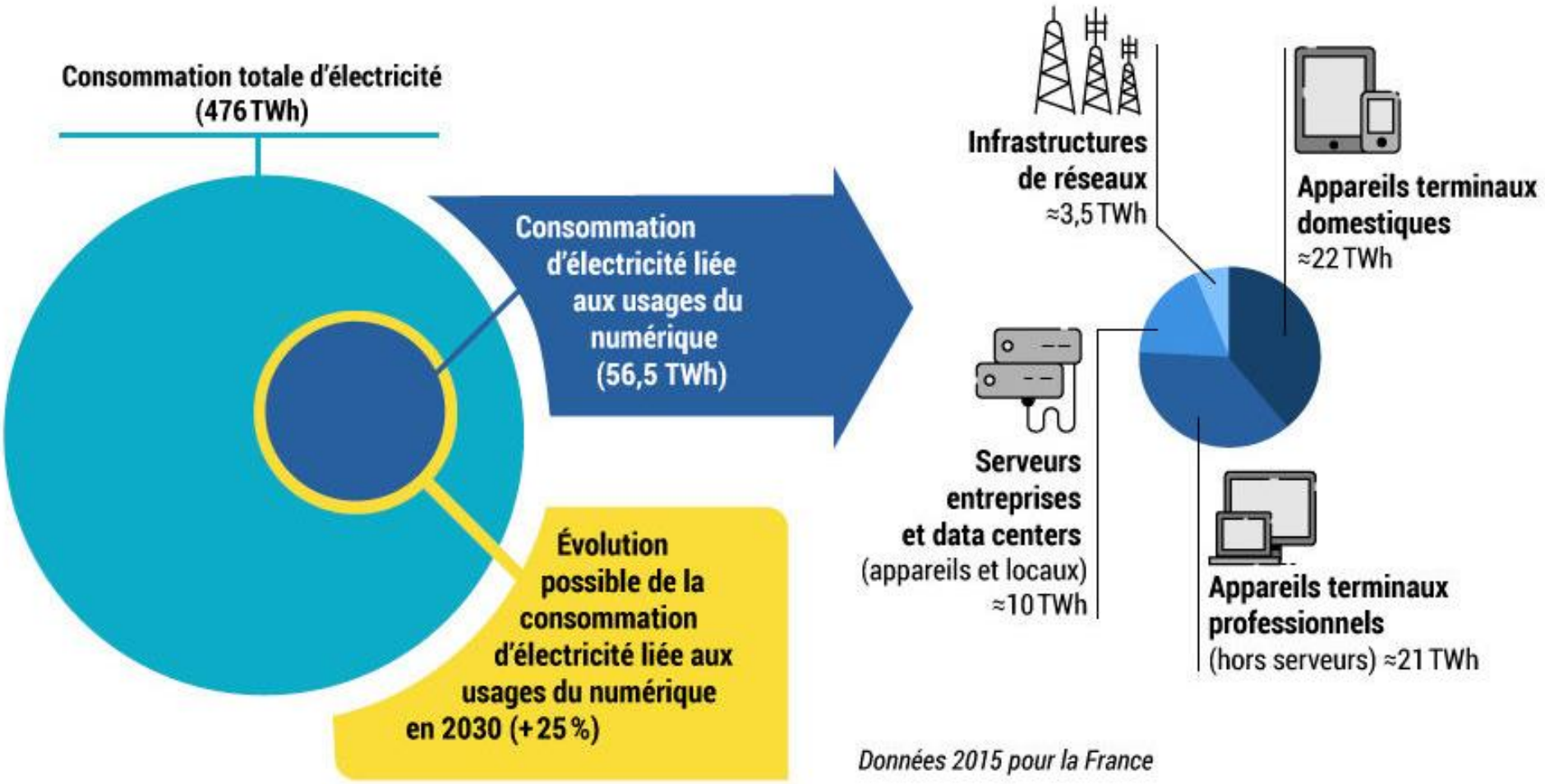
Quels défis pour le développement durable des logiciels ?

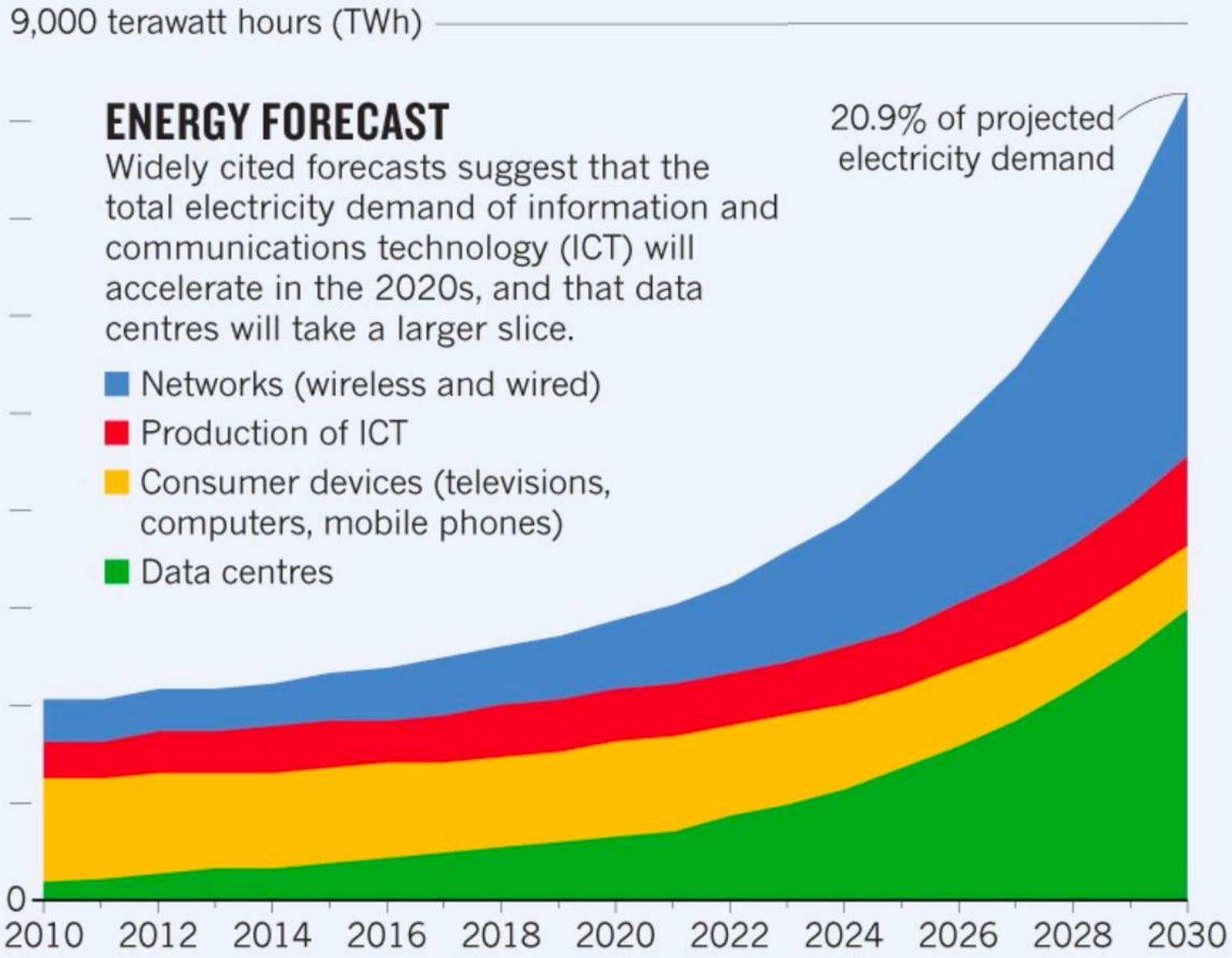
@RomainROUVOY

institut
universitaire
de France

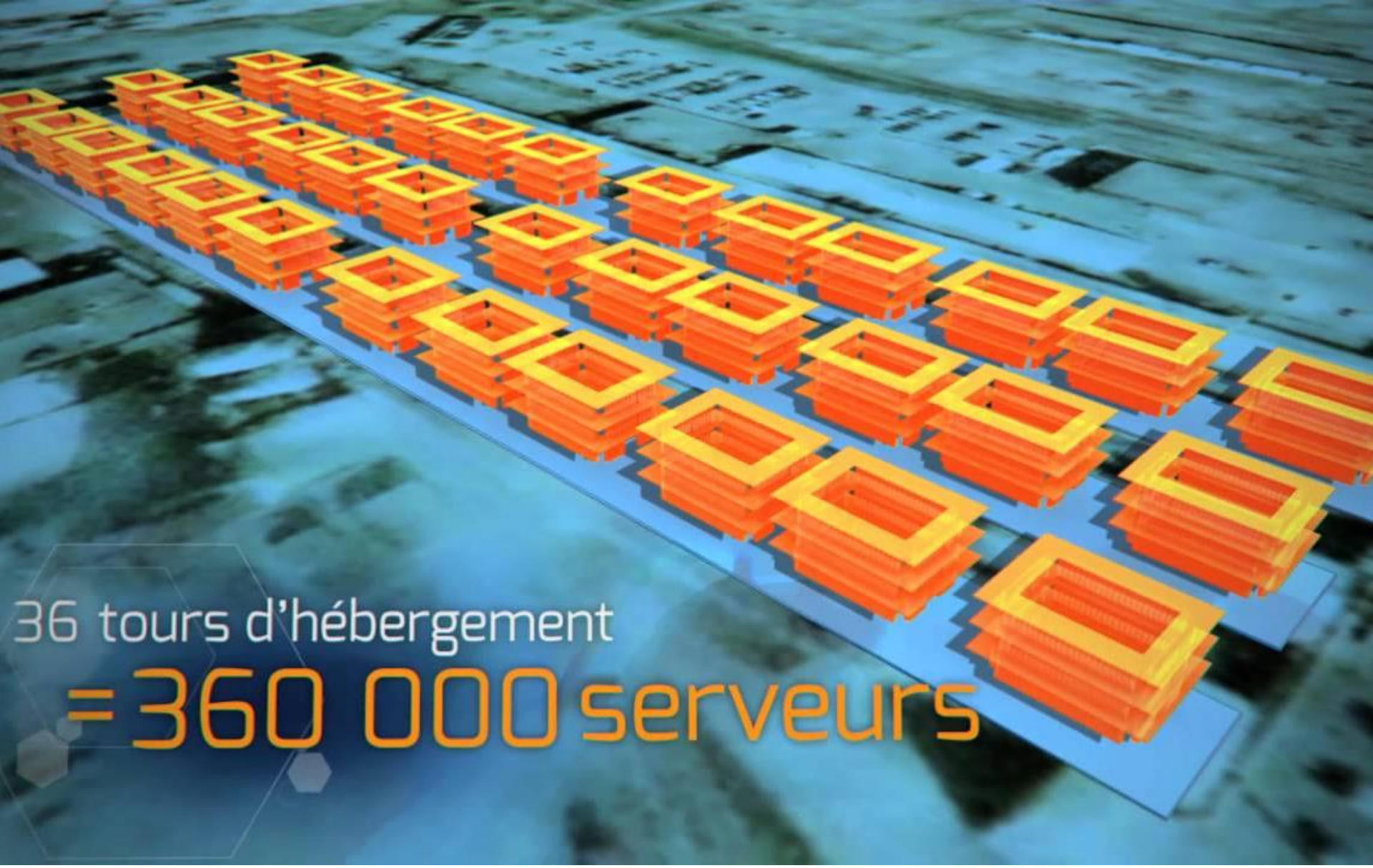
IMPACT
ENVIRONNEMENTAL



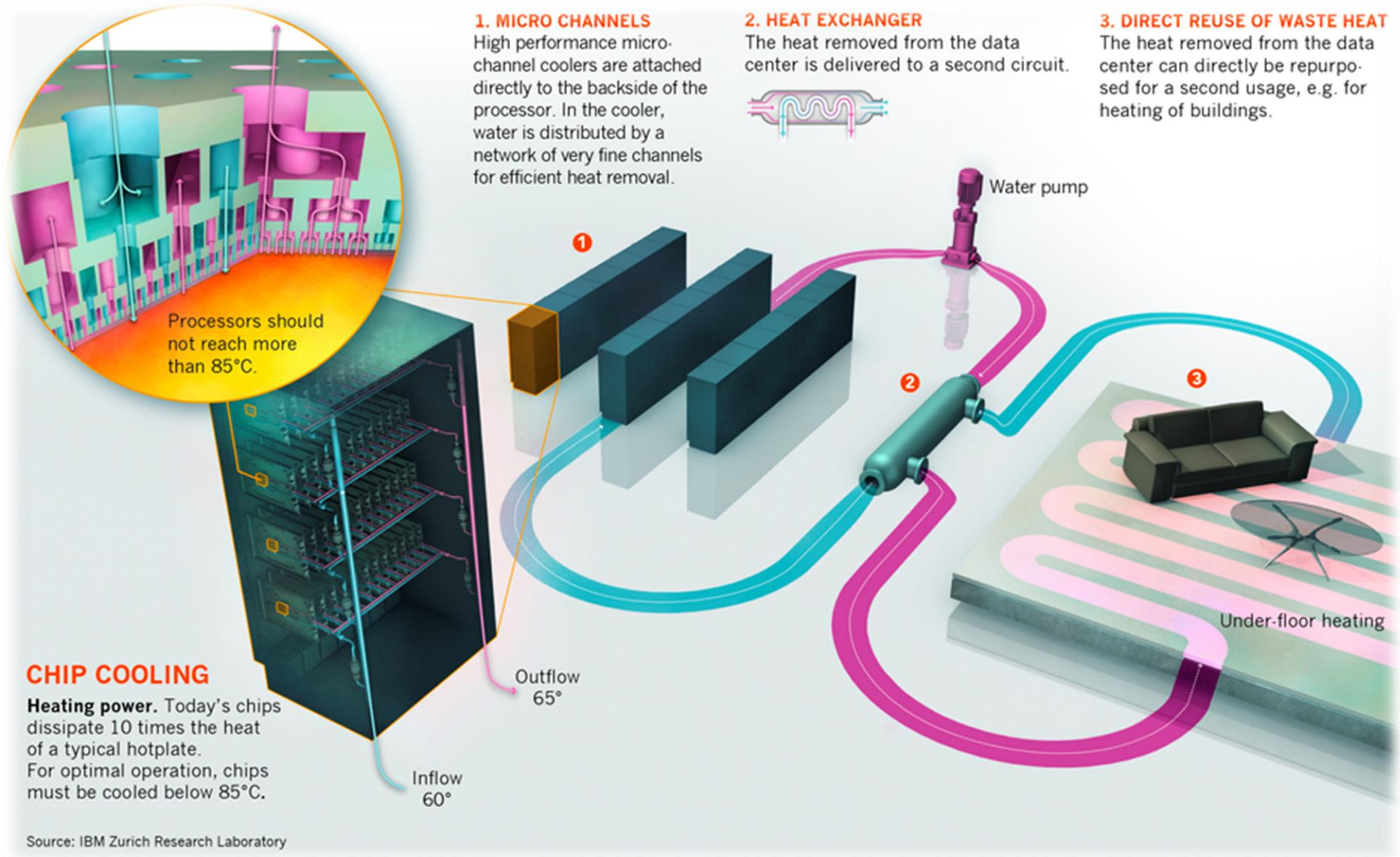
Le développement du numérique aura
UN IMPACT MODÉRÉ SUR LA CONSOMMATION D'ÉLECTRICITÉ
en France



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%.



= 1 data center OVH en 2012


1. MICRO CHANNELS
High performance micro-channel coolers are attached directly to the backside of the processor. In the cooler, water is distributed by a network of very fine channels for efficient heat removal.
2. HEAT EXCHANGER
The heat removed from the data center is delivered to a second circuit.
3. DIRECT REUSE OF WASTE HEAT
The heat removed from the data center can directly be repurposed for a second usage, e.g. for heating of buildings.
Water pump
Processors should not reach more than 85°C.
Outflow 65°
Inflow 60°
Under-floor heating
CHIP COOLING
Heating power. Today's chips dissipate 10 times the heat of a typical hotplate.
For optimal operation, chips must be cooled below 85°C.
Source: IBM Zurich Research Laboratory

Qarnot computing

Pour arrêter ce keynote, envoyez “STOP” au 72 500 (65€ + Prix SMS)

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

Quel peut être l’impact du Génie Logiciel ??

