How green is your cloud?

From measurements to actionable insights…

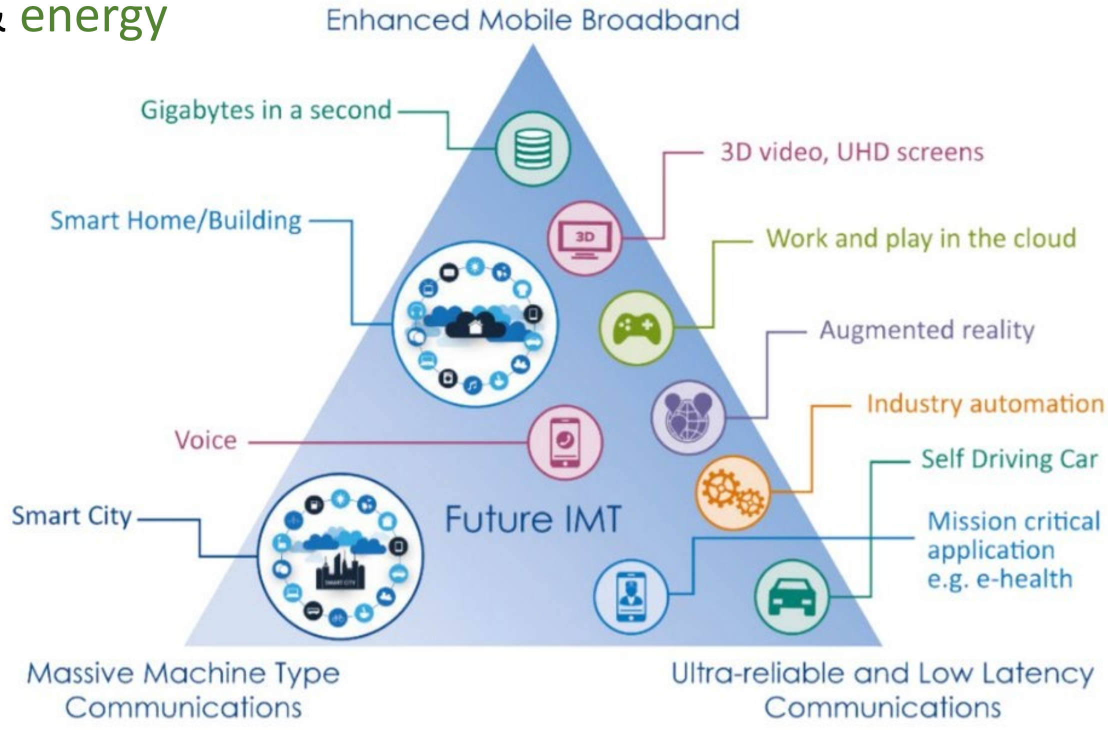
@RomainROUVOY

SPIRALS
Université de Lille
Inria
institut universitaire de France

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

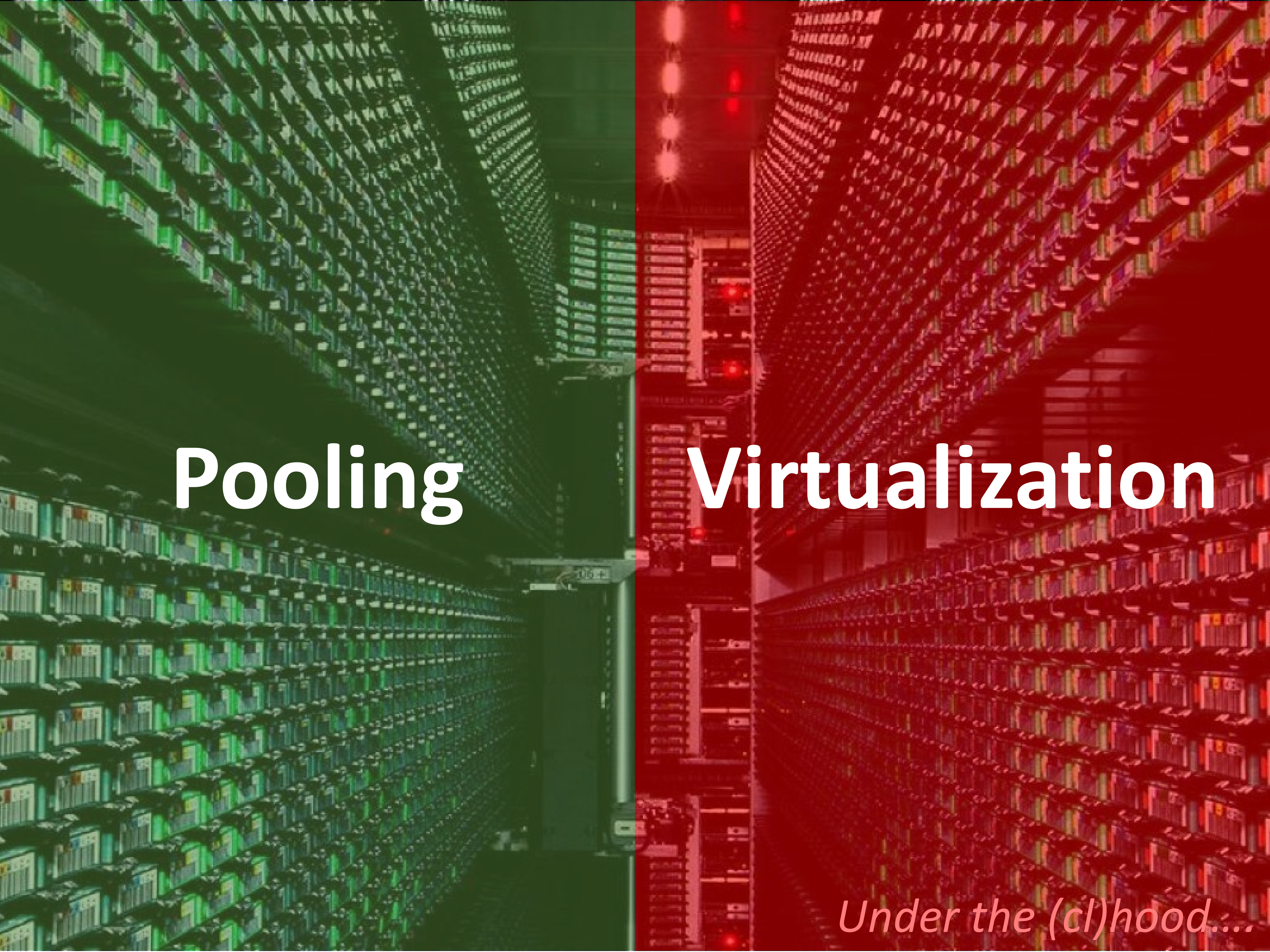
Agence de l'Environnement
et de la Maîtrise de l'Energie

