A Fast Simulator to Enable HPC Scheduling Strategy Comparisons

Alex Wilkinson [1,2], Jess Jones [2], Harvey Richardson [2], Tim Dykes [2], and Utz-Uwe Haus [2]

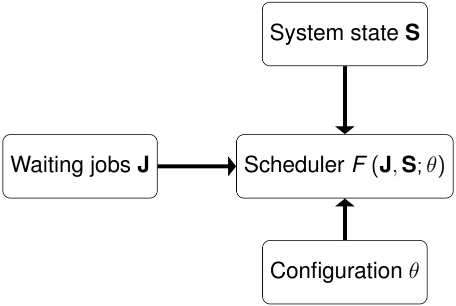
[1]University College London, United Kingdom

[2]HPE HPC/AI EMEA Research Lab, United Kingdom

25 May 2023

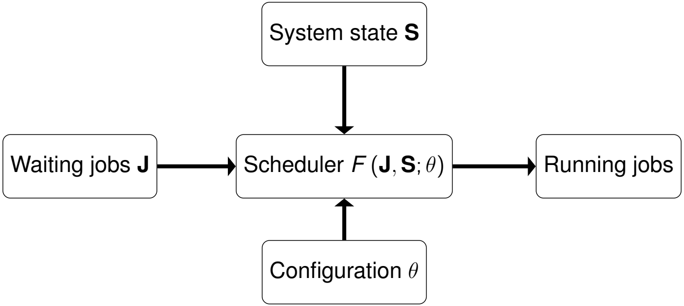
Introduction

- Job schedulers are a vital part of running an efficient HPC system



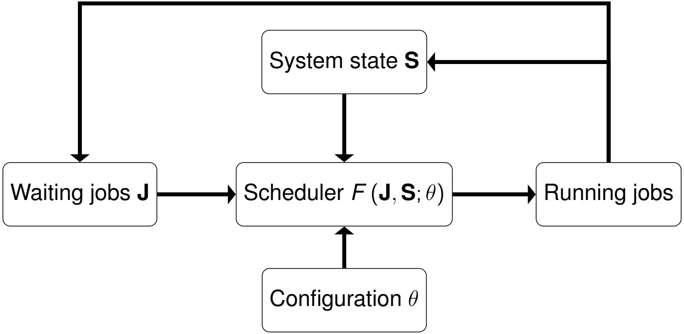
Introduction

- Job schedulers are a vital part of running an efficient HPC system



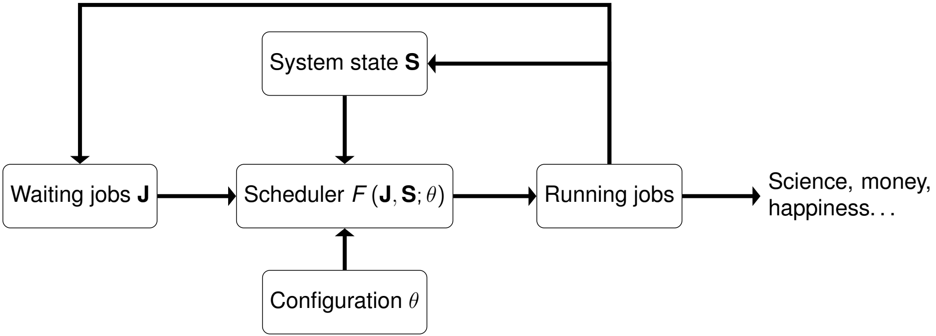
Introduction

- Job schedulers are a vital part of running an efficient HPC system



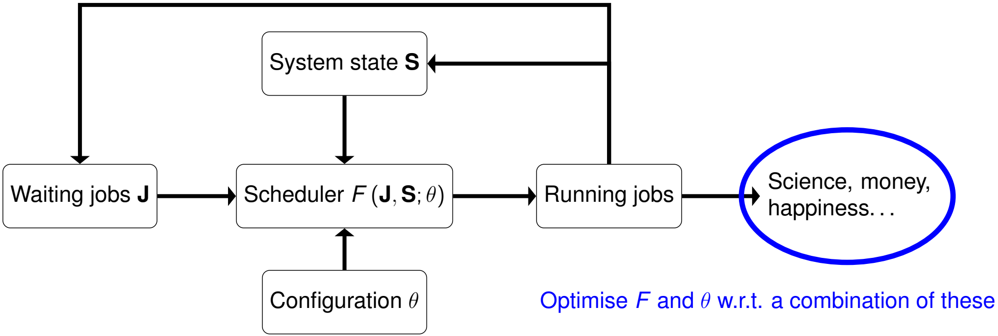
Introduction

- Job schedulers are a vital part of running an efficient HPC system



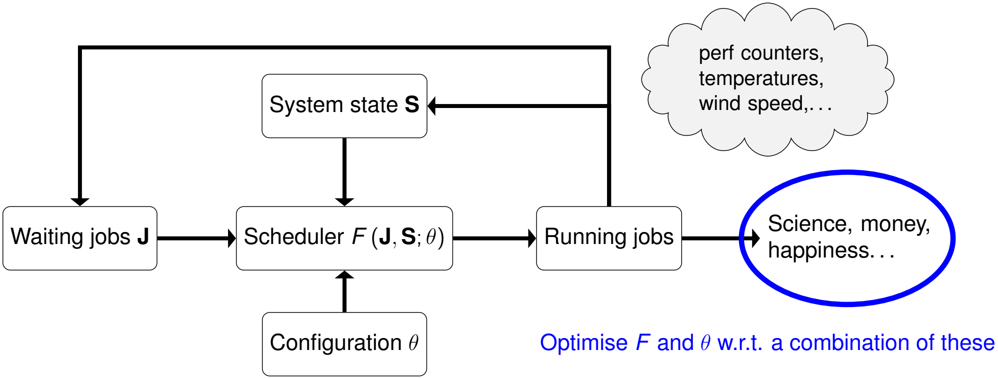
Introduction