Tesla


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

M. Fabrice Brun (député Ardèche) attire l'attention de M. le secrétaire d'État auprès du ministre de l'économie et des finances et du ministre de l'action et des comptes publics, chargé du numérique, sur le sujet de la dépense en énergie et de la **production de CO2 générées par l'utilisation toujours croissante de moyens informatiques en réseau**. Les usages de plateformes dématérialisées de *streaming* audio et vidéo et de jeux vidéo en ligne ont pour conséquence une **hausse exponentielle de la consommation d'énergie**. Avec l'arrivée de nouvelles technologies telles que la 4K, la 8K et la 5G ainsi que l'usage des écrans HD, cette tendance ne fera que s'accélérer. **Il est donc nécessaire de faire évoluer les pratiques des serveurs de stockage afin de favoriser une utilisation plus durable des moyens électriques** et de permettre une politique raisonnée des besoins en bande passante réduisant la facture environnementale des éditeurs de logiciels utilisés par les plateformes précitées. C'est pourquoi il lui demande de bien vouloir préciser la **position du Gouvernement sur l'obligation pour les éditeurs de logiciels de consacrer un budget déterminé de recherche et développement afin de pratiquer une écriture plus vertueuse en terme environnemental du code informatique**. Il souhaiterait savoir si le Gouvernement serait prêt à mobiliser ses partenaires afin **d'intégrer ces questions au programme de la prochaine COP25 qui se tiendra au Chili en novembre 2019.**

Question soumise le 21 mai 2019 (sans réponse)

FEATURE: GREEN SOFTWARE

What Do Programmers Know about Software Energy Consumption?

Candy Pang and Abram Hindle, University of Alberta

Bram Adams, Polytechnique Montréal

Ahmed E. Hassan, Queen's University

// A survey revealed that programmers had limited knowledge of energy efficiency, lacked knowledge of the best practices to reduce software energy consumption, and were unsure about how software consumes energy. These results highlight the need for training on energy consumption. //

WITH THE rising popularity of mobile computing and the advent of large-scale cloud deployments, the nonfunctional requirement of minimizing software energy consumption has become a concern. For mobile devices, energy consumption affects battery life and limits device use. For datacenters, energy consumption limits the number of machines that can be run and cooled. According to an IDC white paper, "Today, for every $1.00 spent on new hardware, an additional $0.50 is spent on power and cooling, more than double the amount of five years ago. Datacenters at their power and cooling thresholds are unable to support new server deployments, a fact that severely limits the expansion of IT resources."[1]

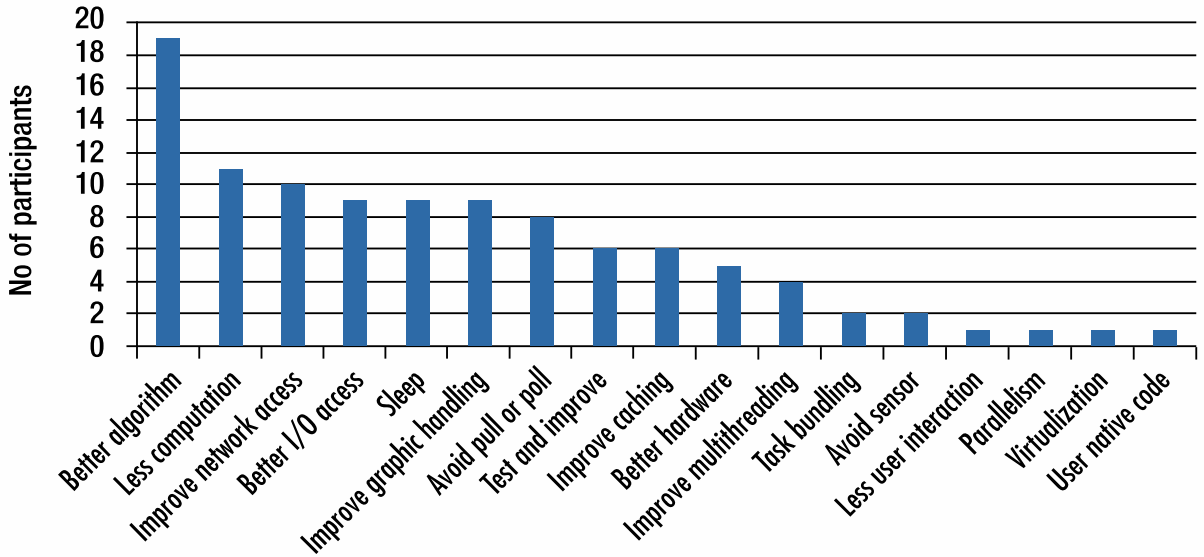
Unfortunately, the demand for energy-efficient computing isn't reflected in the education, training, or knowledge of programmers. Programmer training often focuses on methodologies such as object-oriented programming and nonfunctional requirements such as performance. Performance optimization is often considered a substitute for energy optimization because a faster system likely consumes less energy. Although this is a step in the right direction, it's insufficient and sometimes even incorrect. For instance, parallel processing might improve performance by reducing calculation time. How-

0740-7459/16/$33.00 © 2016 IEEE

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

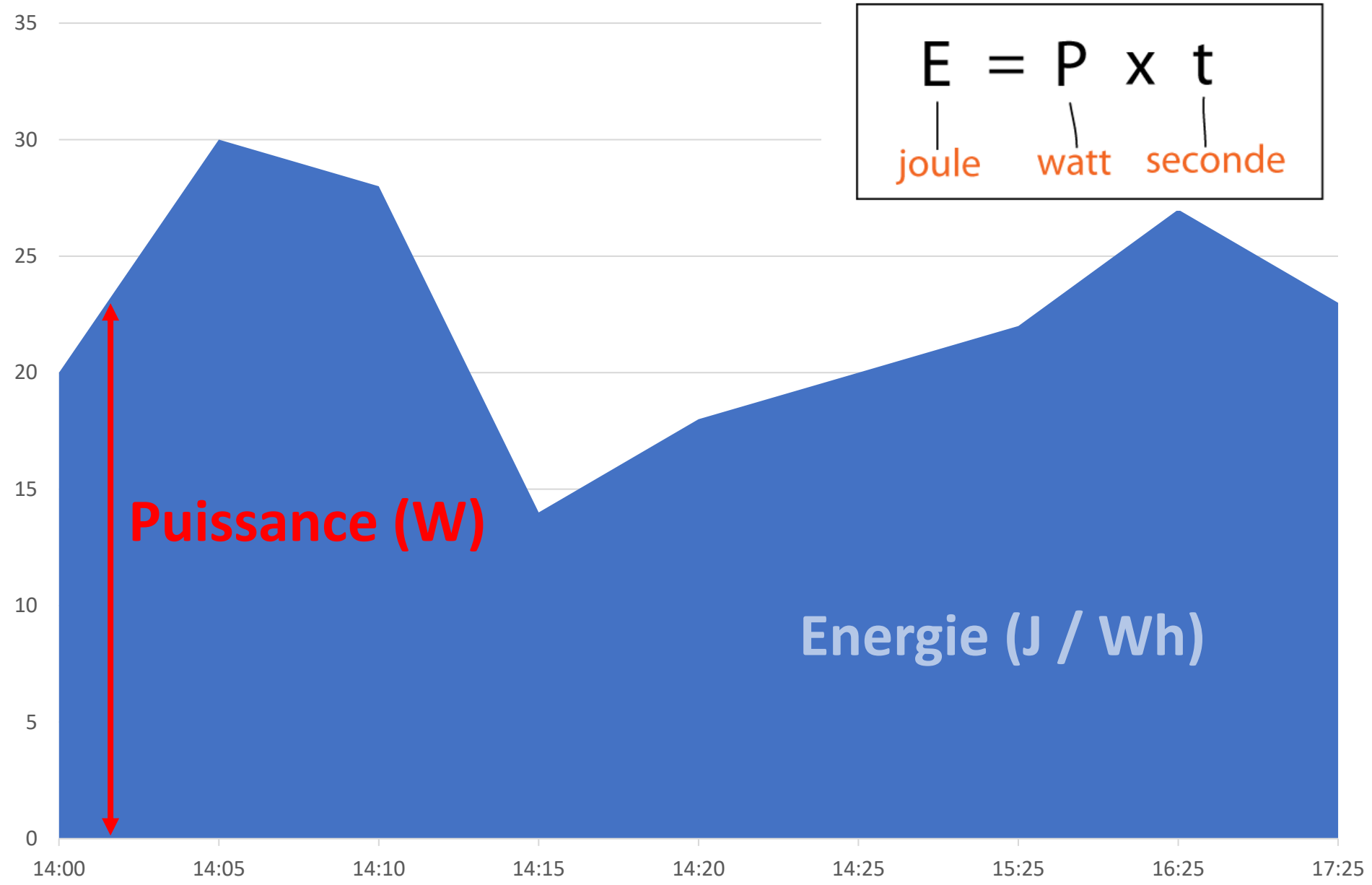
*"It's more often the **hardware rather than the software** that we are interested in when we talk about energy consumption."*

