# GCLab

Hermit: Addressing Resource Underutilization Due to Drain Events in HPC Systems

Joseph Teague, Stephen Herbein, Dong H. Ahn, Michela Taufer

### **Project Overview**

Under certain circumstances, monolithic schedulers are unable to make use of some or all of the resources in HPC clusters, resulting in poor resource utilization. We call these circumstances "drain events." Drain events can occur for a number of reasons including future resource reservations and scheduled maintenance. To address this issue, we introduce Hermit: an extension to the Flux hierarchical scheduler that allows dynamic scheduling via resizable resource allocations. We use Hermit with a PILOT2 workflow called MuMMI and show how it is able to employ otherwise-unused resources during an emulated resource drain, allowing the workflow to get a head start on its preprocessing step.

### **Underutilization on HPC Systems**

During a drain event (e.g., resource reservation), resource utilization can drop significantly.





## Example of decreasing utilization on LLNL's Quartz cluster in the day preceding a resource reservation.

#### **Resource Utilization Without Hermit**





## Tackling Underutilization with Hermit

#### **Hermit Key Features**

- ' Extends Flux scheduler
- Maintains internal work queue, submits pilot jobs to system scheduler queue
- Allows resources to connect as availability changes, allowing work to be performed in any system state



#### Resource Utilization Using Hermit

## Lessons learned

- Resource underutilization can be mitigated in HPC environments using scheduling techniques.
- Multi-level schedulers like Flux are suited to the continuing development of novel solutions to this problem.









Emulation of resource utilization with draining of one node per minute and associated idle times (left) vs. draining of one node per minute with Hermit reclaiming unused nodes (right) on 48 nodes of LLNL cluster.