CNRS - INP - UT3 - UT1 - UT2J
Institut de Recherche en Informatique de Toulouse

DVFS-aware Performance and Energy models, and how to use them in a realistic way

Georges Da Costa
New Challenges in Scheduling Theory
May, 16, 2024

georges.da-costa@irit.fr

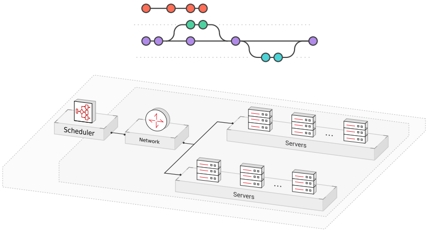
Acknowledgments

- GIS neOCampus of Université Toulouse III Paul Sabatier
- This work was carried out within the MaaS action of the VILAGIL project, a project co-financed by Toulouse Métropole and France 2030 as part of the Territoires d'innovation program operated by the Banque des territoires

Plan

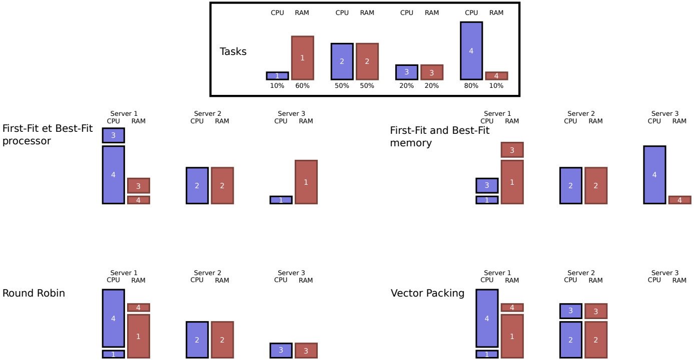
The HPC datacenter scheduling problem

- Typical optimization problem
  - Some tasks, some servers
  - When and where run the tasks
- Several methods
  - NP-Hard problem
  - Large amount of research on heuristics with low competitive ratios



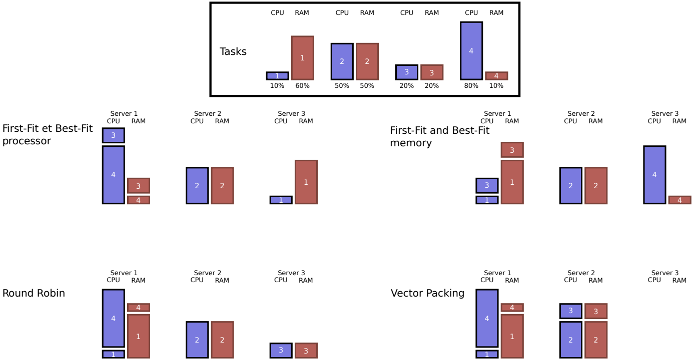
Classical greedy allocation algorithms

- Characteristics
  - Memory
  - Processor
- Sort tasks
- Sort servers
- No coming back on previous decisions

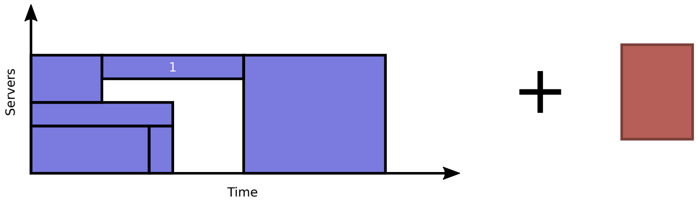


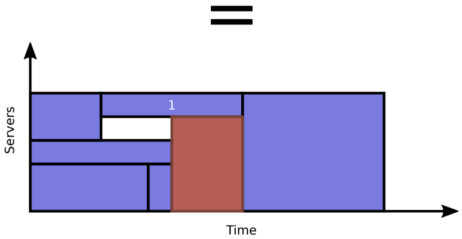
Classical greedy allocation algorithms

- Only maximum and constant values
- O/S assumed as without impact
- Assumed as identical servers



Classical greedy scheduling algorithm

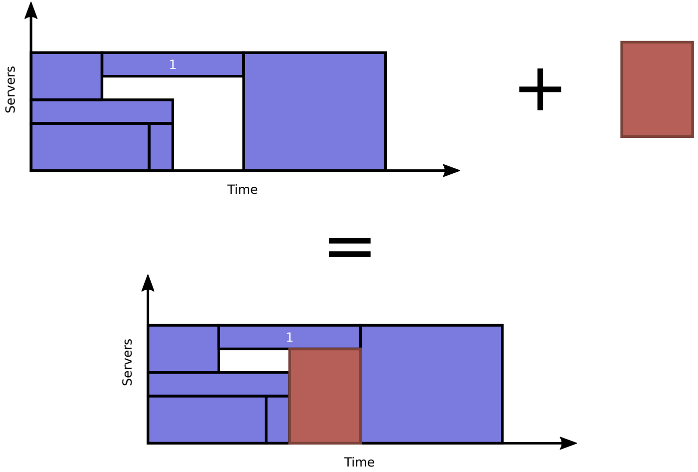




- Characteristics
  - Duration
  - Resources
- Sort tasks
- Sort places

Classical greedy scheduling algorithm

- Only maximum duration (walltime)
- Middleware assumed as without impact
  - Start/stop time



Classical approach

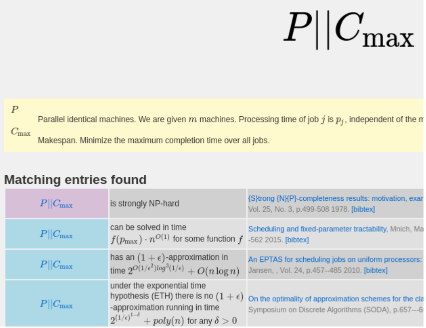
- Full knowledge
  - Tasks: *arrival_time*, *duration*, *resources*
  - Hosts: *computational_speed*
  - Constraints: *deadline*
  - Objectives: ...

Classical approach

- Full knowledge
  - Tasks: *arrival_time*, *duration*, *resources*
  - Hosts: *computational_speed*
  - Constraints: *deadline*
  - Objectives: ...
- Oracle- or distribution-based studies
  - Information are not known in advance

Classical approach

- Full knowledge
  - Tasks: *arrival_time*, *duration*, *resources*
  - Hosts: *computational_speed*
  - Constraints: *deadline*
  - Objectives: ...
- Oracle- or distribution-based studies
  - Information are not known in advance
- Scheduling Zoo
  `http://schedulingzoo.lip6.fr/`



$P||C_{\max}$

P Parallel identical machines. We are given m machines. Processing time of job j is p_j, independent of the m

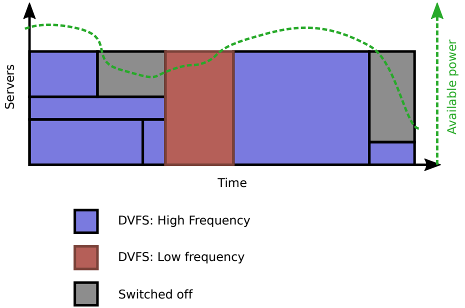
$C_{\max}$ Makespan. Minimize the maximum completion time over all jobs.

