SC26 hpc unites | Chicago | EESP Workshop · Nov 15, 2026

Saving Energy Costs and Time on Supercomputers

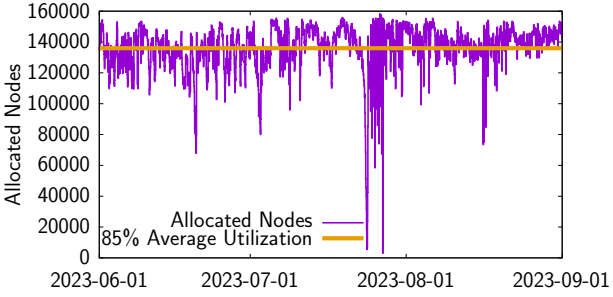
Scheduler-Integrated Recurring Node Shutdown Exploiting Malleable Jobs

Paul Schäfer[1], Anna-Lena Roth[1], Peter Arzt[2], Felix Krull[1], Sebastian Krenz[1], Felix Wolf[2], Jonas Posner[1]

[1]Fulda University of Applied Sciences, Germany
[2]Technical University of Darmstadt, Germany
paul.schaefer@cs.hs-fulda.de

Hochschule Fulda
University of Applied Sciences

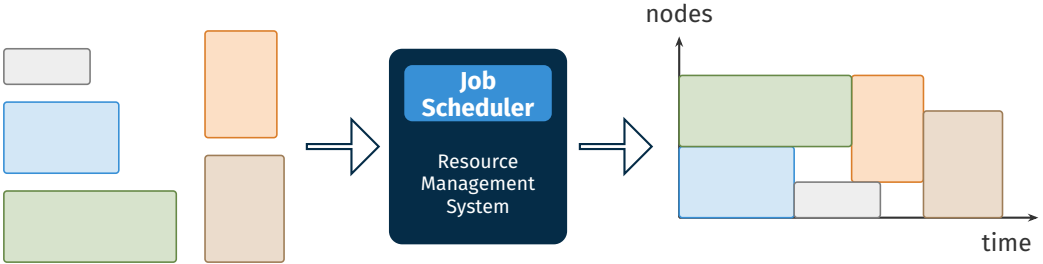
Idle Nodes Still Consume Energy



- Fugaku: 85 % average utilization
- Idle nodes still draw about half of their full-load power
- All nodes stay on around the clock
- Two options: use idle nodes or shut them down

