Mesurer & réduire l'empreinte des services logiciels

@RomainROUVOY

Université de Lille

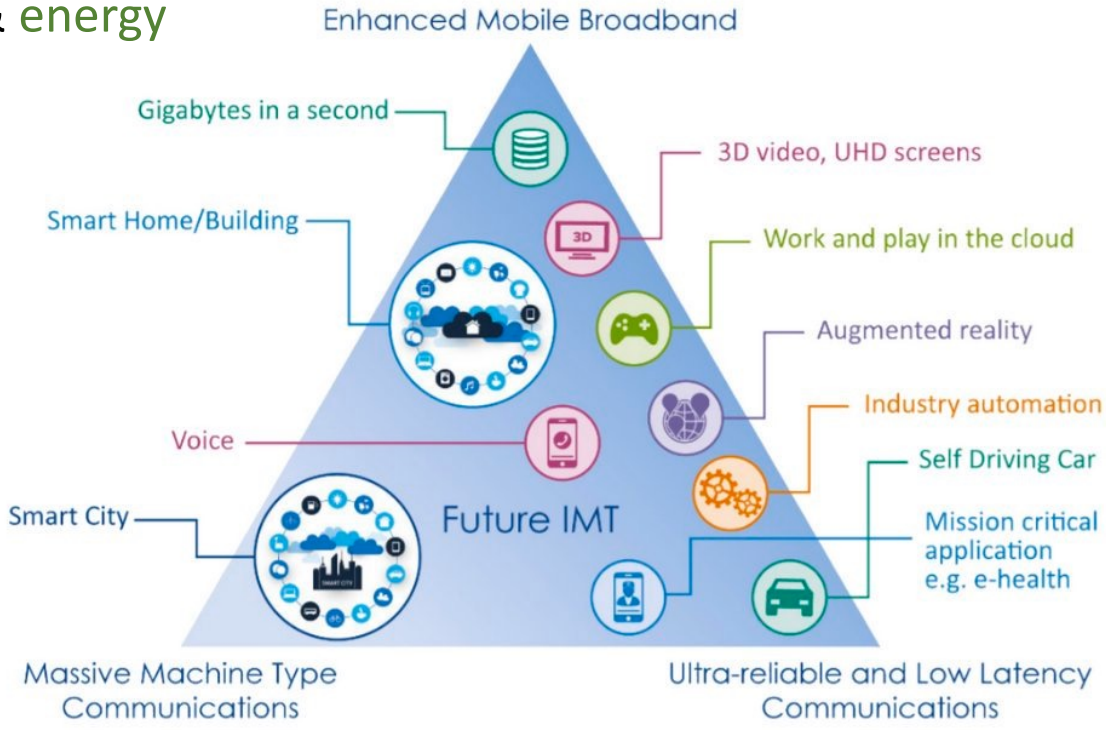
SPIRALS

Inria

SPIRALS project-team

- **Software engineering ⇢⇠ Distributed systems**
  - *Smart Software Systems at Large*
    - *Self-repair & self-optimization*
    - Focus on security & energy

- 40 members :
  - 11 staff members
  - 7 postdocs
  - 17 PhD students
  - 5 engineers



https://team.inria.fr/spirals

ADEME

Agence de l'Environnement
et de la Maîtrise de l'Energie

DAVIDSON

CONSULTING

orange™

OVHcloud

DOI:10.1145/3183582

The future of computing research relies on addressing an array of limitations on a planetary scale.

BY BONNIE NARDI, BILL TOMLINSON, DONALD J. PATTERSON, JAY CHEN, DANIEL PARGMAN, BARATH RAGHAVAN, AND BIRGIT PENZENSTADLER

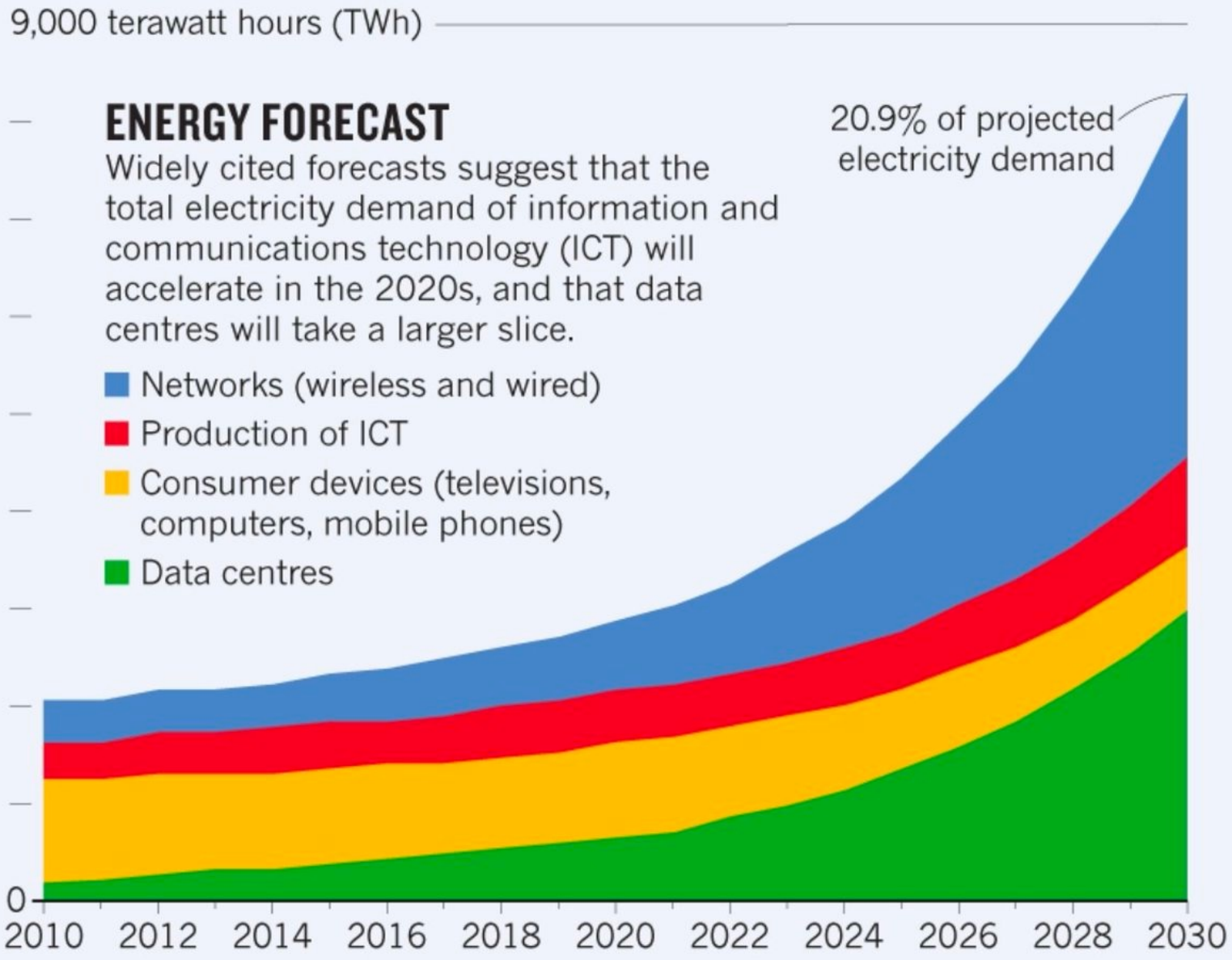
Computing within Limits

COMPUTING RESEARCHERS AND practitioners are c
seen as inventing the future. As such, we are impl
also in the business of predicting the future. We p

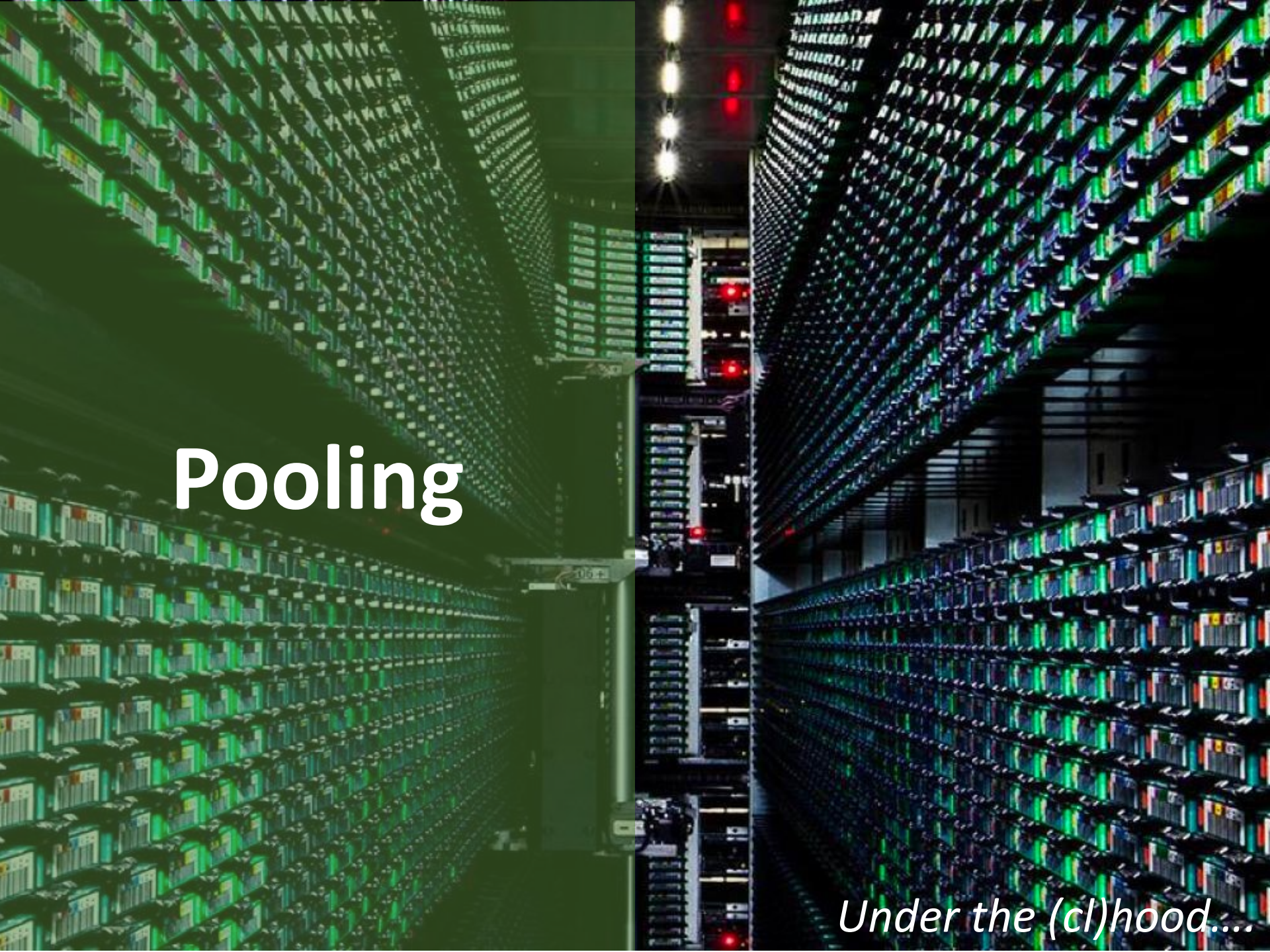
or the amount of pollution an ecosystem can bear, limits are less obvious in computing. Many believe the only limit worth considering is human ingenuity, and that we can surpass any and all other limits if we, as a global communi-