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

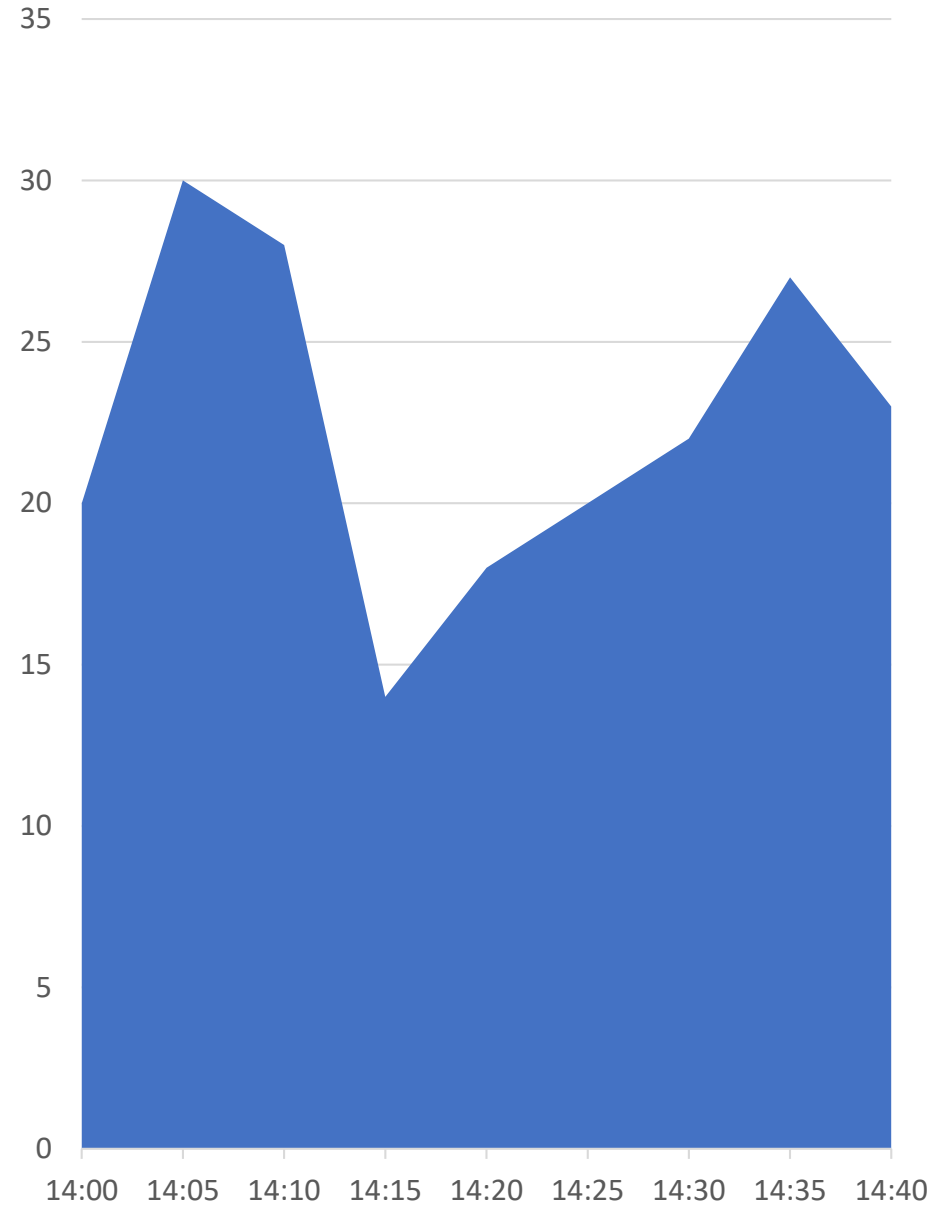


Green computing 101

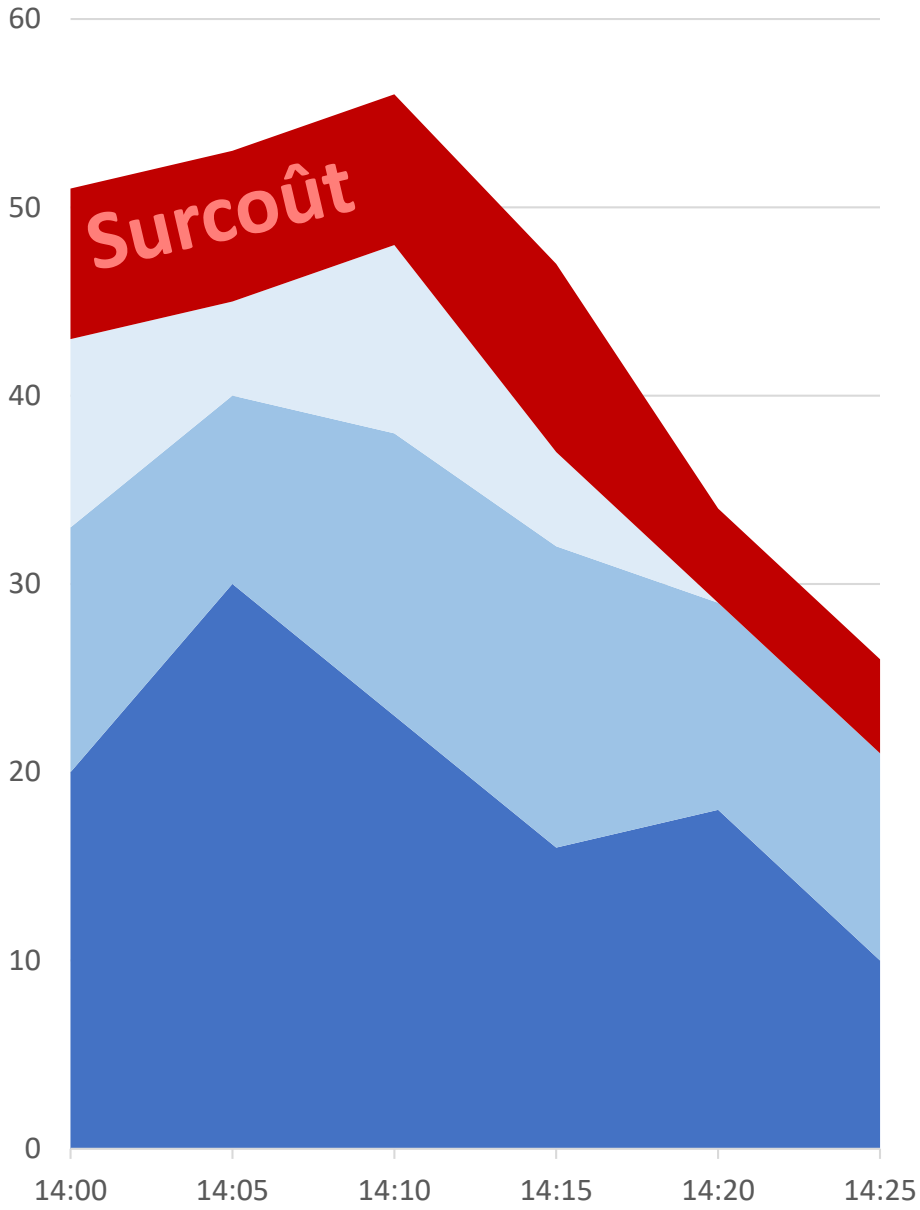
Energie Vs. Puissance



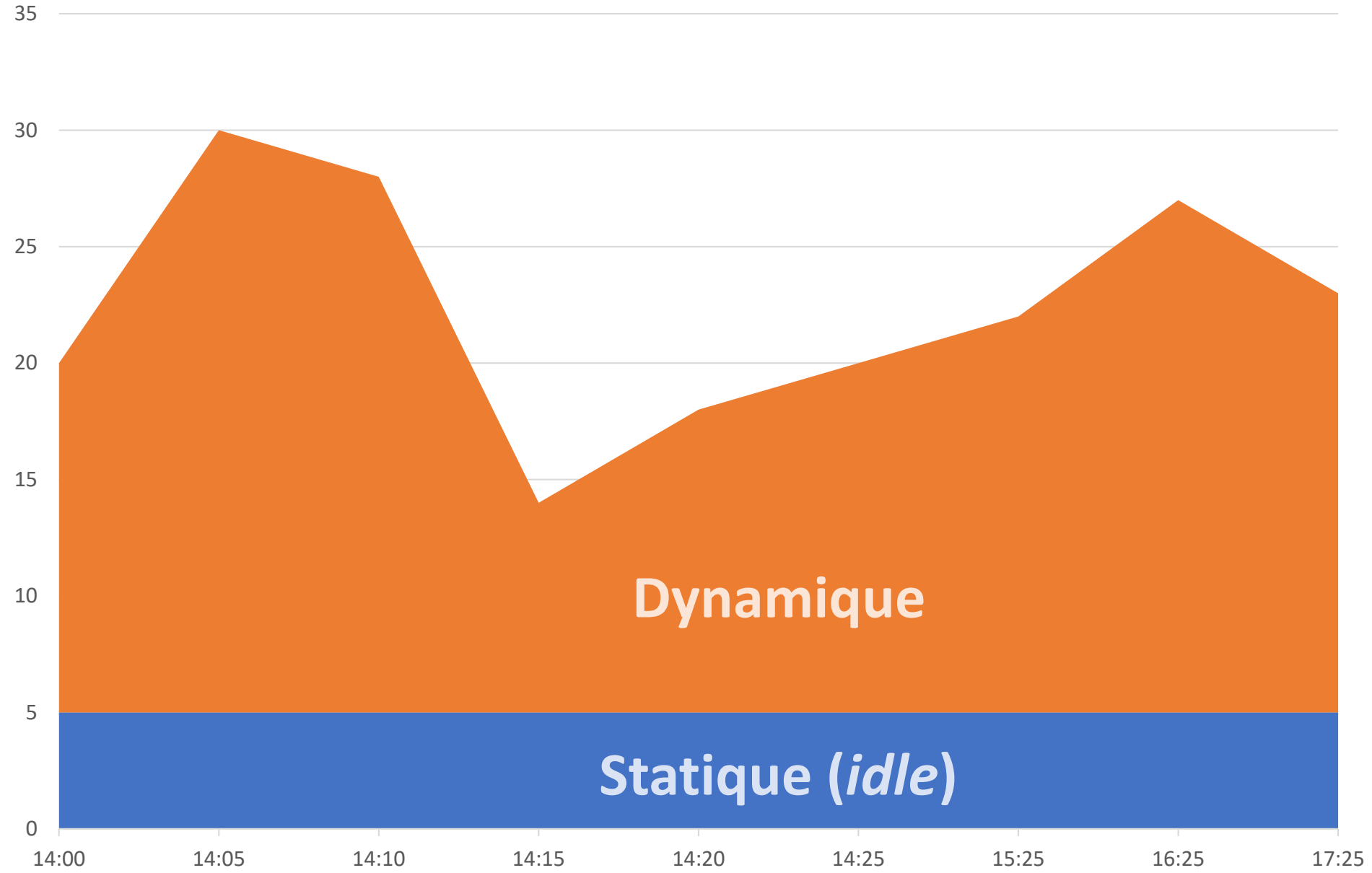
Séquentiel



Parallèle

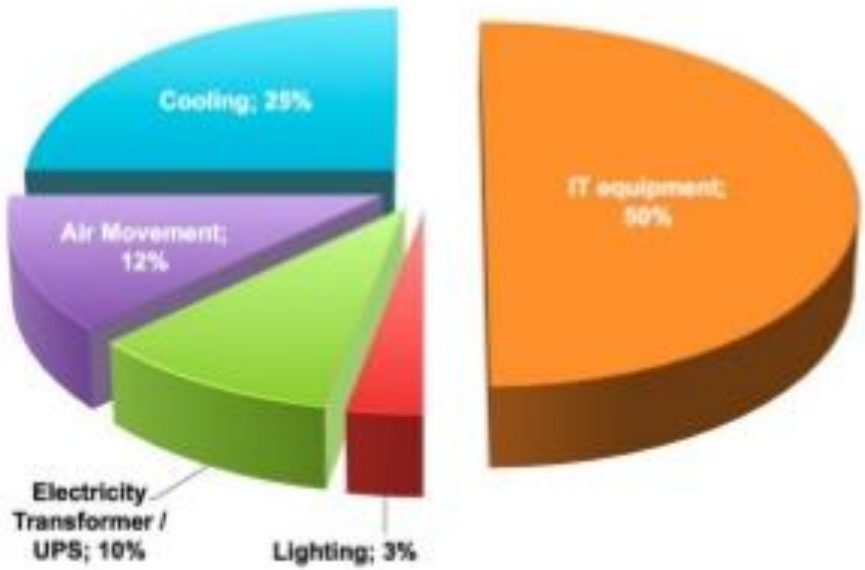


Statique Vs. Dynamique



$$\text{PUE} = \text{Power Usage Effectiveness} = \frac{\text{Total Facility Power}}{\text{IT Equipment Power}}$$

EXAMPLE: Typical 1MW Facility



$$\text{PUE} = \frac{1\ \text{MW}}{0.5\ \text{MW}} = 2.0$$

"My total facility consumes 2x the power of the IT equipment load."

$$\text{DCiE} = \frac{1}{\text{PUE}} = 0.5$$

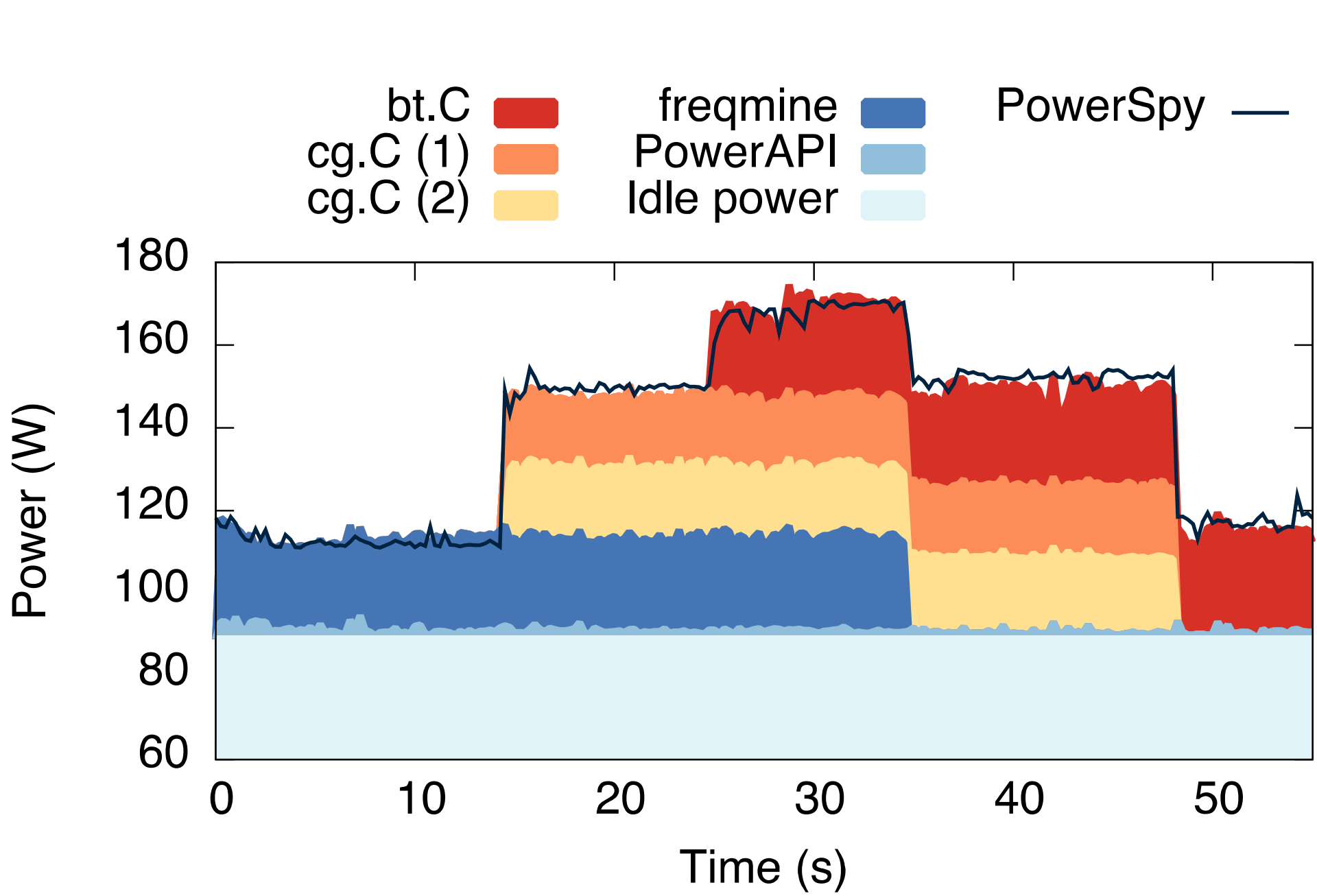
"Roughly 50% of the power in my facility is used to power IT equipment."

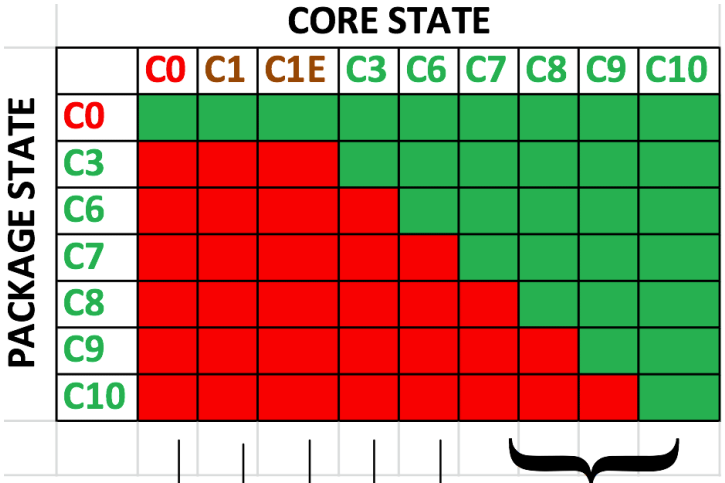
Quelques contributions

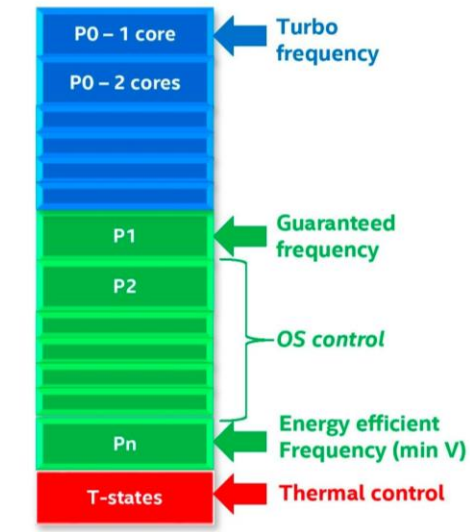
SPIRALS

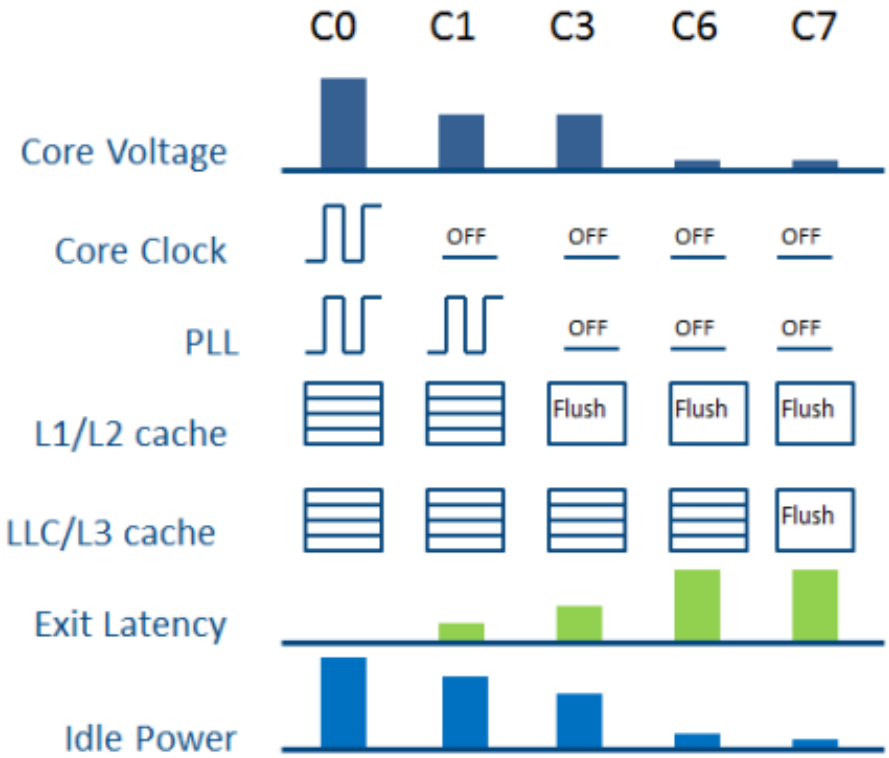
These results show that these programmers lacked knowledge of how to **accurately measure software energy consumption**.