Data: https://status.fugaku.r-ccs.riken.jp

Why Nodes Sit Idle

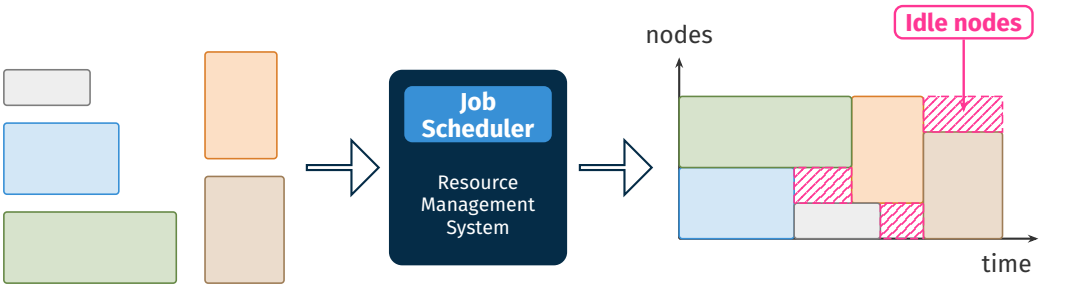


Jobs submitted by users

Possible schedule on a supercomputer

Resource management system in production, e.g., Slurm

Why Nodes Sit Idle



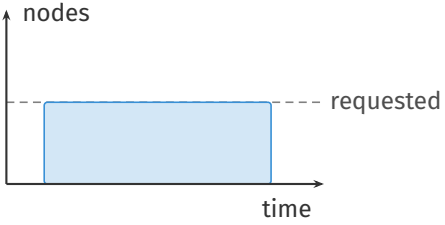
Jobs submitted by users

Possible schedule on a supercomputer

Resource management system in production, e.g., Slurm

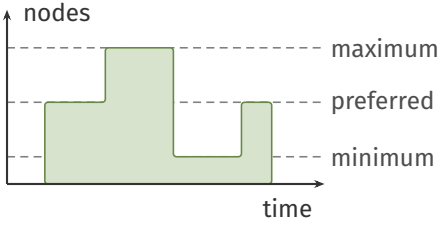
Rigid vs. Malleable Jobs

Rigid jobs



- Fixed node count
- Standard in production

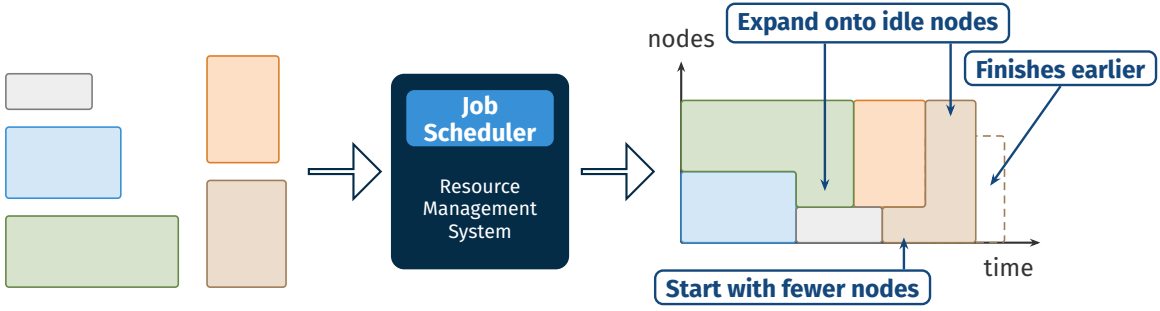
Malleable jobs



- Node range, resized at runtime
- Research

The scheduler decides when a job runs,
and for a malleable job also on how many nodes

Malleable Jobs Can Fill Gaps

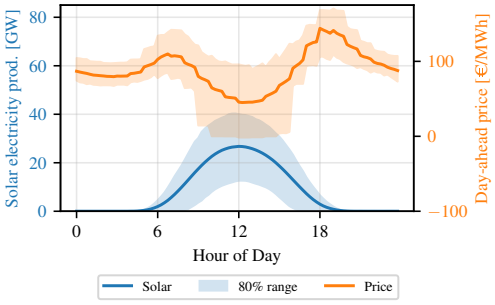


Jobs submitted by users

Possible schedule on a supercomputer

Malleability alone raises utilization, but also electricity cost

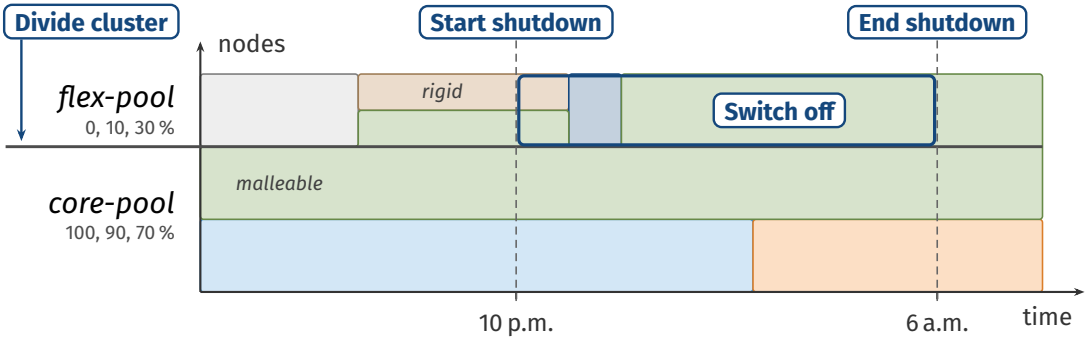
Electricity Supply Varies over the Day



Data: Electricity Maps (Germany, mean over all days of 2025)