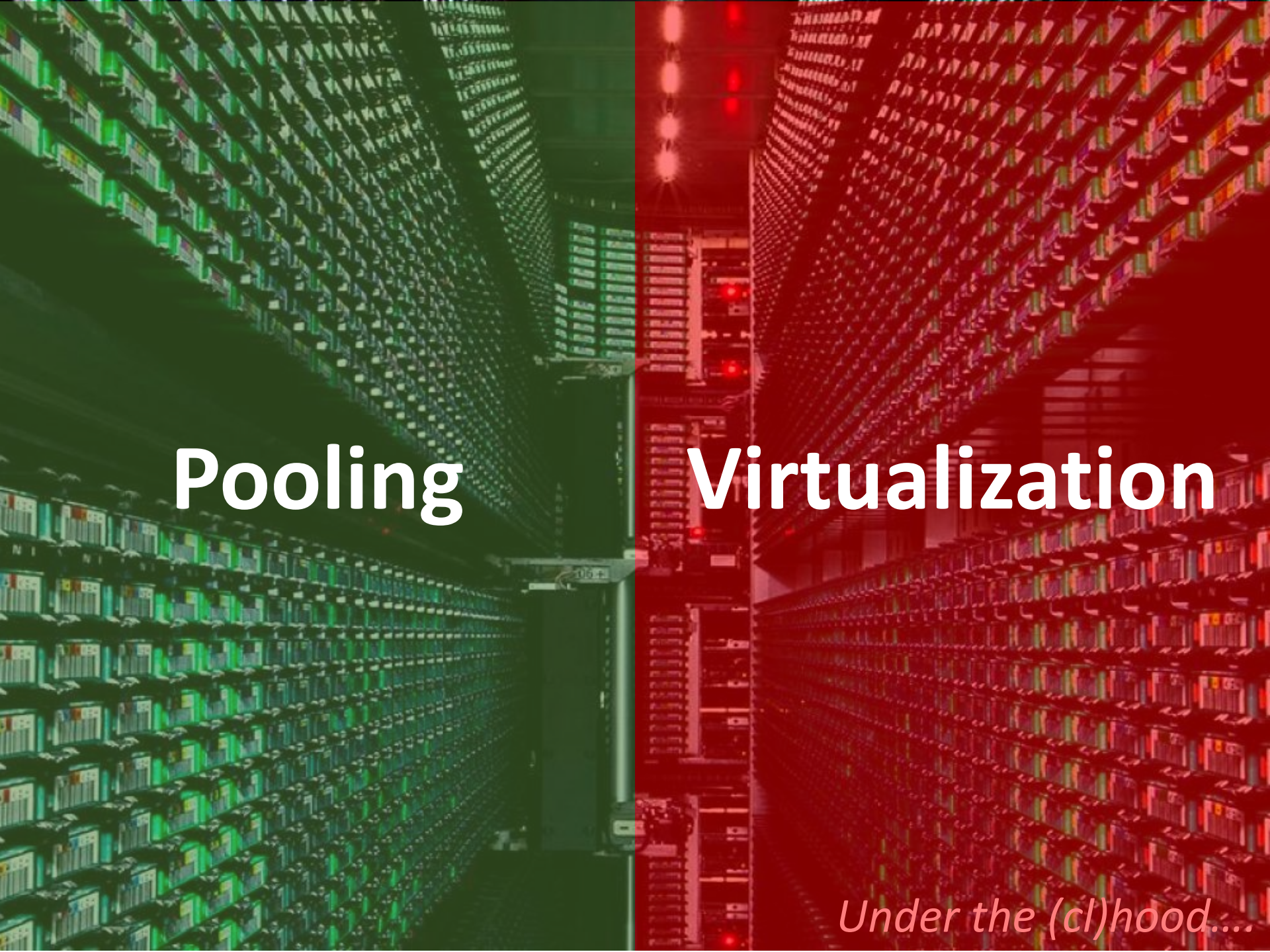
» key insights

- **Most computing work is premised on industrial civilization's default worldview in which ongoing economic growth is both achievable and desirable.**
- **This growth-focused worldview, however, is at odds with findings from many other scientific fields, which see growth as deeply problematic for ecological and social reasons.**
- **We proposed that the computing field transition toward "computing within limits," exploring ways that new forms of computing supported well-being while enabling human civilizations to live within global ecological and material limits.**
- **Computing underlies virtually all the infrastructure of global society, and will therefore be critical in shaping a society that meaningfully adapts to global limits.**



The chart above is an ‘expected case’ projection from Anders Andrae, a specialist in sustainable ICT. In his ‘best case’ scenario, ICT grows to only 8% of total electricity demand by 2030, rather than to 21%.


Pooling
Under the (cl)hood....


Pooling
Virtualization
Under the (cl)hood....


VOUS ÊTES ICI
Software (App)
Software (Framework)
Software (JVM)
Software (Container)
Software (OS)
Software (VM)
Software (Hypervisor)
Software (OS)
Hardware

What about software sustainability??

On Reducing the Energy Consumption of Software: From Hurdles to Requirements

Zakaria Ournani
Orange Labs/ Inria / Univ. Lille
zakaria.ournani@inria.fr

Romain Rouvoy
Univ. Lille / Inria / IUF
romain.rouvoy@univ-lille.fr

Pierre Rust
Orange Labs
pierre.rust@orange.com

Joel Penhoat
Orange Labs
joel.penhoat@orange.com

ABSTRACT

Background. As software took control over hardware in many domains, the question of the energy footprint induced by the software is becoming critical for our society, as the resources powering the underlying infrastructure are finite. Yet, beyond this growing interest, energy consumption remains a difficult concept to master for a developer.

Aims. The purpose of this study is to better understand the root causes that prevent the issue of software energy consumption to be more widely considered by developers and companies.

Method. To investigate this issue, this paper reports on a qualitative study we conducted in an industrial context. We applied an in-depth analysis of the interviews of 10 experienced developers and summarized a set of implications.

Results. We argue that our study delivers *i)* insightful feedback on how green software design is considered among the interviewed developers and *ii)* a set of findings to build helpful tools, motivate further research, and establish better development strategies to promote green software design.

Conclusion. This paper covers an industrial case study of developers' awareness of green software design and how to promote it within the company. While it might not be generalizable for any company, we believe our results deliver a common body of knowledge with implications to be considered for similar cases and further researches.

ACM Reference Format:
Zakaria Ournani, Romain Rouvoy, Pierre Rust, and Joel Penhoat. 2020. On Reducing the Energy Consumption of Software: From Hurdles to Requirements. In *ESEM '20: ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '20), October 8–9, 2020, Bari, Italy.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3382494.3410678

1 INTRODUCTION

The last decade witnessed several attempts to consider green software design as a core development concern to improve the energy efficiency of software systems at large [2, 3, 18, 23, 26]. However, despite previous studies that have contributed to establish guidelines and tools to analyze and reduce the energy consumption [1, 7, 12, 16, 17, 25, 32], these contributions fail to be adopted by practitioners till date [14, 28].

Concretely, both quantitative and qualitative studies [22, 28, 31] previously surveyed developers to establish assumptions about developers' knowledge of green software design. These studies highlight that developers might be aware of software energy consumption problems, but have a very limited knowledge on how to reduce the energy footprint of their software product. For example, Pinto *et al.* [31] mentioned collecting "vague" answers from developers when asked about how to deal with software energy consumption. Fang *et al.* [28] reported that, among 100 developers, a small portion are aware of the primary sources of software energy consumption. Only 10% of the participants try to measure the energy consumption of their software project, while less than 20% take energy into account in the first place. Moreover, the empirical study of Manotas *et al.* [22] reported that energy requirements are often more desires than specific targets. They highlight that developers believe they miss accurate intuitions about the energy usage of their code, and that energy concerns are largely ignored during maintenance.

However, to the best of our knowledge, none of these studies discuss *i)* the hurdles that prevent the broader adoption of green software design, and *ii)* the developers' requirements in terms of tooling in an industrial context. But, we actually believe that both aspects are critical issues to consider when aiming to reach an adoption of such tools and methods among developers.

Contribution. This paper reports on a qualitative investigation on software energy consumption considerations among experienced developers of a large company. Concretely, we conducted interviews with 10 senior/expert developers with the ambition to cover developers' opinions, problems, and requirements to promote the green software design in an industrial context. The key contributions of this paper can, therefore, be summarized as:

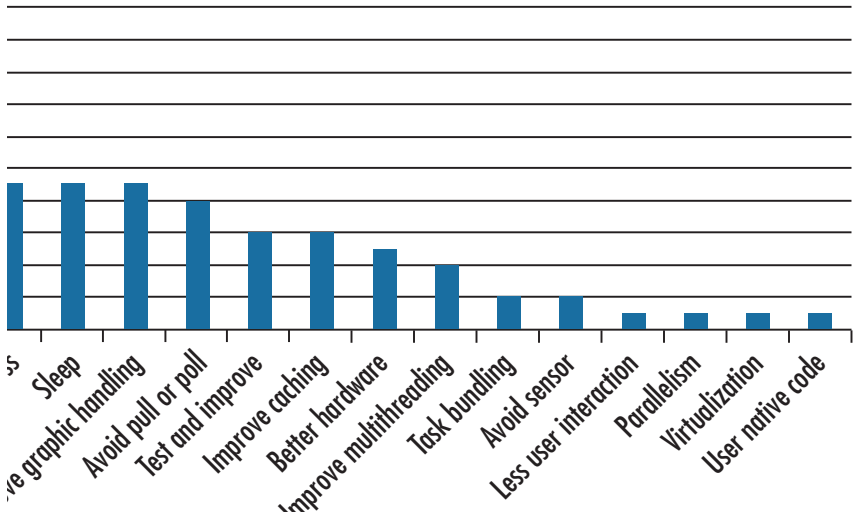
(1) Providing a detailed understanding of the interviewed developers' awareness and knowledge about green software design,
(2) Identifying the main constraints and challenges that developers encounter in their daily development,
(3) Building specifications for the tooling that suits developers expectations and experiences,
(4) Investigating the best ways to keep developers aware of software energy consumption and promote it within a company,
(5) Identifying the exact role and responsibilities of the company to promote green software design,

We believe it can offer a common body of knowledge for researchers, tools creators, companies, and developers, which can be considered to improve awareness and adoption of green software design. For example, our investigations highlight that adoption of green software design in an industrial context requires not only a tight integration of future tools into the software development lifecycle, but also the

[…] clients *"care **first** and foremost about **speed of development**, and secondly about reasonable quality and performance."*

"*It's more often the **hardware rather than the software** that we are interested in when we talk about energy consumption.*"

These results show that these programmers lacked knowledge of how to **accurately measure software energy consumption**.

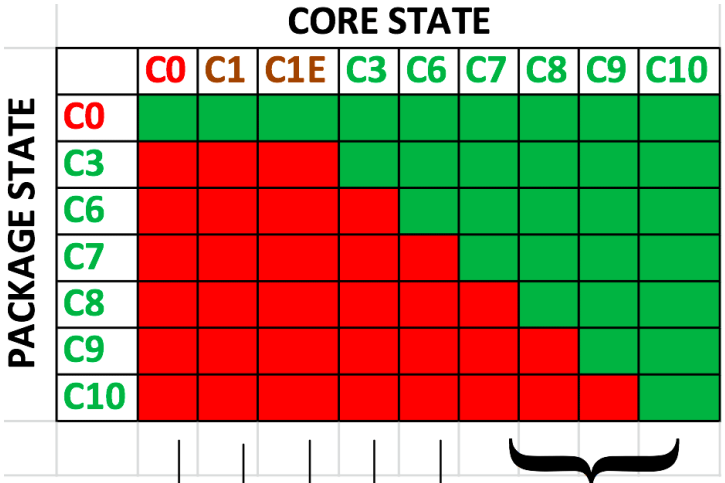


http://doi.org/10.1109/MS.2015.83

« *These results show that these programmers lacked knowledge of how to* ***accurately measure software energy consumption***. »

Enabling power monitoring of software systems

http://powerapi.org

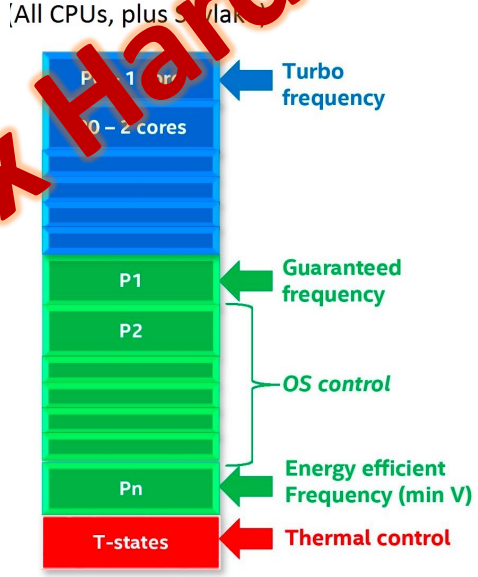


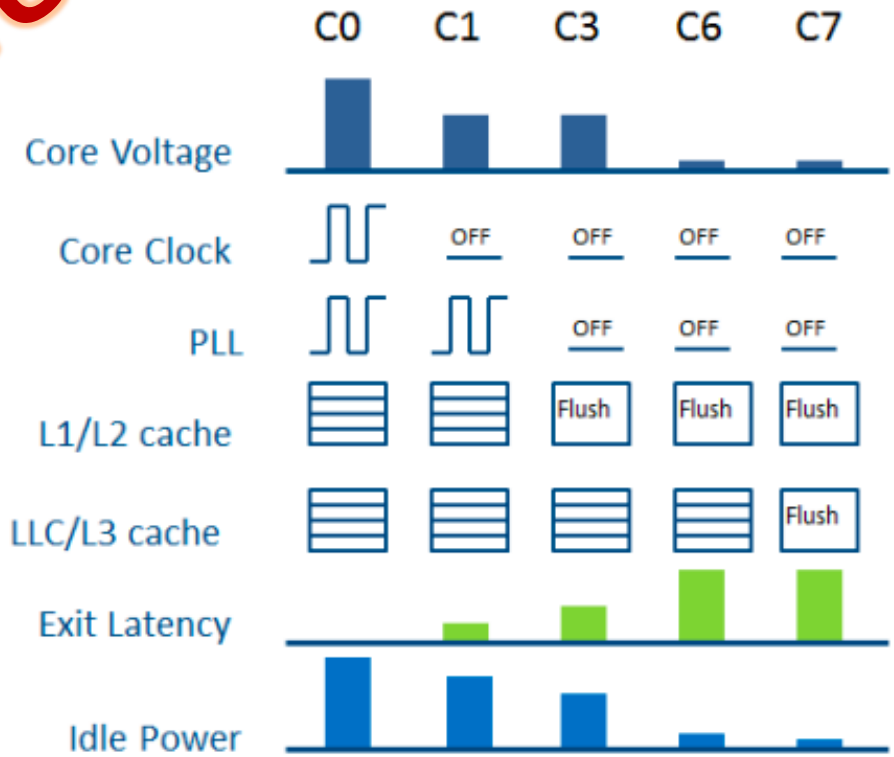
- One or more cores or GT executing instructions
- All cores and GT in C3 or deeper, L3 may be flushed and turned off, memory in self refresh, some Uncore clocks stopped, some Uncore voltages reduced
- All cores and GT in C6 or deeper, L3 may be flushed and turned off, memory in self refresh, all Uncore clocks stopped, some Uncore voltages reduced
- Package C6 + L3 flushed and turned off, additional Uncore voltages reduced
- Package C7 + most Uncore voltages reduced to 0V
- Package C8 + VR12.6 in low power state
- Package C9 + VR12.6 turned off

- Core behaves the same as Core C6 state
- All core clocks are stopped, core state saved and voltage reduce to 0V
- Cores flush L1/L2 into L3, all core clocks are stopped
- Core halted, most core clocks stopped and voltage reduced to Pn
- Core halted, most core clocks stopped
- Core is executing code