Matching entries found

$P\|\|C_{\max}$	is strongly NP-hard	{S}trong {N}{P}-completeness results: motivation, exar Vol. 25, No. 3, p.499-508 1978. [bibtex]
$P\|\|C_{\max}$	can be solved in time $f(p_{\max}) \cdot n^{O(1)}$ for some function f	Scheduling and fixed-parameter tractability, Mnich, Ma -562 2015. [bibtex]
$P\|\|C_{\max}$	has an $(1+\epsilon)$-approximation in time $2^{O(1/\epsilon^2)log^3(1/\epsilon)} + O(n \log n)$	An EPTAS for scheduling jobs on uniform processors: Jansen, , Vol. 24, p.457--485 2010. [bibtex]
$P\|\|C_{\max}$	under the exponential time hypothesis (ETH) there is no $(1+\epsilon)$-approximation running in time $2^{(1/\epsilon)^{1-\delta}} + poly(n)$ for any $\delta > 0$	On the optimality of approximation schemes for the cla Symposium on Discrete Algorithms (SODA), p.657--6

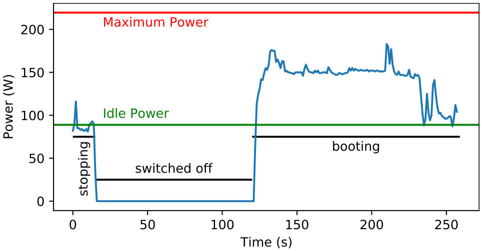
Zoom on energy-efficient scheduling of HPC Datacenters

- Leverages
  - On/Off
  - DVFS (Dynamic voltage and frequency scaling)
  - Choice of server (only for homogeneous system)
- Objectives
  - Power caping
  - Performance
  - Energy
- Each leverage and objective needs its model



Classical models

- On/Off
  - $T_{on}, T_{off}, E_{on}, E_{off}$
- DVFS
  - $Power = P_{static} + C \times Usage \times Freq \times Volt^2$
  - $Time = Time_{Freq_{max}} \times \frac{Freq_{max}}{Freq}$
  - $Energy = Power \times Time$



How to obtain the DVFS model

Dynamic electric power consumed by a CMOS component:

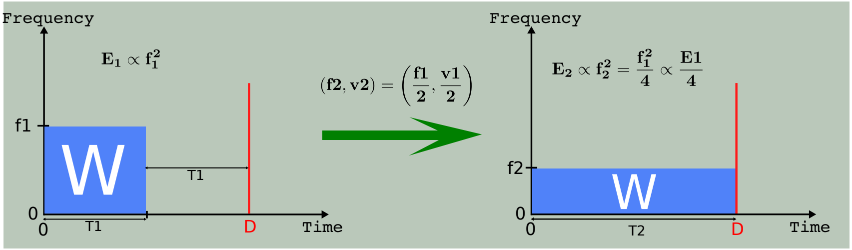
$P_{cmos} = C_{eff} \times V^2 \times f$

with, C_{eff} the effective capacitance *, V the voltage and f the frequency

* physical quantity: capacity of a component to resist to the change of voltage between its pins

Energy consumed for each tasks:

$$\underbrace{E = P \times T \propto T \times V^3 \text{ , with } V \propto f \text{ and } T \propto 1/F\text{, then } E \propto f^2}$$



Realistic models are complex

Electrical power models for a single server:

- Classical : linear (error E∼10-15%)

$$Power = P_{min} + Load \times (P_{max} - P_{min})$$

Realistic models are complex

Electrical power models for a single server:

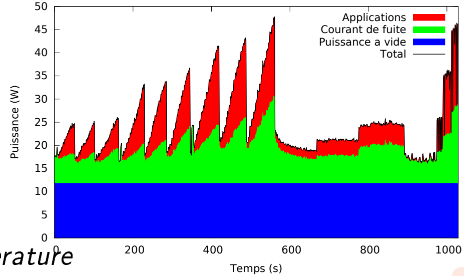
- Classical : linear (error E∼10-15%)
- Finer : Processor voltage/frequency (E∼5-9%)

$$Power = P_{min} + Load \times \alpha Voltage^2 Frequency$$

Realistic models are complex

Electrical power models for a single server:

- Classical : linear (error E∼10-15%)
- Finer : Processor voltage/frequency (E∼5-9%)
- Even finer: Processor temperature (E∼4-7%)



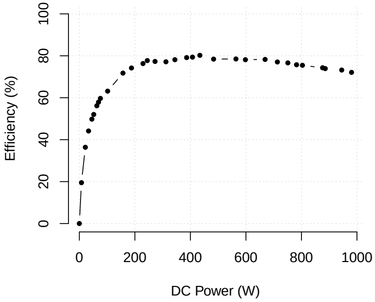
$$Power = P_{min} + Load \times \alpha Voltage^2 Frequency + \lambda Temperature$$

Realistic models are complex

Electrical power models for a single server:

- Classical : linear (error E∼10-15%)
- Finer : Processor voltage/frequency (E∼5-9%)
- Even finer: Processor temperature (E∼4-7%)
- Do not forget about bias: **power supply unit**, cooling, ... E∼2-3%

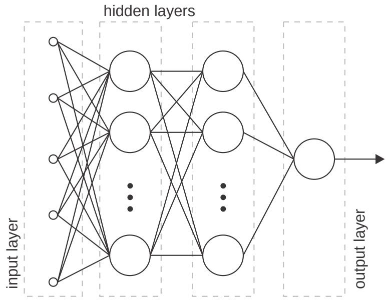
$$Power_{DC} = \omega_0 + \omega_1 Power_{AC} + \omega_2 Power_{AC}^3$$



Realistic models are complex

Electrical power models for a single server:

- Classical : linear (error E∼10-15%)
- Finer : Processor voltage/frequency (E∼5-9%)
- Even finer: Processor temperature (E∼4-7%)
- Do not forget about bias: **power supply unit**, cooling, ... E∼2-3%
- Learning methods (neural networks, E∼2%)*



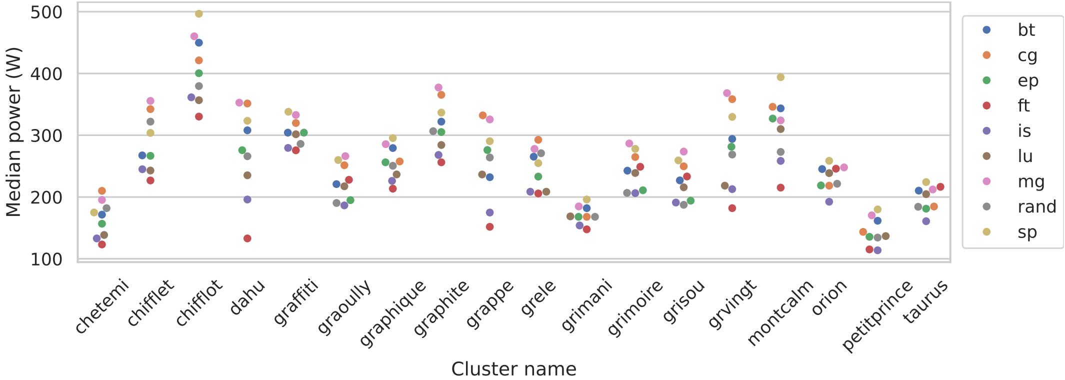
*Da Costa et al., *Effectiveness of neural networks for power modeling for Cloud and HPC: It's worth it!*, Transactions on Modeling and Performance Evaluation of Computing Systems journal, 2020, 10.1145/3388322