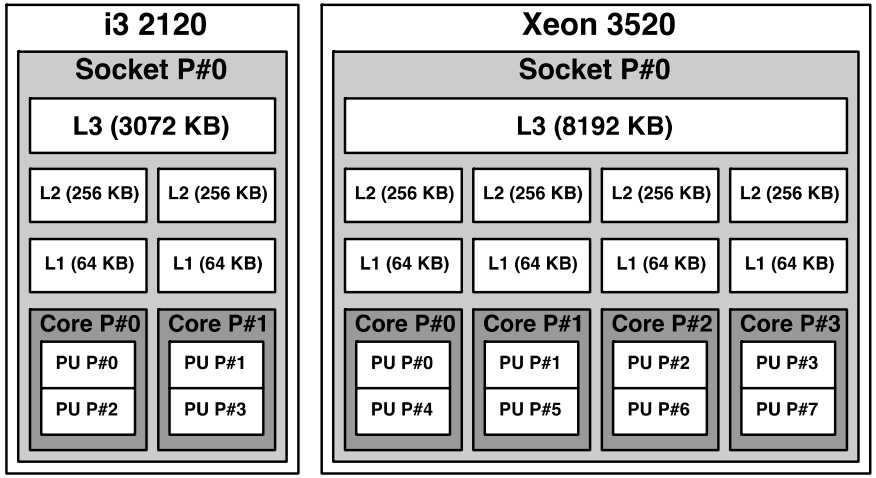
http://powerapi.org

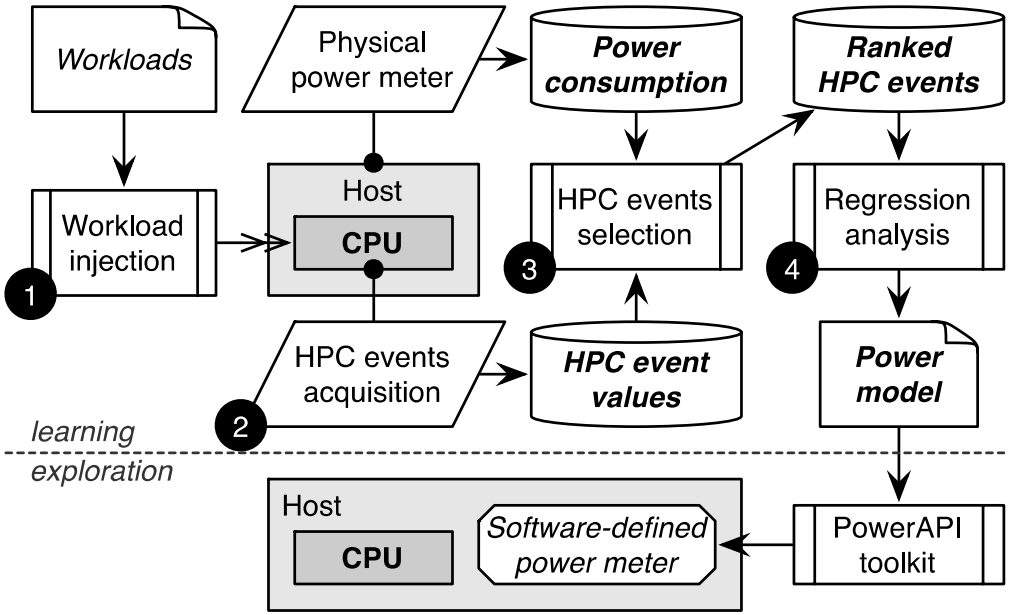

bt.C
cg.C (1)
cg.C (2)
freqmine
PowerAPI
Idle power
PowerSpy
Power (W)
180
160
140
120
100
80
60
0
10
20
30
40
50
Time (s)

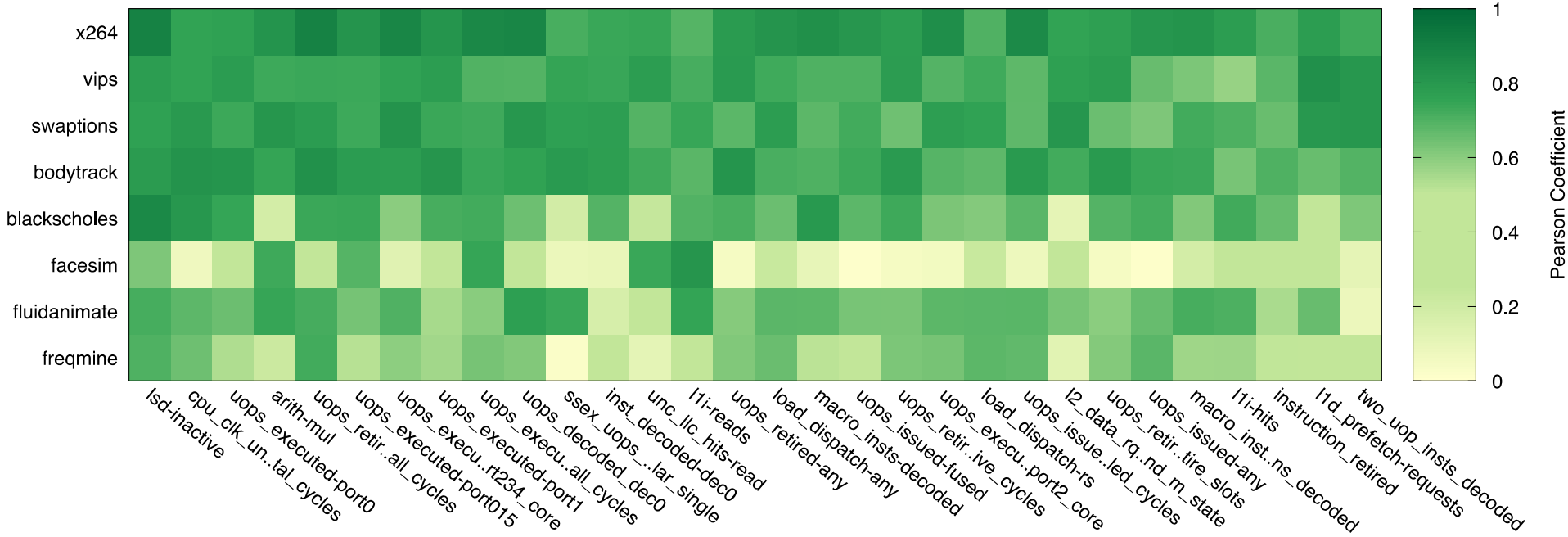

CORE STATE
PACKAGE STATE
C0
C1
C1E
C3
C6
C7
C8
C9
C10
One or more cores or GT executing instructions
All cores and GT in C3 or deeper, L3 may be flushed and turned off, memory in self refresh, some Uncore clocks stopped, some Uncore voltages reduced
All cores and GT in C6 or deeper, L3 may be flushed 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


P-state
(All CPUs, plus Skylake)
P0 – 1 core
P0 – 2 cores
P1
P2
Pn
T-states
Turbo frequency
Guaranteed frequency
OS control
Energy efficient Frequency (min V)
Thermal control

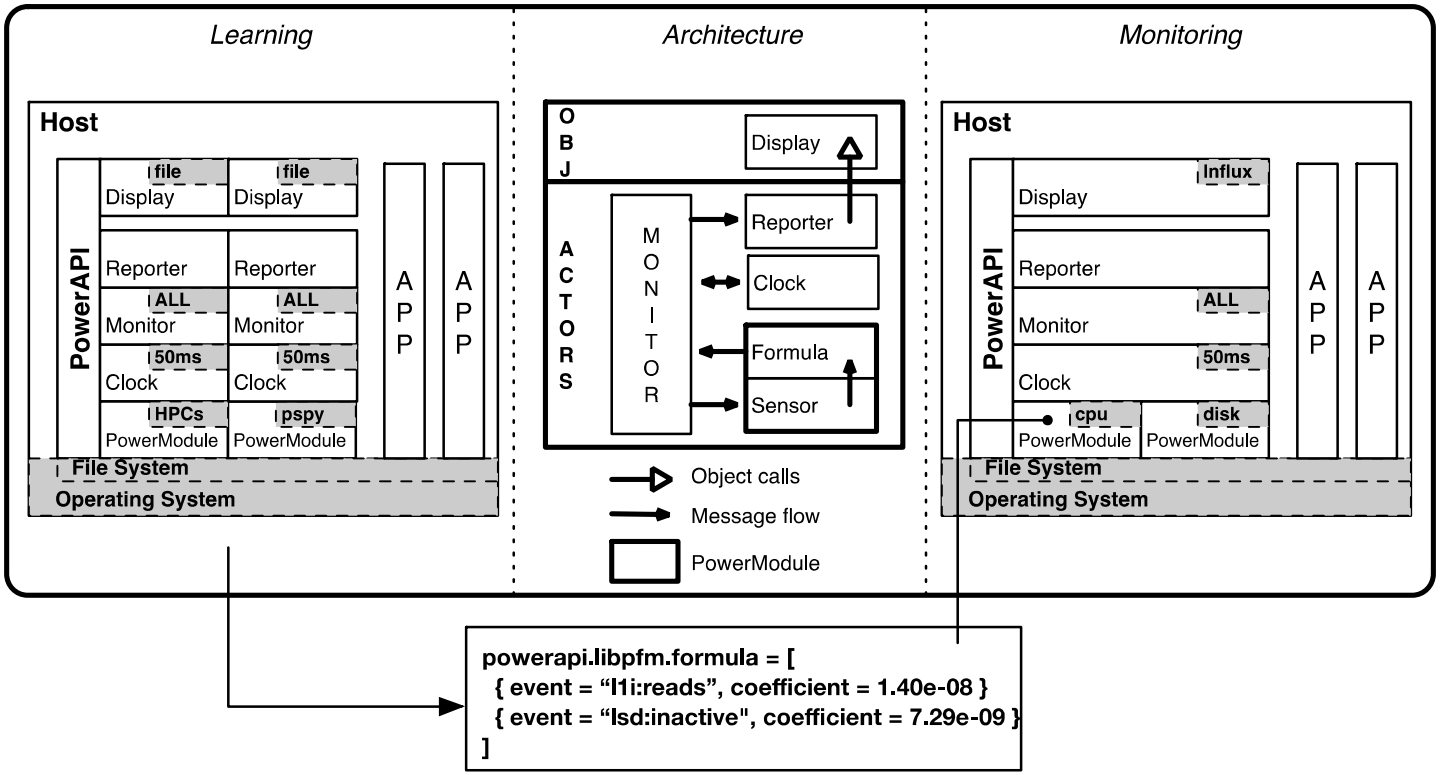

C0
C1
C3
C6
C7
Core Voltage
Core Clock
PLL
L1/L2 cache
LLC/L3 cache
Exit Latency
Idle Power
OFF
Flush

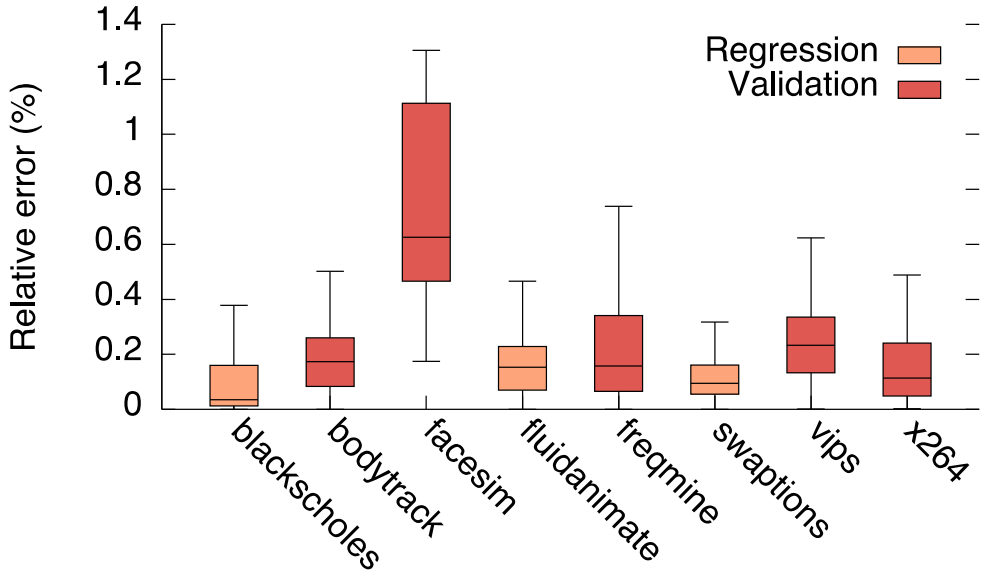

i3 2120
Socket P#0
L3 (3072 KB)
L2 (256 KB)
L2 (256 KB)
L1 (64 KB)
L1 (64 KB)
Core P#0
Core P#1
PU P#0
PU P#2
PU P#1
PU P#3
Xeon 3520
Socket P#0
L3 (8192 KB)
L2 (256 KB)
L2 (256 KB)
L2 (256 KB)
L2 (256 KB)
L1 (64 KB)
L1 (64 KB)
L1 (64 KB)
L1 (64 KB)
Core P#0
Core P#1
Core P#2
Core P#3
PU P#0
PU P#4
PU P#1
PU P#5
PU P#2
PU P#6
PU P#3
PU P#7

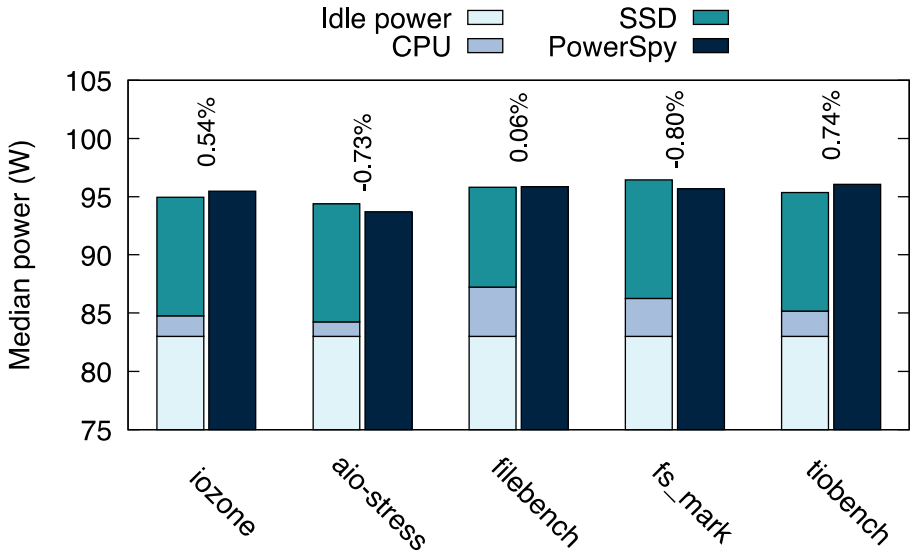

Workloads
Physical power meter
Power consumption
Ranked HPC events
Workload injection
Host
CPU
HPC events selection
Regression analysis
1
3
4
HPC events acquisition
HPC event values
Power model
2
learning
exploration
Host
CPU
Software-defined power meter
PowerAPI toolkit


x264
vips
swaptions
bodytrack
blackscholes
facesim
fluidanimate
freqmine
lsd-inactive
cpu_clk_un..tal_cycles
uops_executed-port0
arith-mul
uops_retir..all_cycles
uops_executed-port015
uops_execu..rt234_core
uops_executed-port1
uops_execu..all_cycles
uops_decoded_dec0
ssex_uops_..lar_single
inst_decoded-dec0
unc_llc_hits-read
l1i-reads
uops_retired-any
load_dispatch-any
macro_insts-decoded
uops_issued-fused
uops_retir..ive_cycles
uops_execu..port2_core
load_dispatch-rs
l2_data_rq..nd_m_state
uops_issue..led_cycles
uops_retir..tire_slots
uops_issued-any
macro_inst..ns_decoded
l1i-hits
instruction_retired
l1d_prefetch-requests
two_uop_insts_decoded
1
0.8
0.6
0.4
0.2
0
Pearson Coefficient