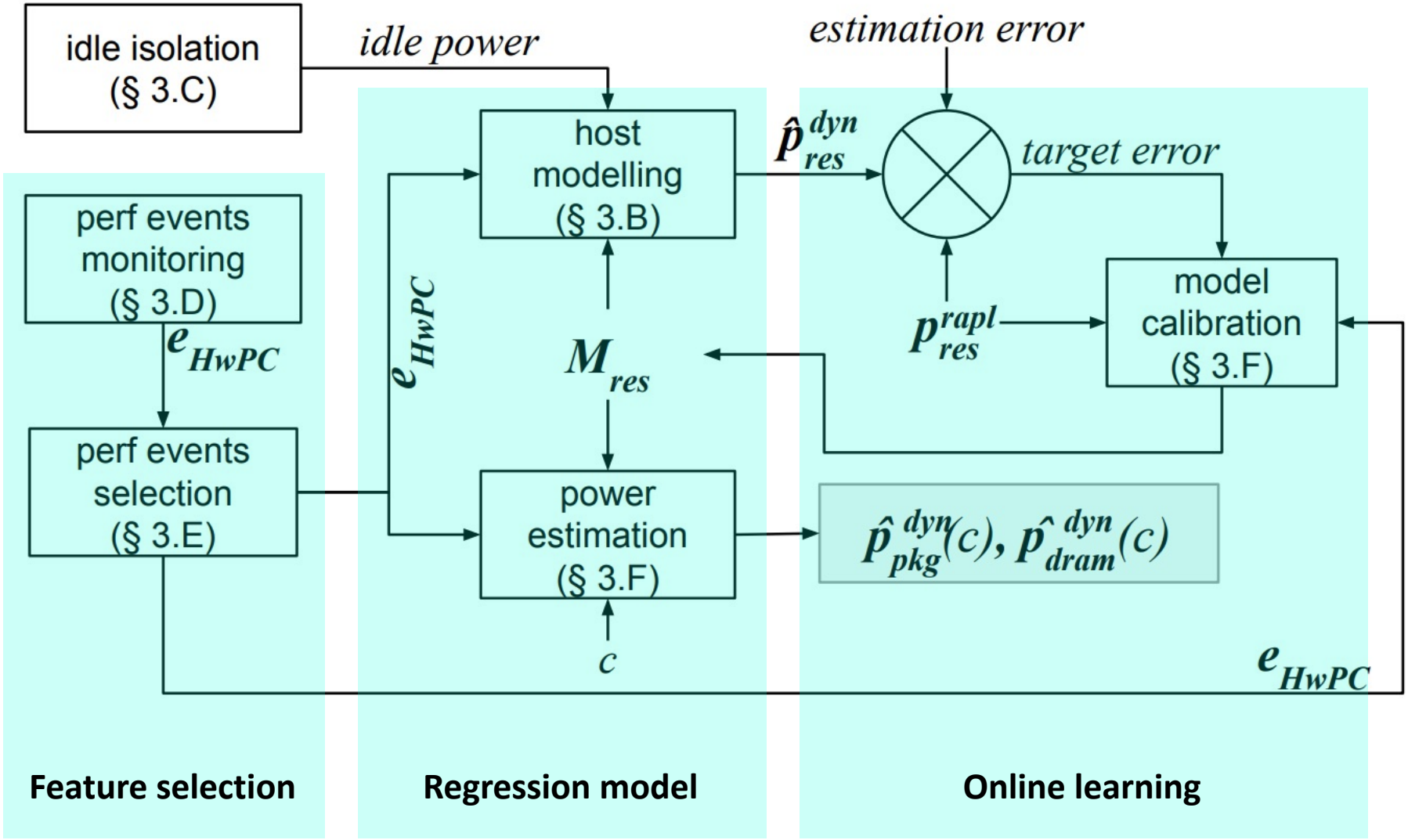
Possible combination of core/package states

Impossible combination of core/package state



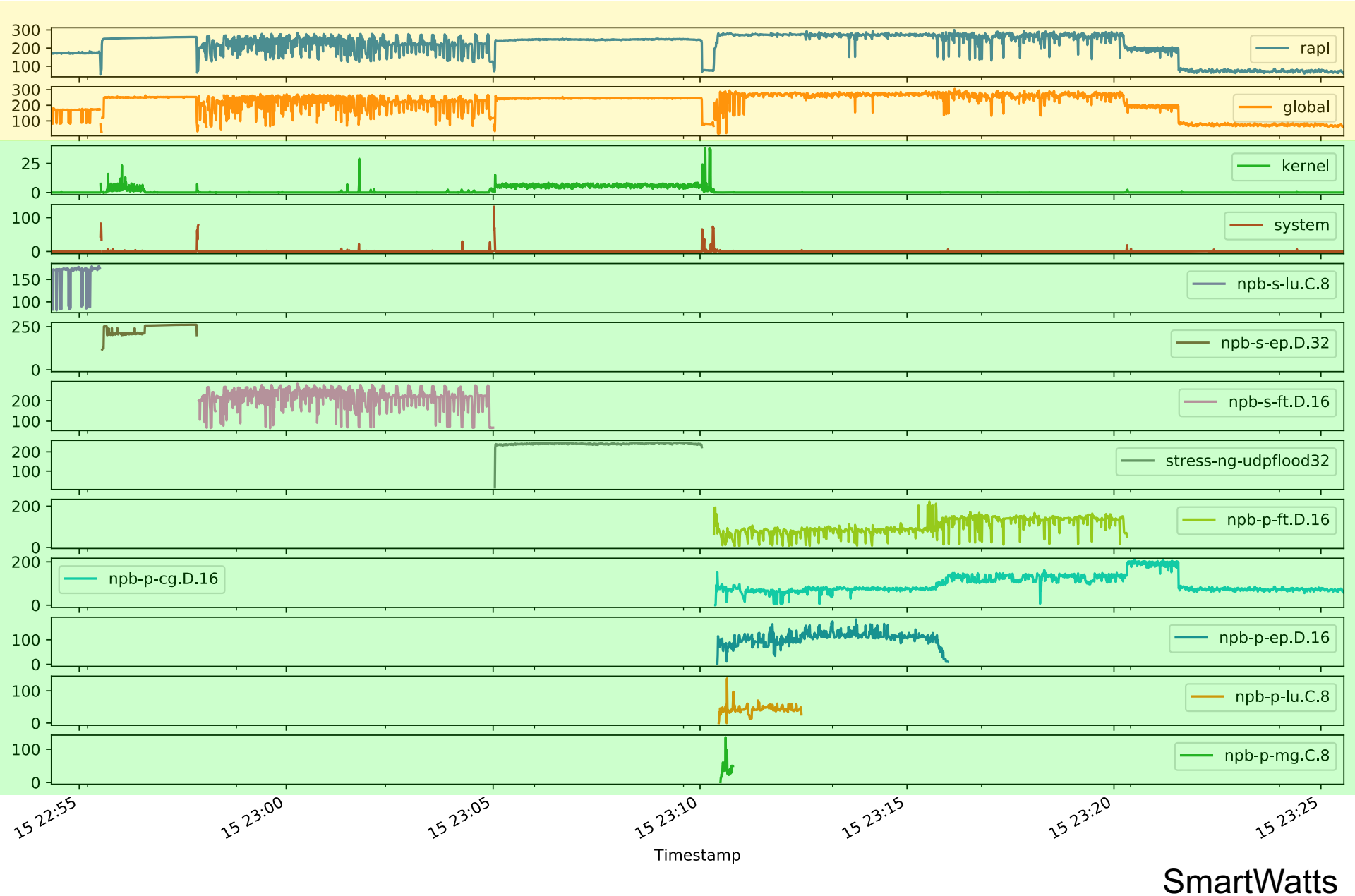


Learning the CPU/DRAM power models from RAPL

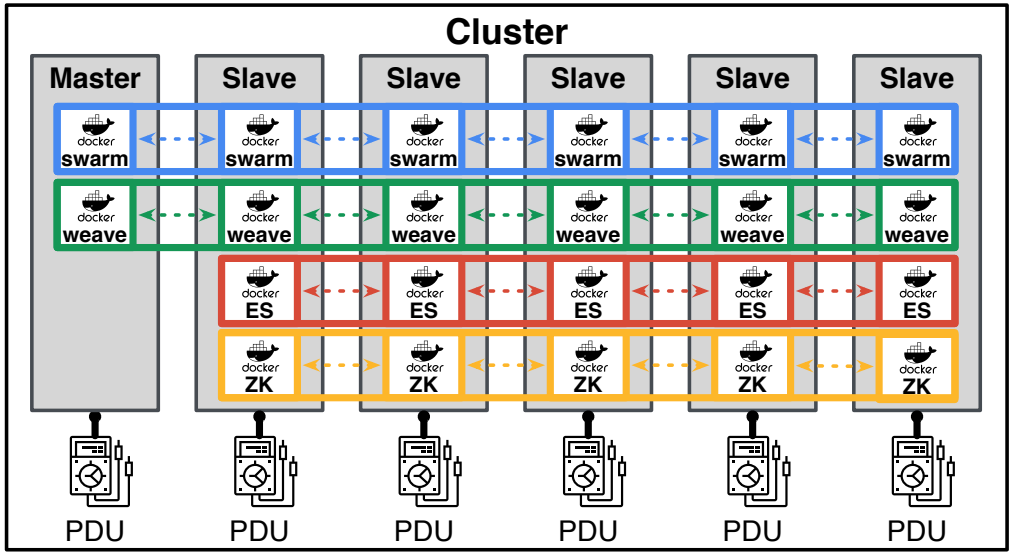


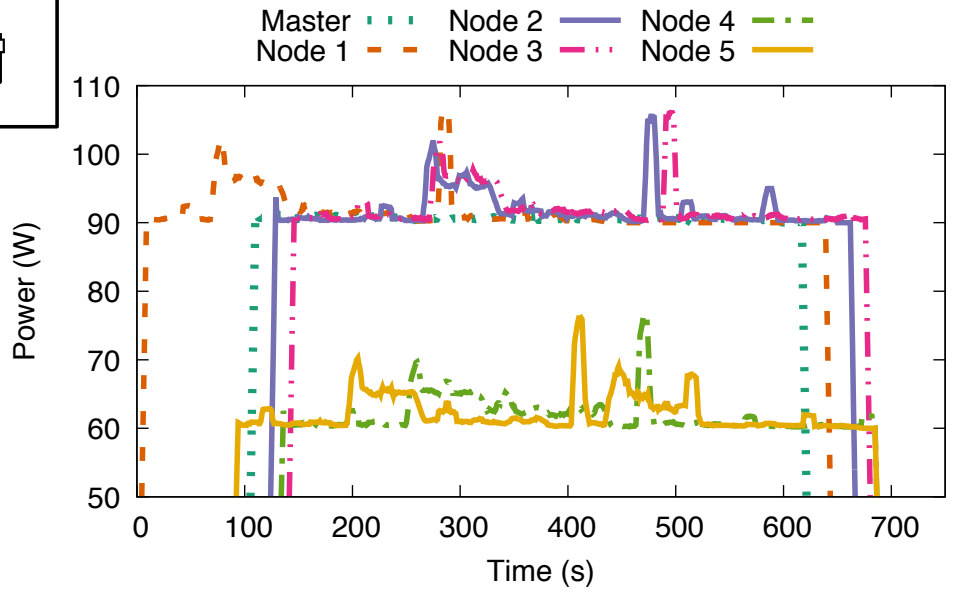
SmartWatts

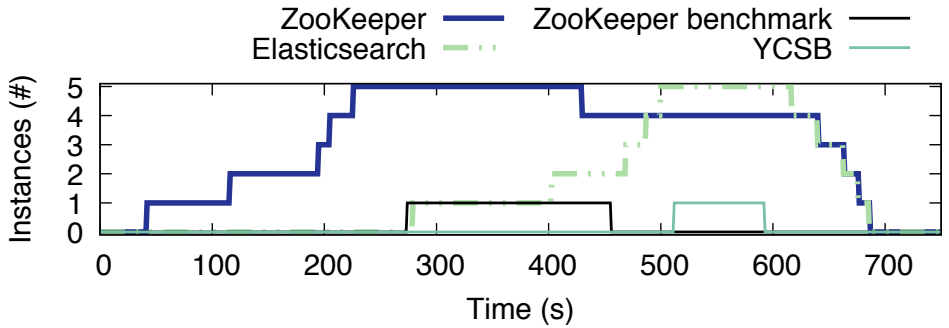
Monitoring the power consumption in real-time



