Simulation approach for resource management

Thesis defense

Millian Poquet

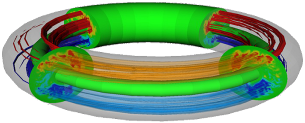
DATAMOVE team
LIG laboratory
Univ. Grenoble Alpes, Inria

Advisors: Denis Trystram and Pierre-François Dutot

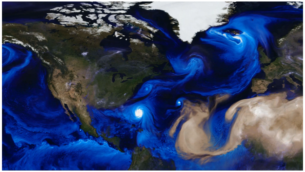
millian.poquet@inria.fr

High Performance Computing

Sunway Taihulight, current TOP500 leader

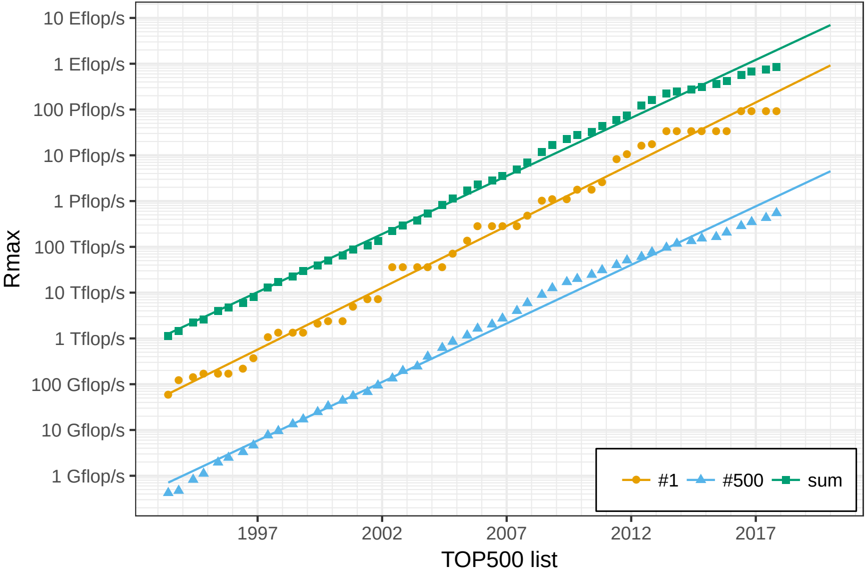


Tokamak plasmas (Gysela5D)

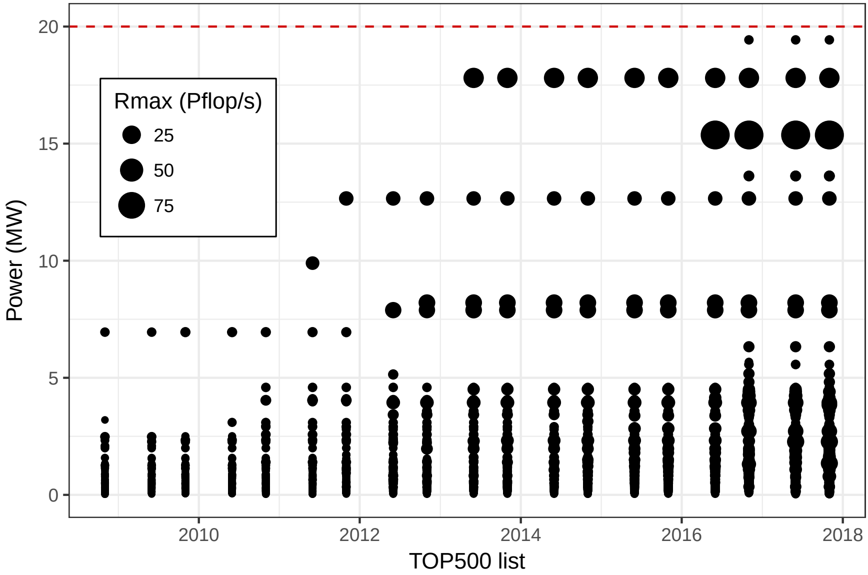


Weather prediction (GEOS)

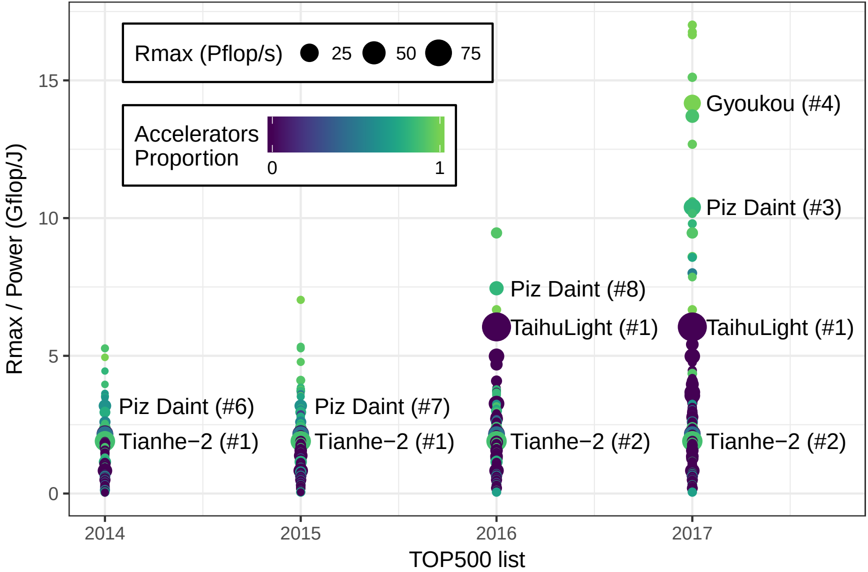
Computational power growth



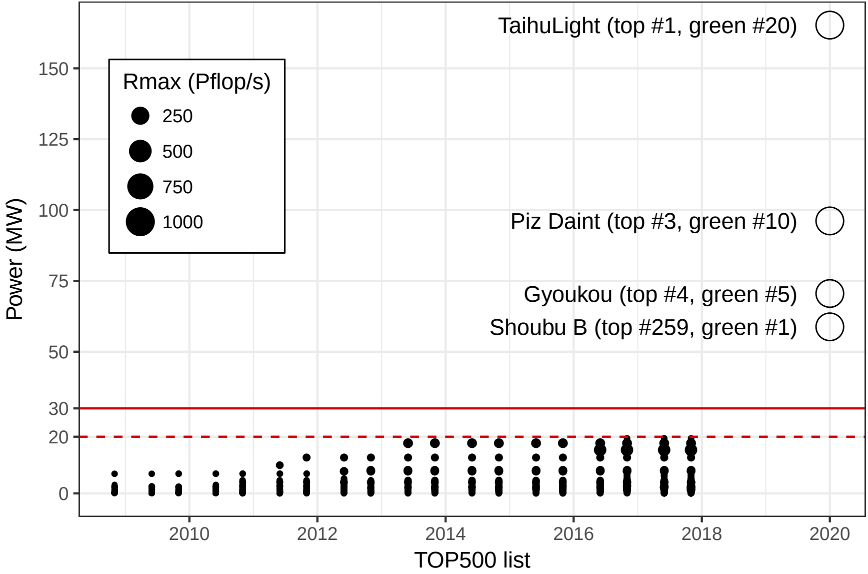
Power consumption evolution



Power efficiency evolution



Power consumption (naive) projection



Exascale challenges

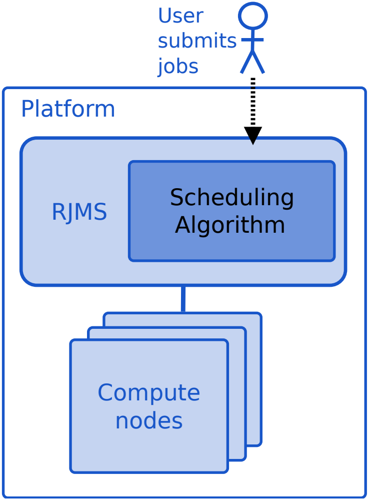
Keywords

- Efficiency
- Energy
- Data movements
- Resilience
- Heterogeneity
- Topologies
- ...

Addressed in this thesis

- How to save energy?
  At which cost?
- More generally, how to study such challenges soundly?

Lever: Resources and Jobs Management System (RJMS)



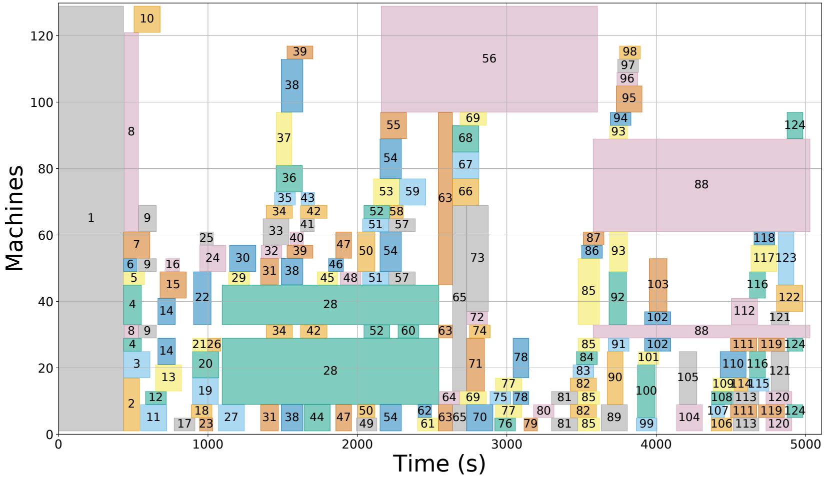
Non-clairvoyant

- Unknown submission times
- Unknown execution times (user-given bound)

Online algorithms

- React to events
- Execute jobs
  Switch machines

Resource management result: Gantt chart



Outline

1 Introduction

2 Batsim

3 Energy/Performance trade-offs

4 Conclusion

Improving RJMS efficiency

Experiments may involve thousands high-end machines for years.

Real platform

- All phenomena ☺
- Real results ☺
- Noisy 😐
- 1:1 speed ☹
- $$$$$$$$ ☹

Simulation

- Modeled phenomena 😐
- Realistic results 😐
- Deterministic ☺
- Fast → Very fast ☺
- $ ☺

Simulators in platform-level resource management

Most papers conduct experiments "by simulation".

- Source code rarely released
- (very) Dedicated
- (very) Short lifecycle
- (very) Simple models

Issues

Credibility [MLW01]

- Implementation robustness?
- Models validation?

Time-efficiency?