- Job schedulers are a vital part of running an efficient HPC system



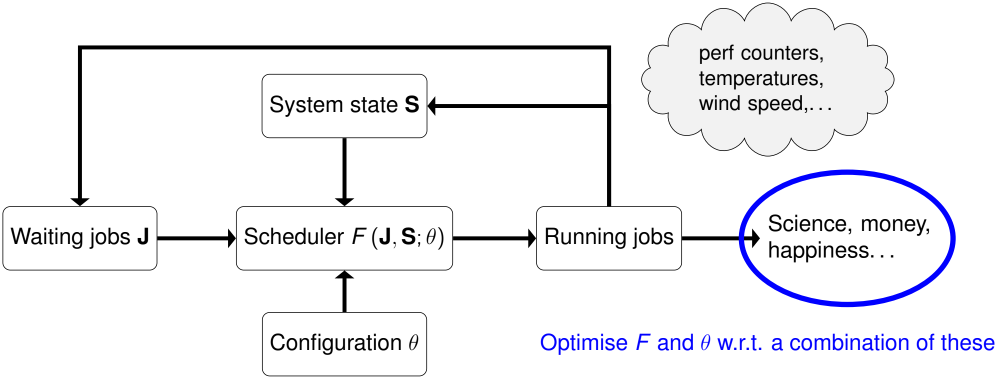
Introduction

- Job schedulers are a vital part of running an efficient HPC system



Introduction

- Job schedulers are a vital part of running an efficient HPC system



- Simulation allows exploration of configurations and scheduling algorithms without risking system efficiency — we will focus on the popular workload manager Slurm

A Lightweight Simulation

- Development of a simulation mode for Slurm started with A. Lucero in 2011[1] and has been iterated on in some other excellent works[2,3]
  → Modify the Slurm source code to emulate communication from nodes and skip through time
  → Limited speed up and extensibility

- Current research into HPC scheduling often uses custom simulations to evaluate algorithms
  → These can be simplistic and not replicate the configuration of a real system