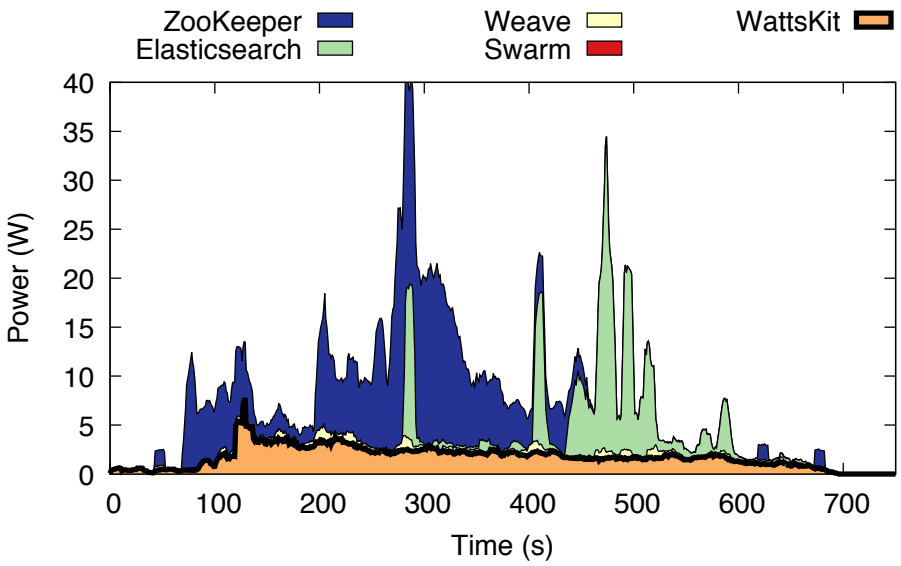
SmartWatts

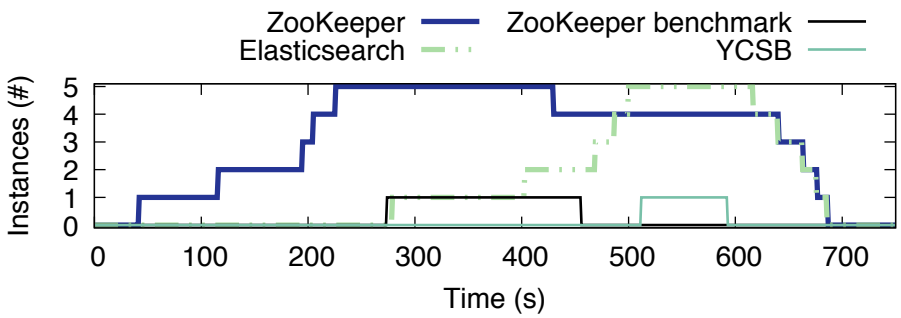

Cluster
Master
Slave
Slave
Slave
Slave
Slave
docker swarm
docker weave
docker ES
docker ZK
PDU
PDU
PDU
PDU
PDU
PDU

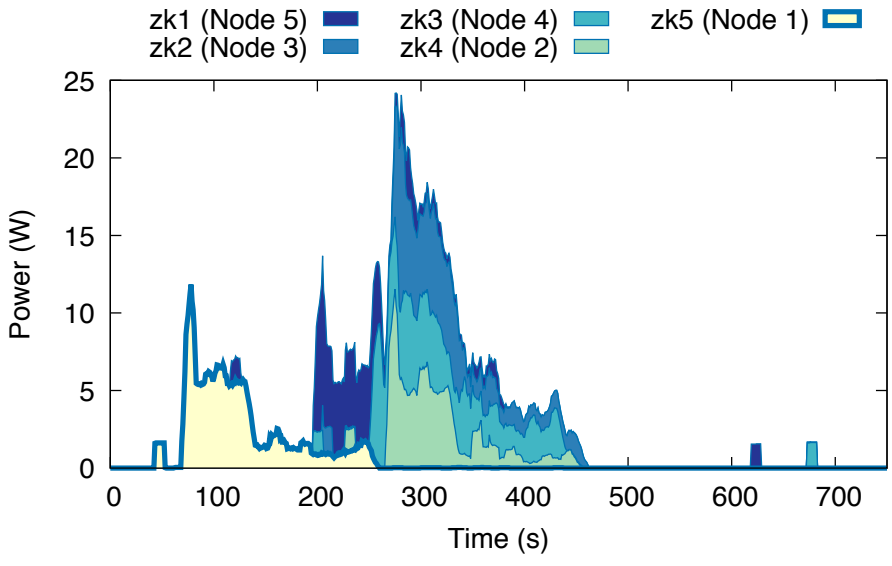

Master
Node 1
Node 2
Node 3
Node 4
Node 5
Power (W)
110
100
90
80
70
60
50
0
100
200
300
400
500
600
700
Time (s)

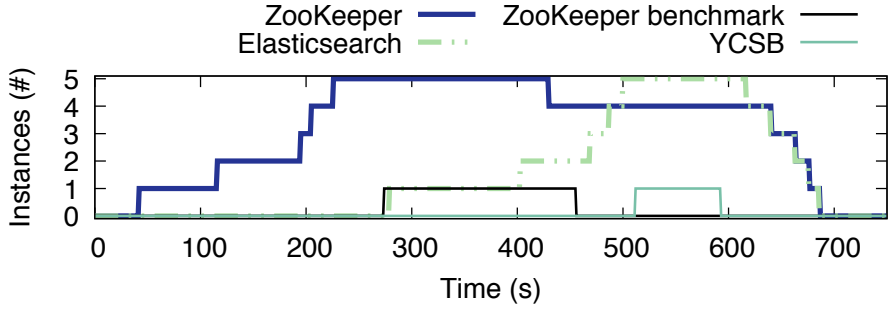

ZooKeeper
Elasticsearch
ZooKeeper benchmark
YCSB
Instances (#)
5
4
3
2
1
0
0
100
200
300
400
500
600
700
Time (s)

WattsKit

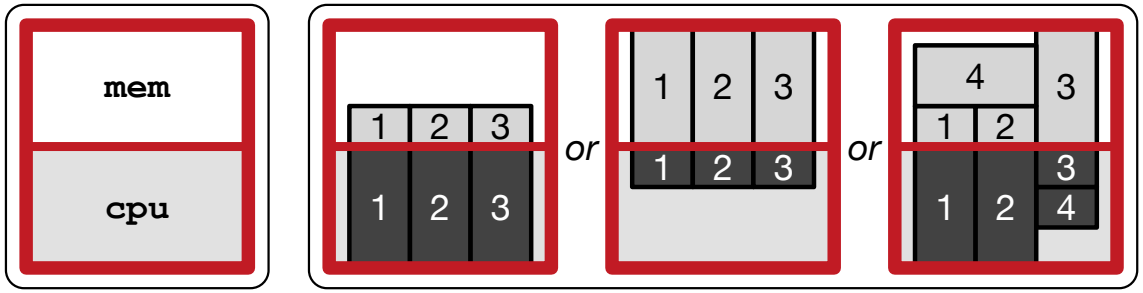

ZooKeeper
Elasticsearch
Weave
Swarm
WattsKit
Power (W)
Time (s)
0
5
10
15
20
25
30
35
40
0
100
200
300
400
500
600
700


ZooKeeper
Elasticsearch
ZooKeeper benchmark
YCSB
Instances (#)
Time (s)
0
1
2
3
4
5
0
100
200
300
400
500
600
700


zk1 (Node 5)
zk2 (Node 3)
zk3 (Node 4)
zk4 (Node 2)
zk5 (Node 1)
Power (W)
Time (s)
0
5
10
15
20
25
0
100
200
300
400
500
600
700


ZooKeeper
Elasticsearch
ZooKeeper benchmark
YCSB
Instances (#)
Time (s)
0
1
2
3
4
5
0
100
200
300
400
500
600
700