Proposition

We need sounder tools

- Adequate models
- Implementation confidence

Related Work

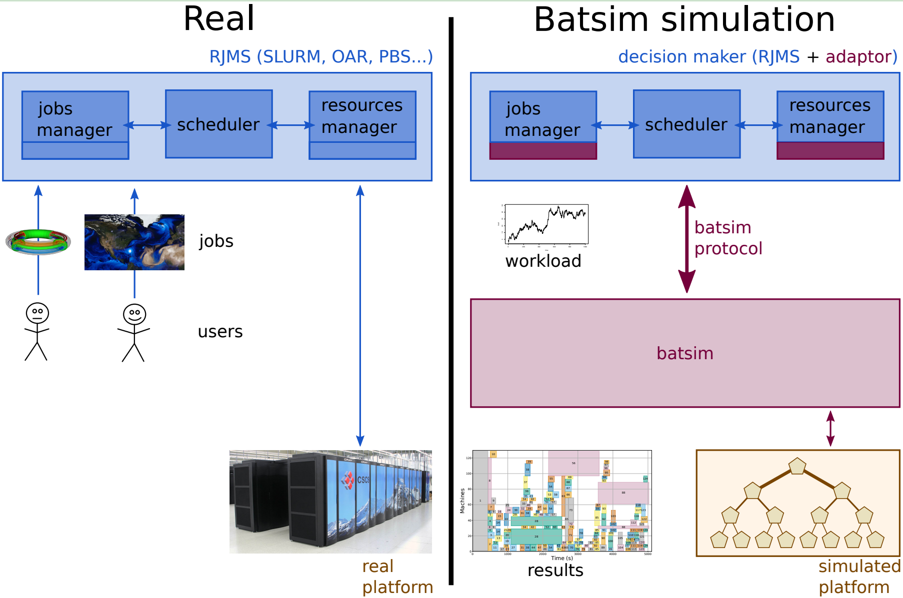
Scheduling community

- Alea [KR10]: GridSim-based
- Simbatch [GC06]: SimGrid-based
- [PML15]: INSEE [PM05] + local placement

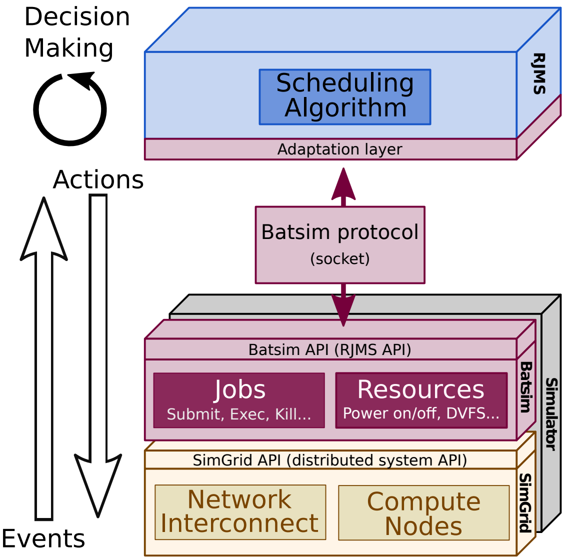
Other communities, similar goals

- NS-3 [@ns3]; [Flo06]
  - Holistic
  - Ease in vivo ↔ in silico
  - Care on implementation (validation, tests...)

Batsim overview



Batsim architecture



Jobs and profiles

Job — Scheduler point of view

- From user request
- Associated to one profile

Profile — Simulator point of view

Multiple models

- Fixed time → no interferences
- Parallel tasks → coarse-grained interferences
  - *Smooth* execution
- MPI traces → fine-grained interferences
  - Detailed execution
- Wrappers, composition...

Batsim validation

How to evaluate the accuracy of Batsim results?

Same scheduler **in vivo** and **in silico**

Deterministic simulation → must be representative

Problem 1

Very little information in traces — e.g. [@Feitelson]

- Resources failures?
- Jobs placements?
- Jobs inners? Parallelism? CPU/IO/network bound? ...

Problem 2

Workloads should be long with thousands of machines

→ huge ecological/financial costs ☹

Batsim evaluation experiment

	in vivo	in silico
Platform	Grid'5000 testbed cluster	SimGrid XML platform
	33 nodes under 1 switch	
Job classes	NAS Parallel Benchmarks Type: IS, FT and LU	Profile generated from real execution instrumentation
	Size: from 1 to 32 nodes Total: **47 different jobs classes**	
Workloads	Contains **800 jobs** randomly picked Inter arrival times: Weibull dist. ($shape = 2$, $scale = 15$) Job size: $2^{\lfloor u \rfloor}$, $u \sim$ Lognormal dist.($\mu = 0.25$, $\sigma = 0.5$) Total: **9 workloads** $\simeq$ **4h each**	
	2 runs/workload	1 deterministic simulation
Scheduler	Kamelot: scheduler from OAR (conservative backfilling)	
	Directly executed by OAR	Bataar: Batsim adaptor for OAR
Resources	10 000+ hour×cores	$\simeq$ 30s on a laptop

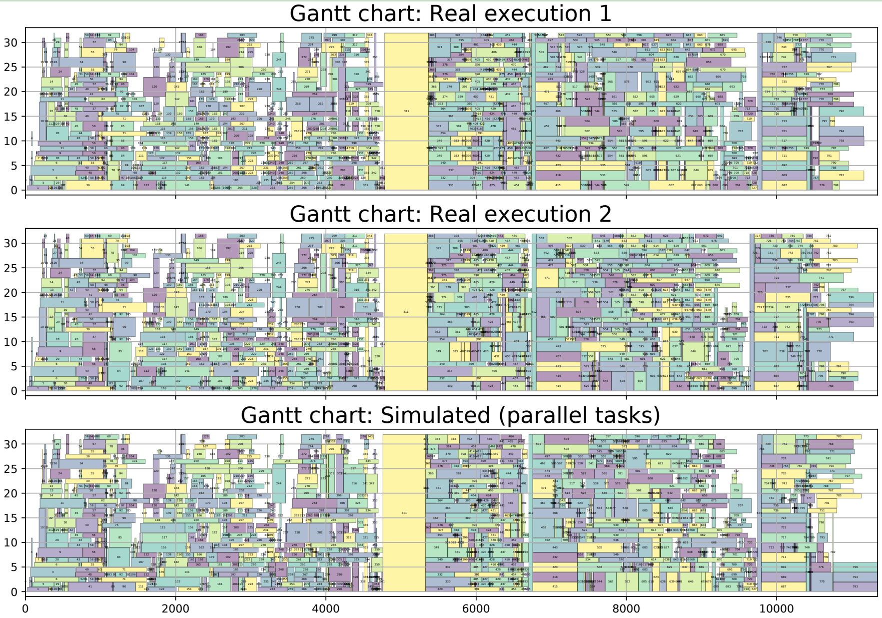
Financial cost estimations

	in vivo	**in silico**
Human resources	$\approx$ 4000 €	$\approx$ 4000 €
IT resources[1]	1 node/machine $1.9 \times node \times hour$ €	1 basic node $0.3 \times f(hour)$ €

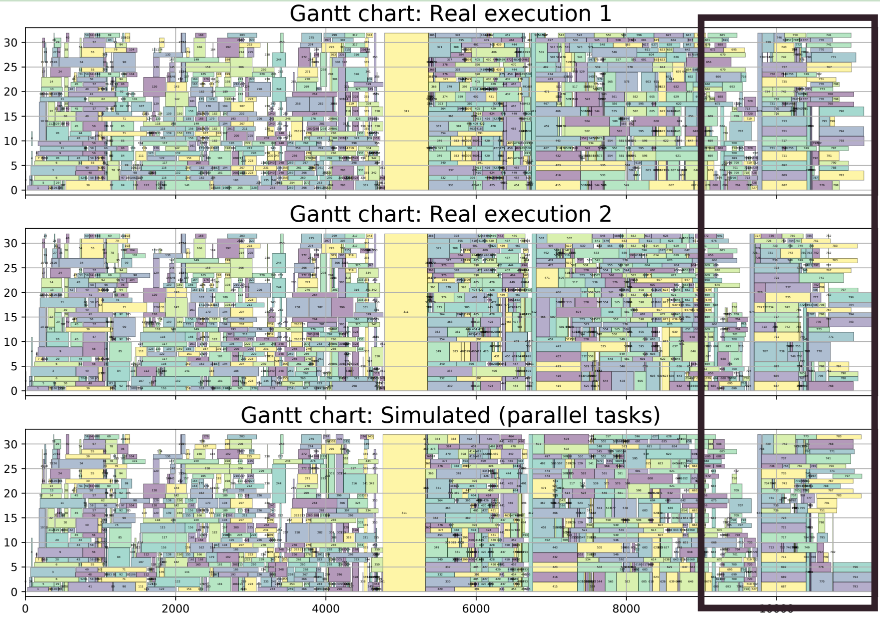
	in vivo	**in silico**
Batsim validation	$\approx$ 4500 €	$\approx$ gratis
Energy (next part)	$\approx$ 3.7 billion €	$\approx$ 9000 €

[1] From Amazon EC2 — December 2017

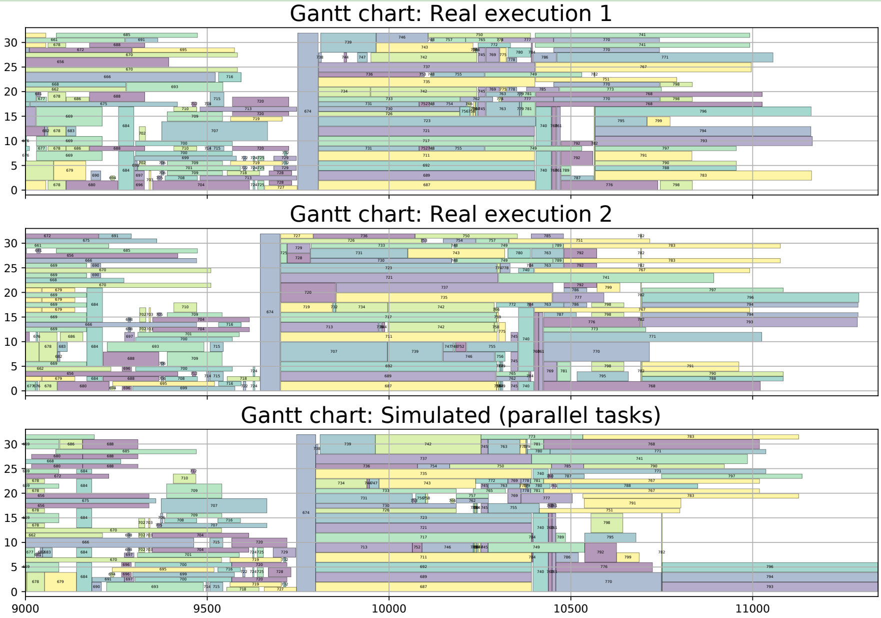
Gantt charts look similar...