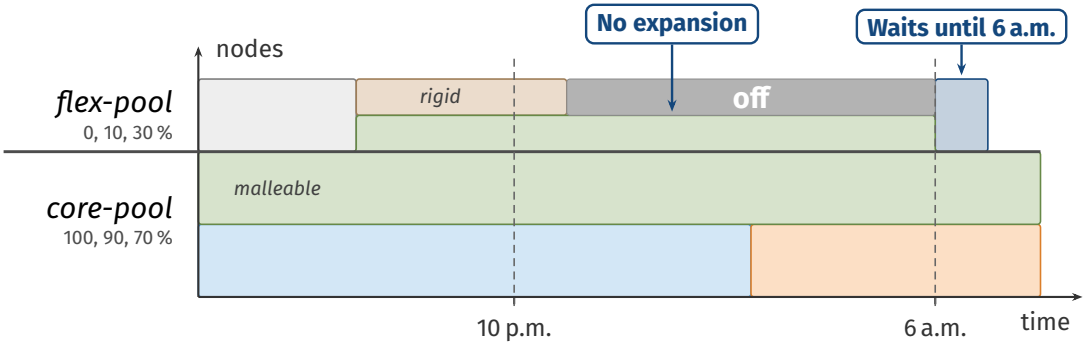
- **By day:** cheaper, cleaner alternative energy covers ≈ 30 % of full load
- **At night:** grid energy only
- **Idea:** shut down parts of the cluster at night
- **Approach:** malleable jobs free these nodes in time
- **Savings:** grid energy of nodes off instead of idle

Flex-Pool Shutdown



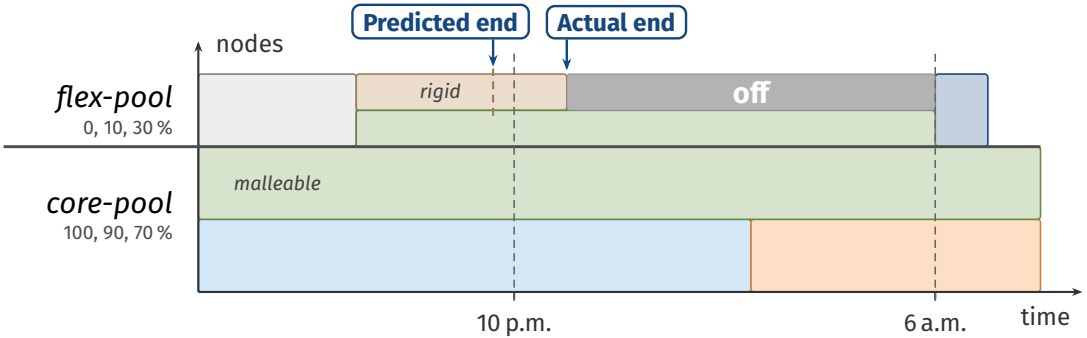
- **No shutdown:** malleable jobs expand onto idle nodes and the flex-pool runs all night

Flex-Pool Shutdown



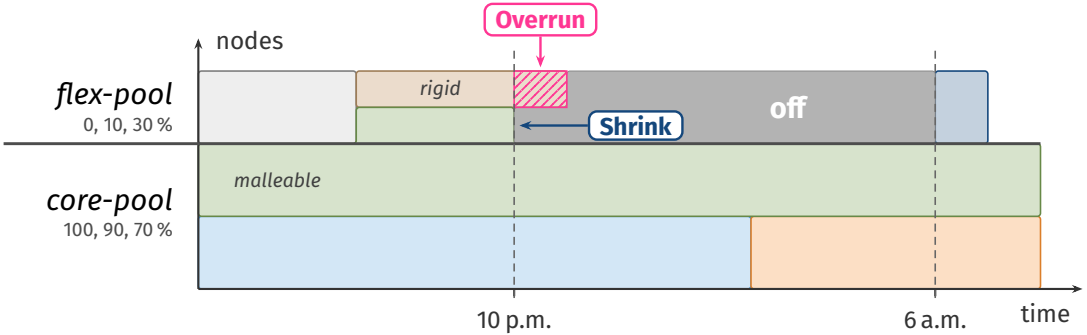
- **Off-window:** no jobs start or expand on the flex-pool, and its free nodes shut down