Hidden hypothesis

- All applications are black boxes with the same internal behavior
- FLOPS of a server allows to evaluate the duration of an application
- Executing two times the same application will result on the same behavior

Plan

Application power ranking depends on the hardware

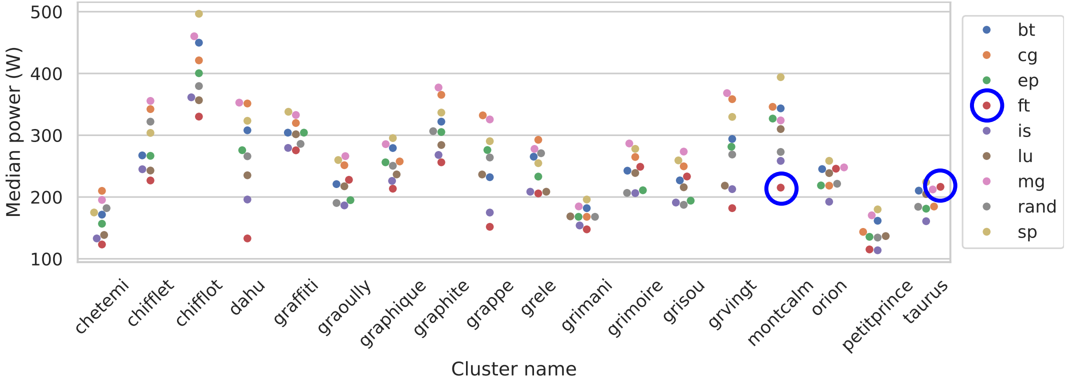


Execution at maximum frequencies for each cluster.

Application power ranking depends on the hardware

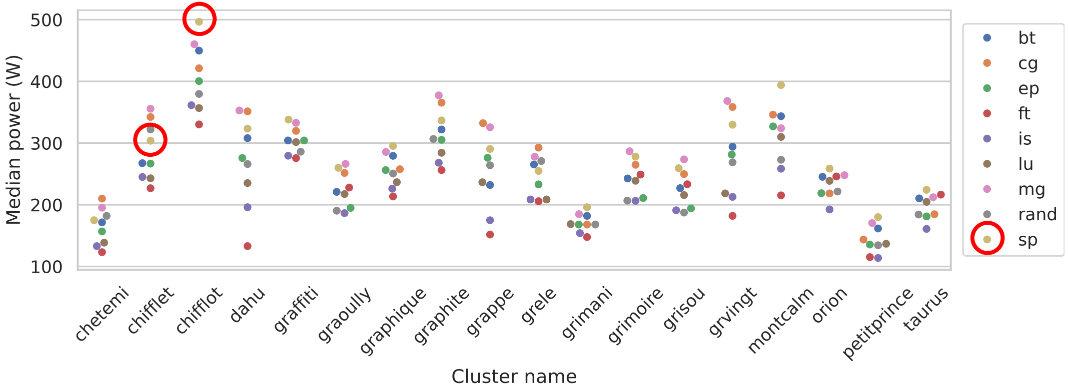
Execution at maximum frequencies for each cluster.
Same architecture: 2 x Intel Xeon E5-2630 v3

Application power ranking depends on the hardware



Execution at maximum frequencies for each cluster.
FT can be first or last

Application power ranking depends on the hardware



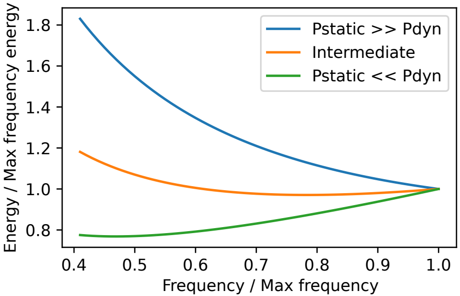
Execution at maximum frequencies for each cluster.
SP can be the overall first or in the middle

DVFS impact on power, time and energy

Classical models

- Inputs: $Time_{fmax}$, $Power_{static}$, $Power_{dynamic}$
- $Time_f = Time_{fmax}\frac{fmax}{f}$
- $Power_f = Power_{static} + Power_{dynamic} \times (\frac{f}{fmax})^2$
- $Energy_f = Time_f \times Power_f$

Only depends on hardware

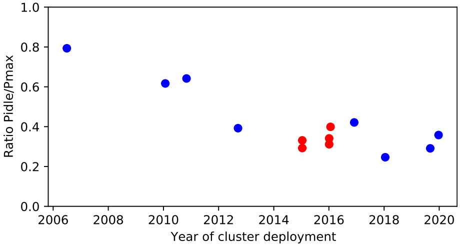


DVFS impact on power, time and energy

Classical models

- Inputs: $Time_{fmax}$, $Power_{static}$, $Power_{dynamic}$
- $Time_f = Time_{fmax} \frac{fmax}{f}$
- $Power_f = Power_{static} + Power_{dynamic} \times (\frac{f}{fmax})^2$
- $Energy_f = Time_f \times Power_f$

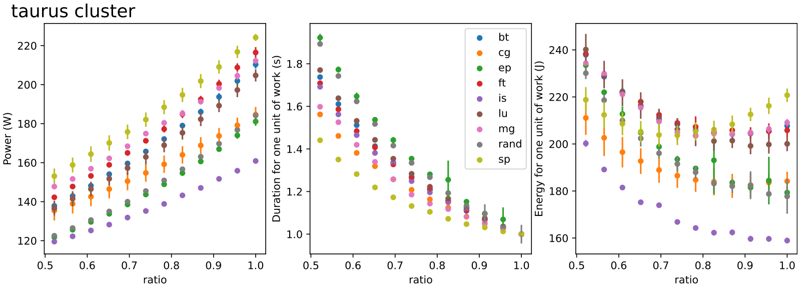
Only depends on hardware



In red: Intel Xeon E5-2630-v3 produced in Q3'2014

DVFS on modern servers is more complex

- NAS Parallel Benchmarks
- Time normalized based on maximum frequency
- Bi-Intel Xeon E5-2630 (2×6 cores)

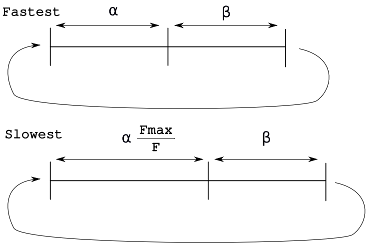


Increasing precision

- State of applications
  - Computing
  - Communications
  - Disk I/O
  - Idle

Increasing precision

- State of applications
  - Computing
  - Communications
  - Disk I/O
  - Idle
- Model
  - Fluid: reacts to DVFS
  - Rigid: does not react to dvfs
  - $Fluid\_percentage = \frac{\alpha}{\alpha+\beta}$