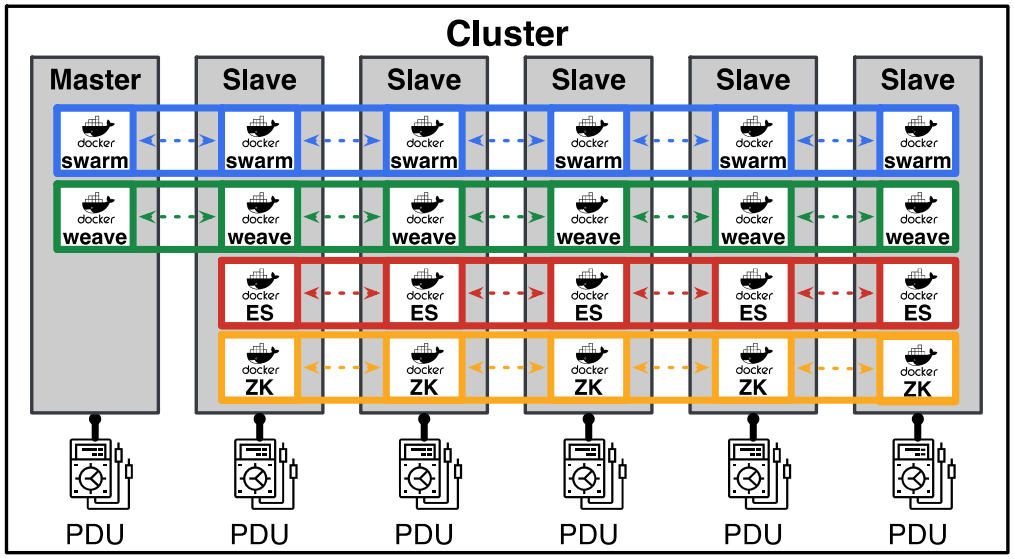
PowerAPI

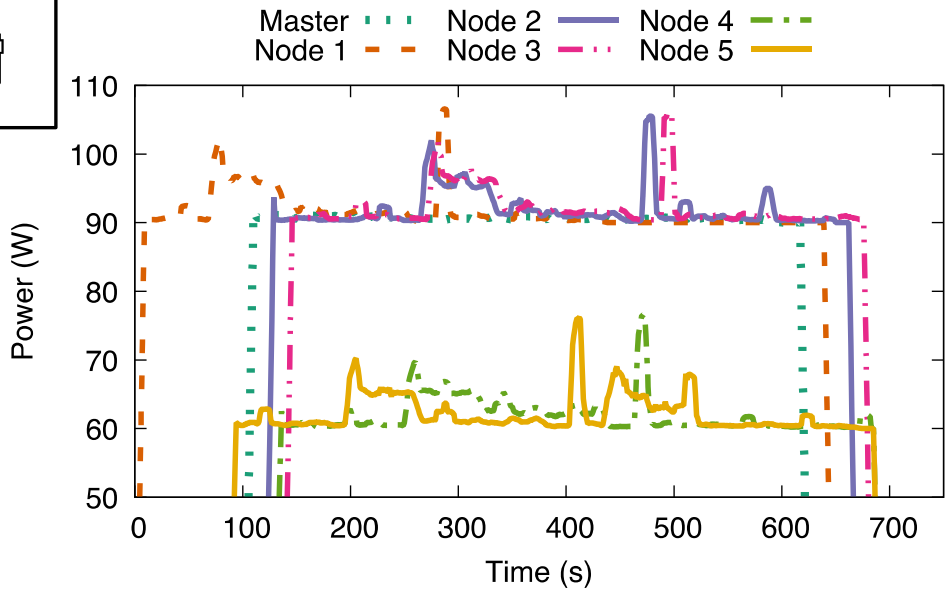

Learning
Architecture
Monitoring
Host
PowerAPI
file
Display
Reporter
ALL
Monitor
50ms
Clock
HPCs
PowerModule
pspy
APP
File System
Operating System
OBJ
ACTORS
MONITOR
Reporter
Clock
Formula
Sensor
Object calls
Message flow
PowerModule
Influx
cpu
disk
powerapi.libpfm.formula = [
{ event = "l1i:reads", coefficient = 1.40e-08 }
{ event = "lsd:inactive", coefficient = 7.29e-09 }
]

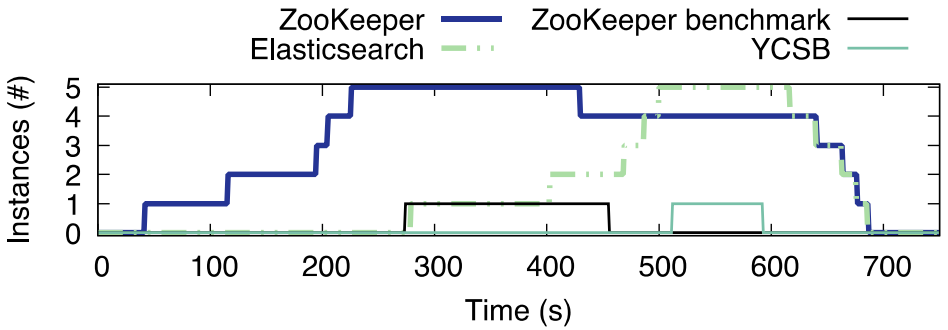

Relative error (%)
Regression
Validation
blackscholes
bodytrack
facesim
fluidanimate
freqmine
swaptions
vips
x264


Idle power
CPU
SSD
PowerSpy
Median power (W)
0.54%
-0.73%
0.06%
-0.80%
0.74%
iozone
aio-stress
filebench
fs_mark
tiobench

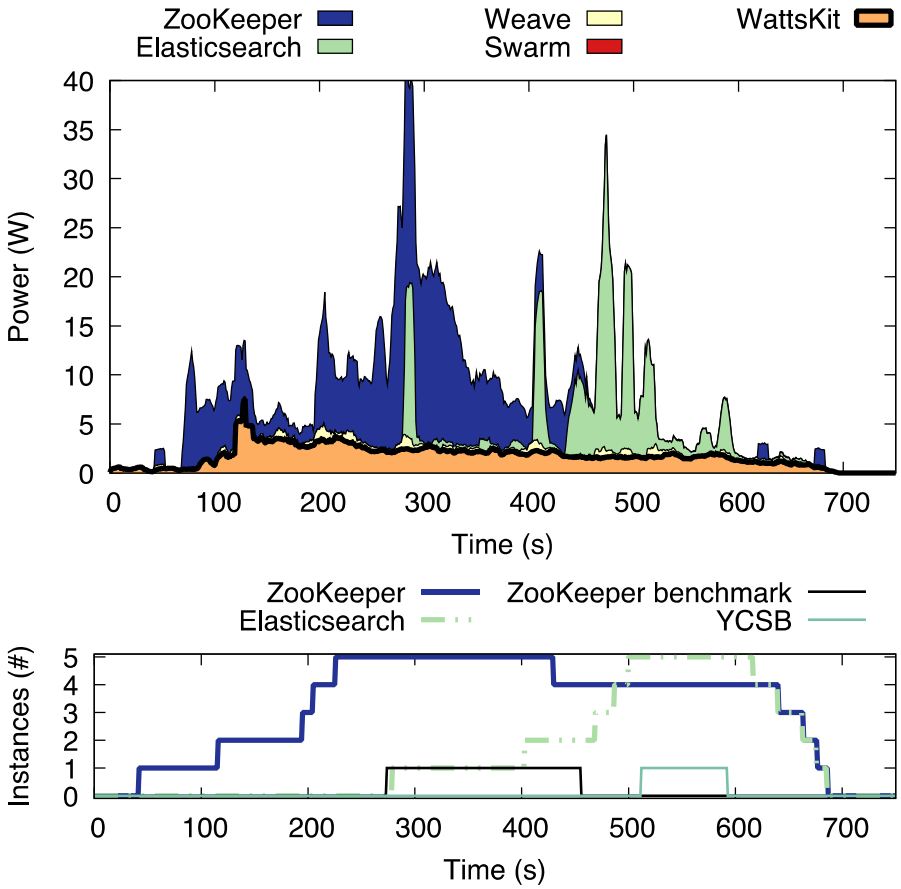
PowerAPI

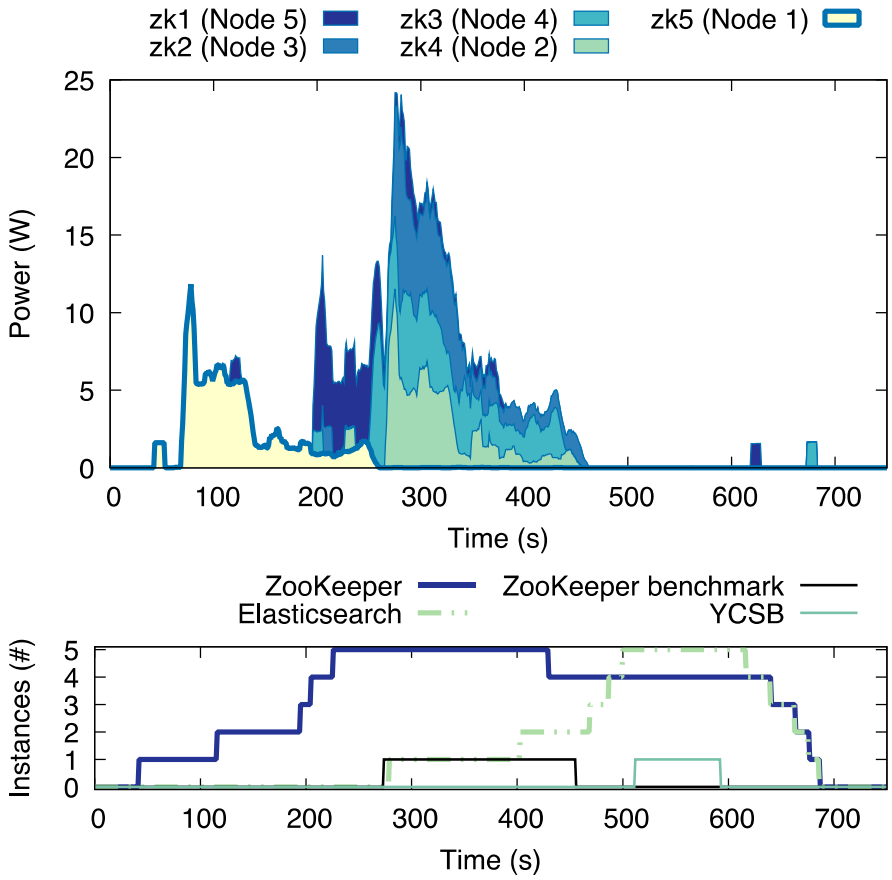

Cluster
Master
Slave
Slave
Slave
Slave
Slave
swarm
weave
ES
ZK
PDU
PDU
PDU
PDU
PDU
PDU


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

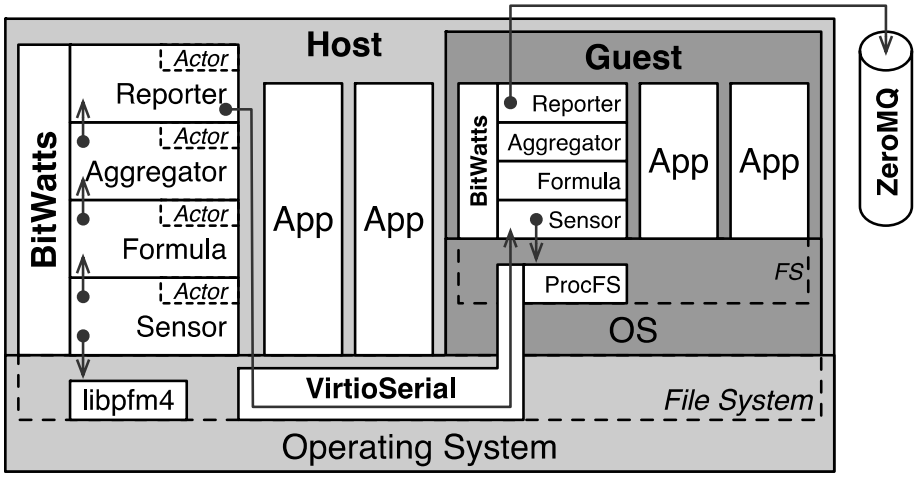

ZooKeeper
Elasticsearch
Weave
Swarm
WattsKit
Power (W)
Time (s)
ZooKeeper
Elasticsearch
ZooKeeper benchmark
YCSB
Instances (#)
Time (s)

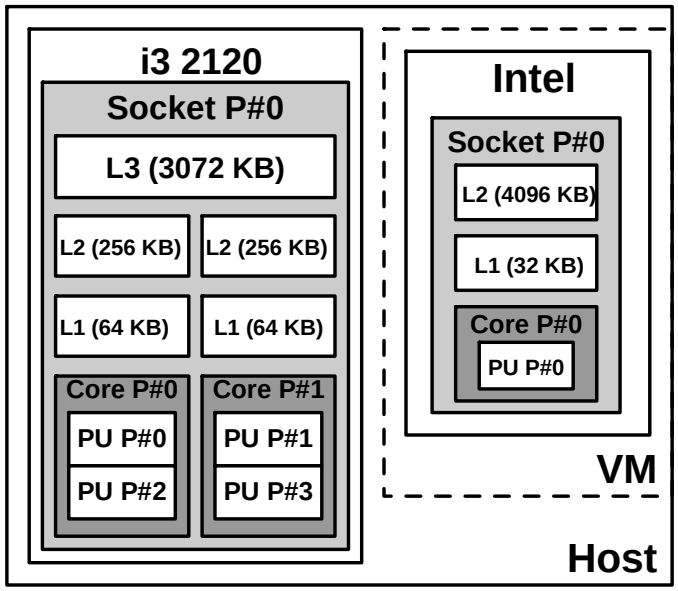

zk1 (Node 5)
zk2 (Node 3)
zk3 (Node 4)
zk4 (Node 2)
zk5 (Node 1)
Power (W)
Time (s)
ZooKeeper
Elasticsearch
ZooKeeper benchmark
YCSB
Instances (#)
Time (s)