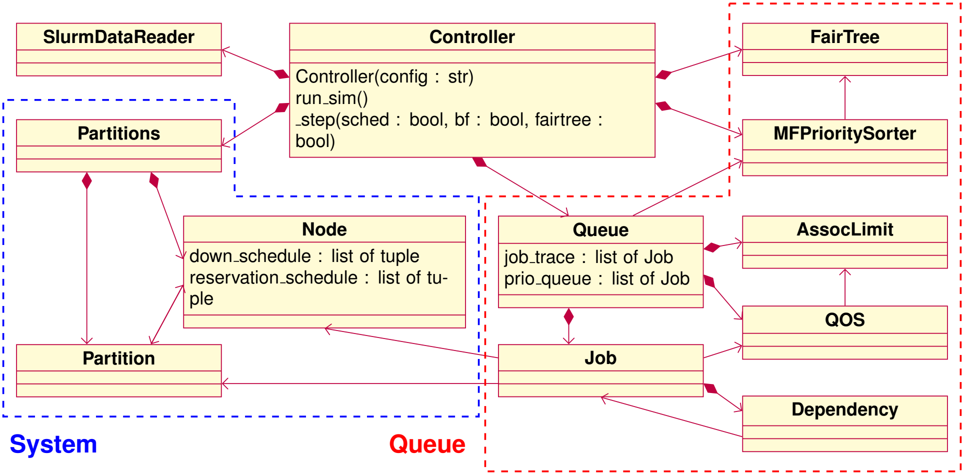
- We propose a fast simulation that can accurately reproduce the dynamics of real Slurm without trying to reproduce the specific design
  → Implement features directly relevant to scheduling from scratch

[1] Lucero, A.: Simulation of batch scheduling using real production-ready software tools (2011)

[2] Jokanovic, A. et al.: Evaluating slurm simulator with real-machine slurm and vice versa (2018)

[3] Simakov, N. et al.: A Slurm Simulator: Implementation and Parametric Analysis (2018)

Simulation Structure



Key Simulation Features

Backfilling

Conservative backfilling algorithm that simulates backfilling thread lock release

Resource Limits

Resource limits tracked at quality of service and association level

MultiFactor Priority and Fairshare

Queue sorted using a hierarchy of job features including a fairshare factor. Fairshare is implemented by sorting a rooted ordered tree of users association by usage and system allocation (Slurm's Fair Tree).

Limitations

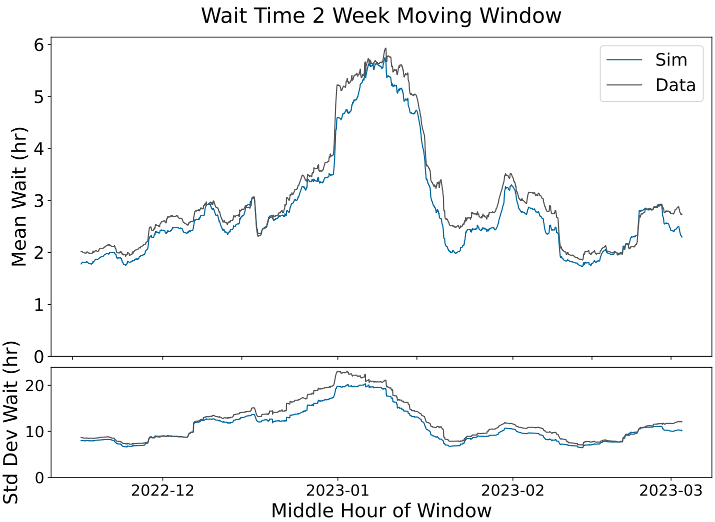
- Recovering full job and system information from Slurm accounting database
  - → Information such as dependencies and requested nodes can only be recovered if submitted via command line rather than in batch script
  - → Completed reservations not stored

- Some scheduling features missing from simulation
  - → Nodes are the only consumable resource
  - → Advanced features: job preemption and heterogeneous jobs