WattsKit

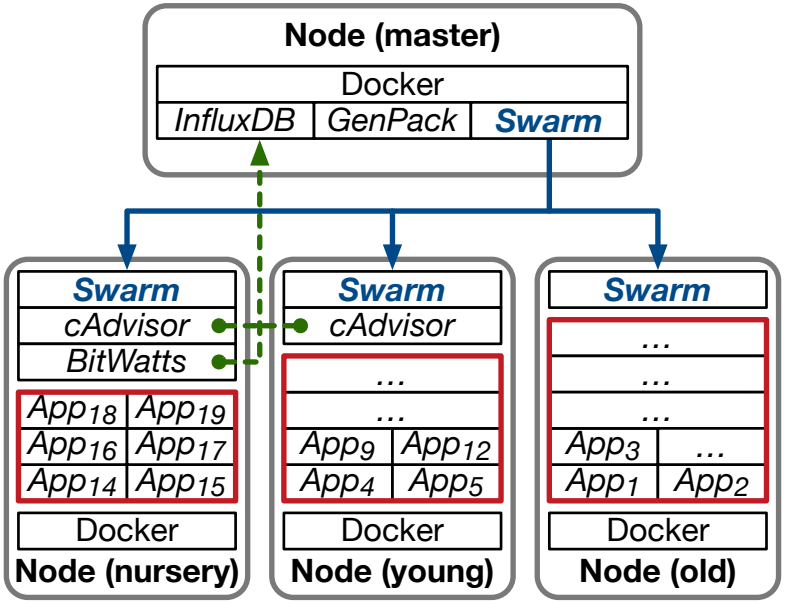


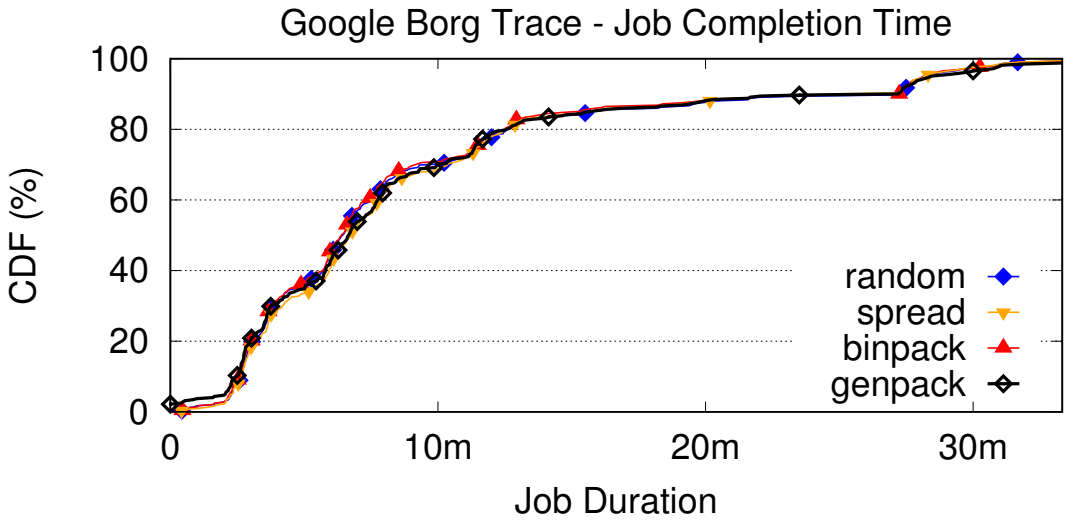
spread

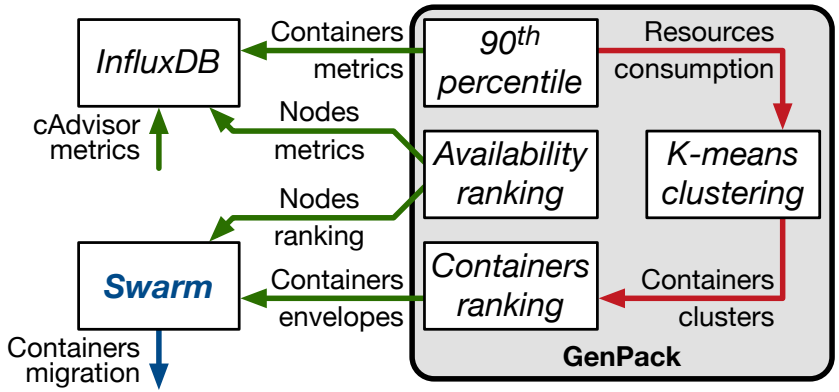
binpack

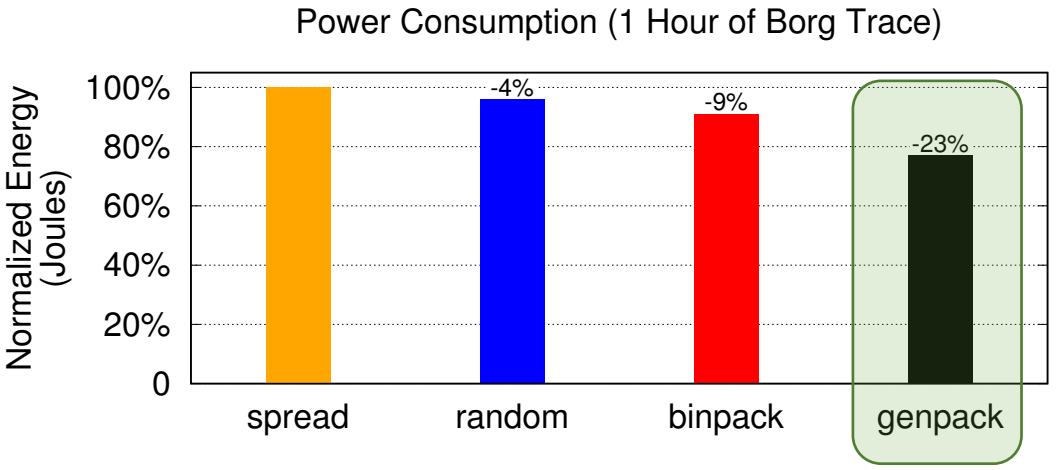
custom

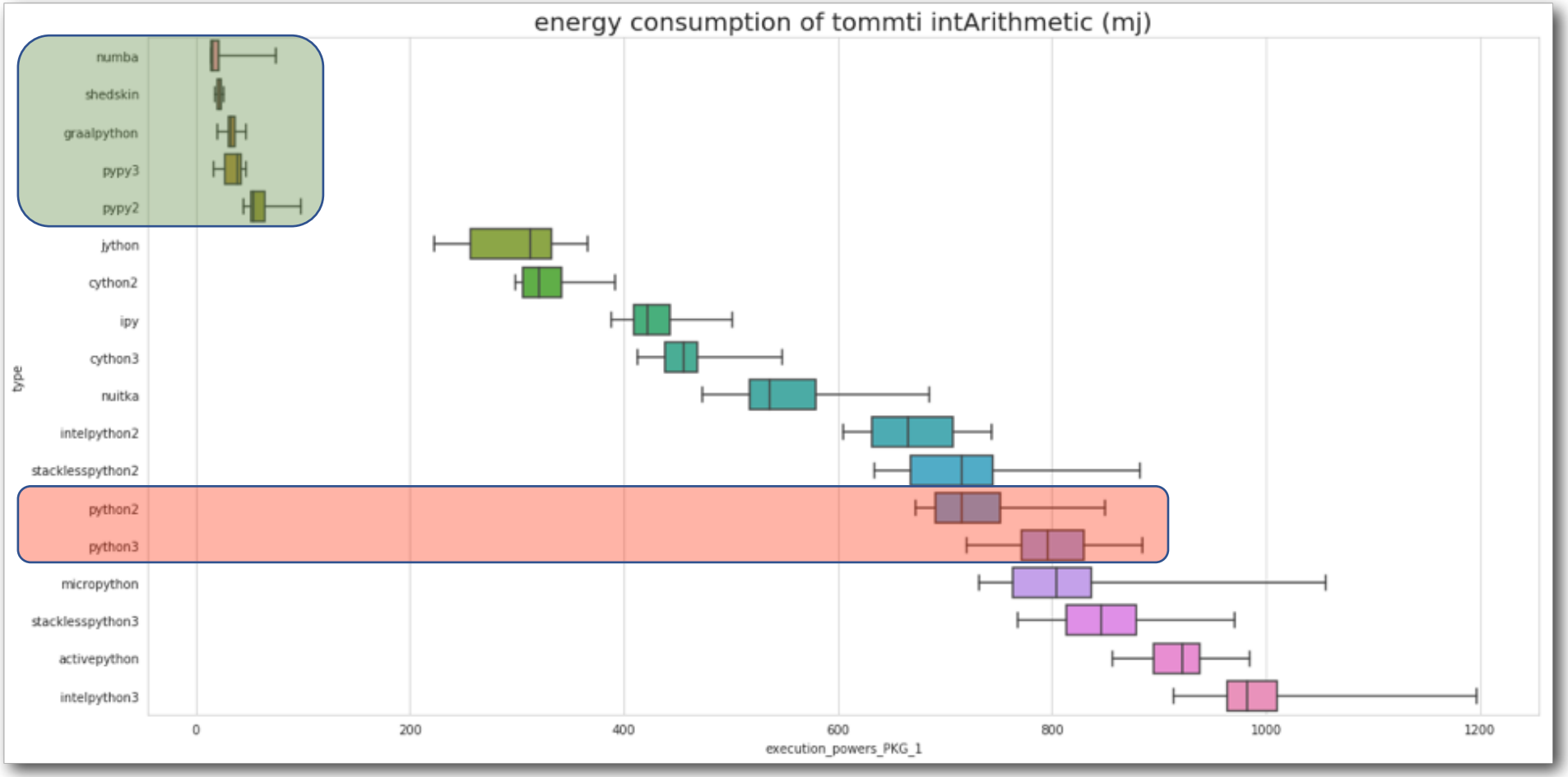
GenPack


Node (master)
Docker
InfluxDB
GenPack
Swarm
Swarm
cAdvisor
BitWatts
App18
App19
App16
App17
App14
App15
Docker
Node (nursery)
Swarm
cAdvisor
...
...
App9
App12
App4
App5
Docker
Node (young)
Swarm
...
...
...
App3
...
App1
App2
Docker
Node (old)


Google Borg Trace - Job Completion Time
CDF (%)
100
80
60
40
20
0
0
10m
20m
30m
Job Duration
random
spread
binpack
genpack

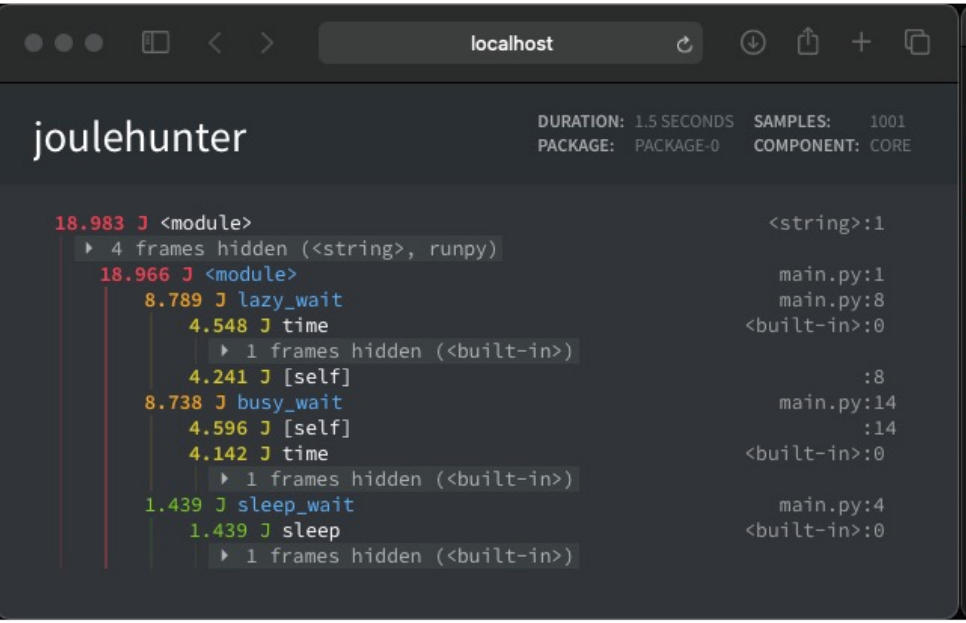

InfluxDB
Containers metrics
90th percentile
Resources consumption
cAdvisor metrics
Nodes metrics
Availability ranking
K-means clustering
Nodes ranking
Swarm
Containers envelopes
Containers ranking
Containers clusters
Containers migration
GenPack

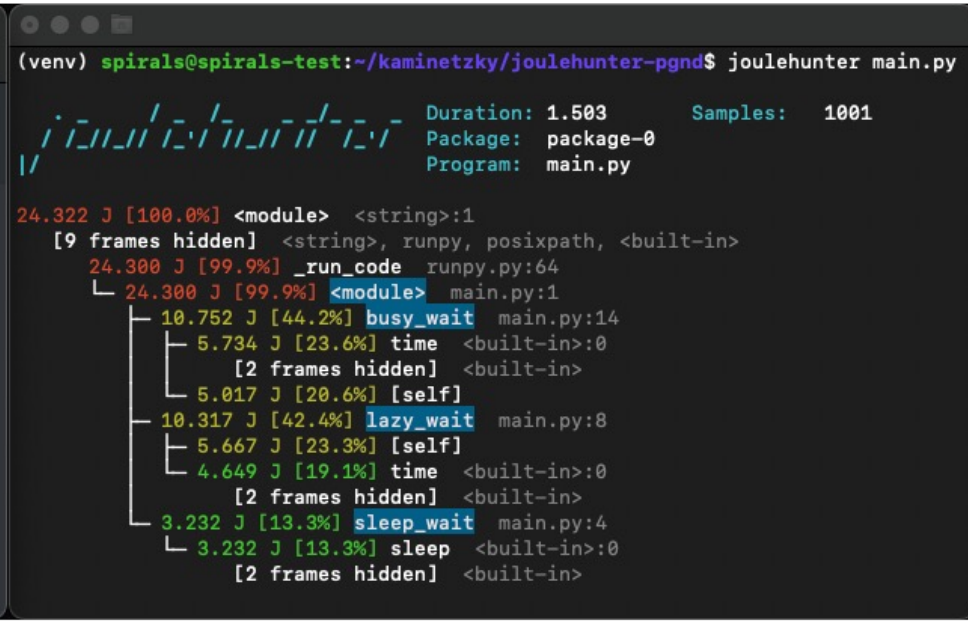

Power Consumption (1 Hour of Borg Trace)
Normalized Energy (Joules)
100%
80%
60%
40%
20%
0
-4%
-9%
-23%
spread
random
binpack
genpack


energy consumption of tommti intArithmetic (mj)
numba
shedskin
graalpython
pypy3
pypy2
jython
cython2
ipy
cython3
nuitka
intelpython2
stacklesspython2
python2
python3
micropython
stacklesspython3
activepython
intelpython3
type
0
200
400
600
800
1000
1200
execution_powers_PKG_1

	source	naive_list_events	optimized_list_events	prefetch_list_events
PKG	**postgres**	41.92453	3.17581	14.85241
	pypysqlitelite	11.81891	1	3.04148
	sqlite	39.78064	3.25259	14.85452
DRAM	**postgres**	57.22666	2.9151	13.87243
	pypysqlitelite	12.78614	1	3.49945
	sqlite	33.46061	2.91994	13.75119
TIME	**postgres**	68.44024	3.15834	14.91648
	pypysqlitelite	12.58561	1	3.06244
	sqlite	39.72118	3.20406	14.88312
AV_POWER	**postgres**	1	1.63947	1.62343
	pypysqlitelite	1.53263	1.63046	1.61948
	sqlite	1.63286	1.6552	1.62725

Energy profiling with **JouleHunter**





https://pypi.org/project/joulehunter/

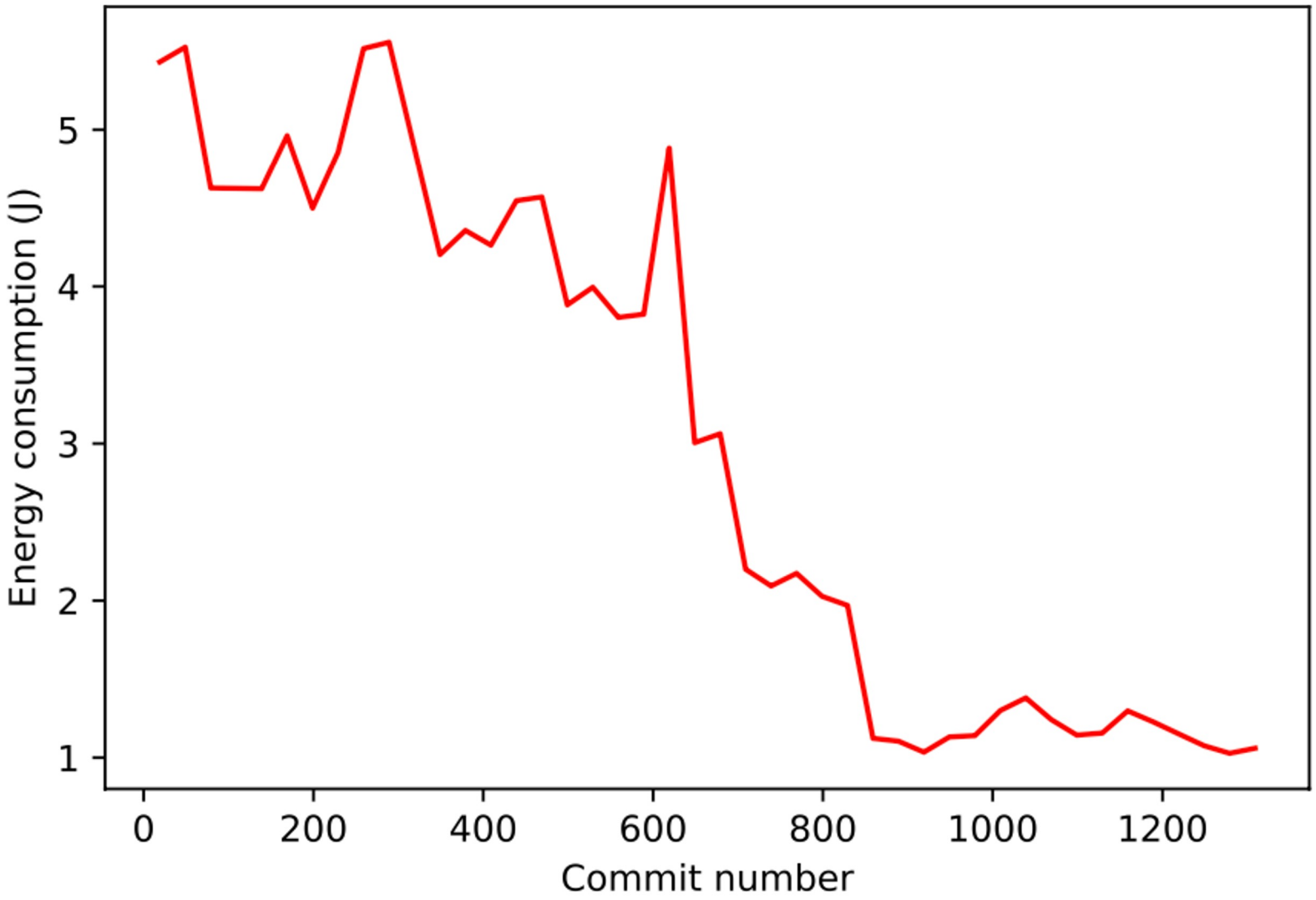
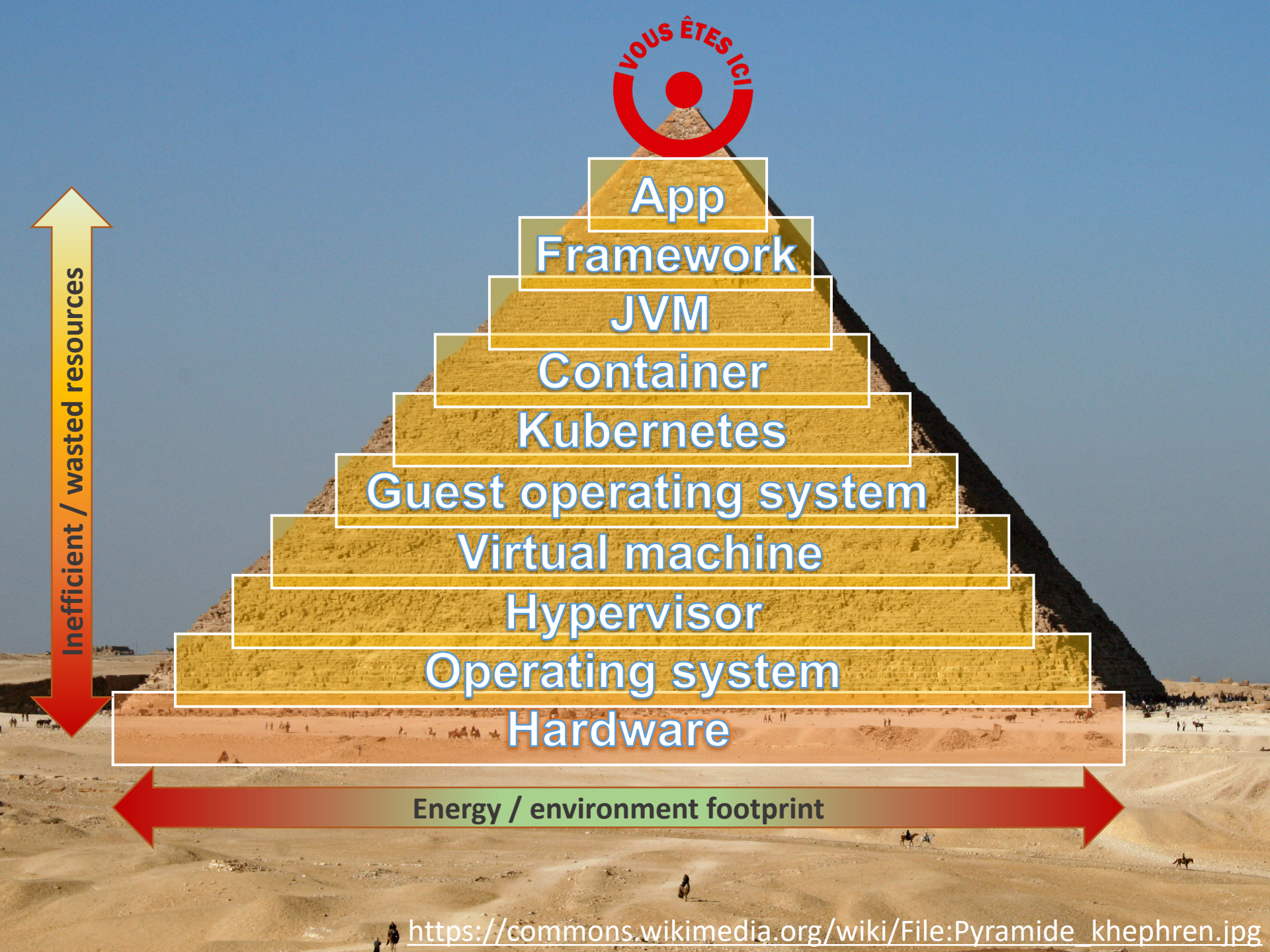