Flex-Pool Shutdown



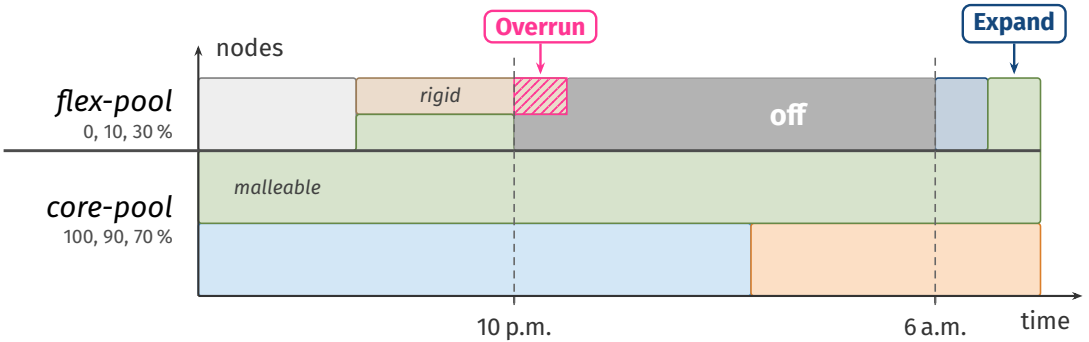
- **Admission control:** jobs start on the flex-pool only if predicted to finish before 10 p.m.

Flex-Pool Shutdown



- **10 p.m.:** malleable jobs shrink off the flex-pool, and jobs that cannot leave cause overrun

Flex-Pool Shutdown



- **6 a.m.:** the flex-pool boots, waiting jobs start, and malleable jobs expand again

Job Scheduling Algorithms

	EASY Backfilling (EBF)	Min Steal Agreement (MSA)
Malleable jobs	Scheduled like rigid jobs	Resized at runtime
Start	Requested node count	Malleable jobs at their *minimum*
Shrink	–	Toward the *minimum*, so that the first waiting job can start
Expand	–	Toward the *maximum*, onto idle nodes
Both	EASY backfilling, same flex-pool shutdown and runtime prediction	

Agreement: the freed nodes are reserved for the first waiting job, and a later job may steal them if it does not delay that job

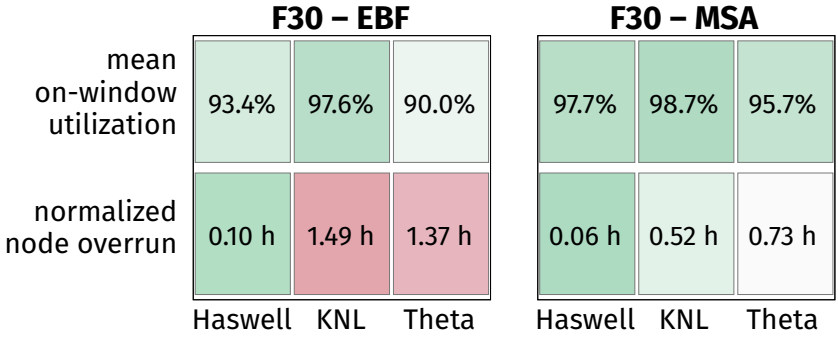
Simulation Setup

ElastiSim — Trace-driven job scheduling simulator, extended with node power states and flex-pool shutdown

3	job sets	Cori Haswell, Cori KNL, Theta (production traces)
5	malleable shares	0, 25, 50, 75, 100 % of the jobs (m0–m100)
2	schedulers	EBF, MSA
3	flex-pool sizes	0, 10, 30 % of the nodes (F0, F10, F30)
3	seeds	which jobs are malleable, prediction errors
270	simulations	$3 \times 5 \times 2 \times 3 \times 3$