DAVIDSON
CONSULTING

orange™

OVHcloud

IMPACT
ENVIRONNEMENTAL


Pooling
Virtualization
Under the (cl)hood…

Joe Armstrong
@joeerl

Should also add that all significant energy gains in the last 50 odd years are result of new hardware NOT software.

> **Joe Armstrong** @joeerl
> Replying to @emidttun and 2 others
>
> Energy usage is *very* complicated - If you want low energy use VLSI or an FPGA and NOT a programming language - true total lifecycle energy costs are very very difficult to calculate - more of a physics/hardware question than a programming problem.

♡ 196 4:43 PM - Apr 10, 2019

45 people are talking about this

« *Why software is eating the world* » *(M. Andreesen, WSJ, 2011)*

Source: https://www.wsj.com/articles/SB10001424053111903480904576512250915629460

What about software sustainability??

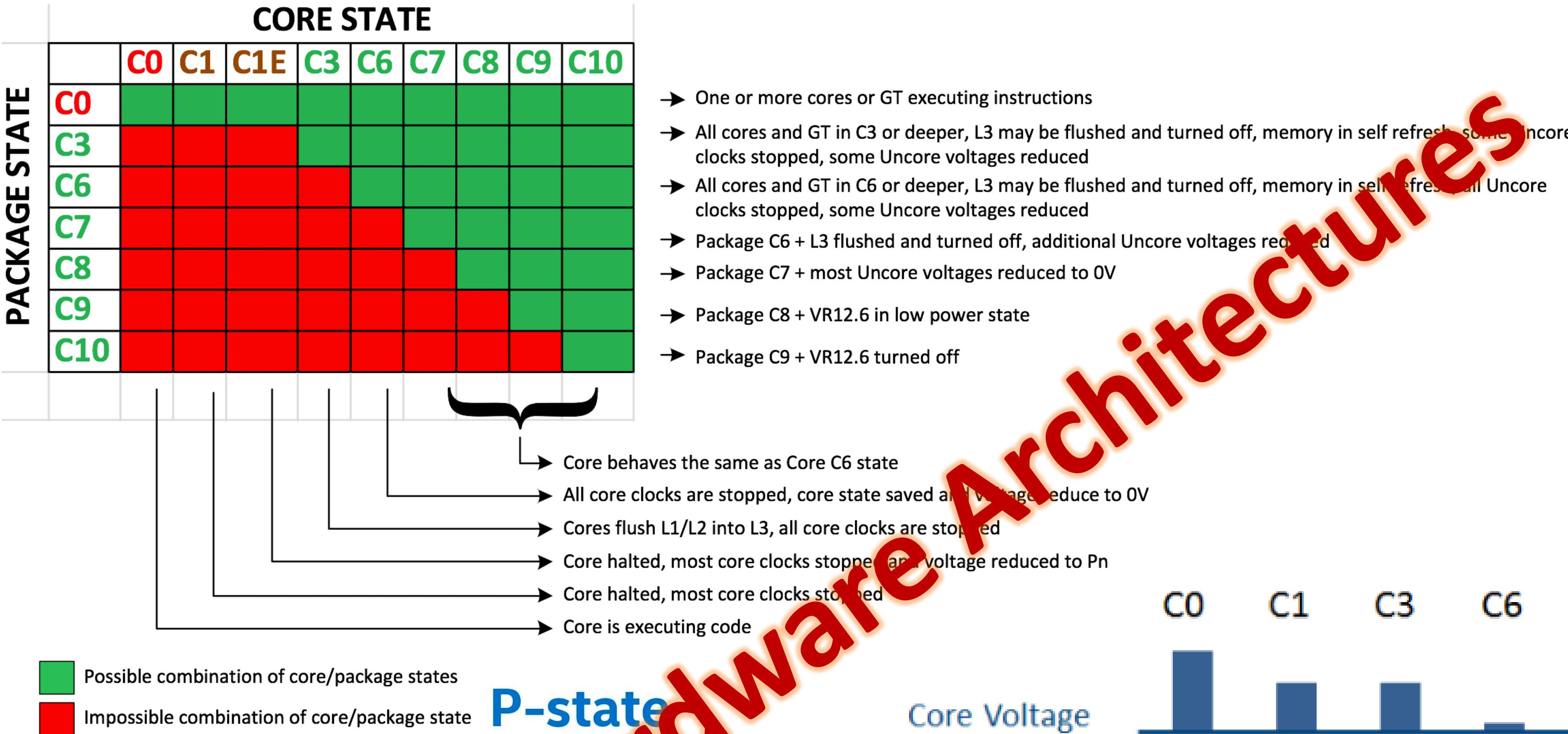
« *These results show that these programmers lacked knowledge of how to* ***accurately measure software energy consumption***. »