Gantt charts look similar...



... but local differences exist



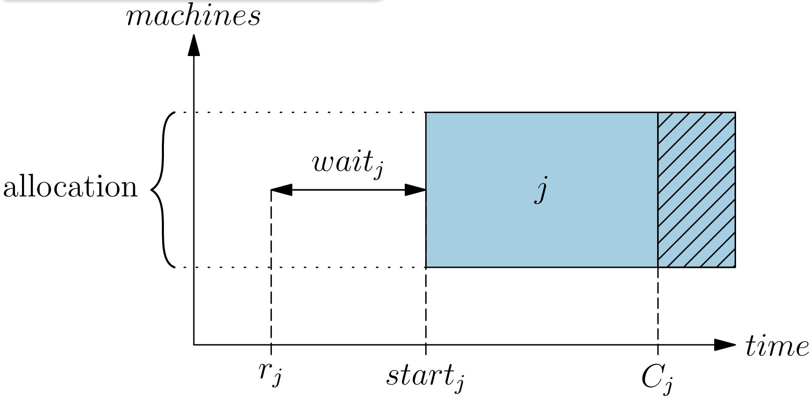
Metrics

Mean waiting time

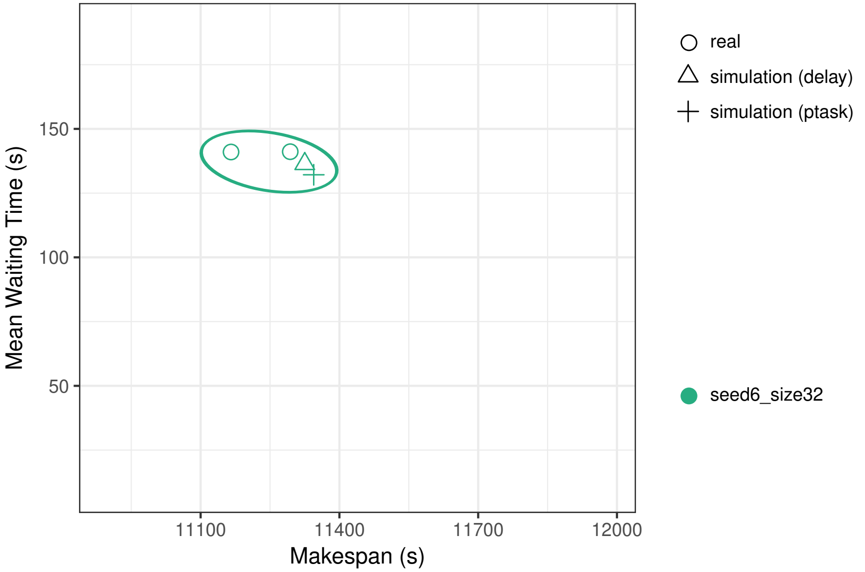
$$\frac{1}{|J|}\sum_j wait_j$$

Makespan (schedule duration)

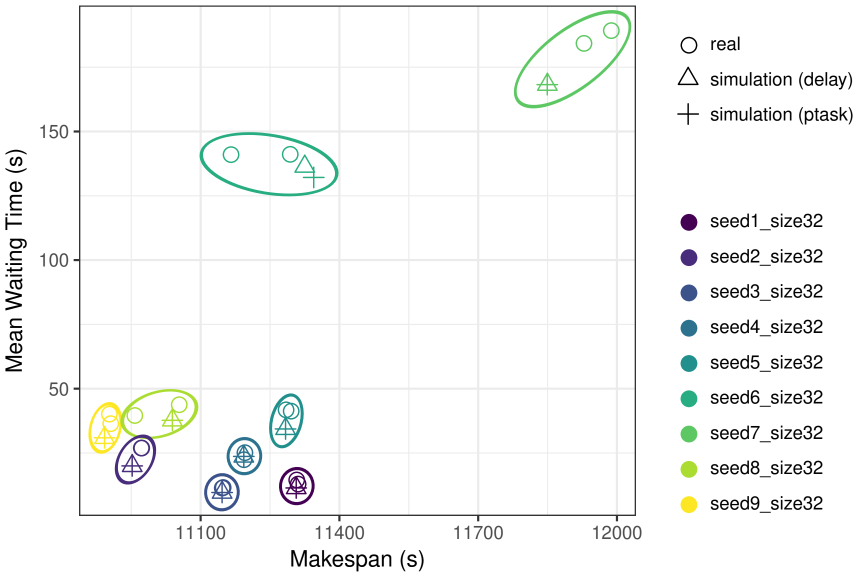
$$\max_j C_j - \min_j r_j$$



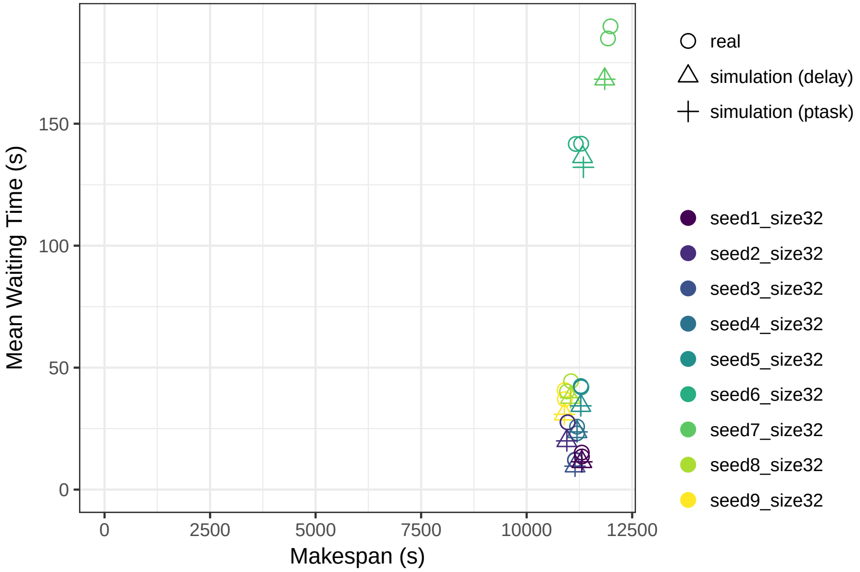
Results: Makespan, mean waiting time



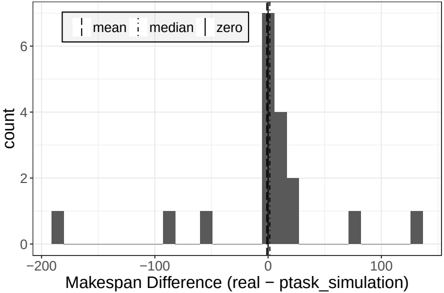
Results: Makespan, mean waiting time

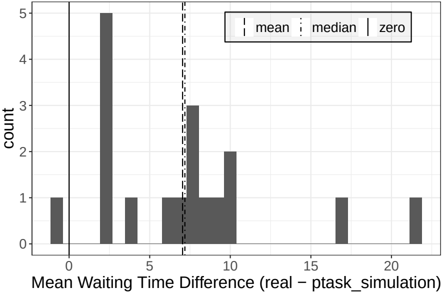


Results: Makespan, mean waiting time



Results: workload per workload differences





Experiment conclusion

Waiting times

- Bias — some underestimation ☹
- Not overfitting OAR's behaviour

Globally

- In this setup, no clear benefit from models 😐
- *diff* (real, simu) ≈ *diff* (*real*, *real*) → representative ☺

Outline

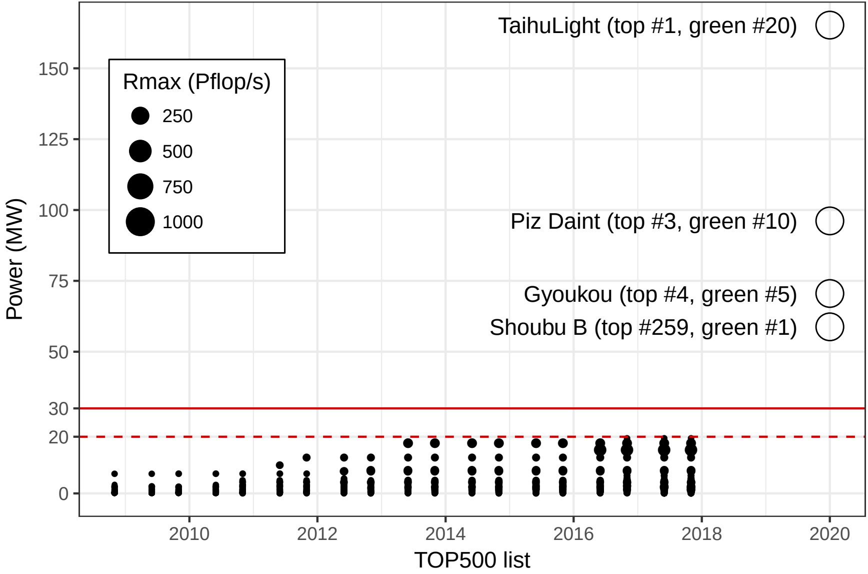
1 Introduction

2 Batsim

3 Energy/Performance trade-offs

4 Conclusion

Reminder: Power consumption (naive) projection



Avenues for saving energy

- Efficient hardware
- Efficient jobs
- Efficient resource management
  - **Unused machines → wasted energy**
  - **Better placement → less energy**
  - Electricity sources (local, renewable...)

Problems studied during the thesis

Execute a given workload on a given platform

- node shutdown

[CC'Grid 2017] (not in this talk)

Optimize performance under an energy budget

[in submission] (following slides)

Optimize energy and performance (trade-off)

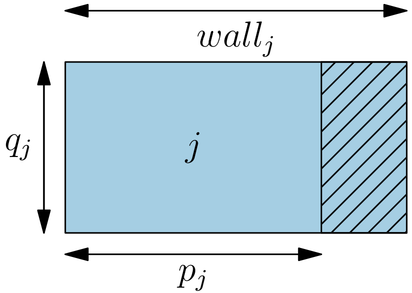
- How much energy can be saved?
- How is performance impacted?