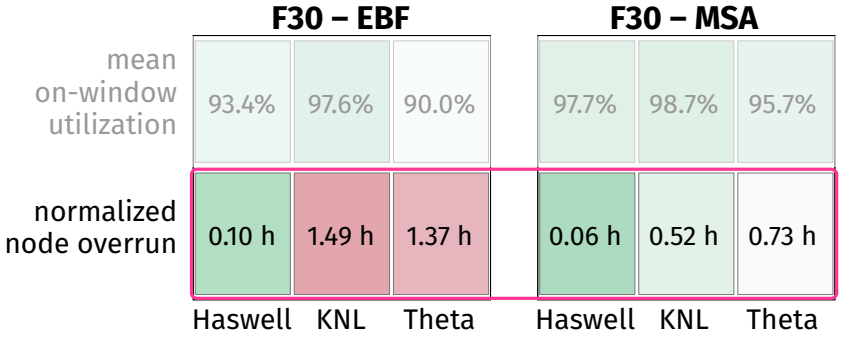
Malleable jobs are synthetic, with runtimes following Amdahl's law
Preferred = trace node count, minimum = 20 % of it, maximum while efficiency > 50 %

Malleable Jobs Free Nodes Reliably



Normalized node overrun: hours per off-window that a flex-pool node is not off mean over m0–m100
F30: 30 % flex-pool shutdown EBF: rigid scheduling MSA: malleable scheduling

Malleable Jobs Free Nodes Reliably

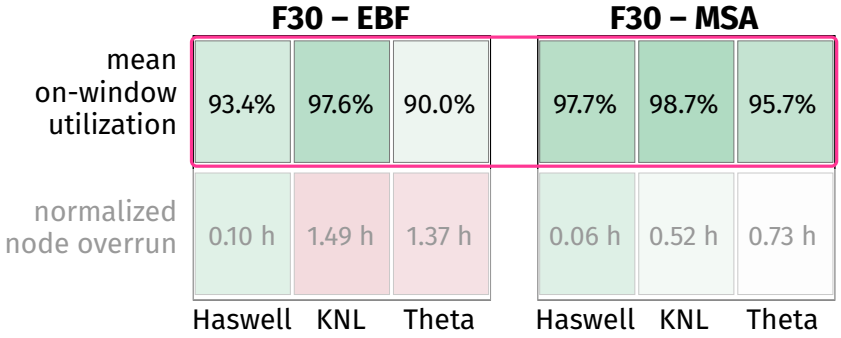


MSA: up to 65.2 % less overrun

Normalized node overrun: hours per off-window that a flex-pool node is not off mean over m0–m100
F30: 30 % flex-pool shutdown EBF: rigid scheduling MSA: malleable scheduling

Malleable Jobs Free Nodes Reliably

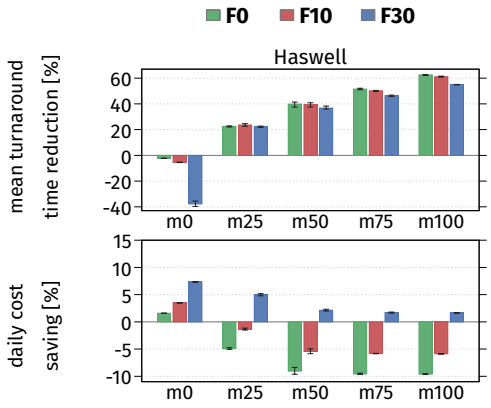
SC26



MSA: up to 65.2 % less overrun, on-window utilization above 95 %

Normalized node overrun: hours per off-window that a flex-pool node is not off mean over m0–m100
F30: 30 % flex-pool shutdown EBF: rigid scheduling MSA: malleable scheduling