Fig. 4: Gson energy consumption across all commits.

Z. Ournani, R. Rouvoy, P. Rust, J. Penhoat. **Tales from the Code #1: The Effective Impact of Code Refactorings on Software Energy Consumption**. 16th International Conference on Software Technologies (ICSOFT), July 2021.



https://commons.wikimedia.org/wiki/File:Pyramide_khephren.jpg

Take away

- ICT energy consumption will keep growing at e^t
  - More and more digital services (no matter the domain)
- Hardware keeps improving energy efficiency
  - But hardware keeps being driven by software
- **Computing resources are going to be limited**
  - One really need to better control computations
- Needs to work on all the layers of an infrastructure
  - Each layer = a software to optimize

The Green Side of the Force

Lionel **Seinturier**

Université de Lille

Daniel **Romero**

Inria

Guillaume **Fieni**

Université de Lille

Chakib **Belgaid**

DAVIDSON CONSULTING

Pierre **Jacquet**

Inria

Thibault **Simon**

orange™

Emile **Cadorel**

Inria

Lauric **Desauw**

Inria

Zakaria **Ournani**

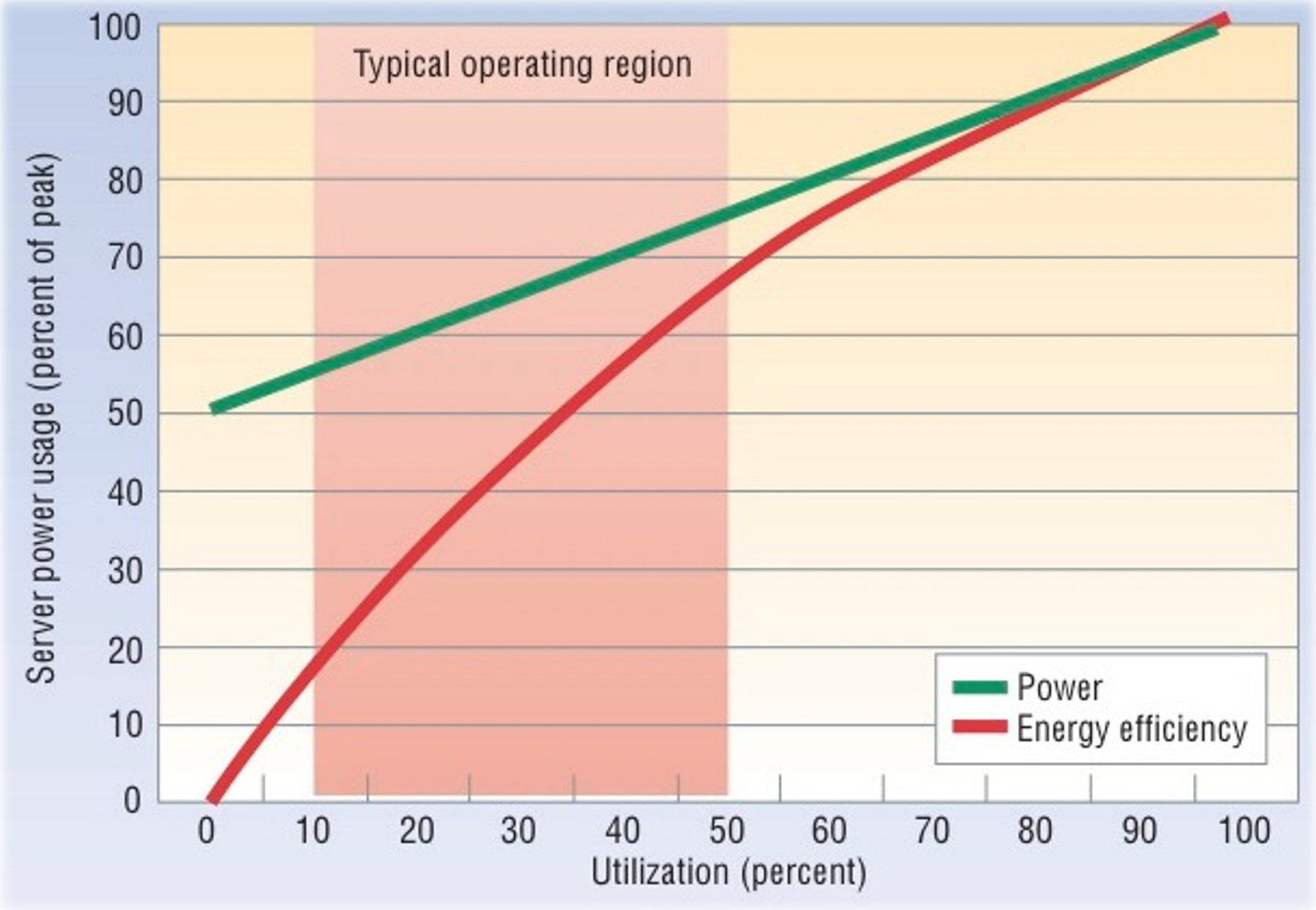
orange™

Adel **Noureddine**

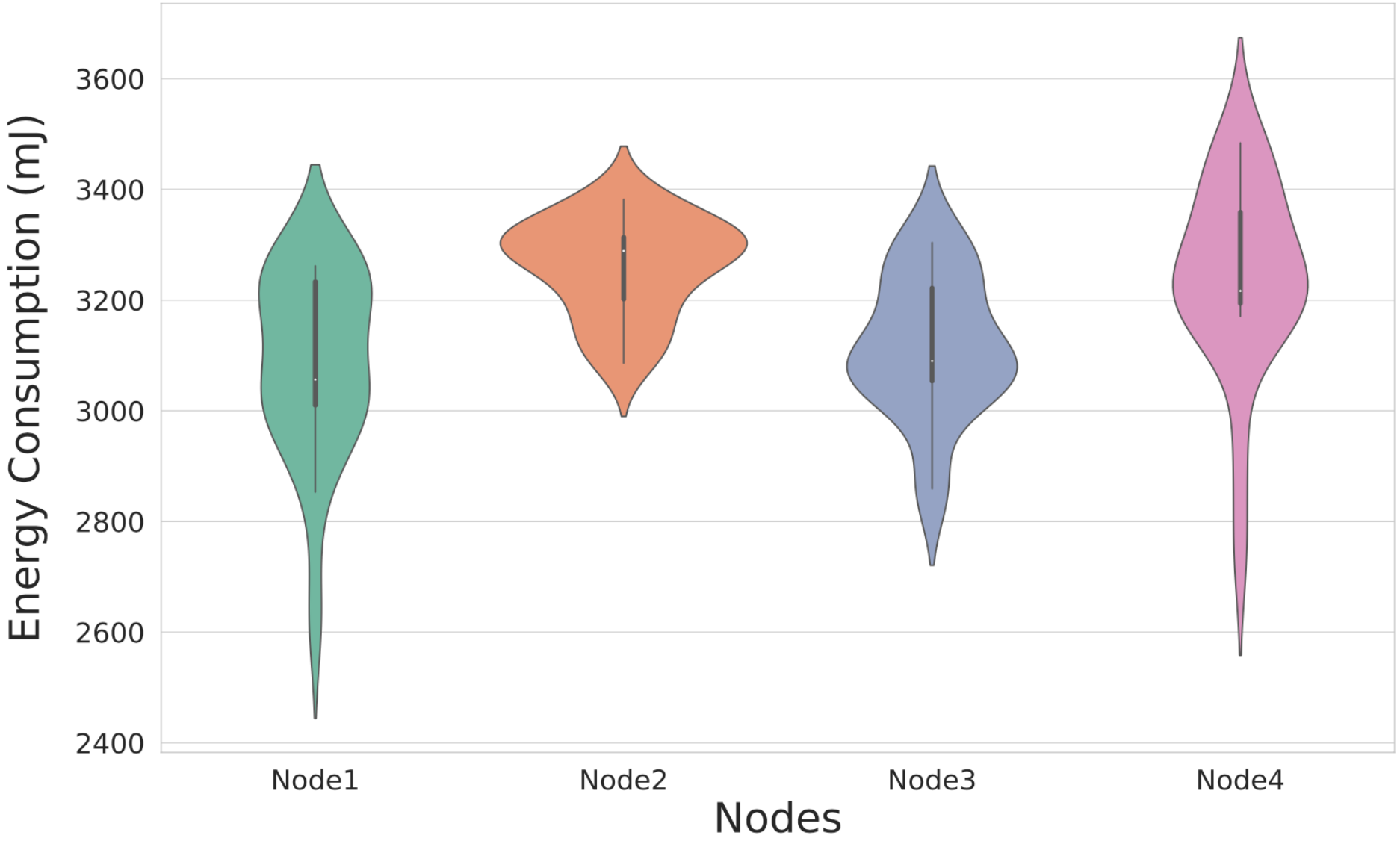
Inria

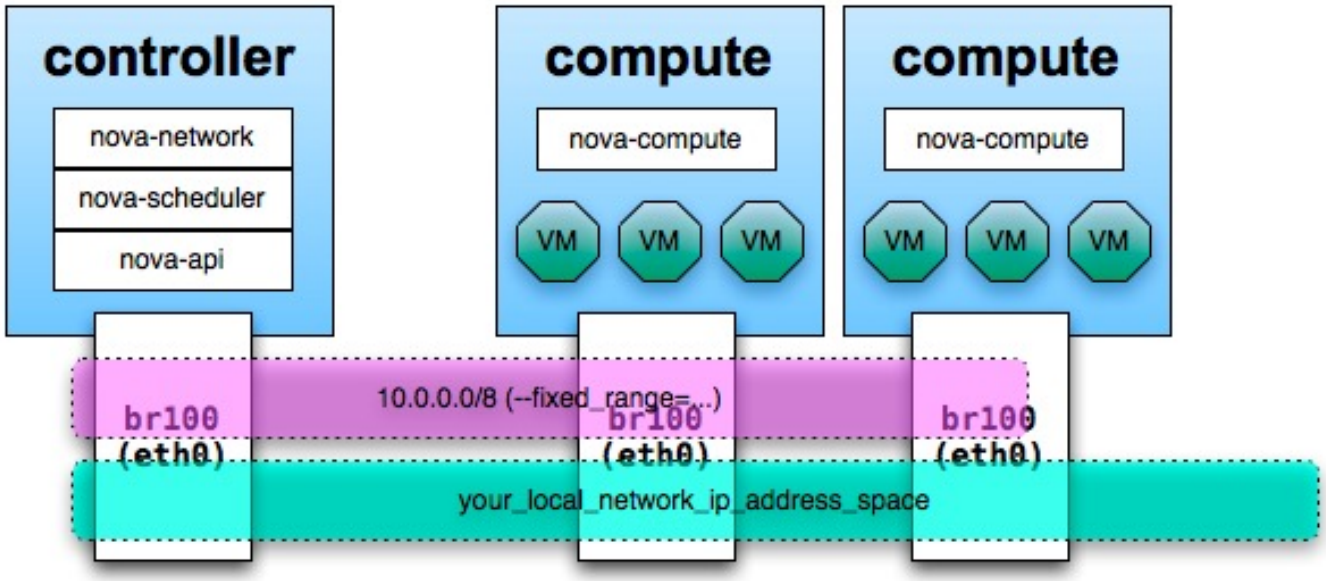
Pascal **Felber**, *Pierre* **Rust**, *Bo* **Zhang**, *Aurélien* **Havet**, *Mascha* **Kurpicz**, *Valerio* **Schiavoni**, *Anita* **Sobe**, *Christof* **Fetzer,** *Yahya* **Al-Dhuraibi**, *Fawaz* **Paraiso**, *Georges-Aaron* **Randrianaina,** *Antoine* **Huyghes,** *Arthur* **D'Azémar,** *Jordan* **Bouchoucha,** *Maxime* **Colmant,** *Loïc* **Huertas,** *Aurélien* **Bourdon**…