Workload definition

$W = \{j_1, j_2, j_3, \ldots\}$. **Unknown** $|W|$

Job j definition:

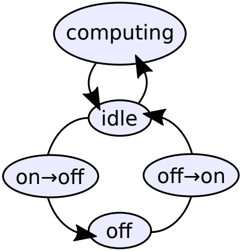
- Submission time r_j (release date). **Unknown** in advance
- Processing time p_j. **Unknown** in advance
- Requested time $wall_j \geq p_j$. **Known** at submission time
- Number of requested resources q_j. **Known** at submission time



Platform definition — Energy model

Platform: ordered set M of identical machines

- $t_{on \to off}$, switching OFF time (s)
- $t_{off \to on}$, switching ON time (s)
- $p_m(t)$, electrical consumption at time t (W)



$$p_M(t) = \sum_m \int_{min(s_j)}^{max(C_j)} p_m(t)\ dt$$

State	Power (W)
computing	p_{comp}
idle	p_{idle}
off	p_{off}
$on \to off$	$p_{on \to off}$
$off \to on$	$p_{off \to on}$

Hypotheses:

- $p_{off} \ll p_{idle} < p_{comp}$
- $p_{off} < p_{* \to *} \leq p_{comp}$

Algorithms: Overview

- Based on EASY backfilling [MF01]
- *Regular* events **and** every T seconds

Two main mechanisms

Opportunistic Shutdown

- Machine idle for $t \geq t_{idle}$ seconds $\rightarrow$ switched off

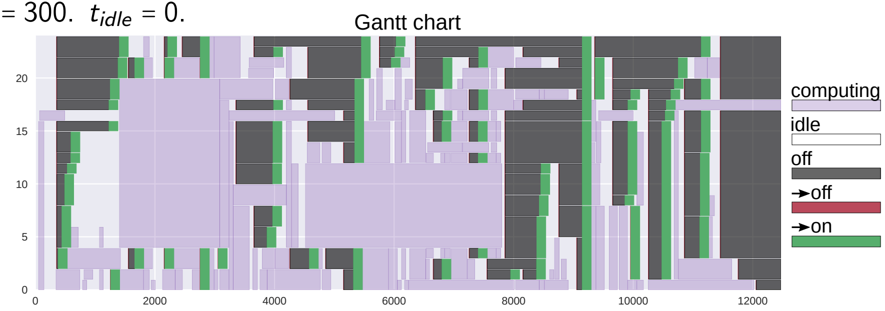
Adjusting the number of usable machines

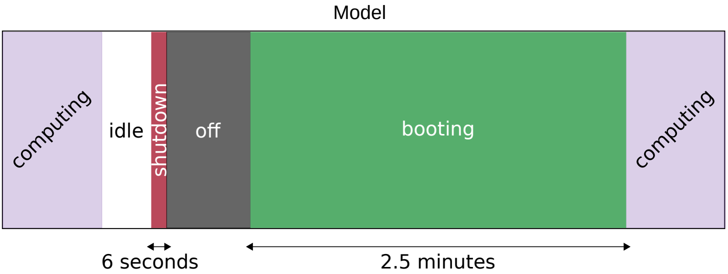
- Statically, no more than $\rho \cdot |M|$ machines
- Dynamically, depending on system *unresponsiveness*

If a job **requires** more machines, they will be switched-on.

Mechanism 1: Opportunistic Shutdown (Grid'5000)

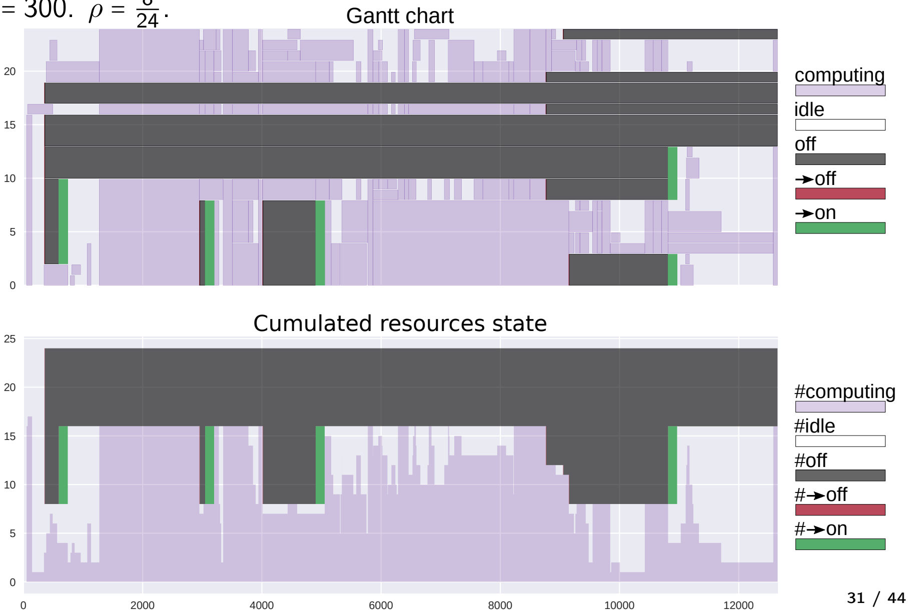
$T = 300$. $t_{idle} = 0$.





Mechanism 2.1: Proportional Shutdown

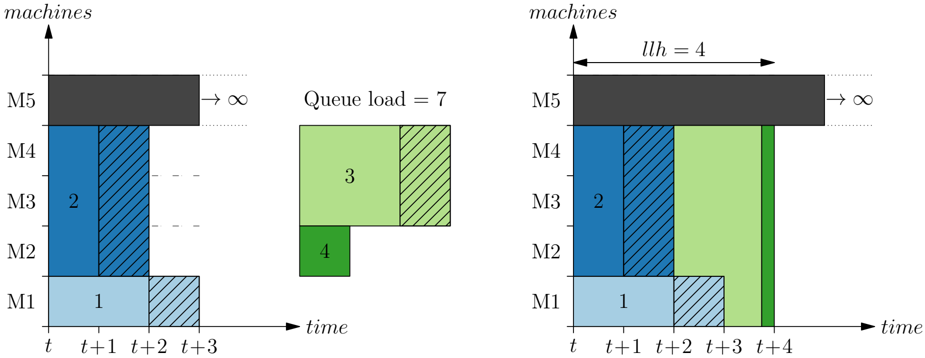
$T = 300$. $\rho = \frac{8}{24}$.



Estimate unresponsiveness? Liquid Load Horizon

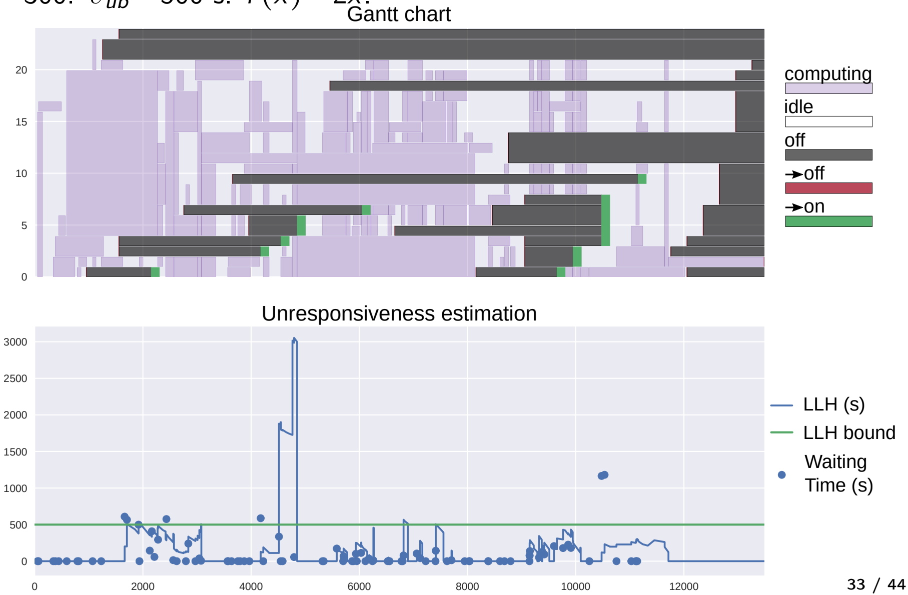
Required time to dump current load in the provisional schedule.

$$\text{load} = \sum_j q_j \times wall_j$$



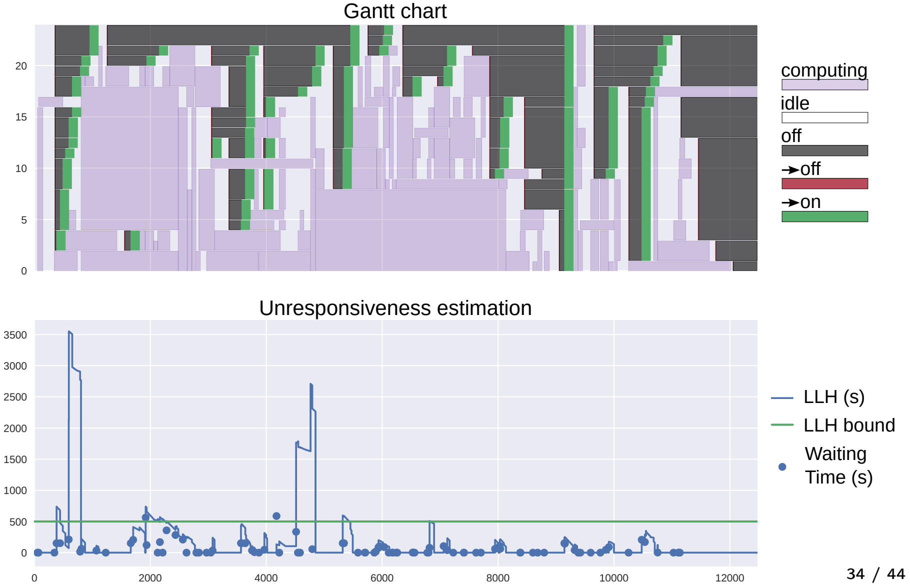
Mechanism 2.2: Inertial Shutdown

$T = 300$. $\bar{\upsilon}_{ub} = 500$ s. $f(x) = 2x$.



Combination: Inertial + Opportunistic

$T = 300$. $\bar{\upsilon}_{ub} = 500$ s. $f(x) = 2x$. $t_{idle} = 0$.