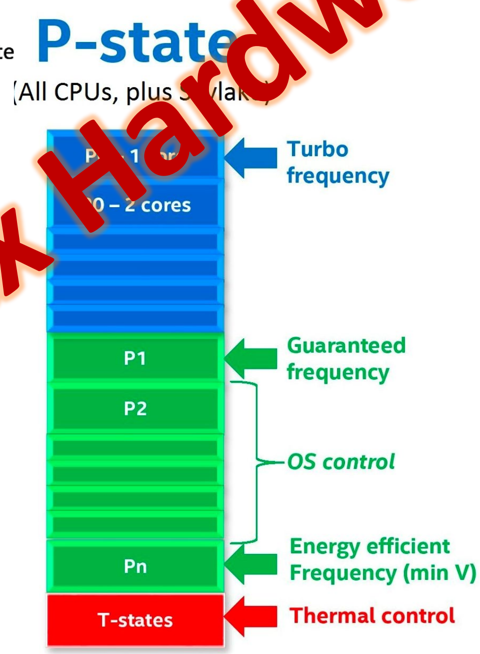
PowerAPI

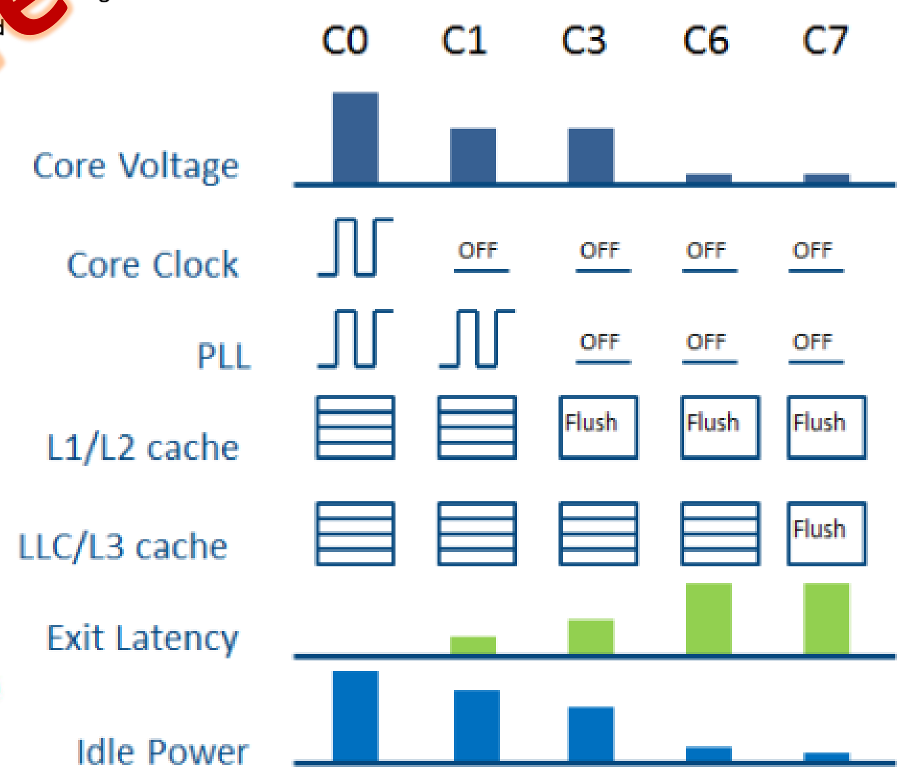
Enabling power monitoring of software systems