Malleable Jobs Absorb the Removed Nodes



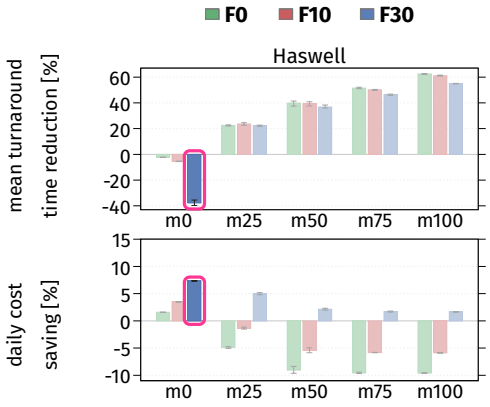
Baseline: rigid scheduling without shutdown

Turnaround time

Electricity cost

MSA runs Fx: *x* % flex-pool shutdown mx: *x* % malleable jobs

Malleable Jobs Absorb the Removed Nodes



Baseline: rigid scheduling without shutdown

Turnaround time

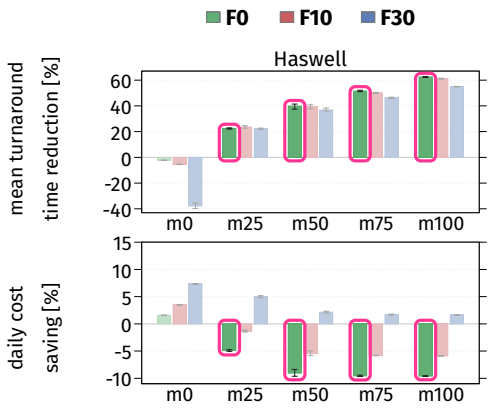
Shutdown only 37.8 % longer

Electricity cost

Shutdown only 7.3 % lower

MSA runs Fx: *x* % flex-pool shutdown mx: *x* % malleable jobs

Malleable Jobs Absorb the Removed Nodes



Baseline: rigid scheduling without shutdown

Turnaround time

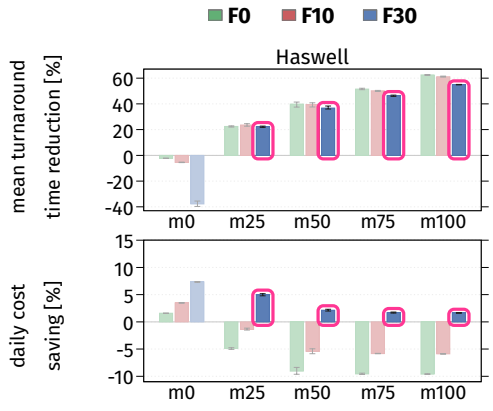
Shutdown only	37.8 % longer
Malleability only	22.5–62.5 % shorter

Electricity cost

Shutdown only	7.3 % lower
Malleability only	4.9–9.6 % higher

MSA runs Fx: *x* % flex-pool shutdown mx: *x* % malleable jobs

Malleable Jobs Absorb the Removed Nodes



Baseline: rigid scheduling without shutdown

Turnaround time

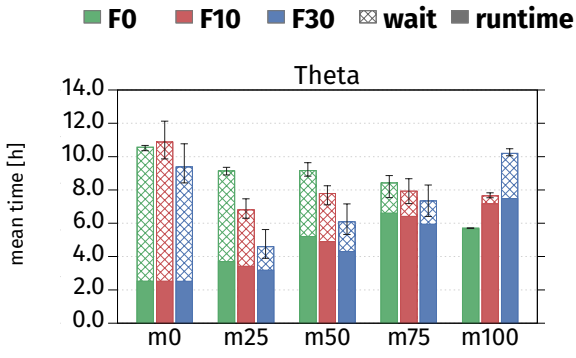
Shutdown only	37.8 % longer
Malleability only	22.5–62.5 % shorter
Both	22.3–54.9 % shorter

Electricity cost

Shutdown only	7.3 % lower
Malleability only	4.9–9.6 % higher
Both	1.6–5.0 % lower