ARCHER2

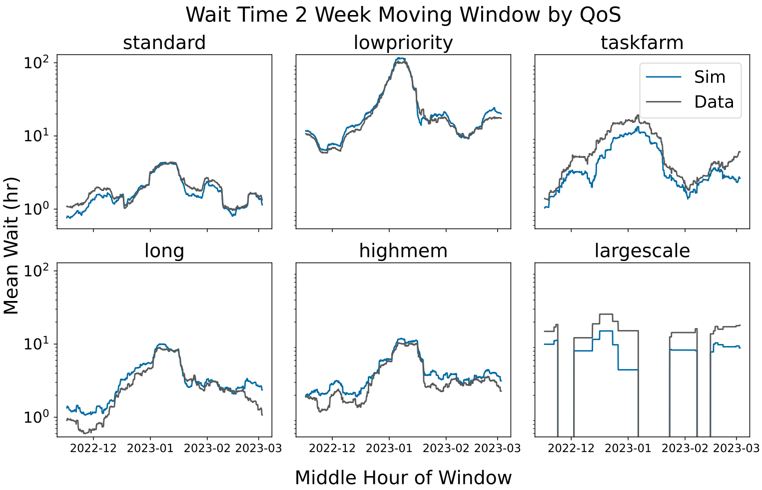
- Development of simulation was closely tied to ARCHER2
- ARCHER2 is the UK's national supercomputer consisting of 23 HPE Cray EX cabinets forming a network of 5,860 CPU compute nodes, 28 in TOP500
- 4 month job trace with ~600,000 jobs used to validate simulation accuracy
  → Dependencies, overlapping partitions, multiple QoS, advanced reservations, record of down nodes,. . .

Wait Times

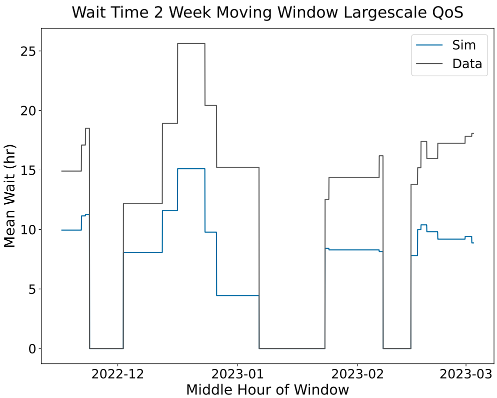


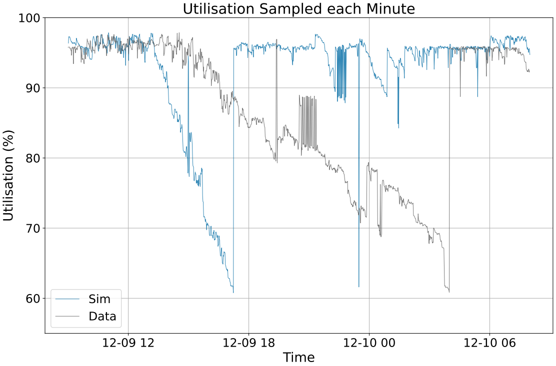
Wait Time 2 Week Moving Window
Sim
Data
Mean Wait (hr)
0
1
2
3
4
5
6
Std Dev Wait (hr)
0
10
20
2022-12
2023-01
2023-02
2023-03
Middle Hour of Window