1. **Comparing the Energy Consumption of Java I/O Libraries and Methods.** Z. Ournani, R. Rouvoy, P. Rust, J. Penhoat: *ICSME'21.*
2. **Evaluating the Impact of Java Virtual Machines on Energy Consumption.** Z. Ournani, M.C. Belgaid, R. Rouvoy, P. Rust, J. Penhoat: *ESEM'21.*
3. ***On Reducing the Energy Consumption of Software Product Lines.*** É. Guégain, C. Quinton, R. Rouvoy: *SPLC'21.*
4. ***Tales from the Code #1: The Effective Impact of Code Refactorings on Software Energy Consumption.*** Z. Ournani, R. Rouvoy, P. Rust, J. Penhoat: *ICSOFT'21.*
5. **SelfWatts: On-the-fly Selection of Performance Events to Optimize Software-defined Power Meters.** G. Fieni, R. Rouvoy, L. Seinturier: *CCGrid'21.*
6. **SmartWatts: Self-Calibrating Software-Defined Power Meter for Containers.** G. Fieni, R. Rouvoy, L. Seinturier: *CCGrid'20.*
7. **On Reducing the Energy Consumption of Software: From Hurdles to Requirements.** Z. Ournani, R. Rouvoy, P. Rust, J. Penhoat: *ESEM'20.*
8. **Power Budgeting of Big Data Applications in Container-based Clusters.** J.Enes, G. Fieni, R. Expósito, R. Rouvoy, J. Tourino: *CLUSTER'20.*
9. **Taming Energy Consumption Variations in Systems Benchmarking.** Z. Ournani, M. C. Belgaid, R. Rouvoy, P. Rust, J. Penhoat, L. Seinturier. *ICPE'20*.
10. **The next 700 CPU power models**. M. Colmant, R.Rouvoy, M. Kurpicz, A. Sobe, P. Felber, L. Seinturier: *Journal of Systems and Software* 144: 382-396 (2018)
11. **WattsKit: Software-Defined Power Monitoring of Distributed Systems**. M. Colmant, P. Felber, R. Rouvoy, L. Seinturier: *CCGrid'17*
12. **GenPack: A Generational Scheduler for Cloud Data Centers**. A. Havet, A. Schiavoni, P. Felber, M. Colmant, R. Rouvoy, C. Fetzer: *IC2E'17*
13. **CloudGC: Recycling Idle Virtual Machines in the Cloud**. B. Zhang, Y. Al-Dhuraibi, R. Rouvoy, F. Paraiso, L. Seinturier: *IC2E'17*
14. **Process-level power estimation in VM-based systems**. M. Colmant, M. Kurpicz, P. Felber, L. Huertas, R. Rouvoy, A. Sobe: *EuroSys'15*
15. **Unit testing of energy consumption of software libraries**. A. Noureddine, R. Rouvoy, L. Seinturier: *SAC'14*
16. **A preliminary study of the impact of software engineering on GreenIT**. A. Noureddine, A. Bourdon, R. Rouvoy, L. Seinturier: *GREENS'12*
17. **Runtime monitoring of software energy hotspots**. A. Noureddine, A. Bourdon, R. Rouvoy, L. Seinturier: *ASE'12*

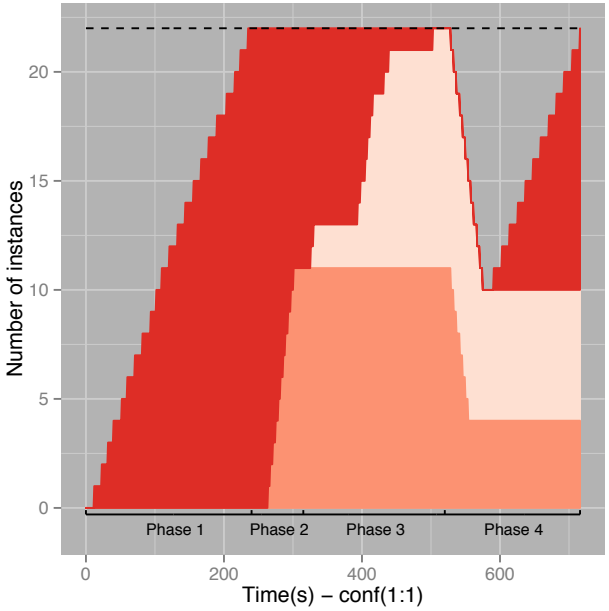


L. A. Barroso, U. Hölzle: *The Case for Energy-Proportional Computing*. IEEE Computer 40(12) 2007

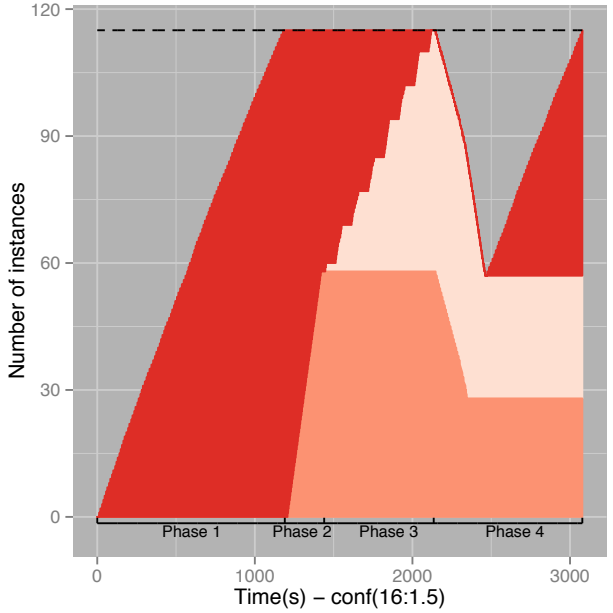

Energy Consumption (mJ)
3600
3400
3200
3000
2800
2600
2400
Node1
Node2
Node3
Node4
Nodes