http://powerapi.org

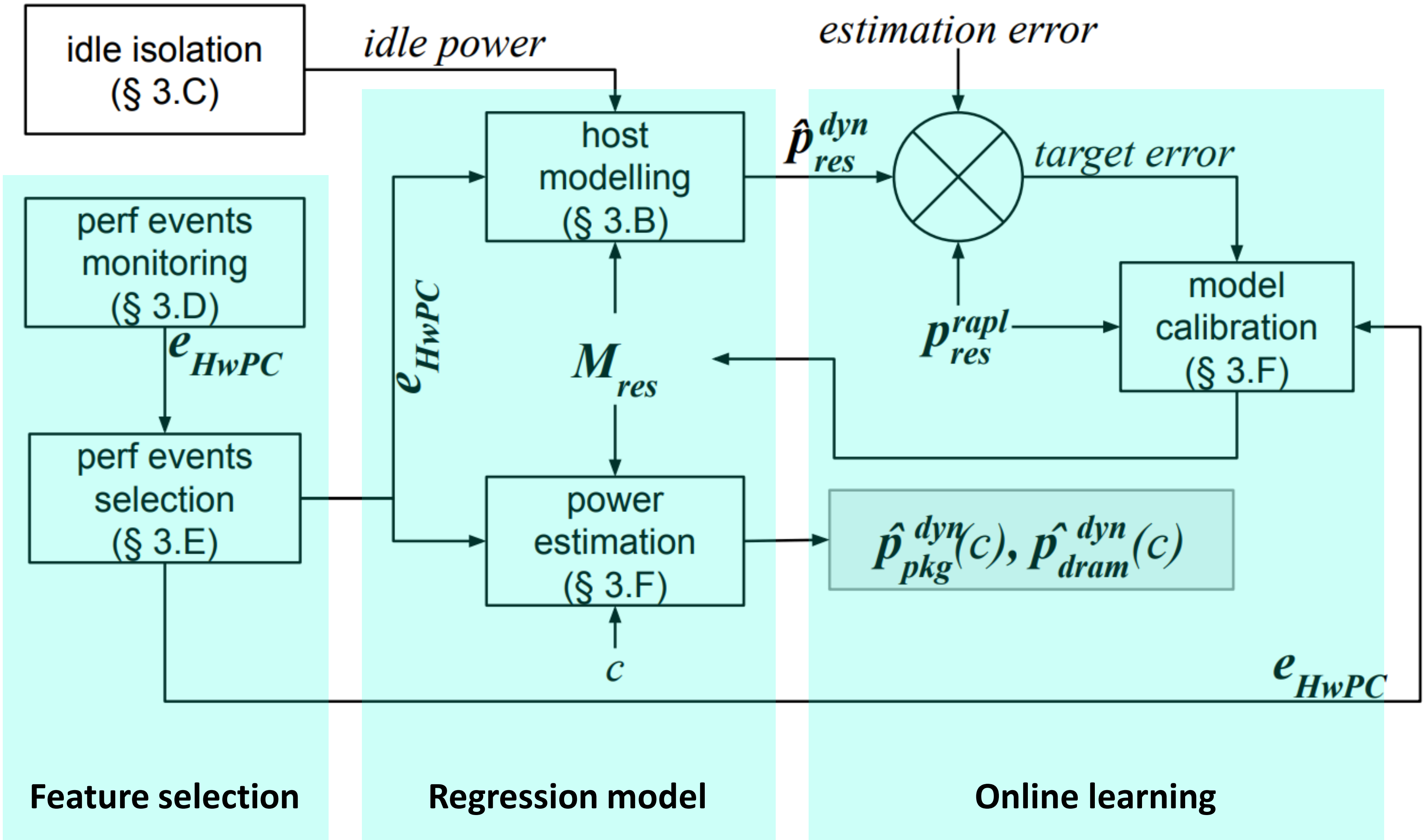
Complex Hardware Architectures





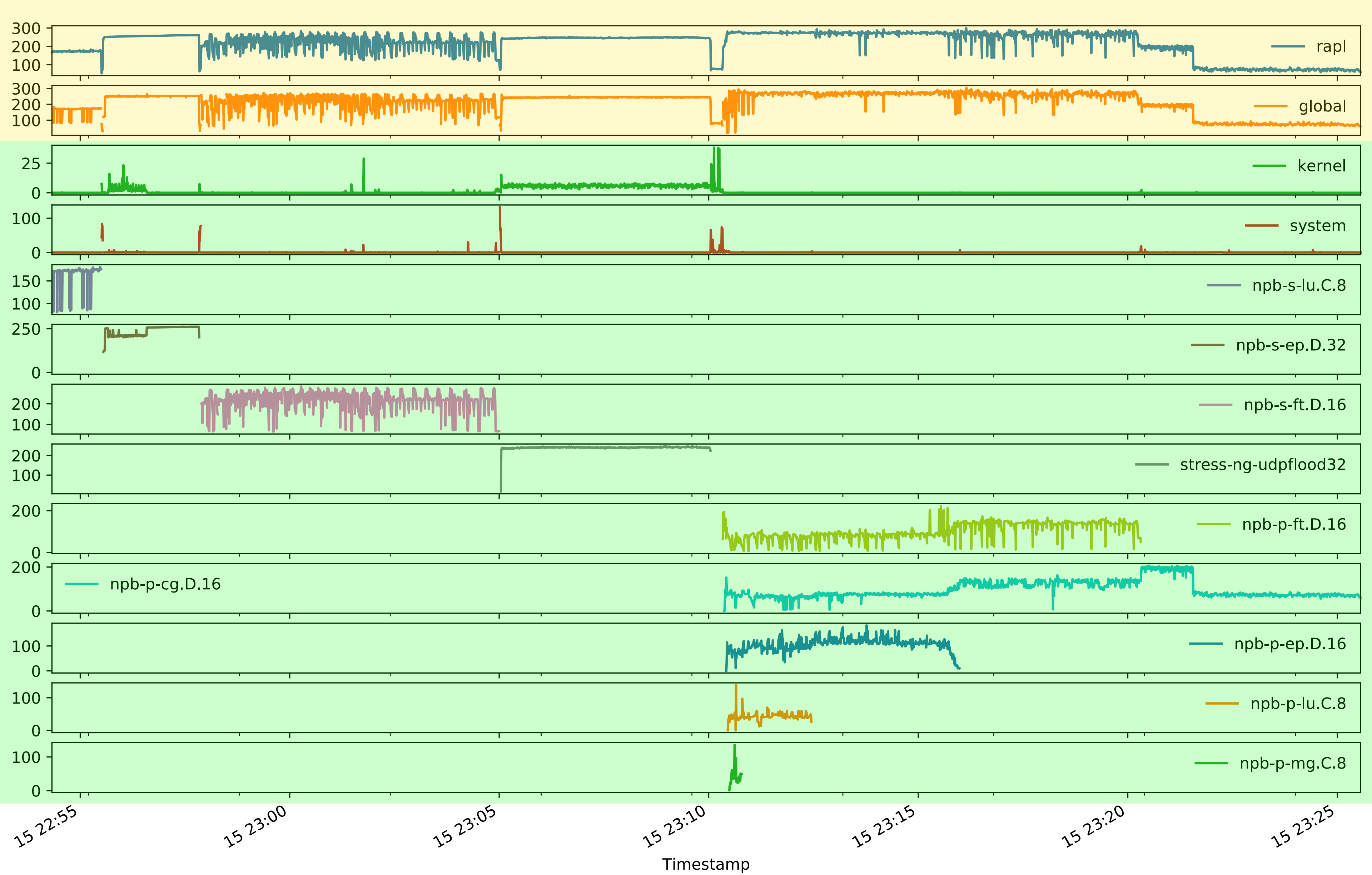


Learning the CPU/DRAM power models from RAPL



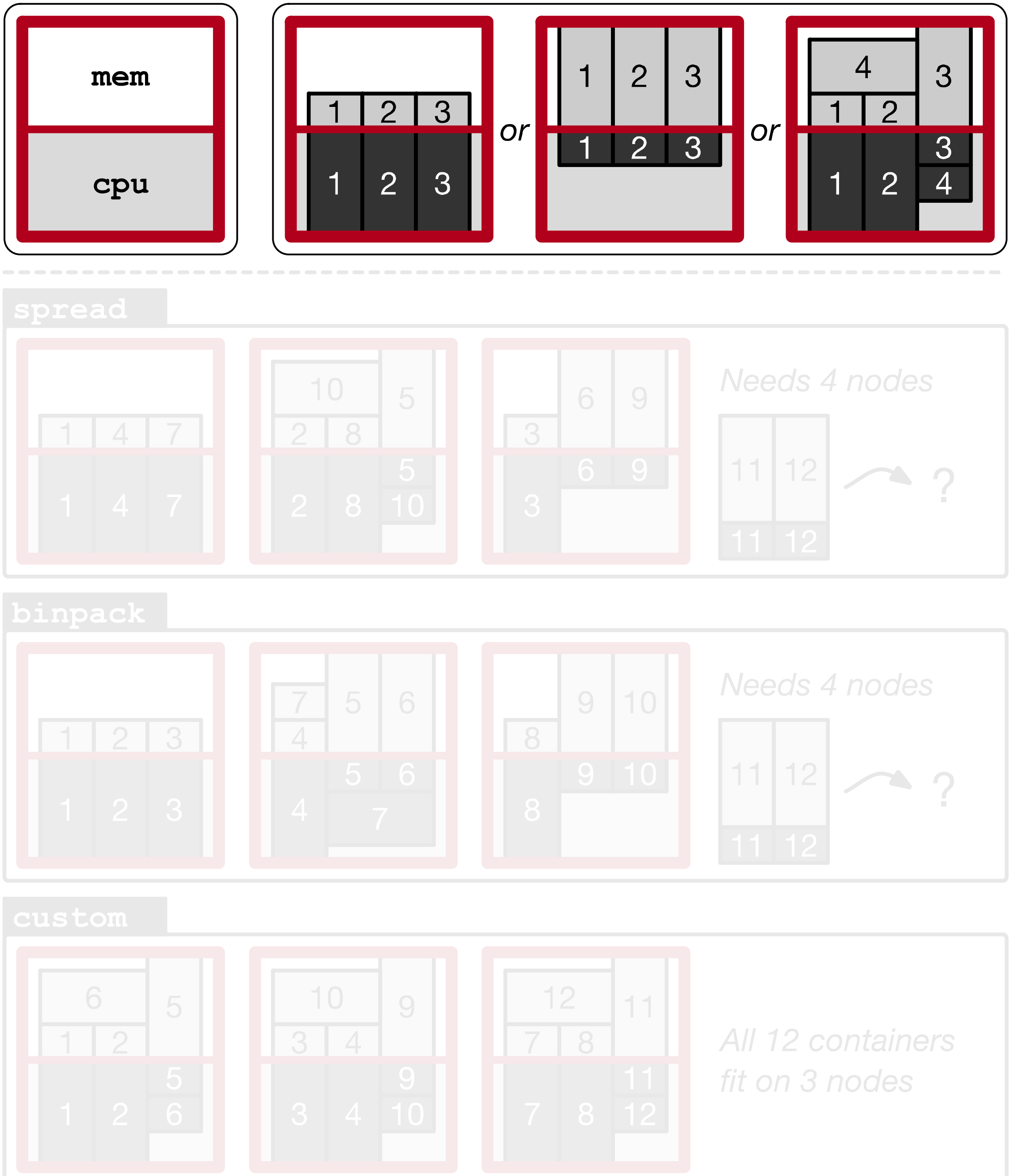
SmartWatts

Monitoring the power consumption in real-time

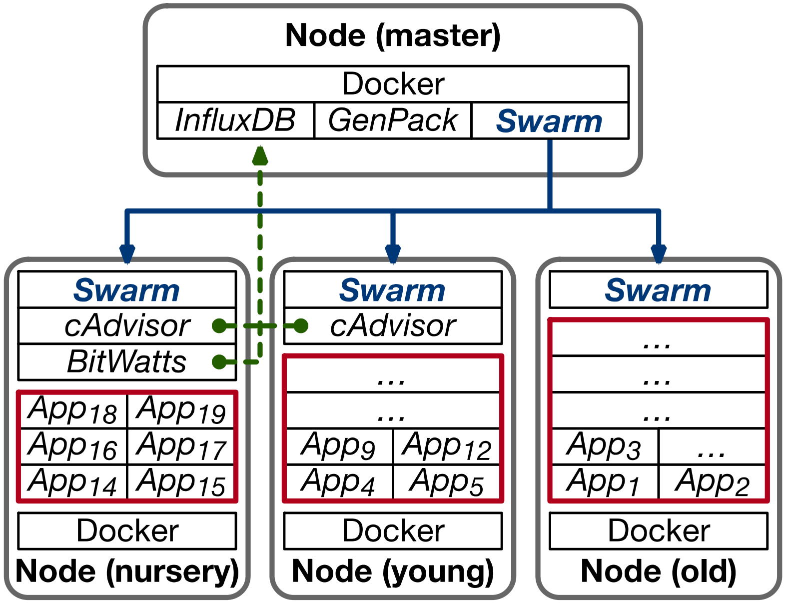


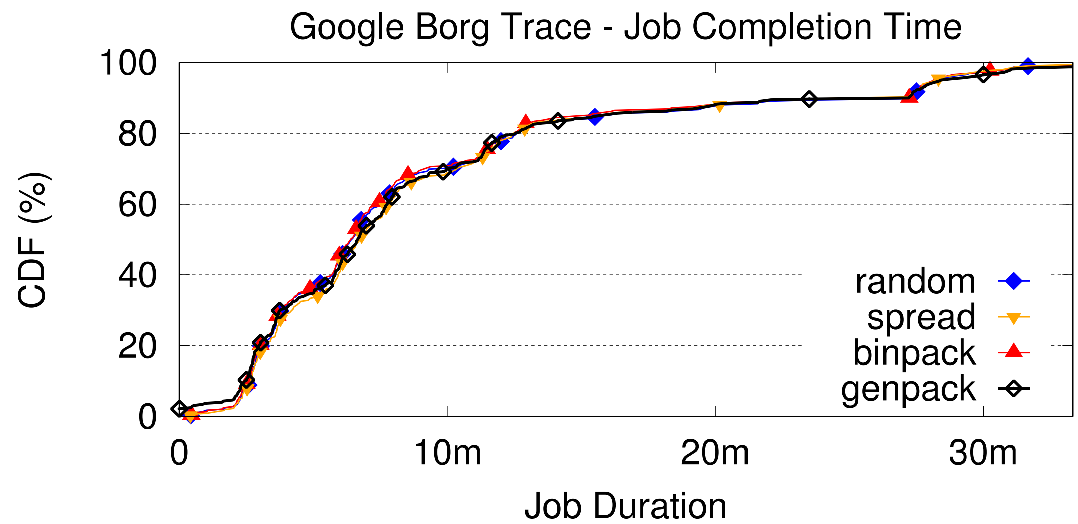
SmartWatts

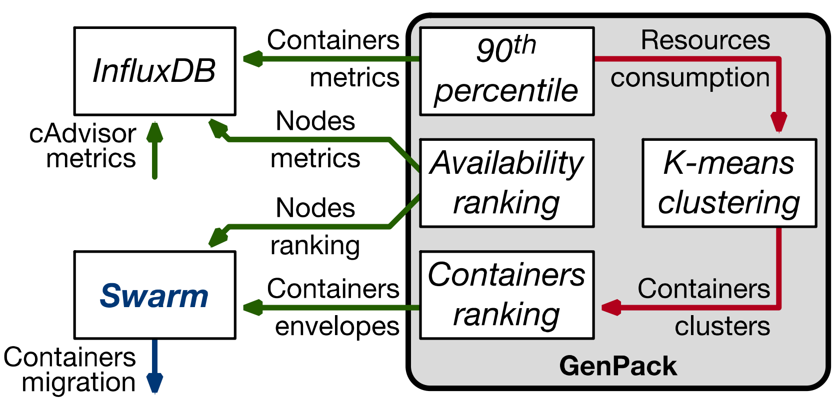

VOUS ÊTES ICI
Software (App)
Software (Framework)
Software (JVM)
Software (Container)
Software (OS)
Software (VM)
Software (Hypervisor)
Software (OS)
Hardware

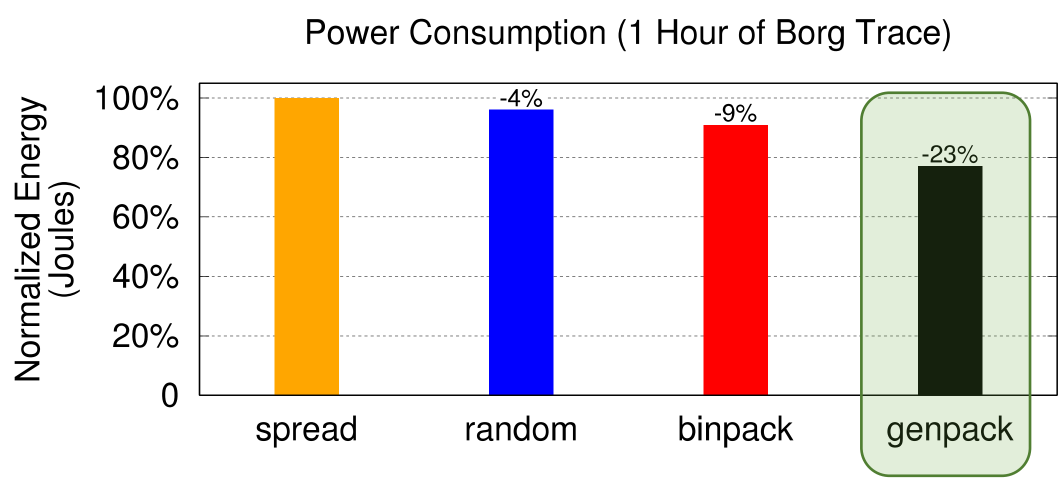

mem
cpu
or
or
spread
Needs 4 nodes
binpack
Needs 4 nodes
custom