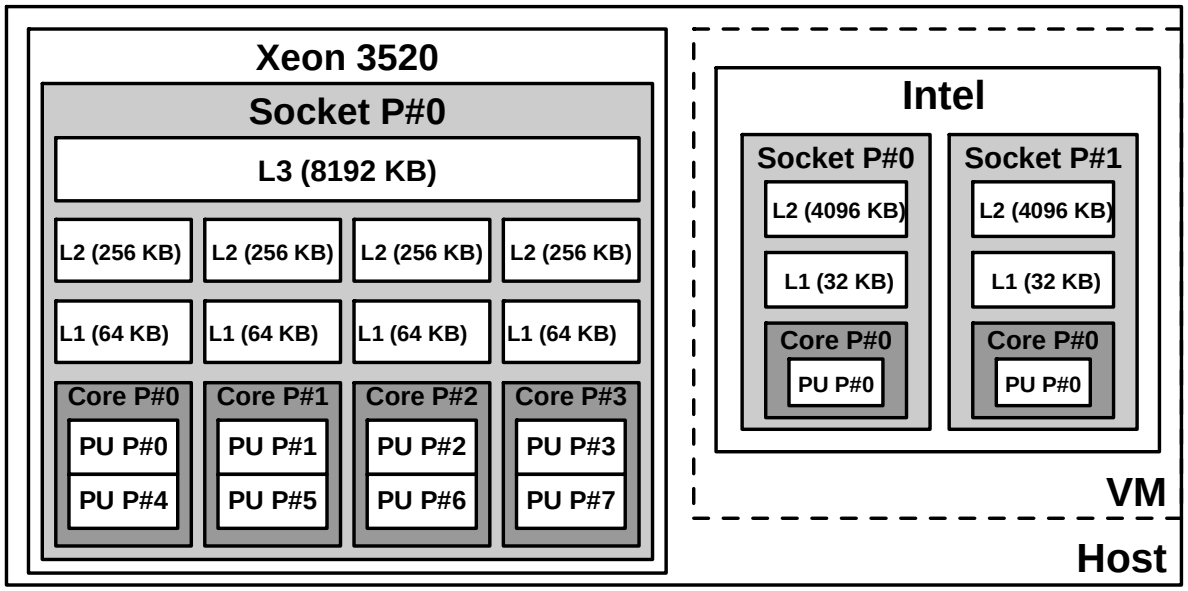
WattsKit

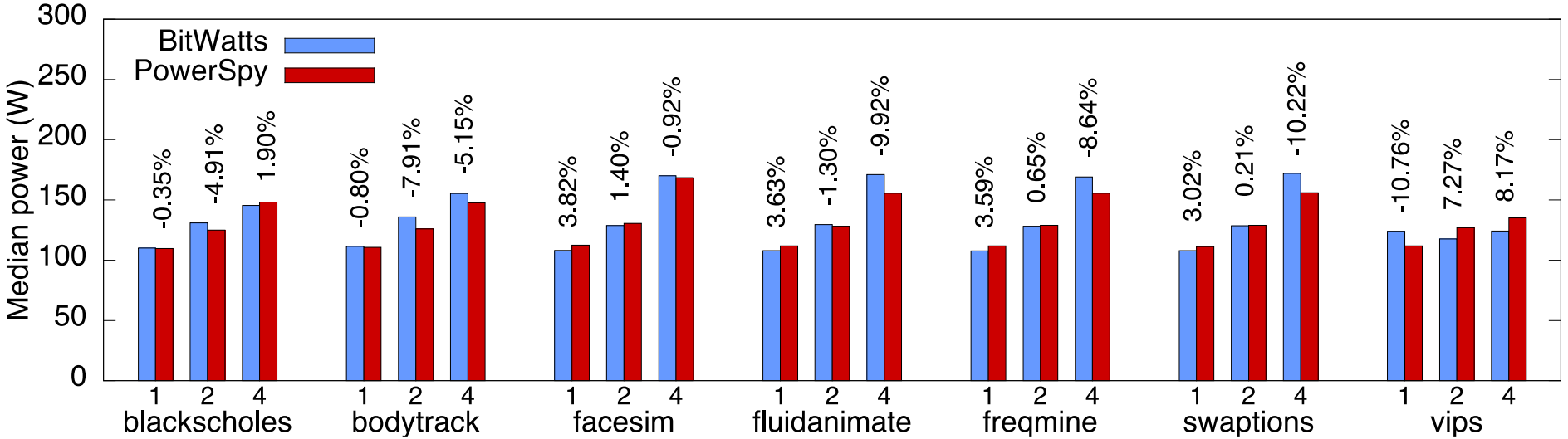
Traditional Architecture

Virtual Architecture

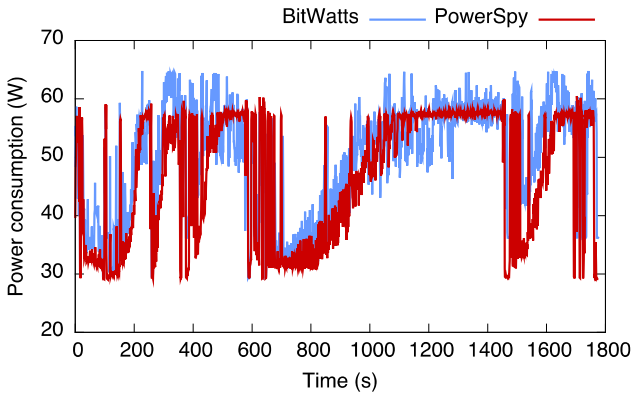


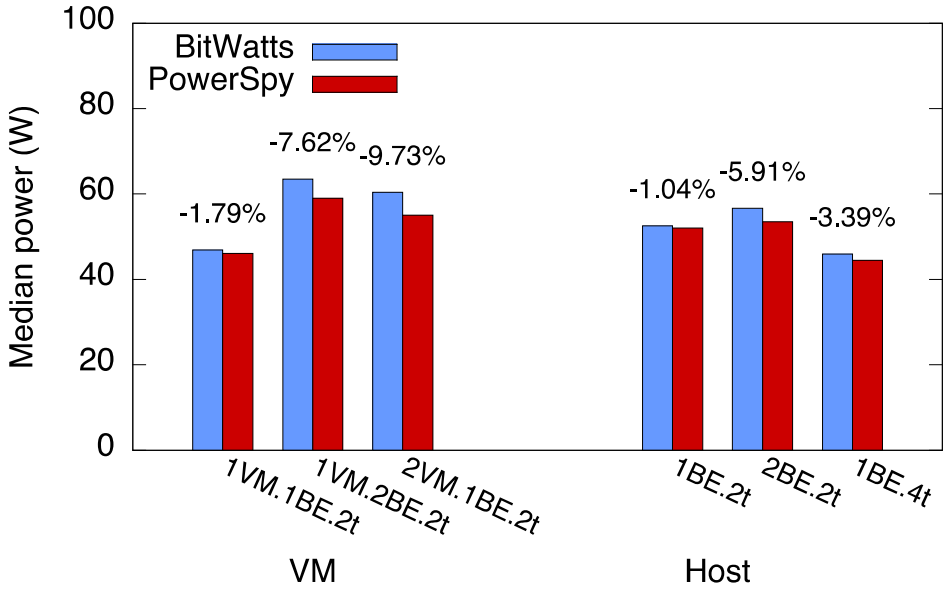






Name	Description
Host	
1BE.2t	1 backend pinned to 2 threads
2BE.2t	2 backends, each pinned to 2 threads
1BE.4t	1 backend with 4 threads
VM	
1BE.1VM.2t	1 backend, 2 threads, 1 VM
1BE.2VM.2t	1 backend, 2 threads, 2 VMs
2BE.1VM.2t	2 backends, each 2 threads, 1 VM
Distributed	
1BE.4t	2 hosts, 1 backend, 4 threads
1BE.1VM.2t	2 hosts, 1 backend, 2 threads, 1 VM





En résumé

PowerAPI

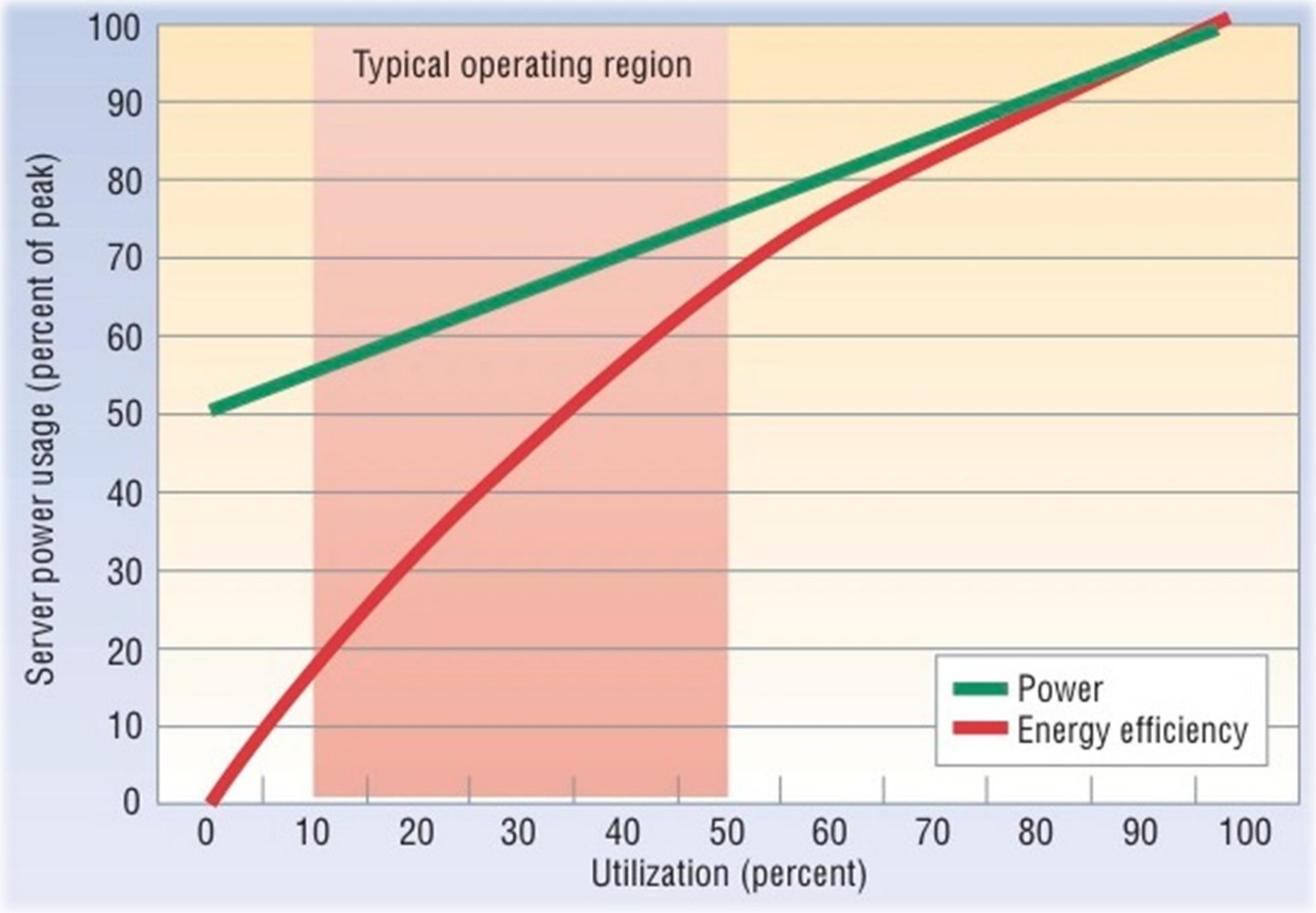
Software-Defined Power-meter

- Aucun matériel requis (fonctionne avec RAPL)
- Mesure à l'echelle de l'app/conteneur
  - Utilisation des c-groups (Docker, K8S, etc.)
- Estimation en temps-réel de la consommation
  - Native : CPU & DRAM
  - Calibration : Disk & Network
- Environnements distribués et virtualisés
  - E.g., Cloud privé (pas AWS)
- Surcoût limité (~1–2W par nœud)
- OSS: http://powerapi.org

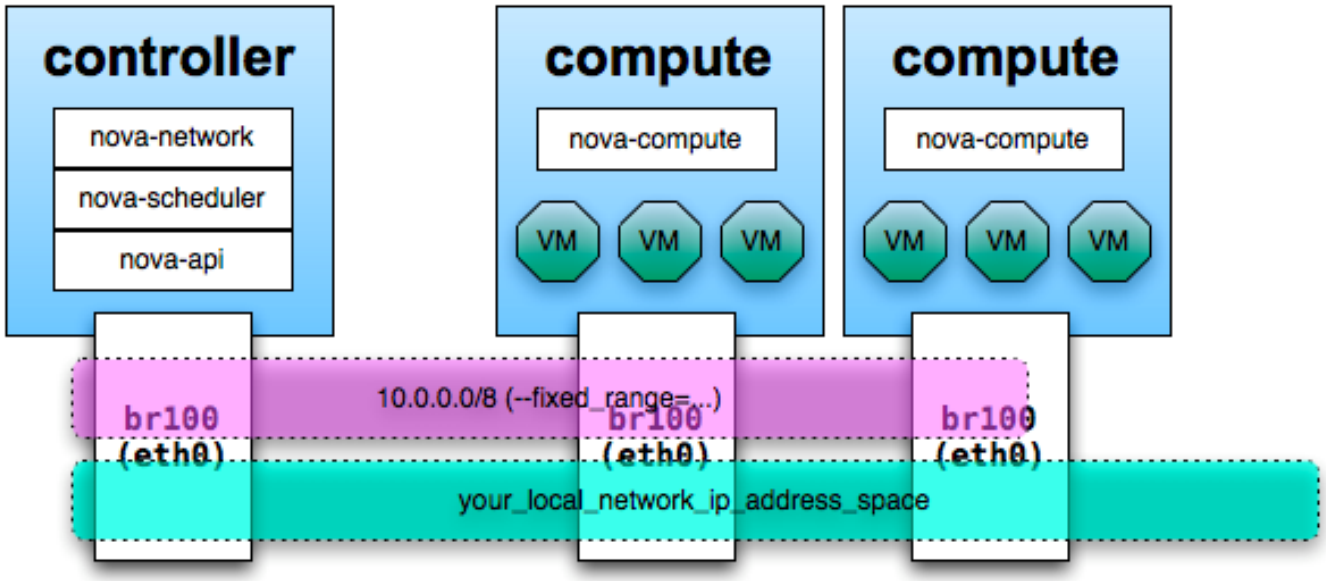
Et maintenant ??