Methodology

Building the model

- For each cluster, application
  - Obtain mean time at *Fmax*
  - Obtain mean time at *Fmin*
  - Obtain $Fluid\_perc(cluster, application)$

Methodology

Building the model

- For each cluster, application
  - Obtain mean time at $Fmax$
  - Obtain mean time at $Fmin$
  - Obtain $Fluid\_perc(cluster, application)$

Using the model

- Impact of DVFS (per unit of work):

$$(1 - Fluid\_perc) + Fluid\_perc \frac{Fmax}{f}$$

Methodology

Building the model

- For each cluster, application
  - Obtain mean time at $Fmax$
  - Obtain mean time at $Fmin$
  - Obtain $Fluid\_perc(cluster, application)$

Using the model

- Impact of DVFS (per unit of work):

$$(1 - Fluid\_perc) + Fluid\_perc\frac{Fmax}{f}$$

Questioning the model

- Is it precise?
- Is $Fluid\_perc$ depending on the cluster?
- Is $Fluid\_perc$ depending on the application?
  - Compare model and measure
  - Use RMSE to compare values

Experimental Framework

- Benchmarks: 8 Nas Parallel Benchmarks (NPB); `rand`: loop of call to rand function
- Platform: 18 clusters from Grid5000, processors from 2012 to 2021
- Grid5000 Power monitoring, DVFS management, experiment management: Expetator[†]
- Hardware performance counters, RAPL: MojitO/S[‡]
- Raw data (soon to be available) 7Go, cleaned data 5.6Go

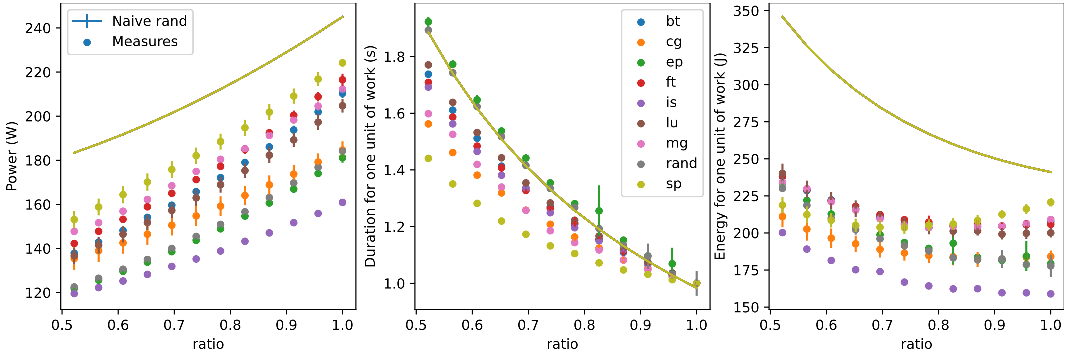
Experiments: All benchmarks, all available frequencies, on all clusters, 10 times

---

[†] https://gitlab.irit.fr/sepia-pub/expetator

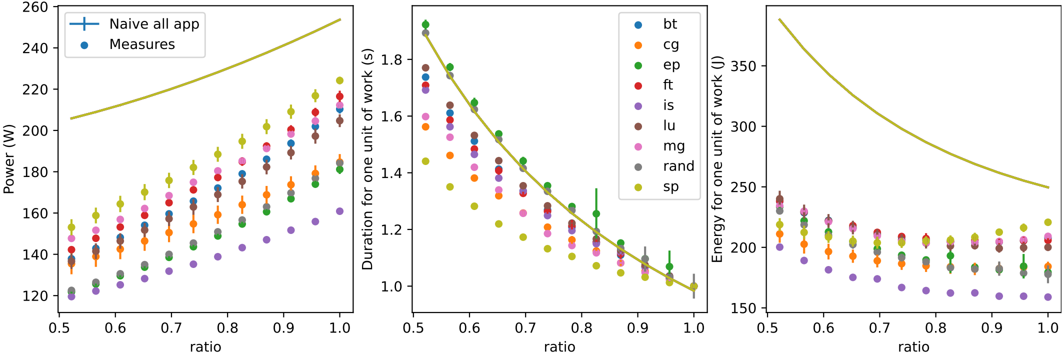
[‡] https://gitlab.irit.fr/sepia-pub/mojitos

Naive rand model



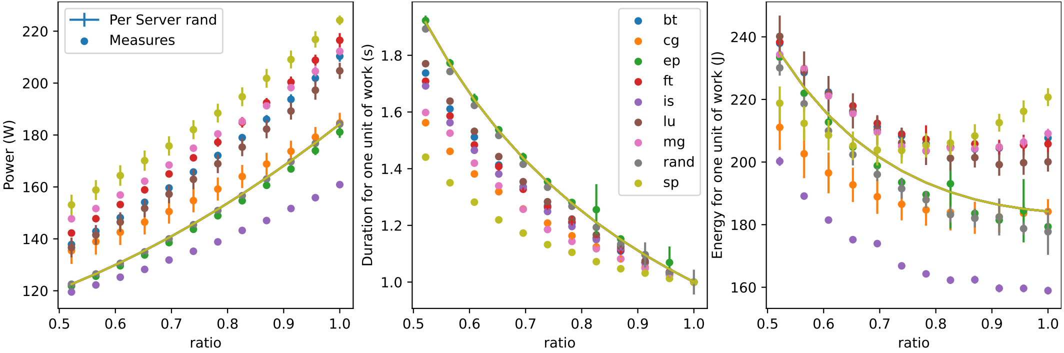
Model: One model: Aggregates all data from all clusters and one application: `rand`

Naive all app model



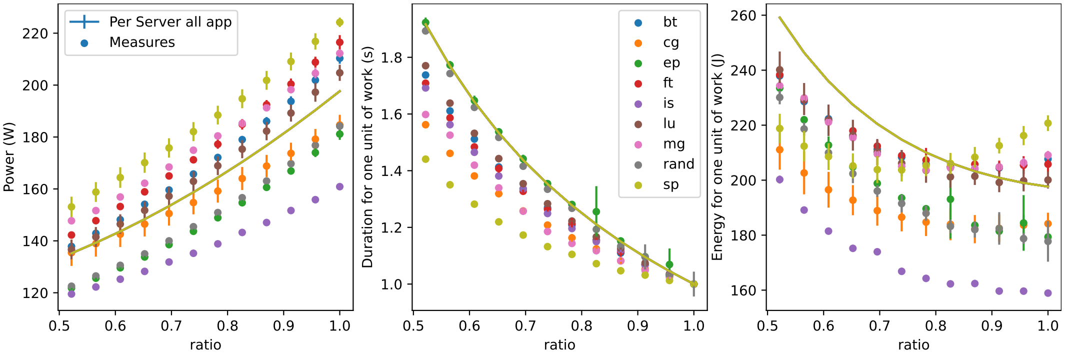
Model: One model: Aggregates all data from all clusters and all applications