All 12 containers fit on 3 nodes

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


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

GenPack

	Energy
(c) C	1.00
(c) Rust	1.03
(c) C++	1.34
(c) Ada	1.70
(v) Java	1.98
(c) Pascal	2.14
(c) Chapel	2.18
(v) Lisp	2.27
(c) Ocaml	2.40
(c) Fortran	2.52
(c) Swift	2.79
(c) Haskell	3.10
(v) C#	3.14
(c) Go	3.23
(i) Dart	3.83
(v) F#	4.13
(i) JavaScript	4.45
(v) Racket	7.91
(i) TypeScript	21.50
(i) Hack	24.02
(i) PHP	29.30
(v) Erlang	42.23
(i) Lua	45.98
(i) Jruby	46.54
(i) Ruby	69.91
(i) Python	75.88
(i) Perl	79.58

	Time
(c) C	1.00
(c) Rust	1.04
(c) C++	1.56
(c) Ada	1.85
(v) Java	1.89
(c) Chapel	2.14
(c) Go	2.83
(c) Pascal	3.02
(c) Ocaml	3.09
(v) C#	3.14
(v) Lisp	3.40
(c) Haskell	3.55
(c) Swift	4.20
(c) Fortran	4.20
(v) F#	6.30
(i) JavaScript	6.52
(i) Dart	6.67
(v) Racket	11.27
(i) Hack	26.99
(i) PHP	27.64
(v) Erlang	36.71
(i) Jruby	43.44
(i) TypeScript	46.20
(i) Ruby	59.34
(i) Perl	65.79
(i) Python	71.90
(i) Lua	82.91

	Mb
(c) Pascal	1.00
(c) Go	1.05
(c) C	1.17
(c) Fortran	1.24
(c) C++	1.34
(c) Ada	1.47
(c) Rust	1.54
(v) Lisp	1.92
(c) Haskell	2.45
(i) PHP	2.57
(c) Swift	2.71
(i) Python	2.80
(c) Ocaml	2.82
(v) C#	2.85
(i) Hack	3.34
(v) Racket	3.52
(i) Ruby	3.97
(c) Chapel	4.00
(v) F#	4.25
(i) JavaScript	4.59
(i) TypeScript	4.69
(v) Java	6.01
(i) Perl	6.62
(i) Lua	6.72
(v) Erlang	7.20
(i) Dart	8.64
(i) Jruby	19.84

