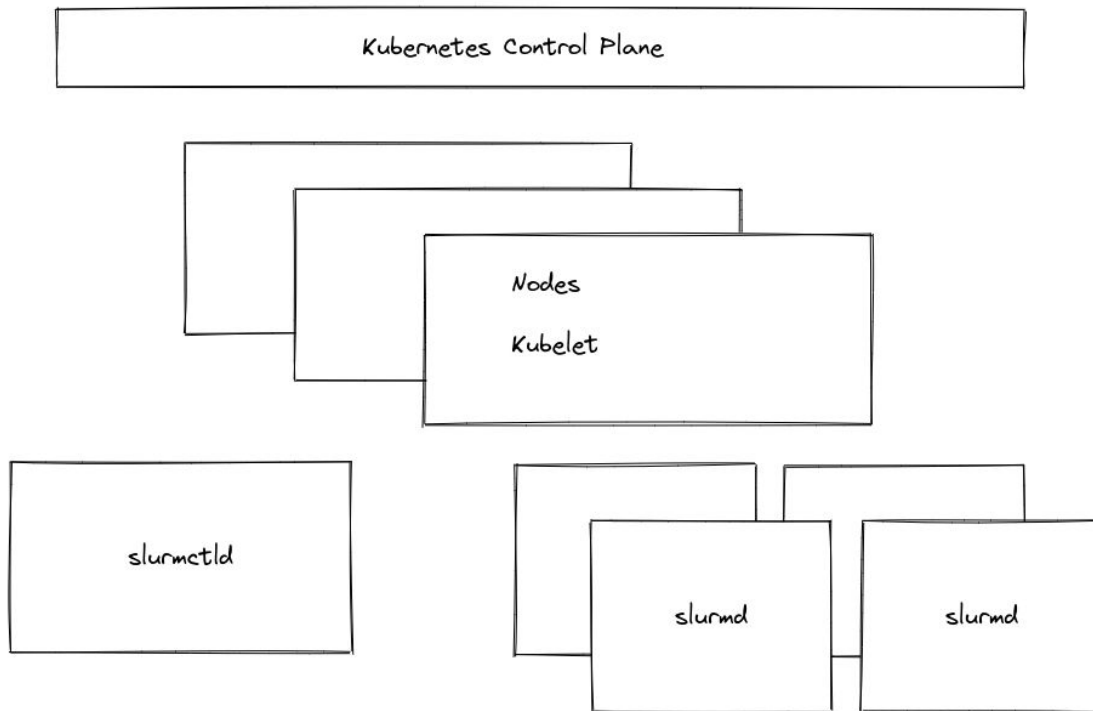
- Overlap both control planes
- Install Slurm Kubernetes scheduler plugin
  - Have Slurm prioritize and schedule both Slurm and Kubernetes workloads
- Kubernetes jobs managed by the kubelet
  - Full access to Kubernetes capabilities - sidecars, operators
- Slurm jobs run through Slurm
  - Manage high-throughput workloads and large-scale MPI workloads
  - Provides traditional CLI interfaces that HPC users expect
    - Alongside RESTful API



# Adjacent

- Current known limitations
  - Kubernetes scheduling is still at node-level granularity
    - DRA driver provides some support for GPU management
    - No further granularity available currently
      - But changes are difficult to push upstream
  - Some Kubernetes scheduling primitives - e.g., affinity/anti-affinity - are difficult to model in Slurm's internals

# Under



# Under

- Run Slurm cluster(s) within a Kubernetes environment
- Kubernetes-native cloud providers are already emerging
  - And all mainstream cloud environments have a managed Kubernetes offering
- Long-lived "login" nodes (Kubernetes pods) provide for traditional user experience
  - While allowing for increased user-to-user isolation
- Auto-scaling - best implemented through a Kubernetes Operator - can be used to shift resources to/from Slurm's control
  - The dynamic nodes feature in Slurm 22.05+ makes this simple
  - Auto-scaling here can also be a bit more nuanced than the existing Slurm power-saving-based cloud bursting model

# Under

- Pros
  - Traditional experience for Slurm users
  - Allows for higher throughput, and full MPI support for those workloads
- Cons
  - Kubernetes workloads run outside of Slurm's view
  - Prioritization between Slurm and Kubernetes workloads difficult
    - All limitations of Kubernetes scheduling apply

# Current Directions

# Current Options

- Dell's Omnia can deploy both Slurm and Kubernetes today
  - Corresponds to the distant model, as previously mentioned

# Work In Progress

- SchedMD is working with a partner to develop a stack that addresses both the under and adjacent models - actually a hybrid of both
  - Kubernetes used to manage and deploy the Slurm cluster on bare metal
  - Kubernetes Operator deployed to monitor Slurm cluster state through the REST API
    - Scale nodes (pods) up-and-down automatically by adding/removing dynamic nodes from the cluster
  - Kubernetes scheduling plugin also allows for Kubernetes workloads to be tracked and managed through that same Slurm cluster

# Work In Progress

- So... where is it? What is it called?
  - ... unfortunately it's not quite ready to go public yet, and I need to defer to our partner on the timing
- Our partner will be presenting on their work at SLUG'23 next week
  - And is working towards open-sourcing their efforts ASAP
  - We expect to have public details available at KubeCon and SC'23 this November



**Questions?**

# Tim Wickberg

tim@schedmd.com

Chief Technology Officer

SchedMD LLC

- Thank You
- Slides will be posted to the Slurm publication archive today:
  - <https://slurm.schedmd.com/publications.html>
- Upcoming events:
  - Slurm User Group Meeting (SLUG'23) at BYU, September 12-13
  - SC'23, November 13-16, Booth 463, as well as at the Slurm Community Birds-of-a-Feather
- Sales:
  - Talk to your Dell account rep, or contact [jacob@schedmd.com](mailto:jacob@schedmd.com) for more information

**SCHEDMD**

The Slurm Company