QoS Wait Times

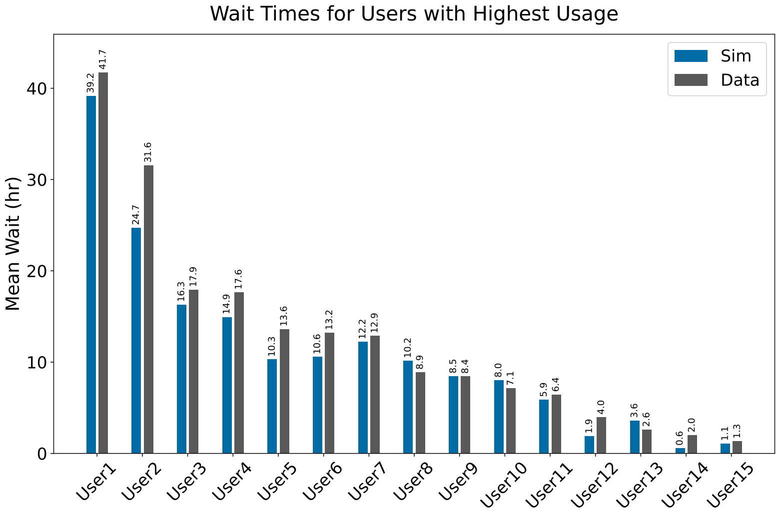


Largescale Jobs Discrepancy

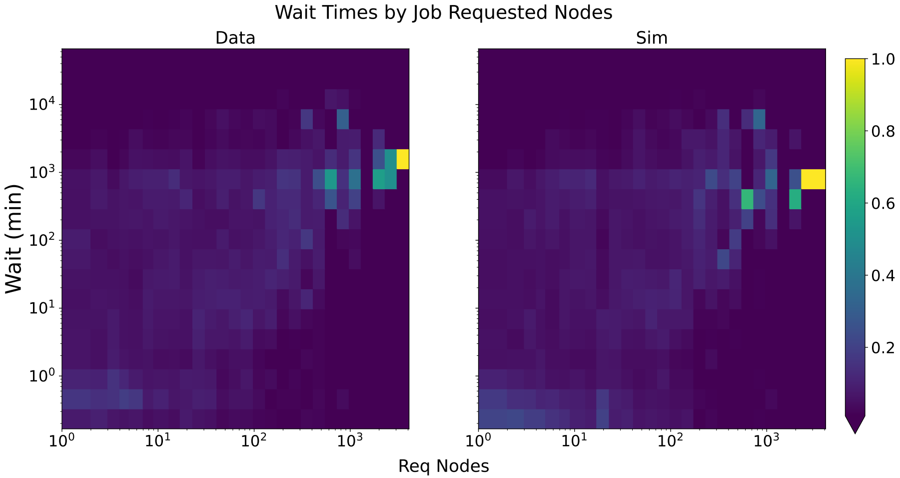




User Wait Times



Job Size Response

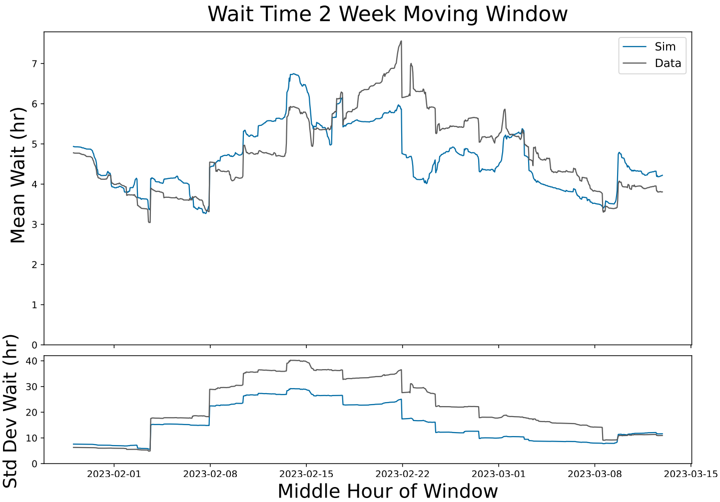


Job Length Response

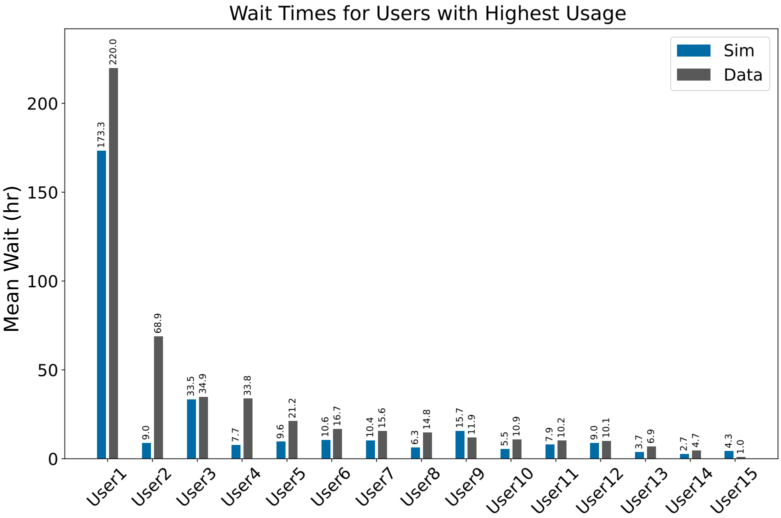
Other Systems: LUMI

- Important to check that the simulation is not tuned to ARCHER2
- LUMI is Europe's fastest supercomputer and part of the EuroHPC Joint Undertaking
  - → We consider the standard partition consisting of 1,022 CPU nodes
- 3 month job trace numbering ∼25,000 jobs
- Difficulty recovering past reservations
  - → Approximate using the maximum utilisation from jobs without reservations in a 2 day window