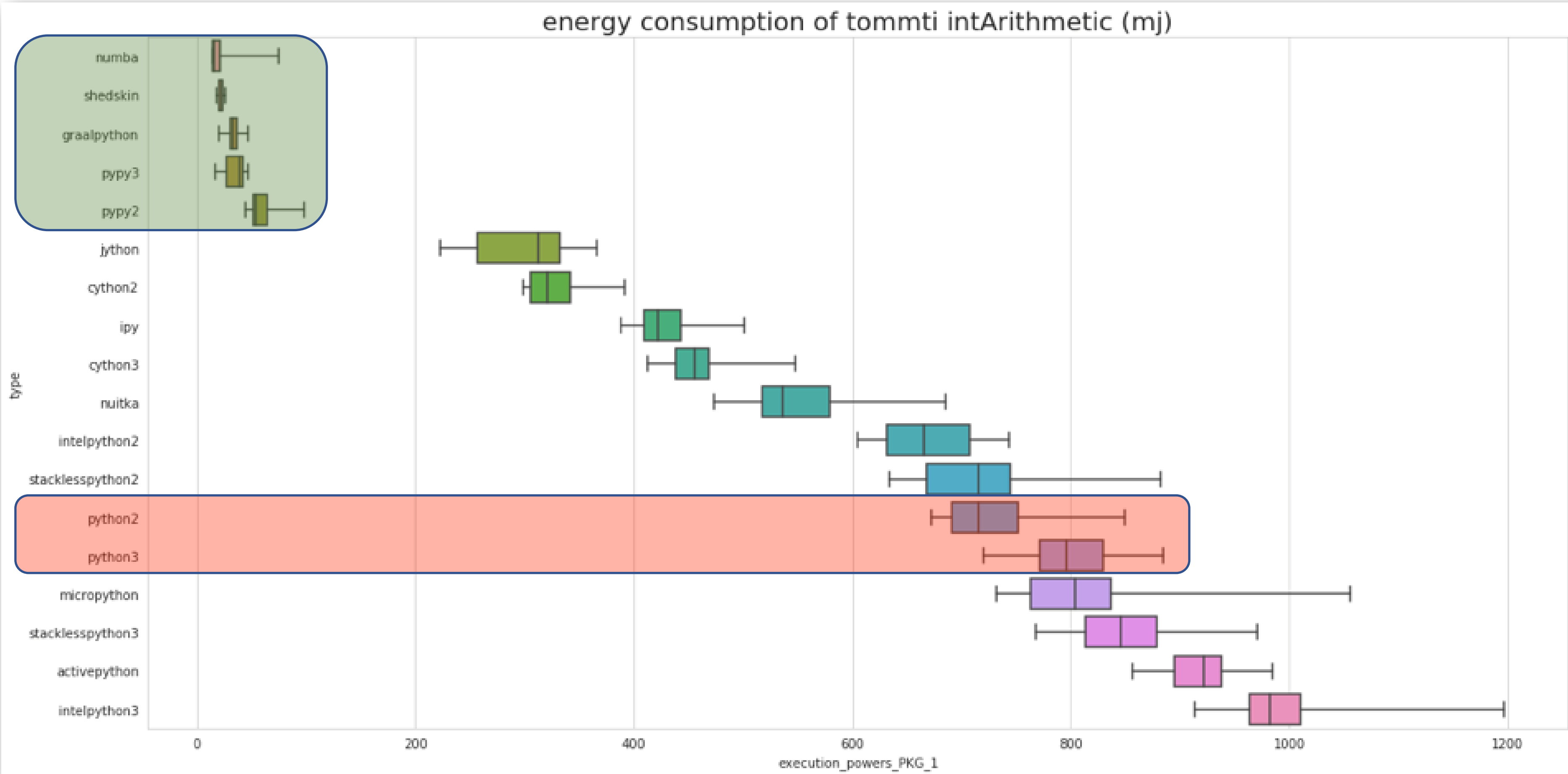
“Only **four languages maintain the same energy and time rank** (OCaml, Haskel, Racket, and Python), while the remainder are completely shuffled.”

when manipulating **strings with regular expression**, three of the five ***most* energy-efficient** languages turn out to be **interpreted languages** (TypeScript, JavaScript, and PHP),