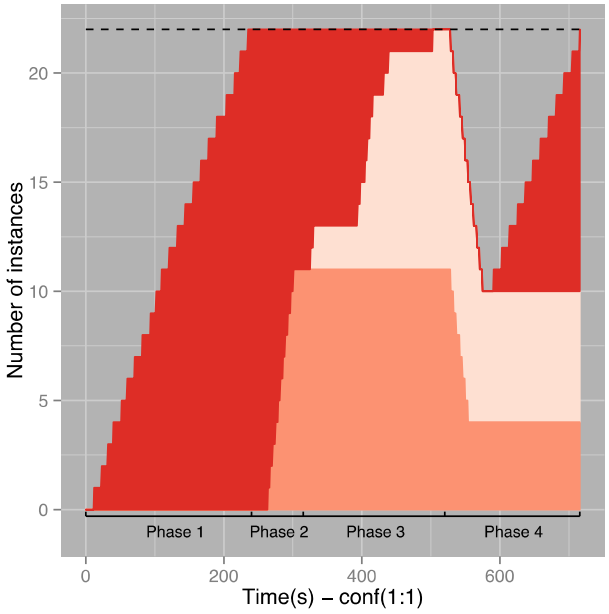
SPIRALS


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

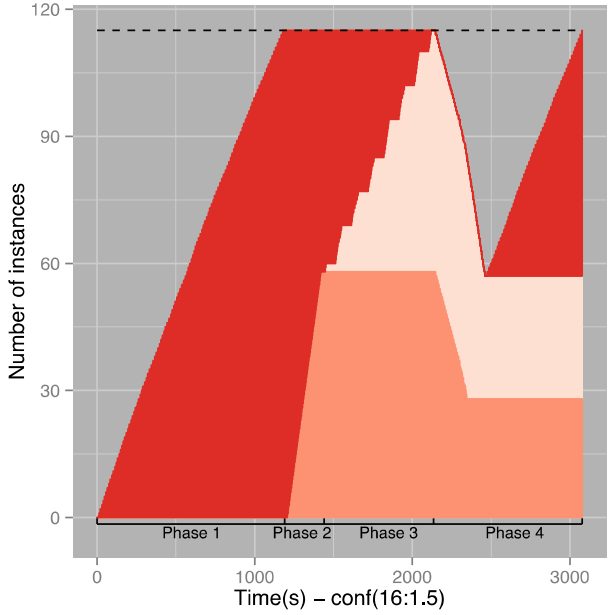


L. A. Barroso, U. Hölzle: *The Case for Energy-Proportional Computing*. IEEE Computer 40(12) 2007

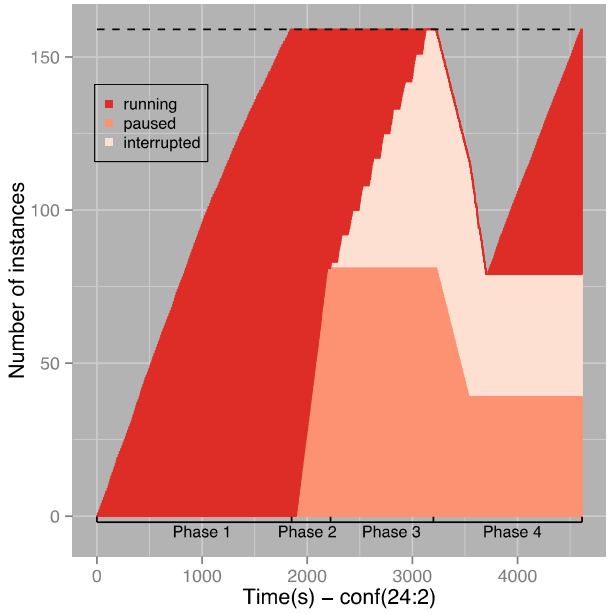




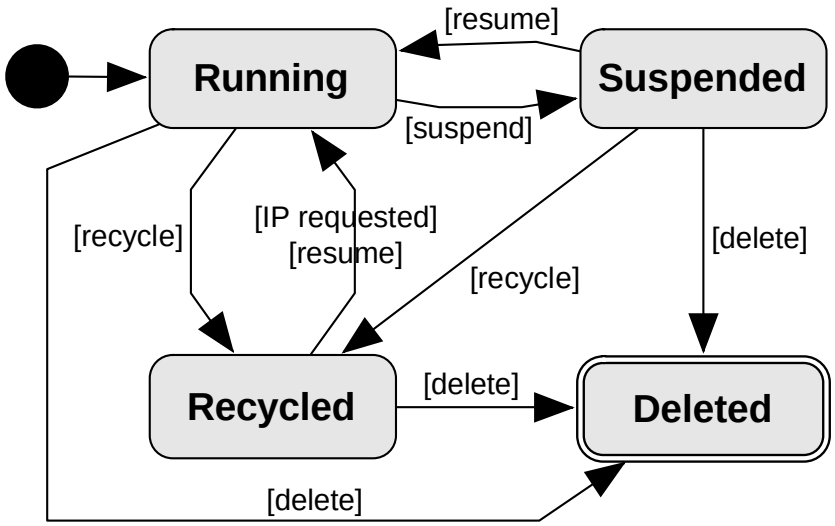
(a) vCPU limitation of straight

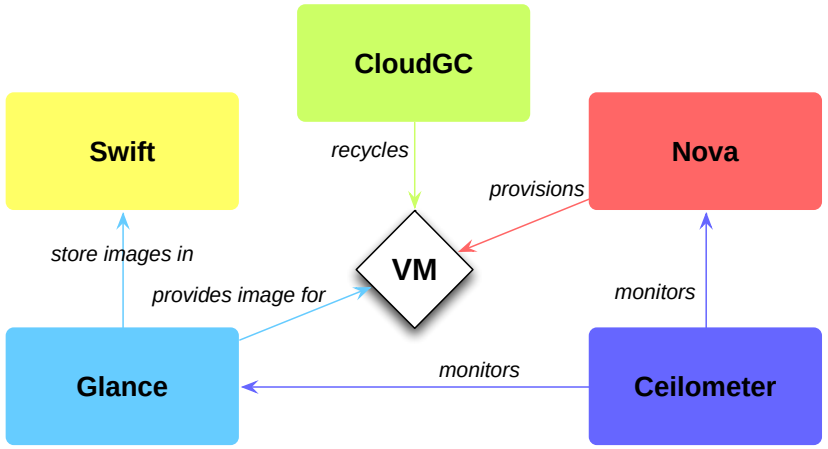


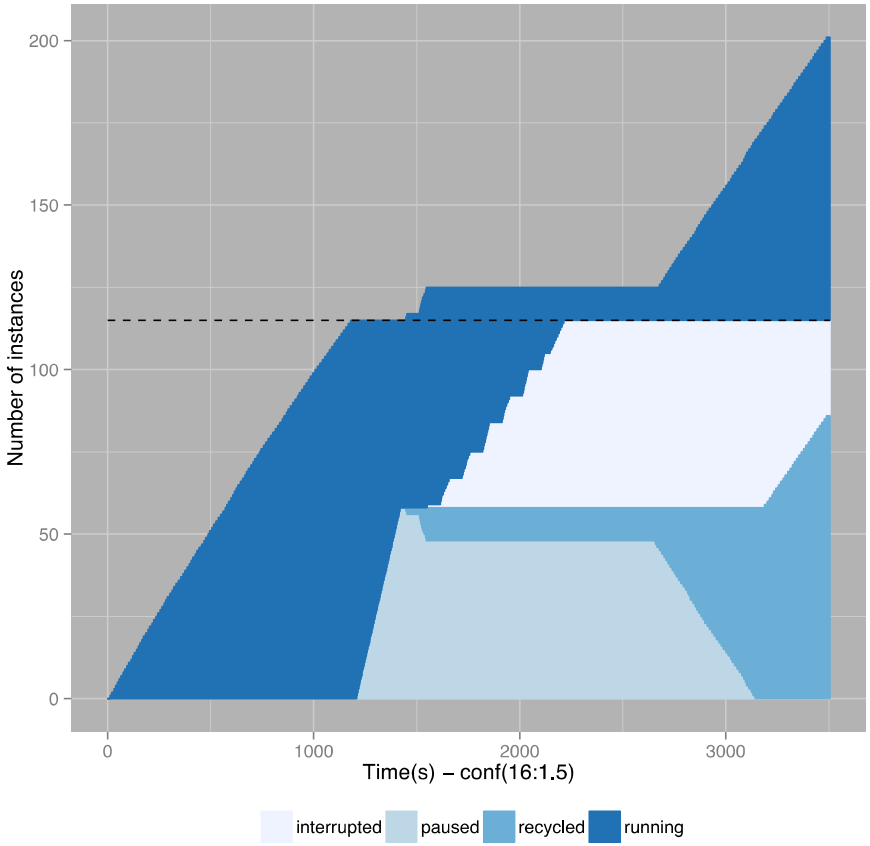
(b) vRAM limitation of standard

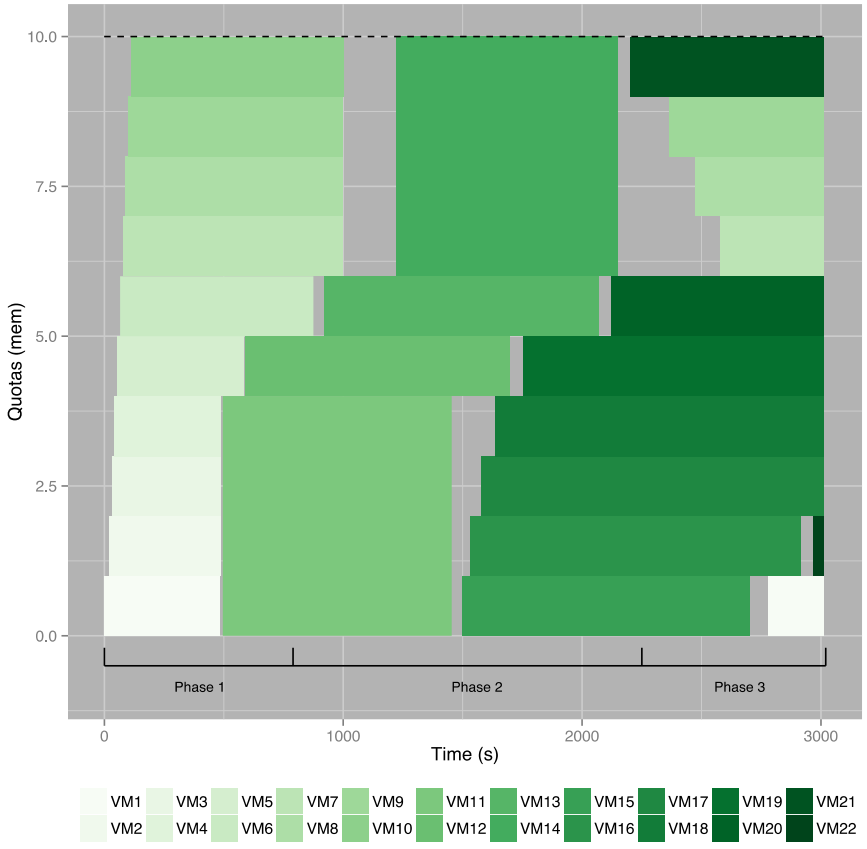


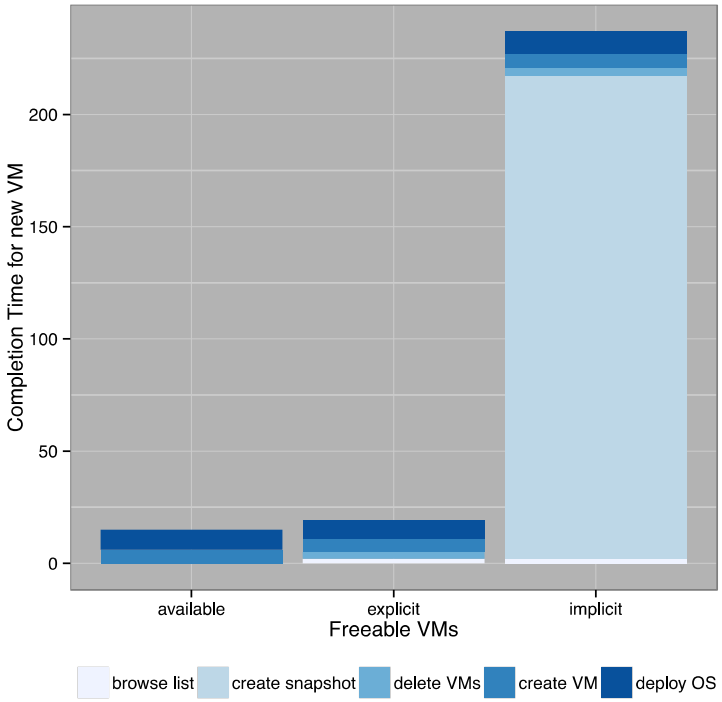
(c) vRAM limitation of over-commit


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

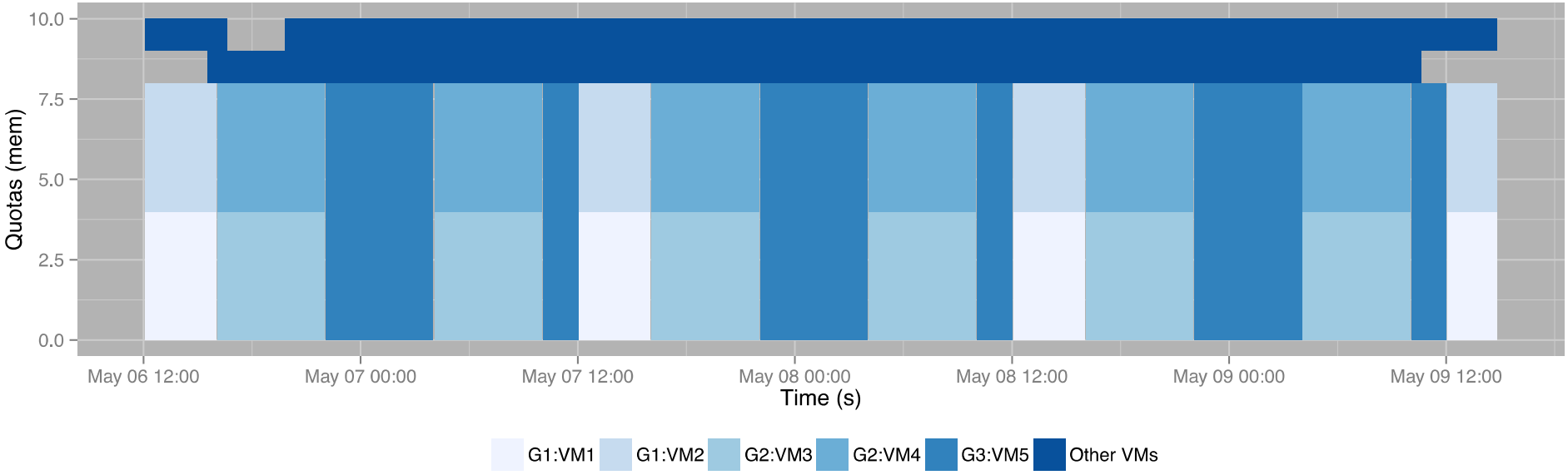

CloudGC
Swift
Nova
recycles
provisions
store images in
VM
provides image for
monitors
Glance
monitors
Ceilometer


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

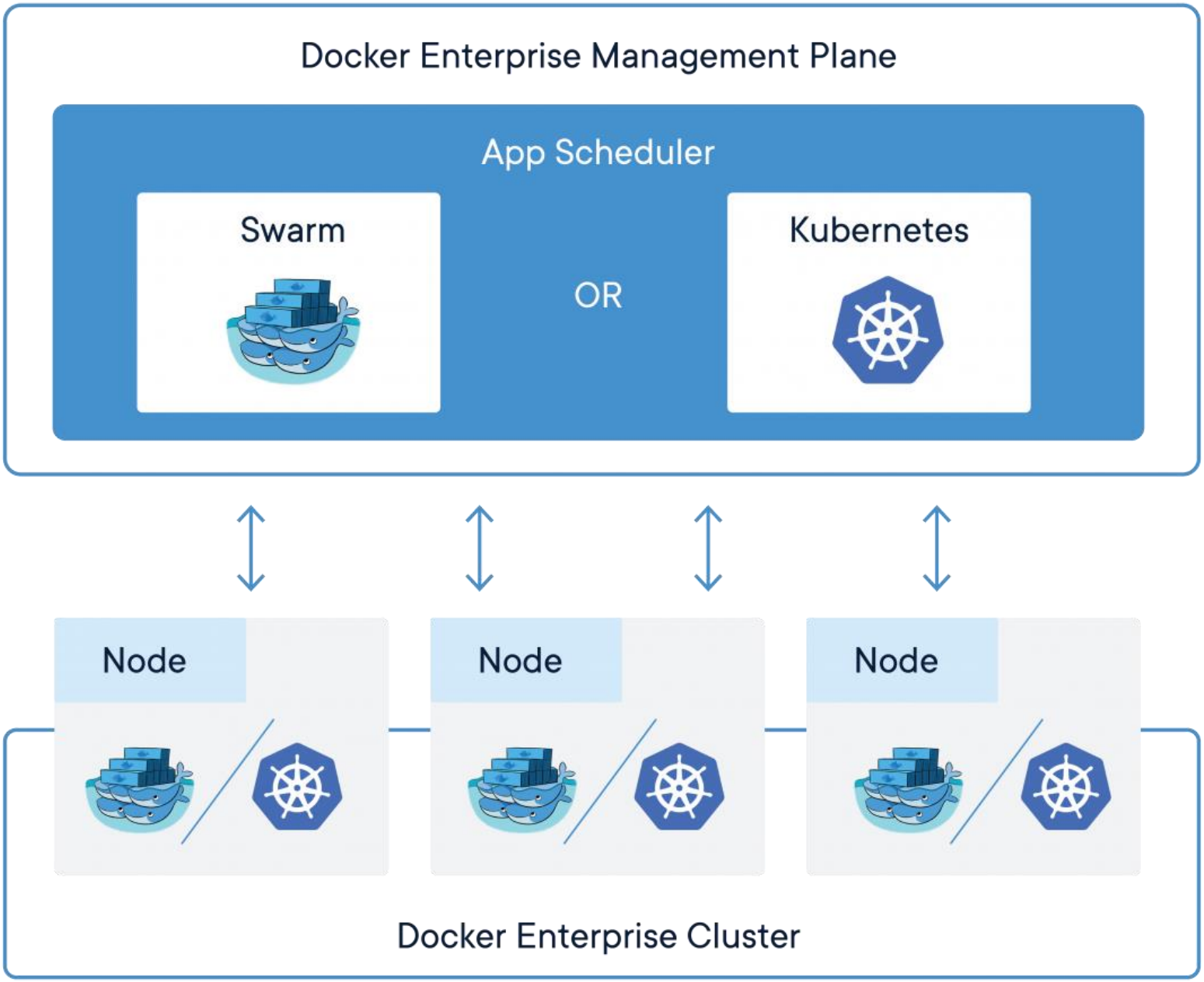

Quotas (mem)
10.0
7.5
5.0
2.5
0.0
Phase 1
Phase 2
Phase 3
0
1000
2000
3000
Time (s)
VM1
VM2
VM3
VM4
VM5
VM6
VM7
VM8
VM9
VM10
VM11
VM12
VM13
VM14
VM15
VM16
VM17
VM18
VM19
VM20
VM21
VM22

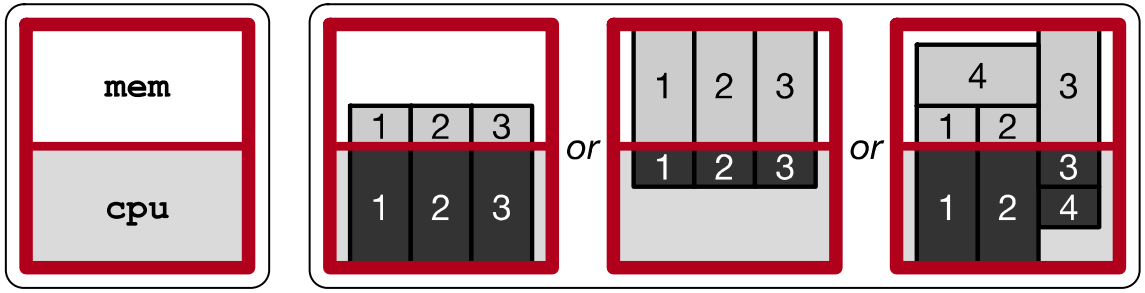


operation	available	explicit	implicit
browse list	-	2 *sec*	2 *sec*
create snapshot	-	-	215 *sec*
delete instance	-	3 *sec*	3 *sec*
create instance	6 *sec*	6 *sec*	6 *sec*
deploy OS	9 *sec*	9 *sec*	9 *sec*
total	15 *sec*	20 *sec*	235 *sec*