Experimental Setup

Simulation

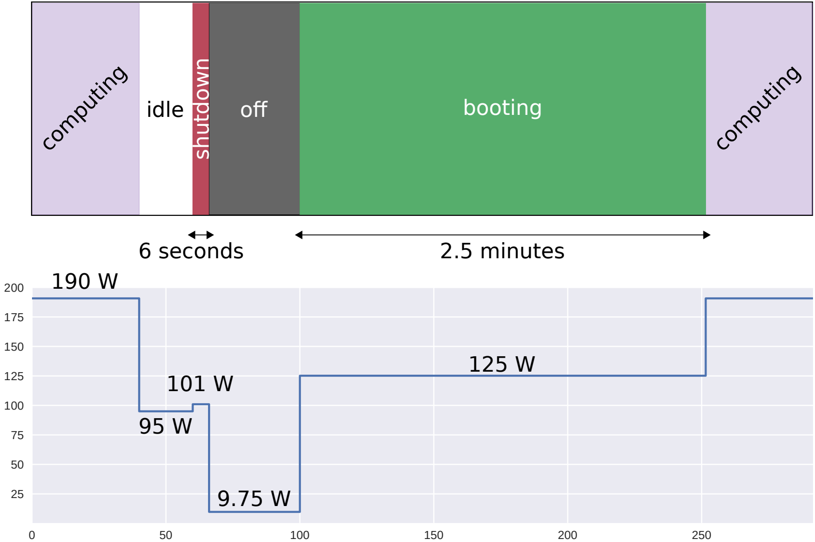
- Batsim
- Algorithms in Batsched (C++)
  - EASY
  - Opportunistic Shutdown (weak/strong aggressiveness)
  - Proportional Shutdown (...)
  - Inertial Shutdown (...)

Workloads

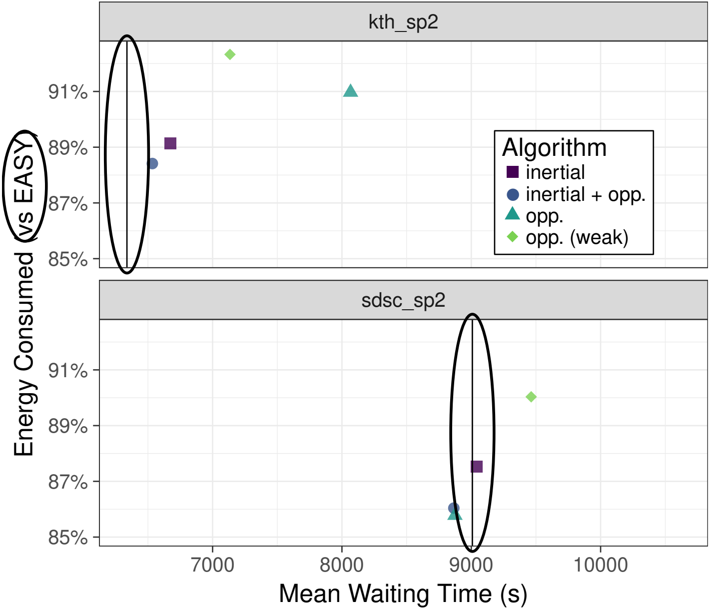
- KTH SP2, SDSC SP2 [@Feitelson]
- Kept valid jobs ($wall_j > p_j$)
- 11, 24 months → assess robustness
- Periodic utilization → room to save energy

Experimental Setup (platform)

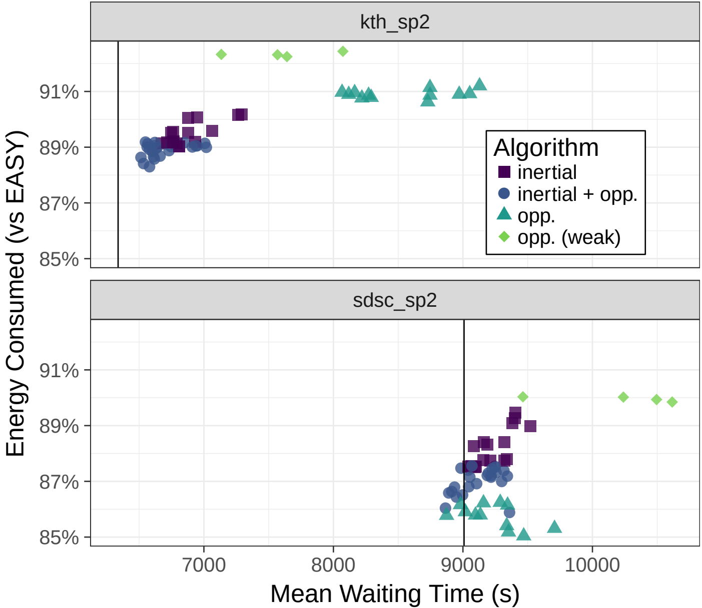
Homogeneous. $|M| \in \{100, 128\}$. G5K Taurus [Dut+16].



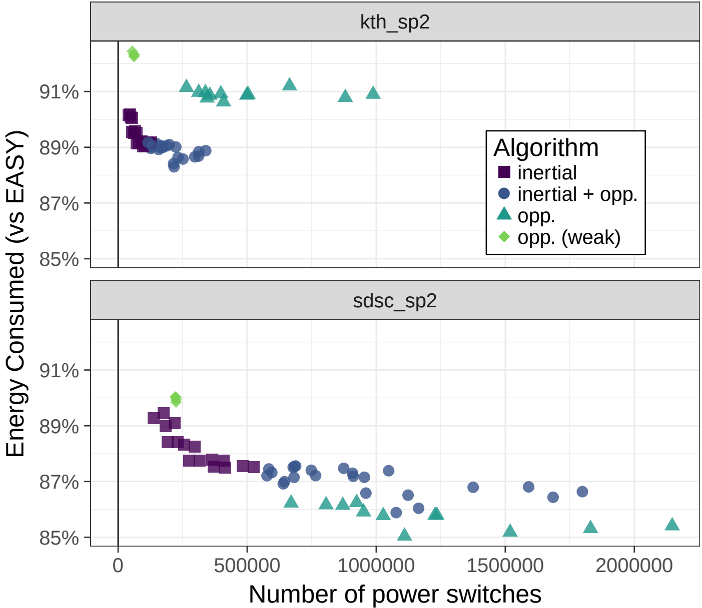
Most interesting trade-offs — Performance



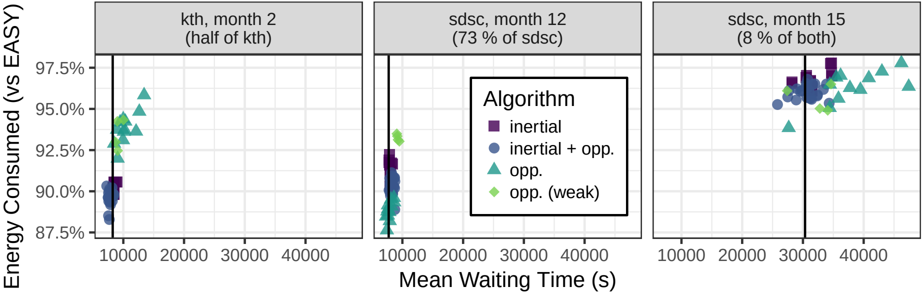
Most interesting trade-offs — Performance



Most interesting trade-offs — #switches



Finer-grain analysis — Common months



- 2 workloads → 38 months
- Opportunistic can take bad decisions (heavy load)
- Inertial is more robust

Brief comparison

	Opp.	Prop.	Inertial	Inertial+Opp
Energy	🙂	☹️	🙂	🙂
Performances	😐	☹️	🙂	🙂
#Switches	☹️	🙂	🙂	☹️
Predictability	😐	🙂	🙂	😐
Implementation cost	🙂	😐	😐	☹️

Outline

1 Introduction

2 Batsim

3 Energy/Performance trade-offs

4 Conclusion

A word on open science

FOSS projects

- Batsim — simulator
- Batsched — algorithms
- Evalys — visualization

Experiments (reproducibility, data, figures)

- Batsim validation
- Energy budget
- Energy trade-offs

Contributions

Evaluation framework for resource management [JSSPP'16]

- Many use cases (platforms, algorithms, phenomena...)
- Separation of Concerns (SimGrid, loose coupling)
- Multiple levels of realism
- Partly validated, tested

Algorithms to save energy

- max(*perf*) under *energy* constraint [CCGRID'17]
- opt(*perf*, *energy*)[in publication]
  - More insight about shutdown techniques
  - Interesting trade-offs

Perspectives

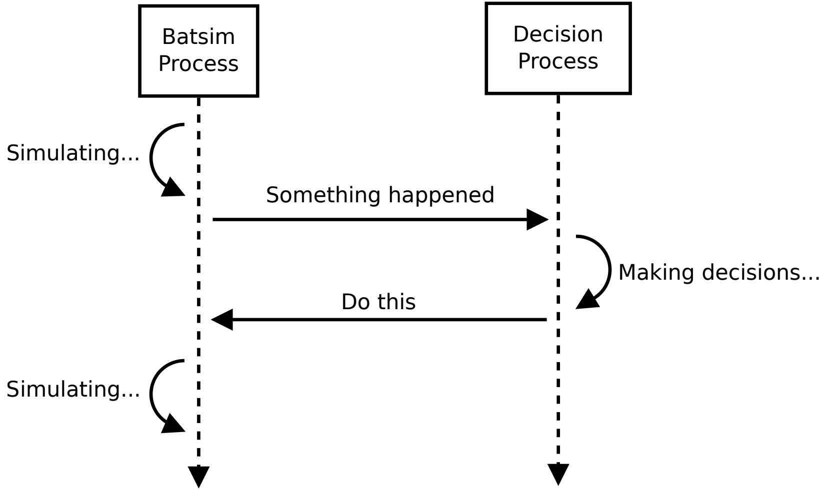
Evaluation of resource management

- Batsim: Features, optimization
- Job modeling and validation — data needed

Saving energy

- Placement — data needed
- Prediction (p_j, r_{j+1}, q_{j+1}...)
- Other bases than EASY

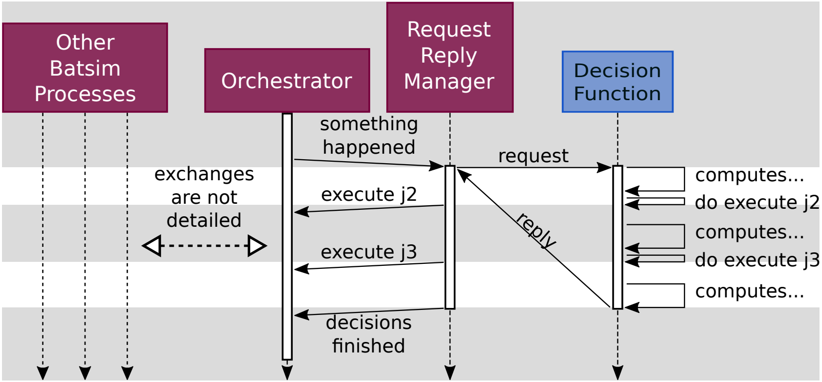
Network protocol — Main idea



Details:

https://github.com/oar-team/batsim/blob/master/doc/proto_description.md

Time Dilation — On-the-fly actions injection



Other metrics [Fei01]