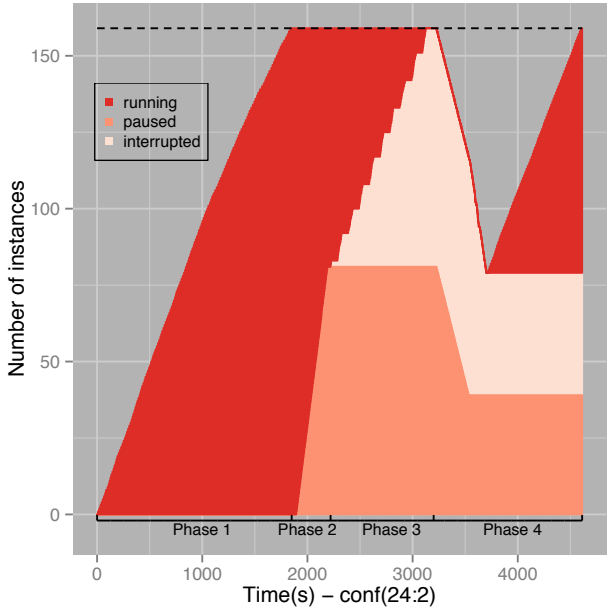
openstack®



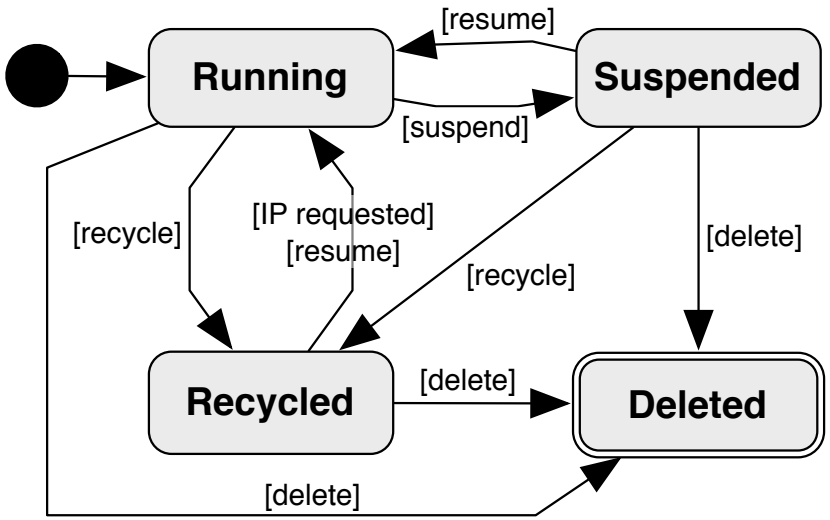
(a) vCPU limitation of straight

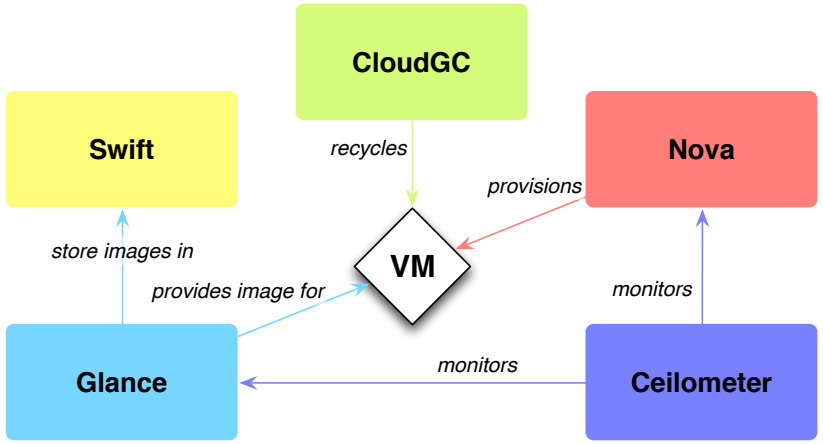


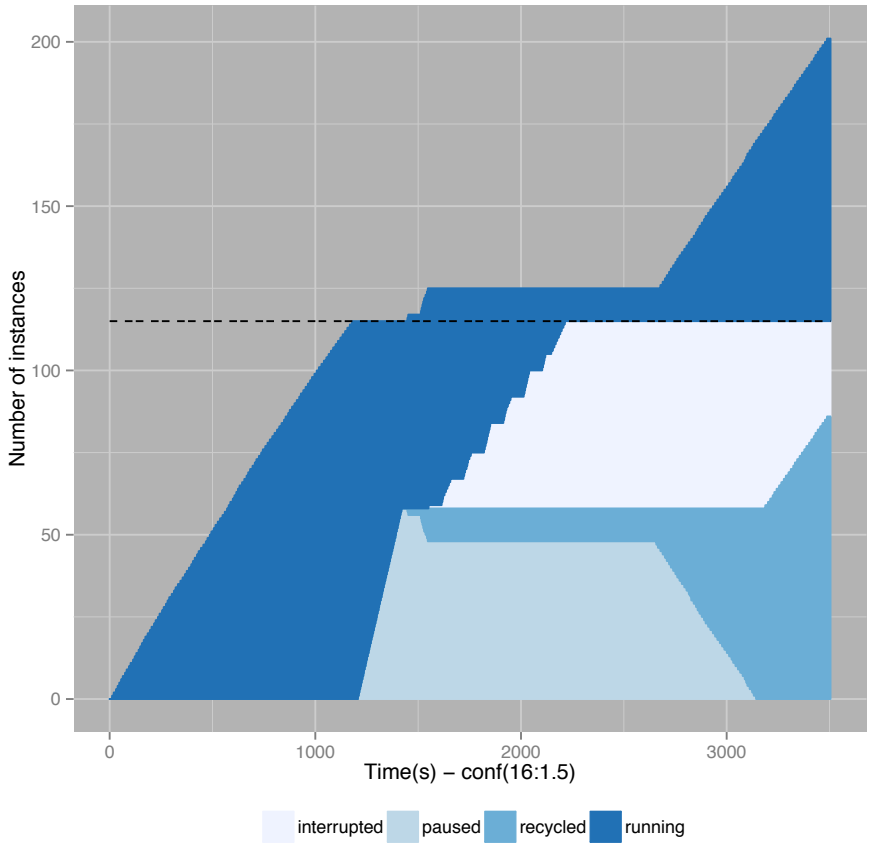
(b) vRAM limitation of standard

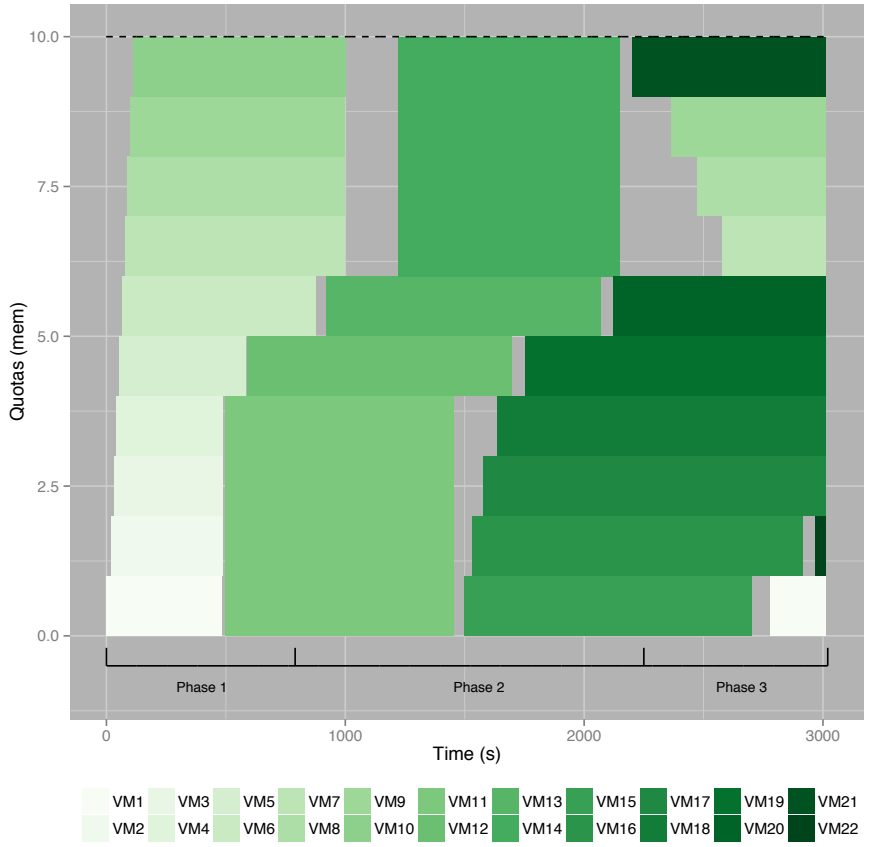


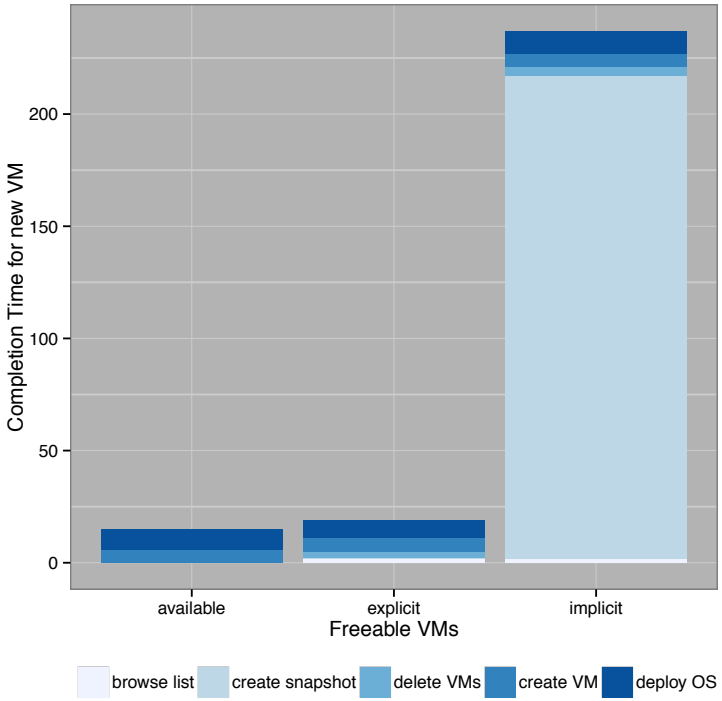
(c) vRAM limitation of over-commit


Running
Suspended
Recycled
Deleted
[resume]
[suspend]
[recycle]
[IP requested]
[resume]
[recycle]
[delete]
[delete]
[delete]

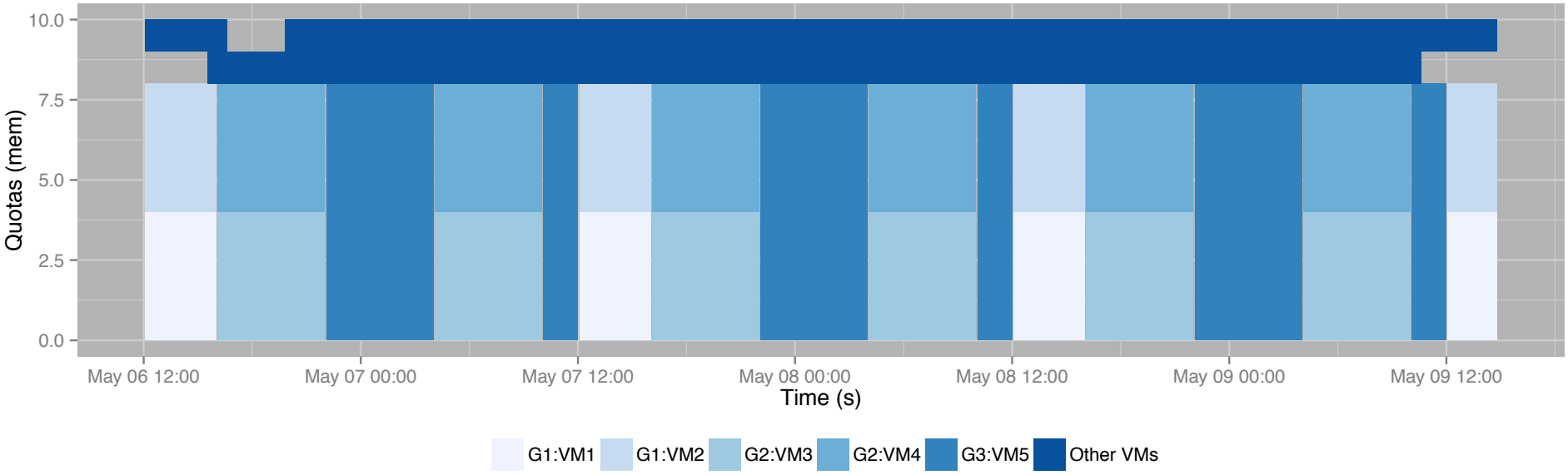

CloudGC
Swift
Nova
recycles
provisions
store images in
VM
provides image for
monitors
Glance
monitors
Ceilometer


Number of instances
200
150
100
50
0
0
1000
2000
3000
Time(s) – conf(16:1.5)
interrupted
paused
recycled
running


Quotas (mem)
10.0
7.5
5.0
2.5
0.0
Phase 1
Phase 2
Phase 3
0
1000
2000
3000
Time (s)
VM1
VM2
VM3
VM4
VM5
VM6
VM7
VM8
VM9
VM10
VM11
VM12
VM13
VM14
VM15
VM16
VM17
VM18
VM19
VM20
VM21
VM22



operation	available	explicit	implicit
browse list	-	2 *sec*	2 *sec*
create snapshot	-	-	215 *sec*
delete instance	-	3 *sec*	3 *sec*
create instance	6 *sec*	6 *sec*	6 *sec*
deploy OS	9 *sec*	9 *sec*	9 *sec*
total	15 *sec*	20 *sec*	235 *sec*



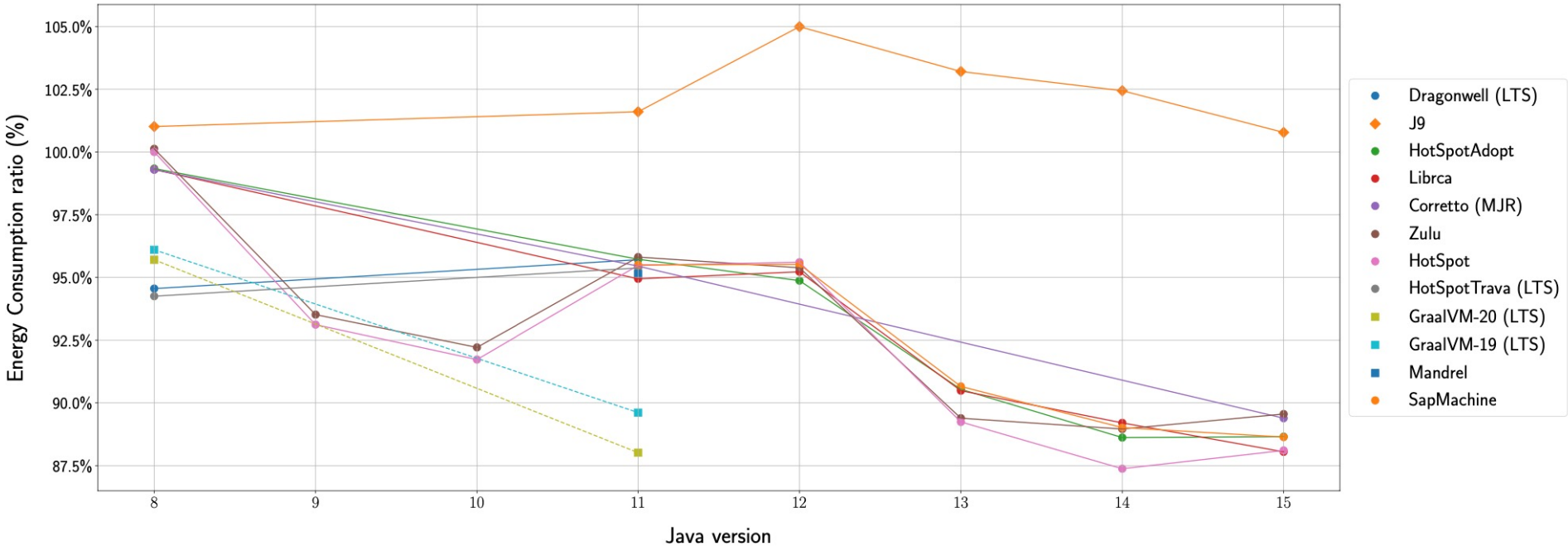
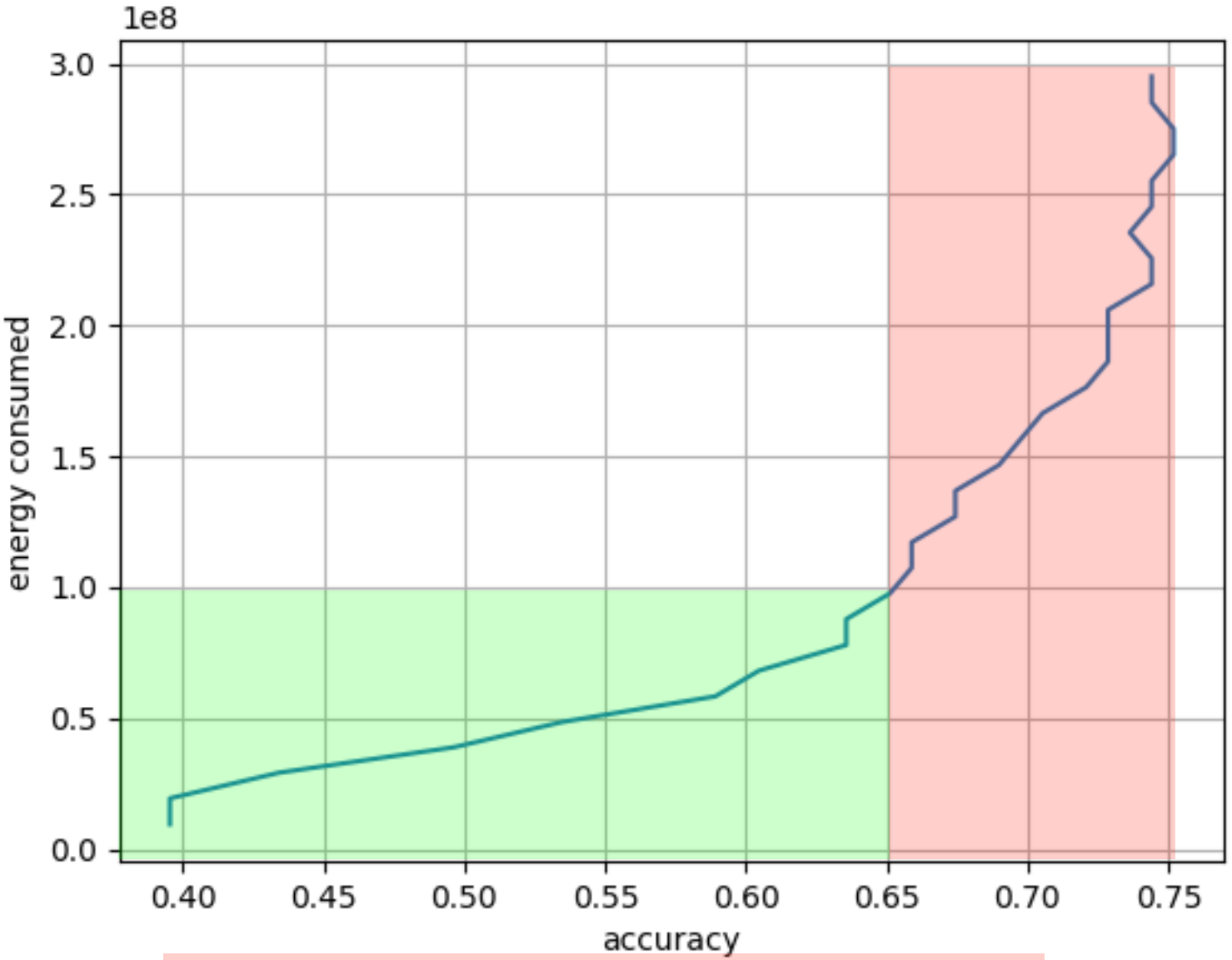


Figure 1: Energy consumption evolution of selected JVM distributions along versions.

Z. Ournani, M.C. Belgaid, R. Rouvoy, P. Rust, J. Penhoat. **Evaluating the Impact of Java Virtual Machines on Energy Consumption**. *15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2021. https://github.com/chakib-belgaid/jreferral

Cost of training a SVM classifier



2x plus d'énergie pour gagner 0,1 de précision…

https://scikit-learn.org/stable/auto_examples/applications/plot_face_recognition.html