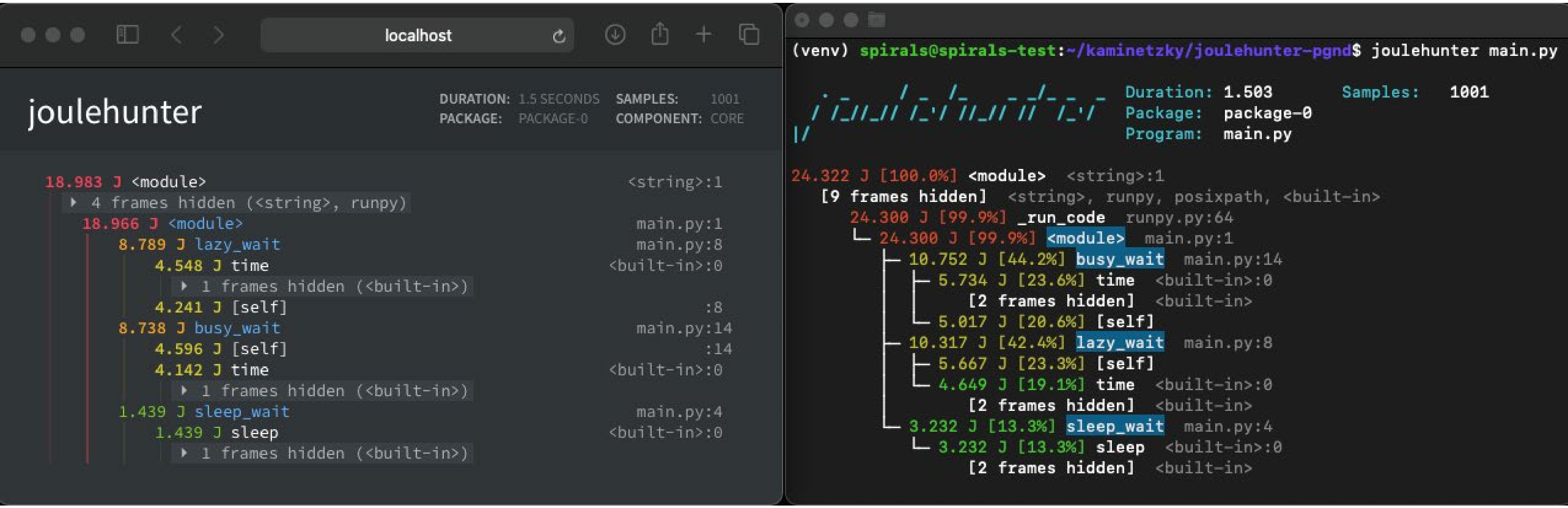
“Although the most energy efficient language in each benchmark is almost always the fastest one, the fact is that there is **no language which is consistently better than the others**,”

R. Pereira et al. SLE 2017

Energy efficiency across programming languages: how do energy, time, and memory relate?


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

Energy profiling with **JouleHunter**



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

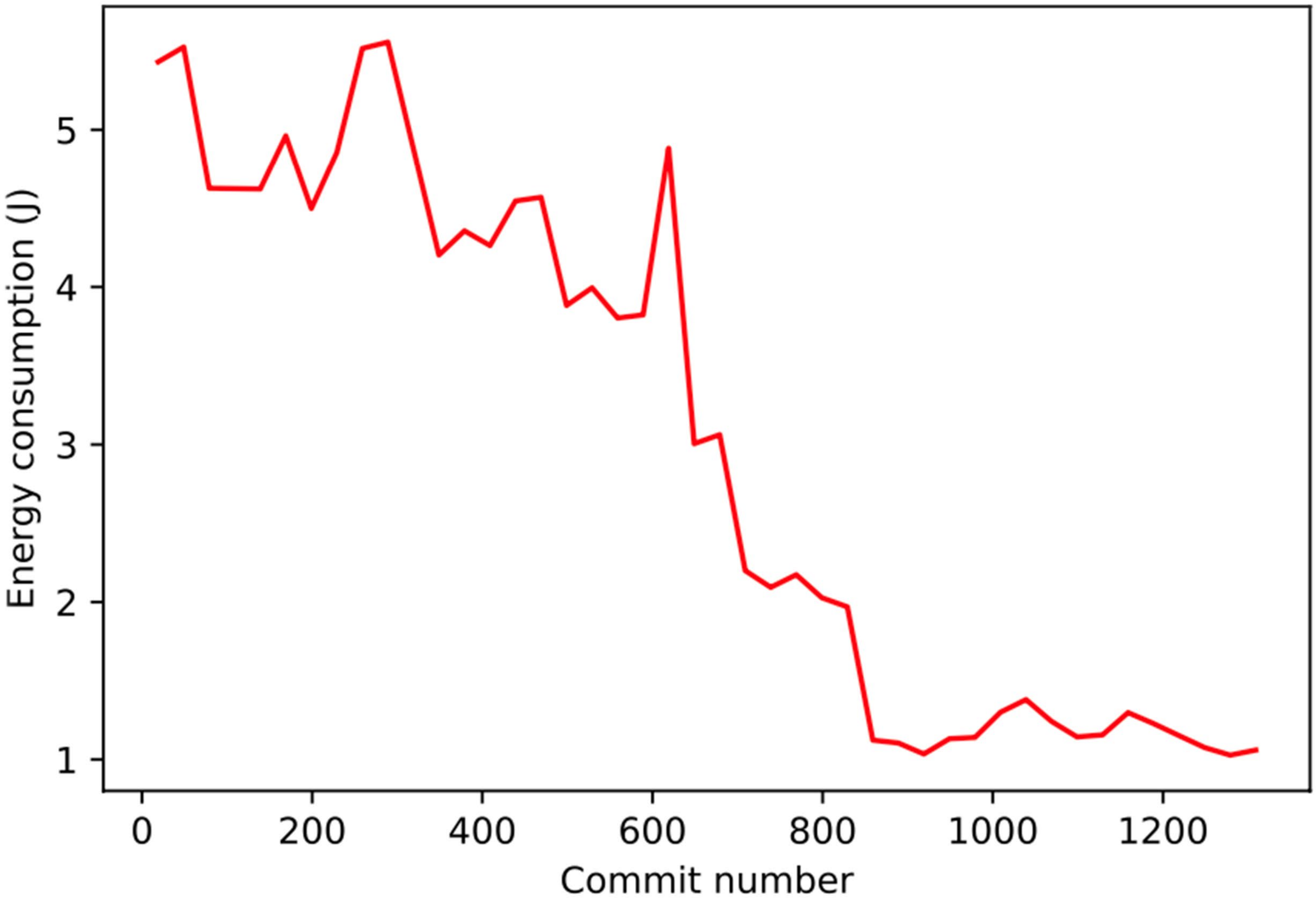


Fig. 4: Gson energy consumption across all commits.

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.

My *talk* in 180 seconds

- ICT energy consumption will keep growing
  - More and more digital services (in all domains)
- Hardware keeps improving energy efficiency
  - But hardware is driven by software
- Software is eating the world, and beyond
  - *Everything is software-defined*
- **Énergy ≈ performance** (time)
  - *Relationship: it's complicated*
- Needs to work on all the layers of an infrastructure
  - Each layer = a software to optimize

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*