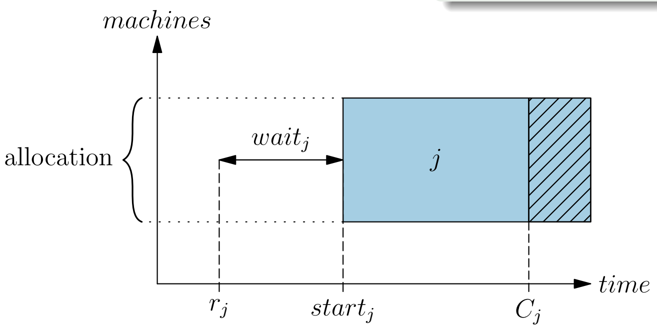
Mean Slowdown

$$\frac{1}{|J|}\sum_j \frac{C_j - r_j}{p_j}$$

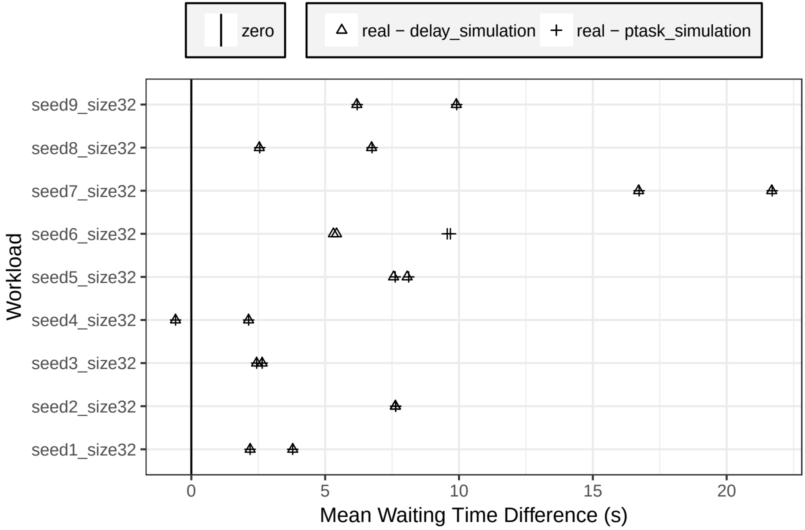
Mean Bounded Slowdown

$$\frac{1}{|J|}\sum_j max\left(\frac{C_j - r_j}{\max(p_j, \tau)}, 1\right)$$

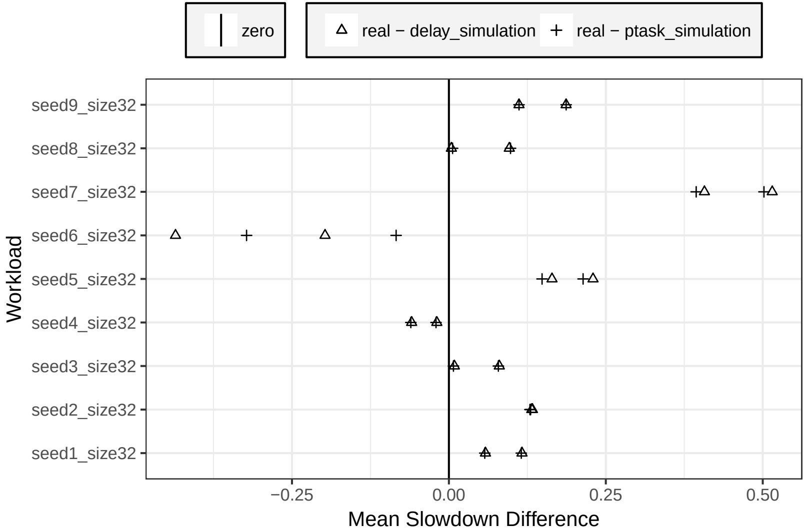
τ : processing time threshold



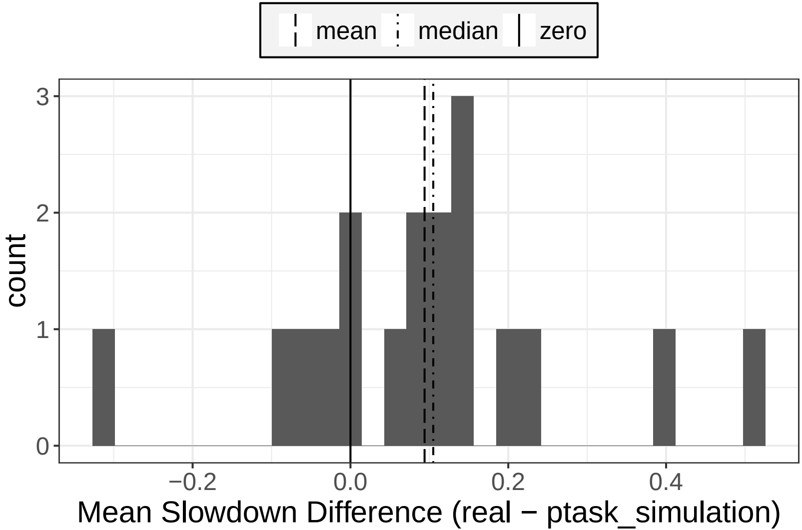
Mean Waiting Time (scatterplot)



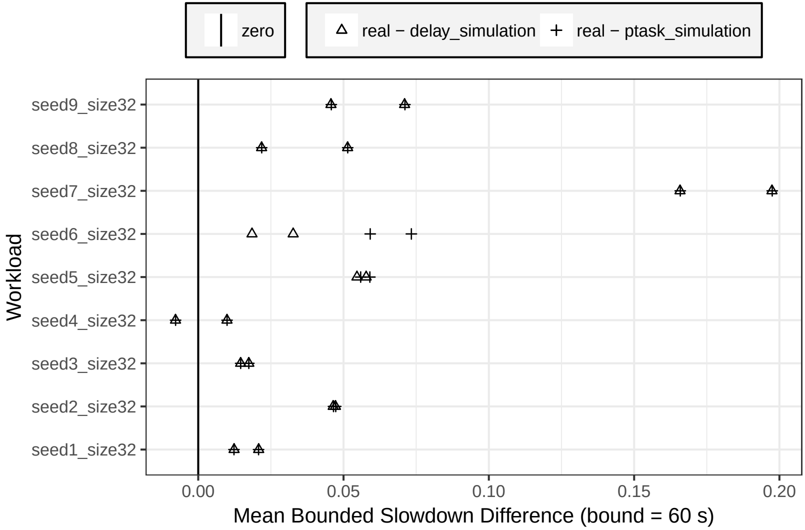
Mean Slowdown (scatterplot)



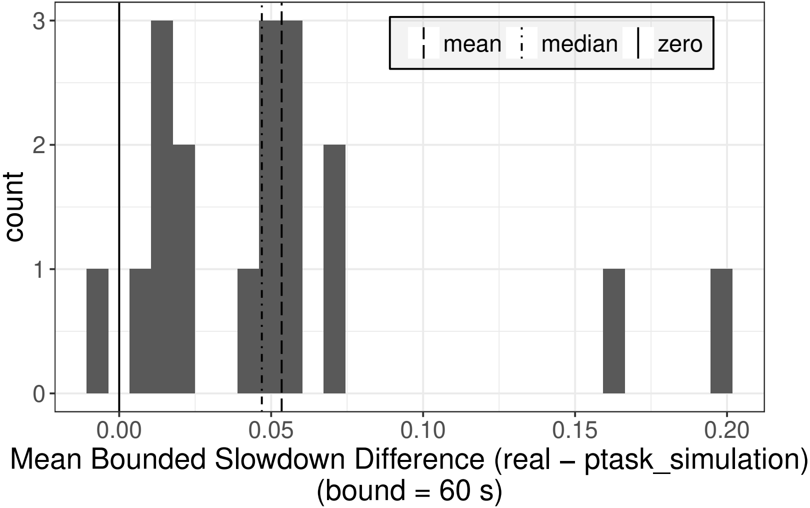
Mean Slowdown (histogram)



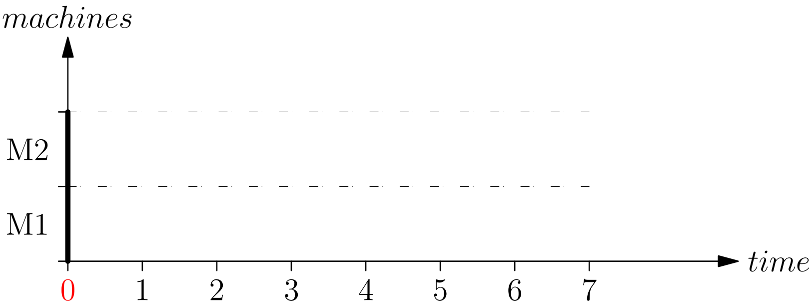
Mean Bounded Slowdown (scatterplot)



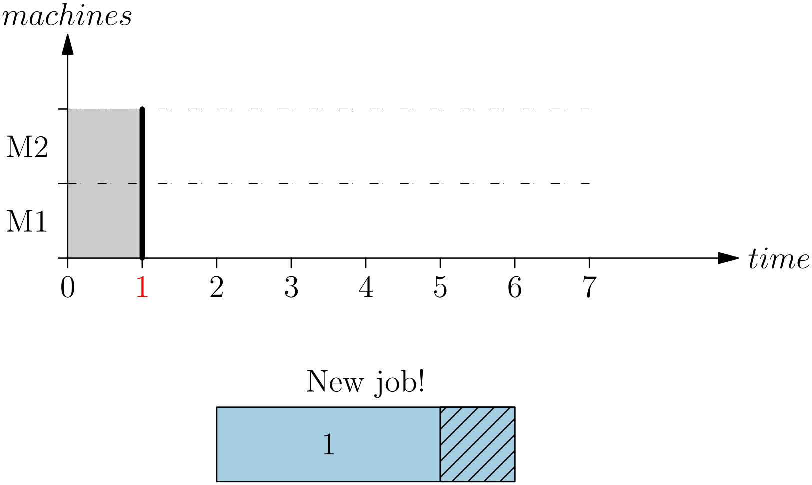
Mean Bounded Slowdown (histogram)