LUMI Wait Times



LUMI User Wait Times



Performance

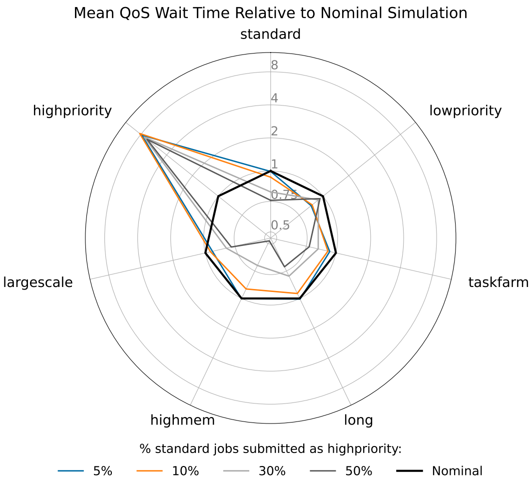
- ARCHER2 simulation takes approximately 7 hours 20 minutes, LUMI 25 minutes
  - $\rightarrow$ Speed up of $\sim$400 for ARCHER2 (400 simulation minutes takes 1 minute)
  - $\rightarrow$ Single threaded, memory usage $\sim$2 Gb depending on job trace size
  - $\rightarrow$ Processing time dominated by backfilling
- Speed ups from simulators in literature are typically between 10 and 25
- Exception is work from Barcelona Supercomputing Center[1] which achieves a 220 speed up with the CAE Curie log from the Parallel Workloads archive
  - $\rightarrow$ $\sim$200,000 jobs over an 8 month period running on 5,040 nodes
  - $\rightarrow$ Archive states 62% utilisation
  - $\rightarrow$ Unclear how performance would translate to modern 90+% utilisation workloads
- Direct comparisons between simulators is important future work

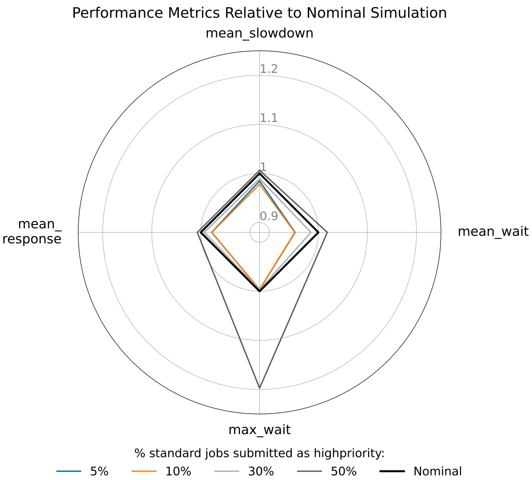
[1] Jokanovic, A. et al.: Evaluating slurm simulator with real-machine slurm and vice versa (2018)

Using the Simulator

- The simulator can be used to understand the effect of changes in scheduler behaviour on a production system
- Start with a simple change to ARCHER2's QoS configuration: adding a high priority QoS
- Consider scenarios with increasing proportions of *standard* QoS jobs being submitted as *highpriority* in the historical job trace