MSA runs Fx: *x* % flex-pool shutdown mx: *x* % malleable jobs

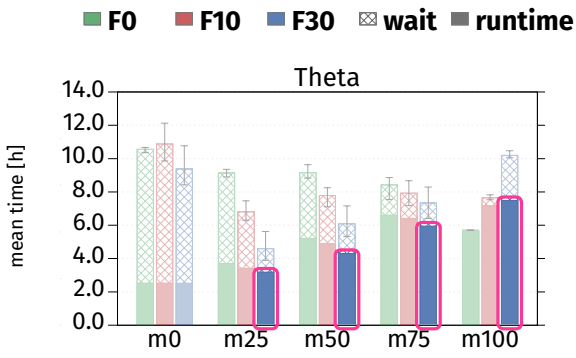
More Malleable Jobs Are Not Always Better



Theta, F30 m25 m100

MSA runs Fx: *x* % flex-pool shutdown mx: *x* % malleable jobs

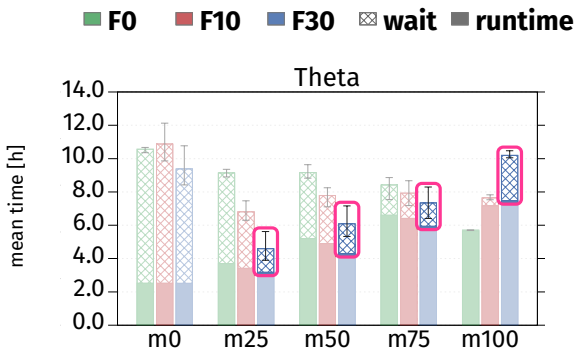
More Malleable Jobs Are Not Always Better



Theta, F30	m25	m100
Runtime	3.2 h	7.5 h

MSA runs Fx: *x* % flex-pool shutdown mx: *x* % malleable jobs

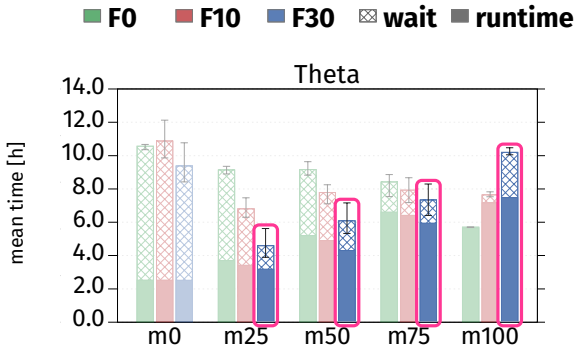
More Malleable Jobs Are Not Always Better



Theta, F30	m25	m100
Runtime	3.2 h	7.5 h
Waiting time	1.4 h	2.7 h

MSA runs Fx: *x* % flex-pool shutdown mx: *x* % malleable jobs

More Malleable Jobs Are Not Always Better



Theta, F30	m25	m100
Runtime	3.2 h	7.5 h
Waiting time	1.4 h	2.7 h
Turnaround time	**4.6 h**	**10.2 h**

MSA: jobs start at and shrink toward their minimum

MSA runs Fx: *x* % flex-pool shutdown mx: *x* % malleable jobs

Conclusion

- Idle nodes waste energy: our scheduler shrinks malleable jobs to shut nodes down
- **30 % of the nodes off at night, only 25 % malleable jobs: lower electricity cost *and* shorter turnaround time on all job sets**
- Malleable jobs free the flex-pool more reliably than rigid jobs
- More malleable jobs are not always better

Artifact



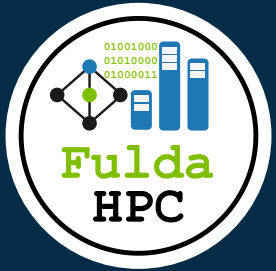
github.com/FuldaHPC/
artifact-sc26-node-shutdown

Next steps: realistic malleable applications, dynamic prices and off-windows, I/O and accelerators

SC26 hpc unites | Chicago

Thank you for your attention!

Questions?



paul.schaefer@cs.hs-fulda.de

github.com/FuldaHPC

doi.org/10.5281/zenodo.22846509

zenodo.org/communities/fuldahpc