Per Server rand model



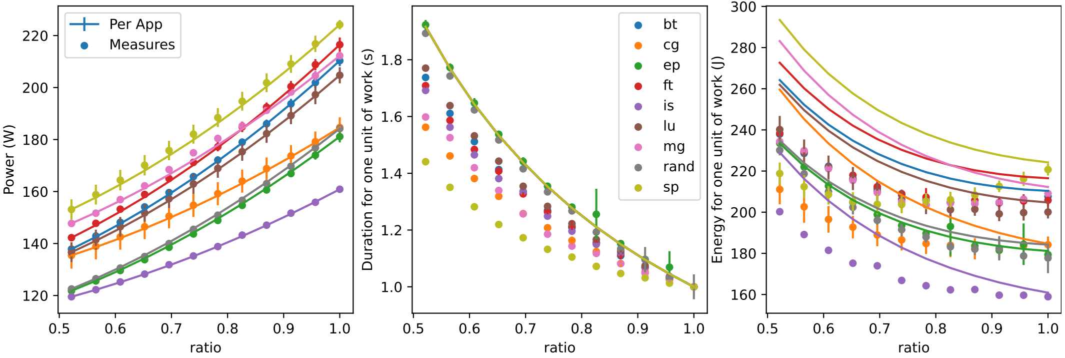
Model: One model per server: Aggregates all data for each clusters using one application: `rand`

Per Server all app model



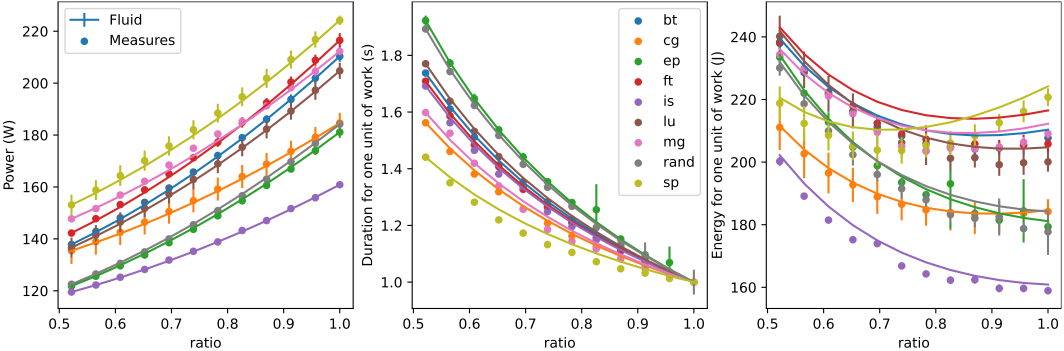
Model: One model per server: Aggregates all data for each clusters using data of all applications

Per App model



Model: One model per server per application

Fluid model



Model: One model per server using one alpha (obtained using data from all applications)

There is one alpha per server per application

Is it precise? Using RMSE as comparison metric

- Mean measured power: 229W
- Mean duration : 1.1
- Mean energy : 213J

name	Power	Duration	Energy
Naive rand	67.04	0.32	91.58
Naive all app	65.74	0.32	103.48
Per Server rand	43.90	0.30	57.98
Per Server all app	41.24	0.30	82.18
Per App	28.96	0.30	91.56
Fluid	28.96	0.05	36.98

Fluid percentages

bench cluster	bt	cg	ep	ft	is	lu	mg	rand	sp
chetemi	0.73	0.26	1.02	0.68	0.64	0.75	0.28	1.05	0.25
chifflet	0.59	0.28	1.00	0.52	0.58	0.56	0.15	1.00	0.14
chifflot	0.75	0.22	1.00	0.65	0.44	0.83	0.42	0.97	0.49
dahu	0.81	0.28	0.99	0.66	0.40	0.81	0.12	1.00	0.36
graffiti	0.73	0.22	0.96	0.63	0.41	0.82	0.28	1.01	0.30
graoully	0.66	0.18	1.02	0.62	0.50	0.77	0.09	1.11	0.11
graphique	0.65	0.34	0.99	0.58	0.55	0.67	0.18	1.00	0.10
graphite	0.83	0.67	0.98	0.79	0.75	0.89	0.62	0.96	0.47
grappe	0.74	0.31	0.94	0.51	0.26	0.69	0.12	1.00	0.19
grele	0.76	0.32	0.99	0.66	0.68	0.71	0.34	1.04	0.36
grimani	0.67	0.29	1.02	0.61	0.44	0.72	0.20	1.00	0.13
grimoire	0.66	0.18	0.99	0.62	0.50	0.76	0.09	0.95	0.11
grisou	0.67	0.20	0.91	0.64	0.50	0.76	0.09	0.96	0.10
grvingt	0.81	0.27	1.00	0.65	0.40	0.81	0.13	1.00	0.36
montcalm	0.67	0.24	1.13	0.43	0.24	0.59	0.03	1.00	0.16
orion	0.81	0.61	1.04	0.77	0.76	0.84	0.65	1.01	0.47
petitprince	0.77	0.59	0.96	0.81	0.76	0.77	0.70	0.93	0.45
taurus	0.80	0.61	1.01	0.77	0.75	0.84	0.65	0.97	0.48

Fluid percentages

- 100% CPU: similar to theoretical model

bench cluster	bt	cg	ep	ft	is	lu	mg	rand	sp
chetemi	0.73	0.26	1.02	0.68	0.64	0.75	0.28	1.05	0.25
chifflet	0.59	0.28	1.00	0.52	0.58	0.56	0.15	1.00	0.14
chifflot	0.75	0.22	1.00	0.65	0.44	0.83	0.42	0.97	0.49
dahu	0.81	0.28	0.99	0.66	0.40	0.81	0.12	1.00	0.36
graffiti	0.73	0.22	0.96	0.63	0.41	0.82	0.28	1.01	0.30
graoully	0.66	0.18	1.02	0.62	0.50	0.77	0.09	1.11	0.11
graphique	0.65	0.34	0.99	0.58	0.55	0.67	0.18	1.00	0.10
graphite	0.83	0.67	0.98	0.79	0.75	0.89	0.62	0.96	0.47
grappe	0.74	0.31	0.94	0.51	0.26	0.69	0.12	1.00	0.19
grele	0.76	0.32	0.99	0.66	0.68	0.71	0.34	1.04	0.36
grimani	0.67	0.29	1.02	0.61	0.44	0.72	0.20	1.00	0.13
grimoire	0.66	0.18	0.99	0.62	0.50	0.76	0.09	0.95	0.11
grisou	0.67	0.20	0.91	0.64	0.50	0.76	0.09	0.96	0.10
grvingt	0.81	0.27	1.00	0.65	0.40	0.81	0.13	1.00	0.36
montcalm	0.67	0.24	1.13	0.43	0.24	0.59	0.03	1.00	0.16
orion	0.81	0.61	1.04	0.77	0.76	0.84	0.65	1.01	0.47
petitprince	0.77	0.59	0.96	0.81	0.76	0.77	0.70	0.93	0.45
taurus	0.80	0.61	1.01	0.77	0.75	0.84	0.65	0.97	0.48