High Priority

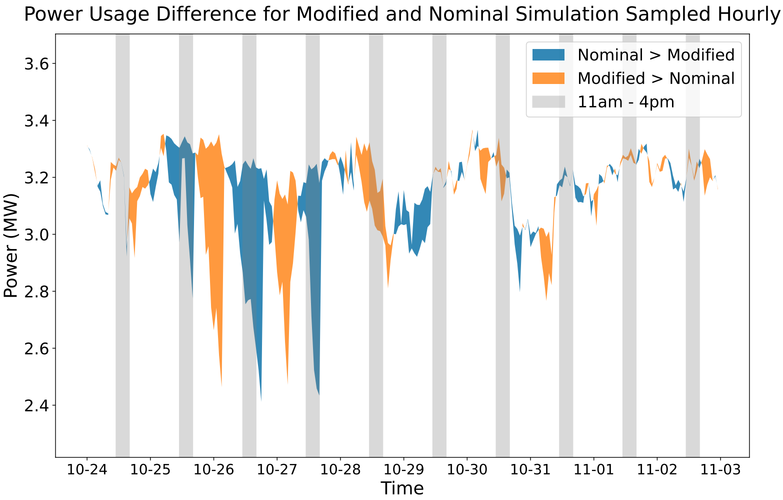




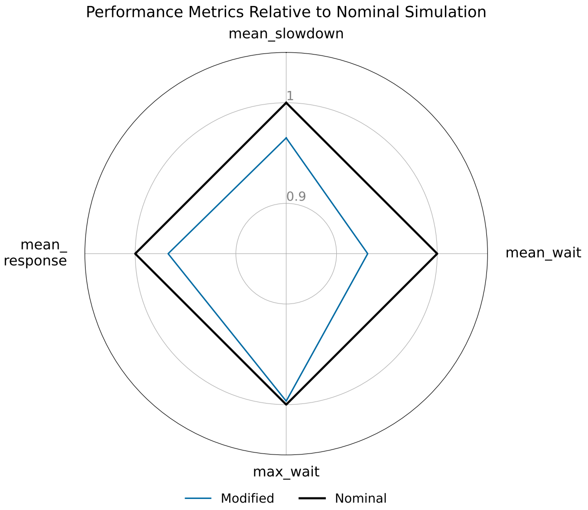
Large Jobs at Peak Times

- Slurm can be configured to associate energy counters from nodes with the jobs running on them
  - → System power usage can then be estimated from the jobs running at any given time in the simulation
- Consider scheduling jobs to minimise power usage during peak times of day
  - → Even with backfilling large jobs will require the system to partially drain in order to be scheduled
  - → Hold *largescale* jobs until morning, specific time depending on size

Power Usage


Power Usage Difference for Modified and Nominal Simulation Sampled Hourly
Nominal > Modified
Modified > Nominal
11am - 4pm
Power (MW)
3.6
3.4
3.2
3.0
2.8
2.6
2.4
10-24
10-25
10-26
10-27
10-28
10-29
10-30
10-31
11-01
11-02
11-03
Time

System Efficiency

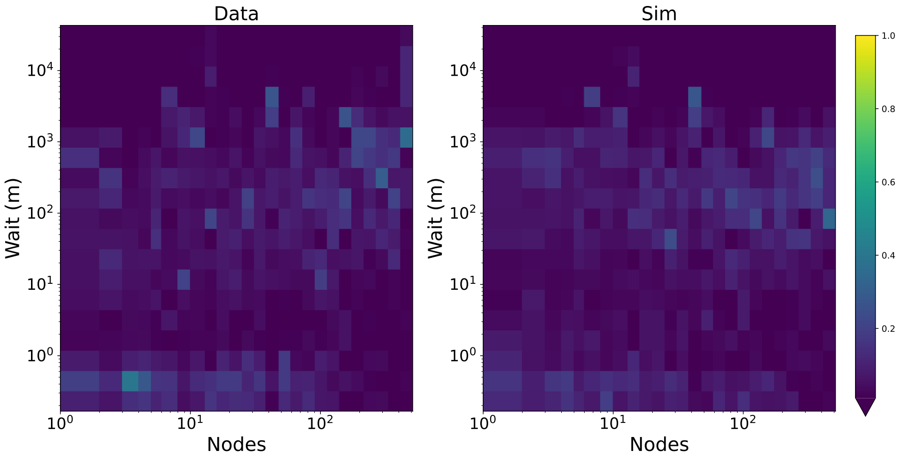


Summary and Future Work

- A fast and easily extendable scheduling simulation that incorporates many features of Slurm
- Validated with modern production systems
- Potential of simulation to provide insight into scheduling strategies demonstrated
- Future work:
  - → Direct comparisons with existing simulations
  - → Improving feature coverage of simulation to validate with a wider range of HPC systems

Backup

Job Size Response LUMI



Job Length Response LUMI