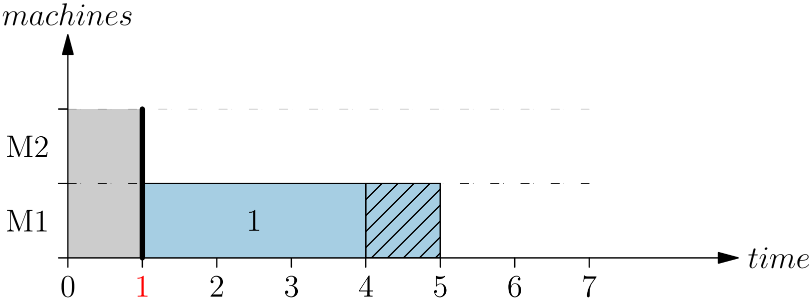
Online scheduling with backfilling



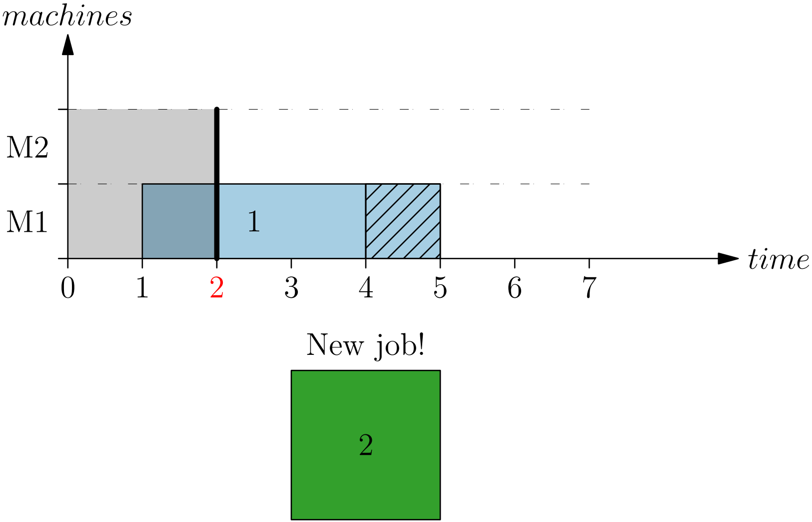
Online scheduling with backfilling



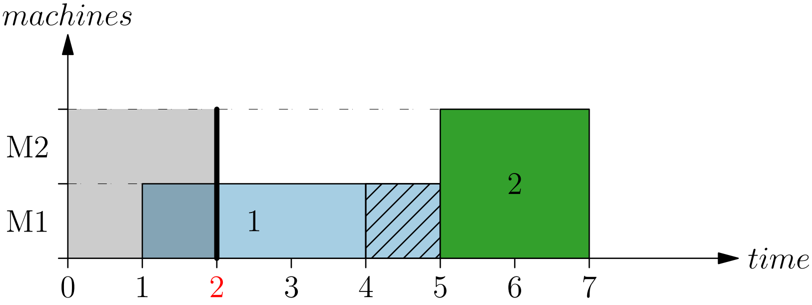
Online scheduling with backfilling



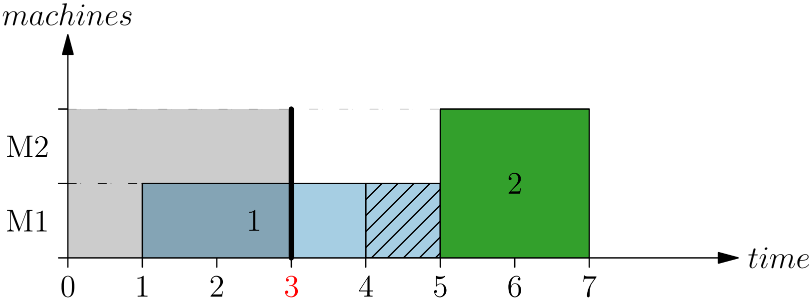
Online scheduling with backfilling



Online scheduling with backfilling

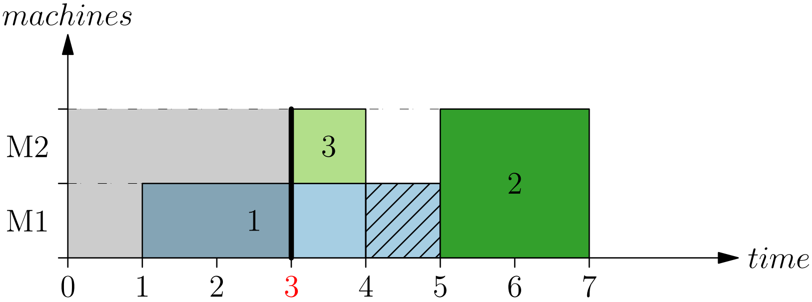


Online scheduling with backfilling

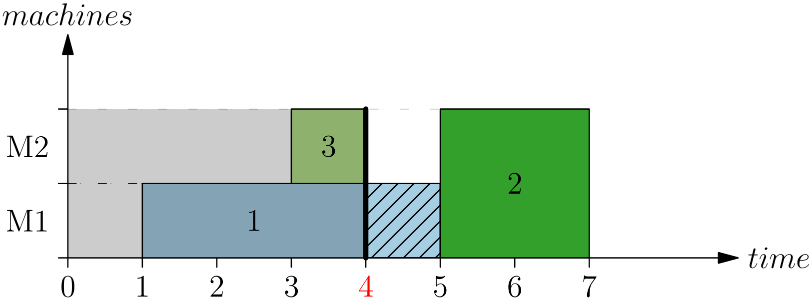


New job!

Online scheduling with backfilling

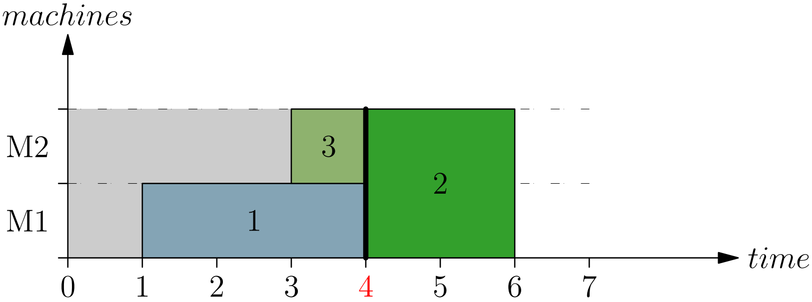


Online scheduling with backfilling

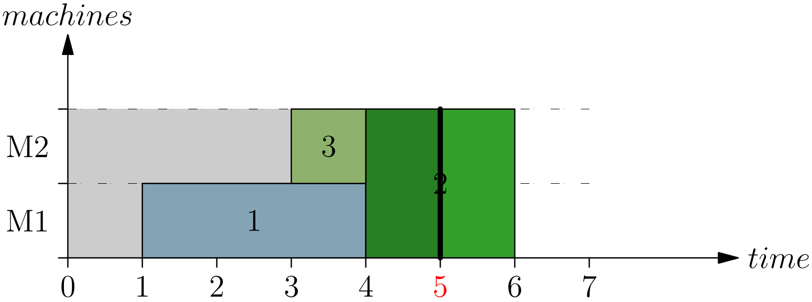


Jobs 1 and 3 finished

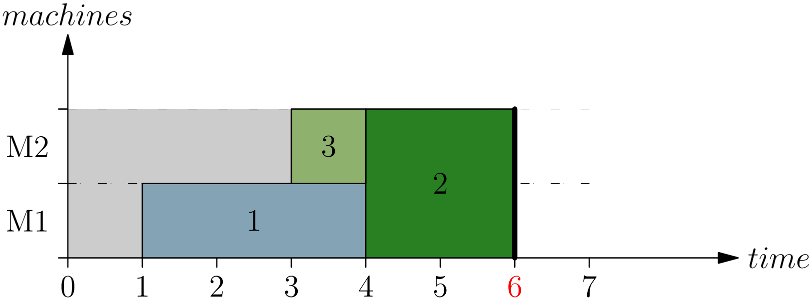
Online scheduling with backfilling



Online scheduling with backfilling



Online scheduling with backfilling



Experimental Setup (exploration space)

Shared by all algorithms	
Workloads	KTH_SP2, SDSC_SP2
Shared by Proportional and Inertial	
T (s)	60, 120, 300, 600
t_{idle} (s)	0, 30, 60, 600, 6000, $+\infty$
Make run decisions on period	true, false
Proportional-specific	
ρ	1.00, 0.95, 0.90, 0.85
Inertial-specific	
$f(n)$	$n+1$, $n \times 2$
$\overline{\upsilon}_{ub}$ (s)	$1 \cdot 10^4$, $1 \cdot 10^5$, $2 \cdot 10^5$
Allow future switches	true, false

All these parameters combinations have been tested

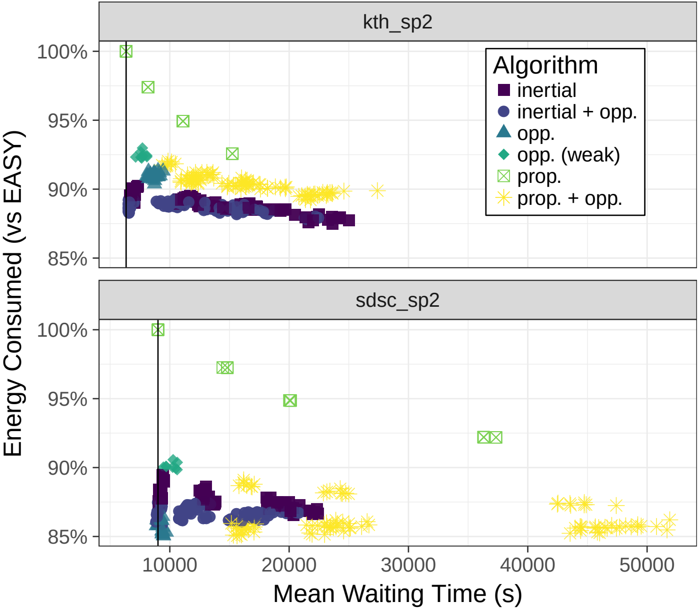
Algorithm Nomenclature

Opportunistic shutdown aggressiveness:

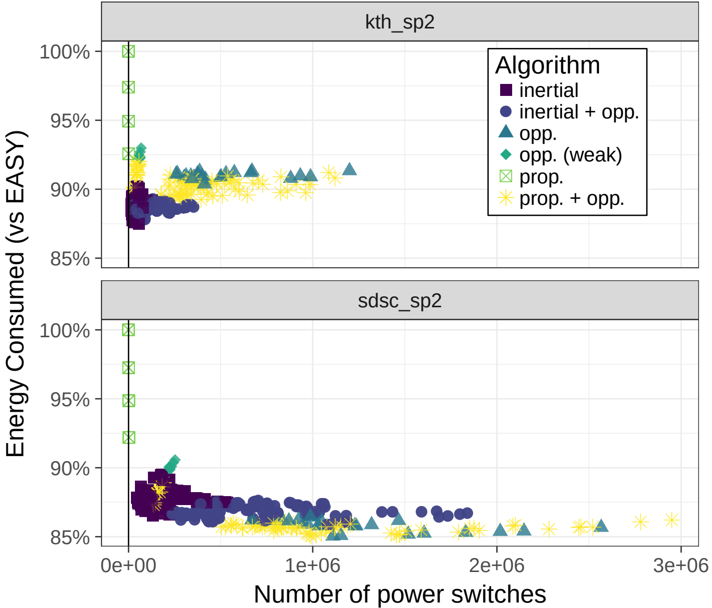
- strong: $t_{idle} \in \{\ 0,\ 30,\ 60,\ 600\ \}$
- weak: $t_{idle} \in \{\ 6000,\ +\infty\ \}$

Name	Opp.?	Proportional?	Inertial?
EASY			
opp. (weak)	weak		
prop.	weak	✓	
inertial	weak		✓
opp.	strong		
prop. + opp.	strong	✓	
inertial + opp.	strong		✓

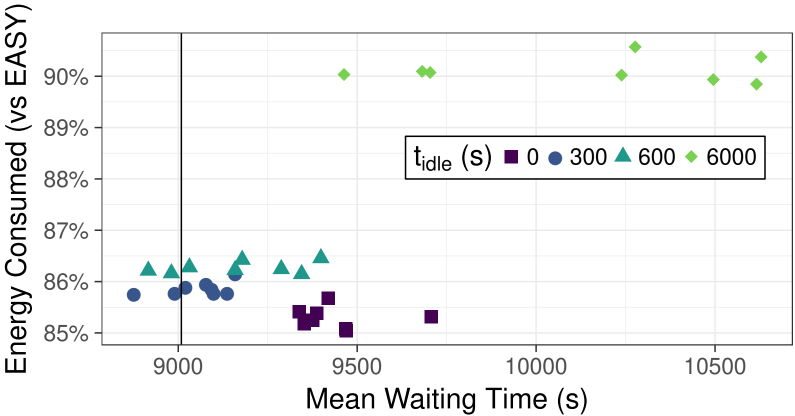
All trade-offs — Performance



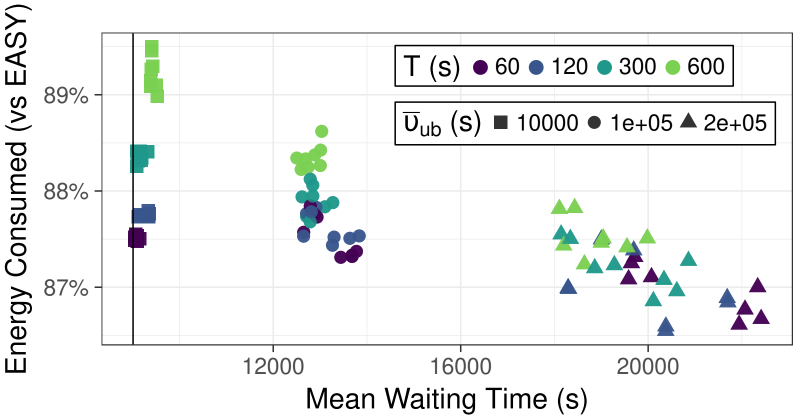
Most interesting trade-offs — #switches



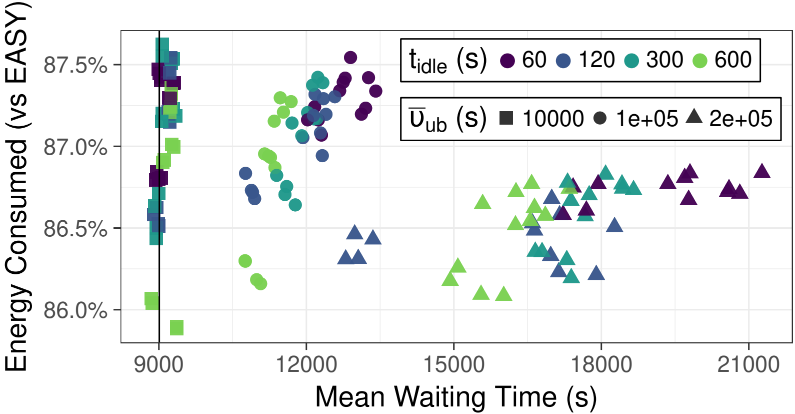
Opportunistic Shutdown — Impact of t_{idle} (SDSC)



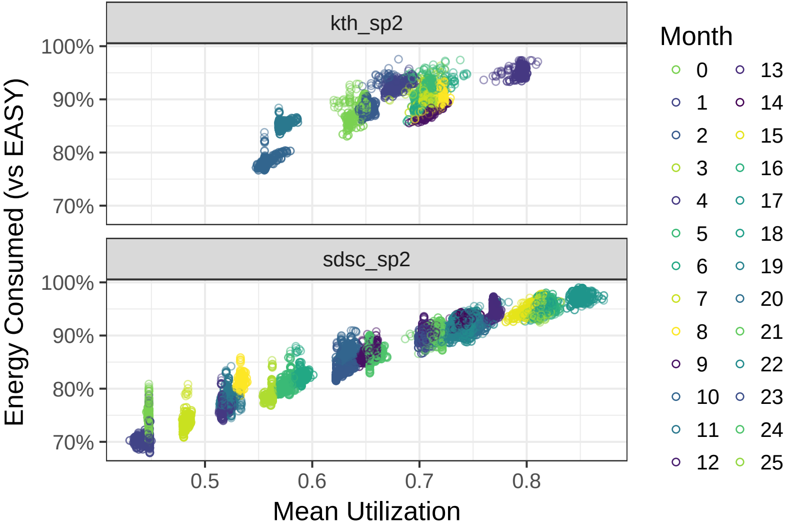
Inertial Shutdown — Impact of $\bar{\upsilon}_{ub}$ and T (SDSC)



Inertial+Opportunistic — Impact of $\bar{\upsilon}_{ub}$ and t_{idle} (SDSC)



Finer-grain analysis — Energy and utilization



Some related work

Theoretical:

- DVFS/shutdown models and algo [Alb10]
- Markov Chains [HK12]

Practical:

- Shutdown, reservations, predictions [OLG08]
- Shutdown, physical constraints [Ben+17]
- DVFS/shutdown in SLURM [GGT15]
- Applications [Eti+12]

Overprovisioning:

- Max throughput, power budget [Sar+14]

References I

Dror Feitelson. *Parallel Workload Archive*. http://www.cs.huji.ac.il/labs/parallel/workload/. 2017. (Visited on 01/17/2017).

The ns-3 network simulator. url: https://www.nsnam.org (visited on 09/30/2017).

Susanne Albers. "Energy-efficient algorithms". In: *Communications of the ACM* 53.5 (2010), pp. 86–96.

Anne Benoit et al. "Reducing the energy consumption of large scale computing systems through combined shutdown policies with multiple constraints". In: *International Journal of High Performance Computing Applications* (2017). url: https://hal.inria.fr/hal-01557025.

References II

Pierre-François Dutot et al. "Towards Energy Budget Control in HPC". In: *Cluster, Cloud and Grid Computing (CCGrid), 2016 16th IEEE/ACM International Symposium on*. IEEE. 2016.

Maja Etinski et al. "Understanding the future of energy-performance trade-off via DVFS in HPC environments". In: *Journal of Parallel and Distributed Computing* 72.4 (2012), pp. 579–590.

Dror G Feitelson. "Metrics for parallel job scheduling and their convergence". In: *Workshop on Job Scheduling Strategies for Parallel Processing*. Springer. 2001, pp. 188–205.

Sally Floyd. "Maintaining a critical attitude towards simulation results (invited talk)". In: (2006).

References III

Jean-Sébastien Gay and Yves Caniou. *Simbatch: an API for simulating and predicting the performance of parallel resources and batch systems*. Research Report RR-6040. INRIA, 2006, p. 15. url: https://hal.inria.fr/inria-00115880.

Yiannis Georgiou, David Glesser, and Denis Trystram. "Adaptive Resource and Job Management for Limited Power Consumption". In: *IEEE International Parallel and Distributed Processing Symposium Workshop, IPDPS 2015, Hyderabad, India, May 25-29*. Hyderabad, India, 2015, pp. 863–870. doi: 10.1109/IPDPSW.2015.118. url: https://hal.archives-ouvertes.fr/hal-01230292.

References IV

Matthias Herlich and Holger Karl. "Average and Competitive Analysis of Latency and Power Consumption of a Queuing System with a Sleep Mode". In: *Proceedings of the 3rd International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet*. e-Energy '12. Madrid, Spain: ACM, 2012, 14:1–14:10. isbn: 978-1-4503-1055-0. doi: 10.1145/2208828.2208842. url: http://doi.acm.org/10.1145/2208828.2208842.

References V

Dalibor Klusáček and Hana Rudová. “Alea 2: job scheduling simulator”. In: *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). 2010, p. 61.

Ahuva W. Mu’alem and Dror G. Feitelson. “Utilization, predictability, workloads, and user runtime estimates in scheduling the IBM SP2 with backfilling”. In: *Parallel and Distributed Systems, IEEE Transactions on* 12.6 (2001), pp. 529–543.

References VI

Paul R Muessig, Dennis R Laack, and John J Wrobleski. *An integrated approach to evaluating simulation credibility*. Tech. rep. NAVAL AIR WARFARE CENTER WEAPONS DIV CHINA LAKE CA, 2001.

Anne-Cécile Orgerie, Laurent Lefèvre, and Jean-Patrick Gelas. "Save Watts in your Grid: Green Strategies for Energy-Aware Framework in Large Scale Distributed Systems". In: *IEEE International Conference on Parallel and Distributed Systems (ICPADS)*. Melbourne, Australia, Dec. 2008, pp. 171–178. doi: 10.1109/ICPADS.2008.97. url: https://hal.inria.fr/ensl-00474726.

References VII

Fco Javier Ridruejo Perez and José Miguel-Alonso. "INSEE: An interconnection network simulation and evaluation environment". In: *Euro-Par 2005 Parallel Processing*. Springer, 2005, pp. 1014–1023.

Jose A Pascual, Jose Miguel-Alonso, and Jose A Lozano. "Locality-aware policies to improve job scheduling on 3D tori". In: *The Journal of Supercomputing* 71.3 (2015), pp. 966–994.

Osman Sarood et al. "Maximizing throughput of overprovisioned hpc data centers under a strict power budget". In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE Press. 2014, pp. 807–818.