Fluid percentages

- 100% CPU: similar to theoretical model
- Depends on the architecture

bench cluster	bt	cg	ep	ft	is	lu	mg	rand	sp
chetemi	0.73	0.26	1.02	0.68	0.64	0.75	0.28	1.05	0.25
chifflet	0.59	0.28	1.00	0.52	0.58	0.56	0.15	1.00	0.14
chifflot	0.75	0.22	1.00	0.65	0.44	0.83	0.42	0.97	0.49
dahu	0.81	0.28	0.99	0.66	0.40	0.81	0.12	1.00	0.36
graffiti	0.73	0.22	0.96	0.63	0.41	0.82	0.28	1.01	0.30
graoully	0.66	0.18	1.02	0.62	0.50	0.77	0.09	1.11	0.11
graphique	0.65	0.34	0.99	0.58	0.55	0.67	0.18	1.00	0.10
graphite	0.83	0.67	0.98	0.79	0.75	0.89	0.62	0.96	0.47
grappe	0.74	0.31	0.94	0.51	0.26	0.69	0.12	1.00	0.19
grele	0.76	0.32	0.99	0.66	0.68	0.71	0.34	1.04	0.36
grimani	0.67	0.29	1.02	0.61	0.44	0.72	0.20	1.00	0.13
grimoire	0.66	0.18	0.99	0.62	0.50	0.76	0.09	0.95	0.11
grisou	0.67	0.20	0.91	0.64	0.50	0.76	0.09	0.96	0.10
grvingt	0.81	0.27	1.00	0.65	0.40	0.81	0.13	1.00	0.36
montcalm	0.67	0.24	1.13	0.43	0.24	0.59	0.03	1.00	0.16
orion	0.81	0.61	1.04	0.77	0.76	0.84	0.65	1.01	0.47
petitprince	0.77	0.59	0.96	0.81	0.76	0.77	0.70	0.93	0.45
taurus	0.80	0.61	1.01	0.77	0.75	0.84	0.65	0.97	0.48

Fluid percentages

- 100% CPU: similar to theoretical model
- Depends on the architecture
- Depends on the benchmark

bench cluster	bt	cg	ep	ft	is	lu	mg	rand	sp
chetemi	0.73	0.26	1.02	0.68	0.64	0.75	0.28	1.05	0.25
chifflet	0.59	0.28	1.00	0.52	0.58	0.56	0.15	1.00	0.14
chifflot	0.75	0.22	1.00	0.65	0.44	0.83	0.42	0.97	0.49
dahu	0.81	0.28	0.99	0.66	0.40	0.81	0.12	1.00	0.36
graffiti	0.73	0.22	0.96	0.63	0.41	0.82	0.28	1.01	0.30
graoully	0.66	0.18	1.02	0.62	0.50	0.77	0.09	1.11	0.11
graphique	0.65	0.34	0.99	0.58	0.55	0.67	0.18	1.00	0.10
graphite	0.83	0.67	0.98	0.79	0.75	0.89	0.62	0.96	0.47
grappe	0.74	0.31	0.94	0.51	0.26	0.69	0.12	1.00	0.19
grele	0.76	0.32	0.99	0.66	0.68	0.71	0.34	1.04	0.36
grimani	0.67	0.29	1.02	0.61	0.44	0.72	0.20	1.00	0.13
grimoire	0.66	0.18	0.99	0.62	0.50	0.76	0.09	0.95	0.11
grisou	0.67	0.20	0.91	0.64	0.50	0.76	0.09	0.96	0.10
grvingt	0.81	0.27	1.00	0.65	0.40	0.81	0.13	1.00	0.36
montcalm	0.67	0.24	1.13	0.43	0.24	0.59	0.03	1.00	0.16
orion	0.81	0.61	1.04	0.77	0.76	0.84	0.65	1.01	0.47
petitprince	0.77	0.59	0.96	0.81	0.76	0.77	0.70	0.93	0.45
taurus	0.80	0.61	1.01	0.77	0.75	0.84	0.65	0.97	0.48

Fluid percentages

- 100% CPU: similar to theoretical model
- Depends on the architecture
- Depends on the benchmark
- Similar architectures

bench cluster	bt	cg	ep	ft	is	lu	mg	rand	sp
chetemi	0.73	0.26	1.02	0.68	0.64	0.75	0.28	1.05	0.25
chifflet	0.59	0.28	1.00	0.52	0.58	0.56	0.15	1.00	0.14
chifflot	0.75	0.22	1.00	0.65	0.44	0.83	0.42	0.97	0.49
dahu	0.81	0.28	0.99	0.66	0.40	0.81	0.12	1.00	0.36
graffiti	0.73	0.22	0.96	0.63	0.41	0.82	0.28	1.01	0.30
graoully	0.66	0.18	1.02	0.62	0.50	0.77	0.09	1.11	0.11
graphique	0.65	0.34	0.99	0.58	0.55	0.67	0.18	1.00	0.10
graphite	0.83	0.67	0.98	0.79	0.75	0.89	0.62	0.96	0.47
grappe	0.74	0.31	0.94	0.51	0.26	0.69	0.12	1.00	0.19
grele	0.76	0.32	0.99	0.66	0.68	0.71	0.34	1.04	0.36
grimani	0.67	0.29	1.02	0.61	0.44	0.72	0.20	1.00	0.13
grimoire	0.66	0.18	0.99	0.62	0.50	0.76	0.09	0.95	0.11
grisou	0.67	0.20	0.91	0.64	0.50	0.76	0.09	0.96	0.10
grvingt	0.81	0.27	1.00	0.65	0.40	0.81	0.13	1.00	0.36
montcalm	0.67	0.24	1.13	0.43	0.24	0.59	0.03	1.00	0.16
orion	0.81	0.61	1.04	0.77	0.76	0.84	0.65	1.01	0.47
petitprince	0.77	0.59	0.96	0.81	0.76	0.77	0.70	0.93	0.45
taurus	0.80	0.61	1.01	0.77	0.75	0.84	0.65	0.97	0.48

Takeaway

- All applications and hardware are different
- Fluid/Rigid is a good approximation
  - Two measures needed
  - 10 times more precise (RMSE) than the black box model

Takeaway

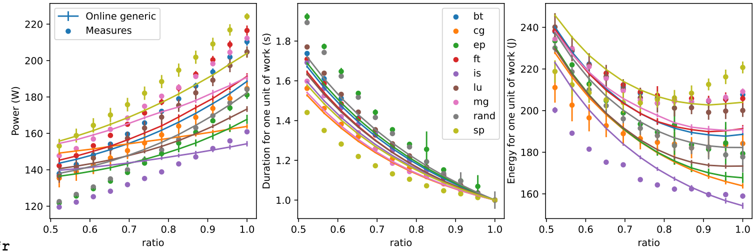
- All applications and hardware are different
- Fluid/Rigid is a good approximation
  - Two measures needed
  - 10 times more precise (RMSE) than the black box model

$$Power_f^{c,a} = Pstatic^{c,a} + Pdyn^{c,a}\left(\frac{f}{Fmax^c}\right)^{Pcoef}$$

$$Duration_f^{c,a} = \left((1-\alpha^{c,a}) + \alpha^{c,a}\frac{Fmax^c}{f}\right) \times Duration_{fmax}^{c,a}$$

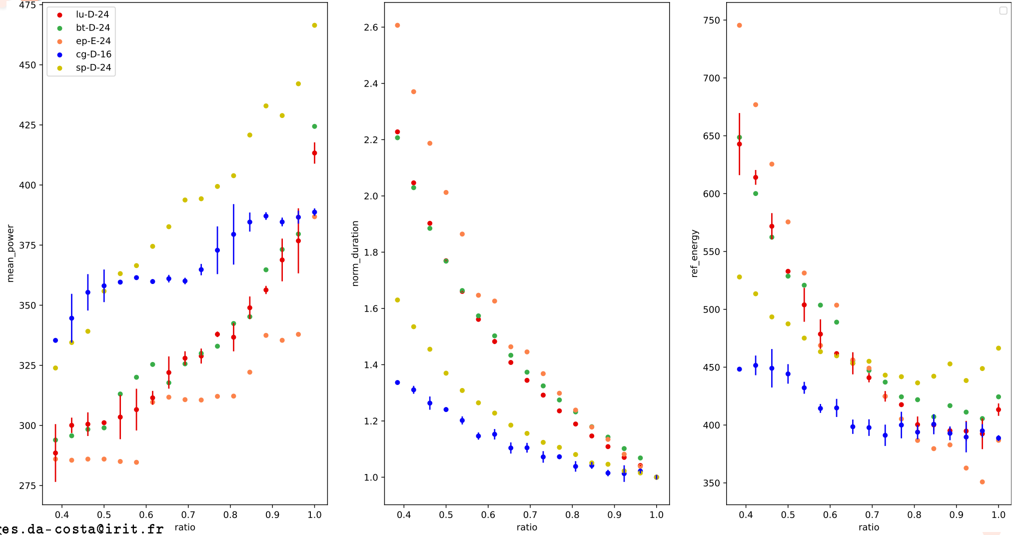
Takeaway

- All applications and hardware are different
- Fluid/Rigid is a good approximation
  - Two measures needed
  - 10 times more precise (RMSE) than the black box model
- Can be done online using system measures
  - No application knowledge
  - hardware performance counters and RAPL
  - Only 10 to 25% increased error (RMSE)



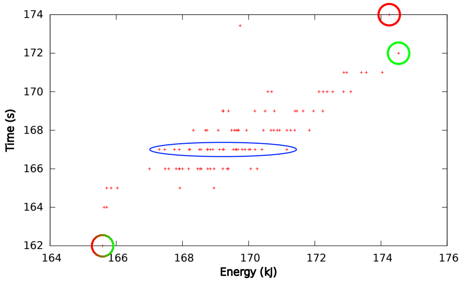
Plan

Data acquisition is sometime difficult

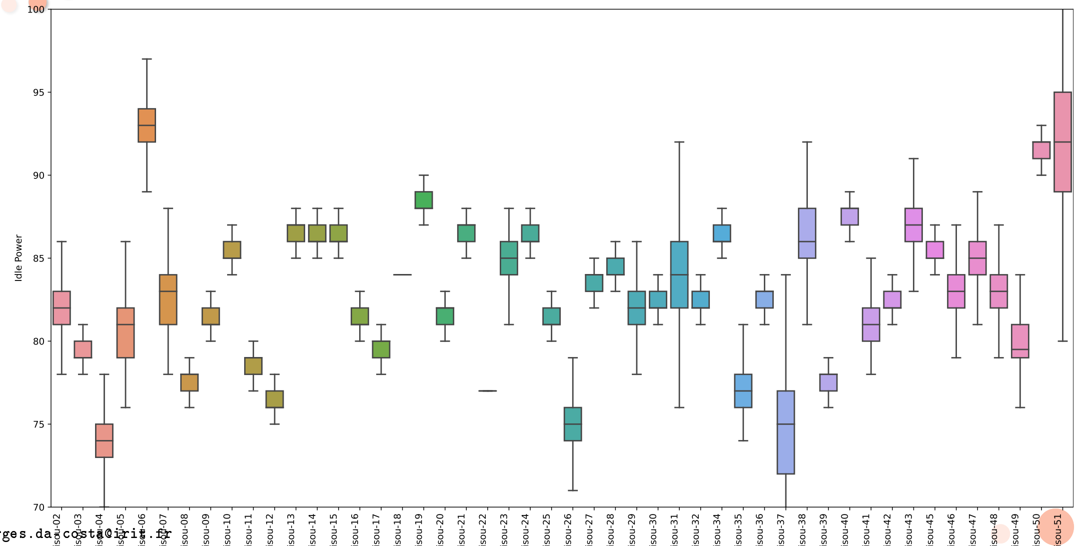


No stability of experiments

- *Simple* experiment of Fast Fourier Transform (NPB)
- 100 experiments on exactly the same 4 servers (Grid'5000)
- Large variations
  - Time: 12s, 7% (Std. Dev. 3.2s)
  - Energy: 9.3kJ, 5.5% (3kJ)
- For the same time, 167s
  - Difference of 4kJ
- Time $\neq$ Energy

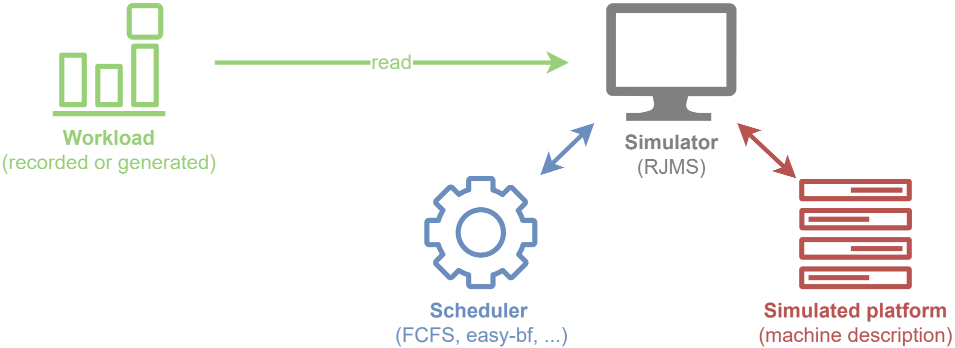


Identical servers after boot (10 minutes)

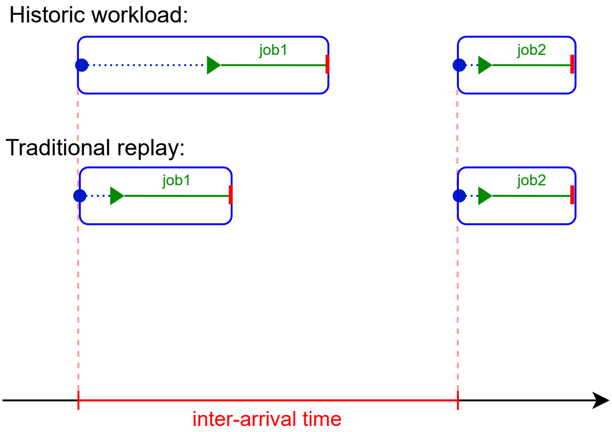


Plan

Traditional replay: principle

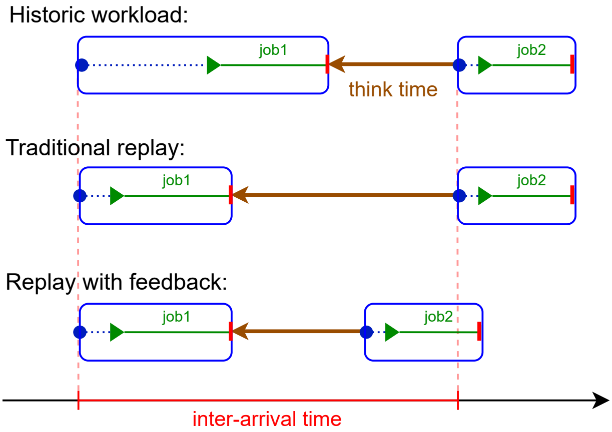


Traditional replay: shortcomings



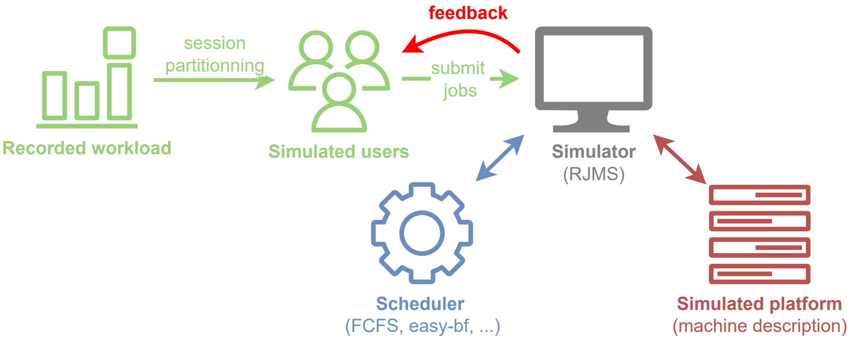
(original work from Zakay and Feitelson 2015)

Traditional replay: shortcomings



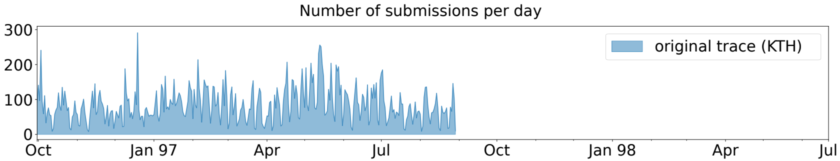
(original work from Zakay and Feitelson 2015)

Replay with feedback



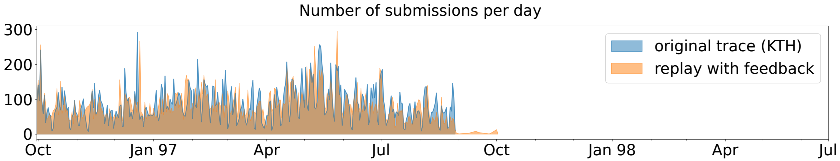
- Implementation available in **Batmen**:
  `https://gitlab.irit.fr/sepia-pub/mael/batmen`
- Reproducible experimental campaign:
  `https://gitlab.irit.fr/sepia-pub/open-science/expe-replay-feedback`

Distribution of jobs' submission times[§]



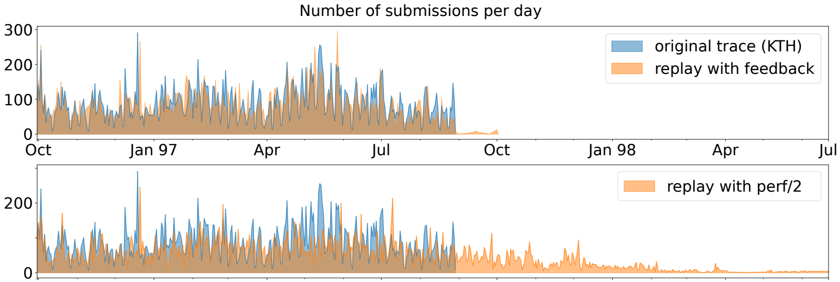
[§] M. Madon, G. Da Costa, and J.-M. Pierson, *Replay with Feedback: How Does the Performance of HPC System Impact User Submission Behavior?*, in Future Generation Computer Systems 2024, 10.1016/j.future.2024.01.024

Distribution of jobs' submission times[§]



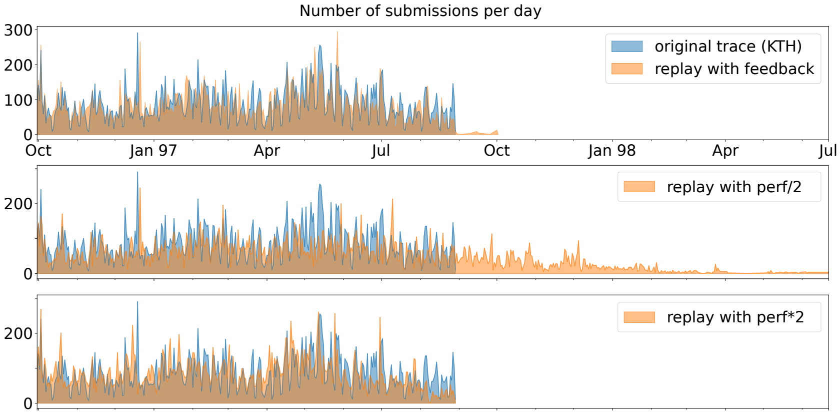
[§] M. Madon, G. Da Costa, and J.-M. Pierson, *Replay with Feedback: How Does the Performance of HPC System Impact User Submission Behavior?*, in Future Generation Computer Systems 2024, 10.1016/j.future.2024.01.024

Distribution of jobs' submission times§



§M. Madon, G. Da Costa, and J.-M. Pierson, *Replay with Feedback: How Does the Performance of HPC System Impact User Submission Behavior?*, in Future Generation Computer Systems 2024, 10.1016/j.future.2024.01.024

Distribution of jobs' submission times[§]



[§] M. Madon, G. Da Costa, and J.-M. Pierson, *Replay with Feedback: How Does the Performance of HPC System Impact User Submission Behavior?*, in Future Generation Computer Systems 2024, 10.1016/j.future.2024.01.024

Plan

Conclusion

- Takeaway
  - Applications behave differently depending of servers
  - Online algorithms provide quite precise models

https://www.irit.fr/~Georges.Da-Costa/

Conclusion

- Takeaway
  - Applications behave differently depending of servers
  - Online algorithms provide quite precise models
- Impact on energy-efficient scheduling of datacenters
  - Too tight heuristics might have difficulties on actual systems
  - Offline: Resilient scheduling is a must
  - Online: Reactive systems with measures

https://www.irit.fr/~Georges.Da-Costa/

Conclusion

- Takeaway
  - Applications behave differently depending of servers
  - Online algorithms provide quite precise models
- Impact on energy-efficient scheduling of datacenters
  - Too tight heuristics might have difficulties on actual systems
  - Offline: Resilient scheduling is a must
  - Online: Reactive systems with measures
- Replay with feedback
  - Improves the realism of scheduling experiments
  - Use local products, simulate with batmen and batsim :)

`https://www.irit.fr/~Georges.Da-Costa/`