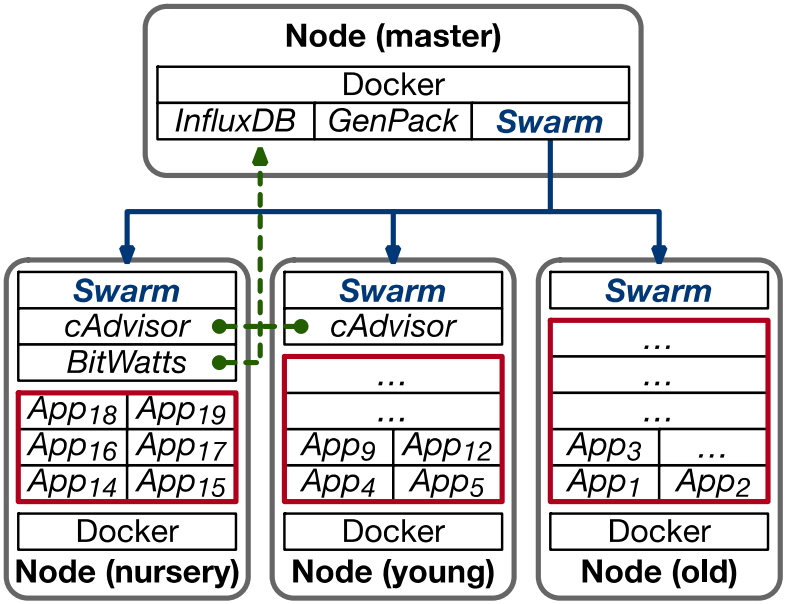


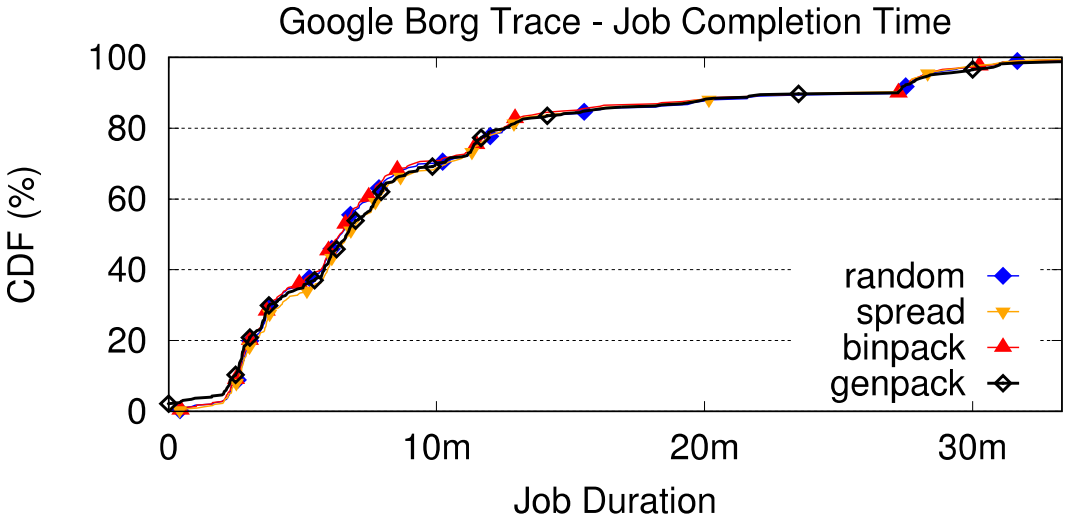

VOUS ÊTES ICI
Software (Framework)
Software (JVM)
Software (Container)
Software (OS)
Software (VM)
Software (Hypevisor)
Software (OS)
Hardware

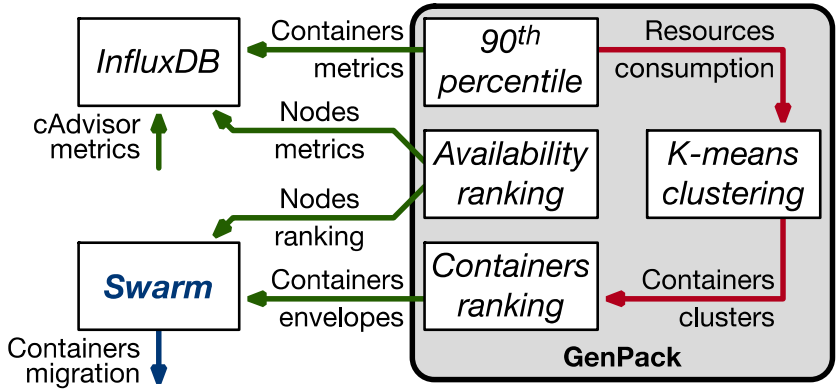

Docker Enterprise Management Plane
App Scheduler
Swarm
OR
Kubernetes
Node
Node
Node
Docker Enterprise Cluster

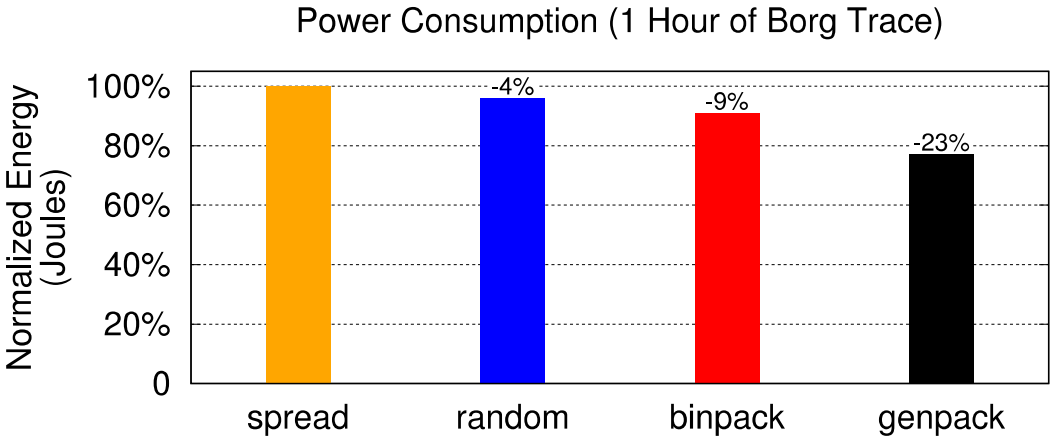


GenPack

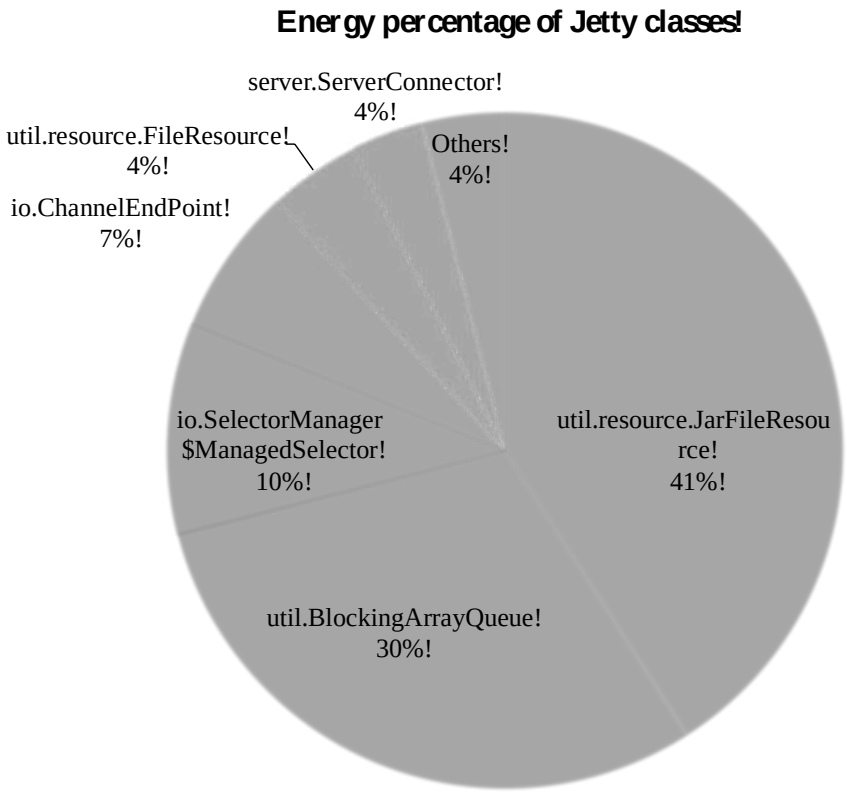

Node (master)
Docker
InfluxDB
GenPack
Swarm
Swarm
cAdvisor
BitWatts
App18
App19
App16
App17
App14
App15
Docker
Node (nursery)
Swarm
cAdvisor
...
...
App9
App12
App4
App5
Docker
Node (young)
Swarm
...
...
...
App3
...
App1
App2
Docker
Node (old)

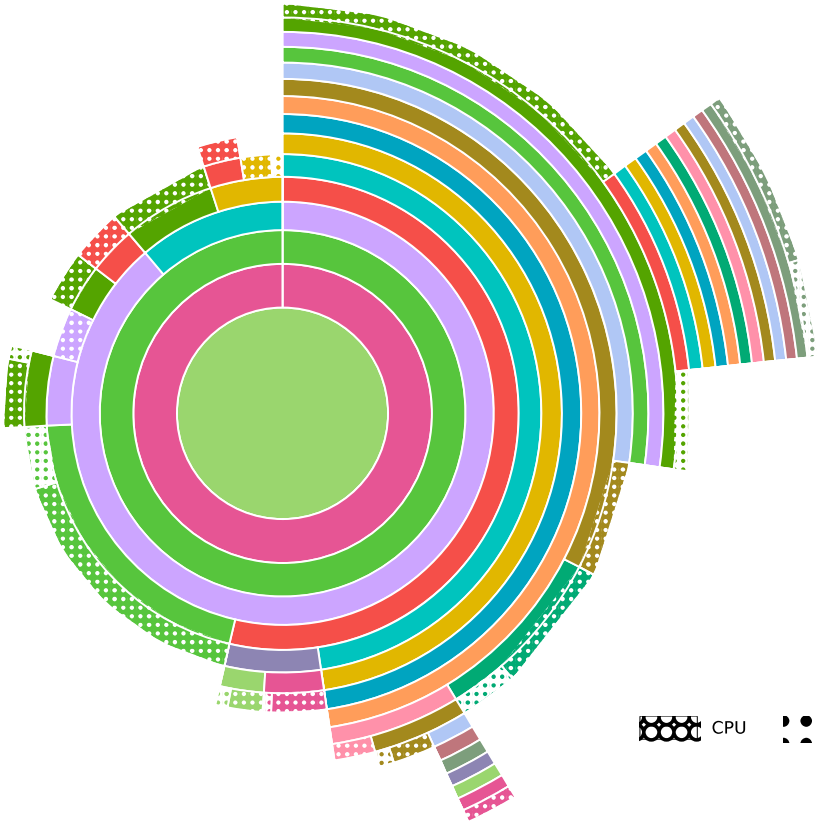

Google Borg Trace - Job Completion Time
CDF (%)
100
80
60
40
20
0
0
10m
20m
30m
Job Duration
random
spread
binpack
genpack

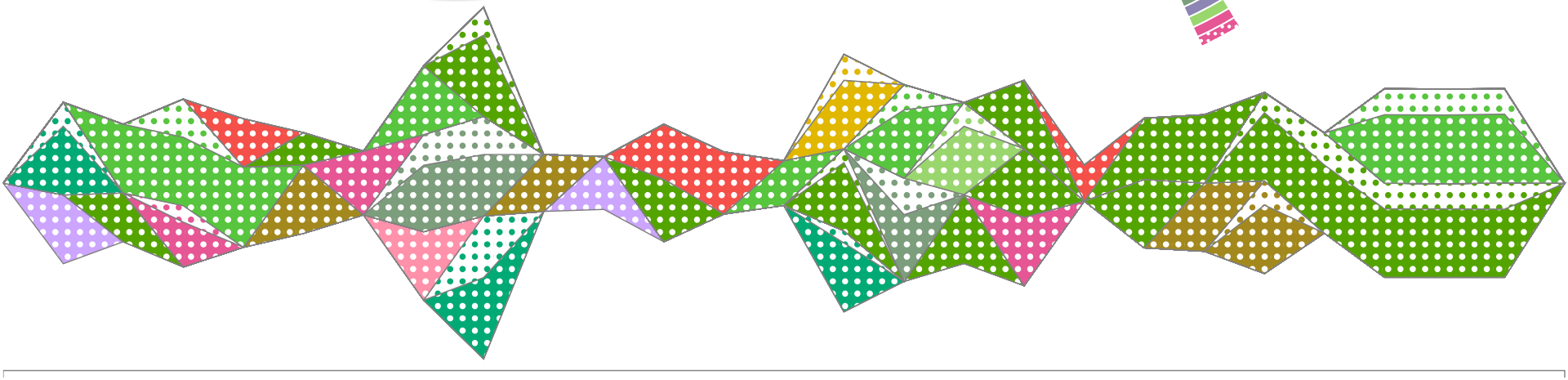

InfluxDB
Containers metrics
90th percentile
Resources consumption
cAdvisor metrics
Nodes metrics
Availability ranking
K-means clustering
Nodes ranking
Swarm
Containers envelopes
Containers ranking
Containers clusters
Containers migration
GenPack


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

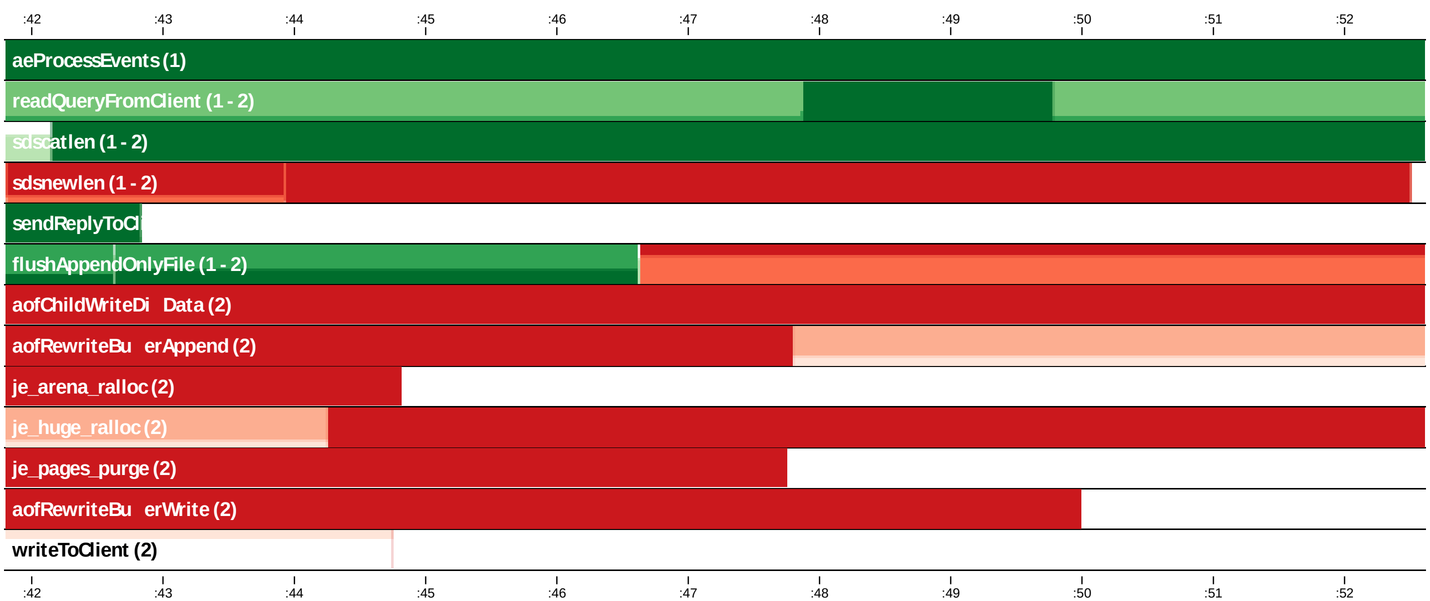
Peut-on étudier la consommation du code?



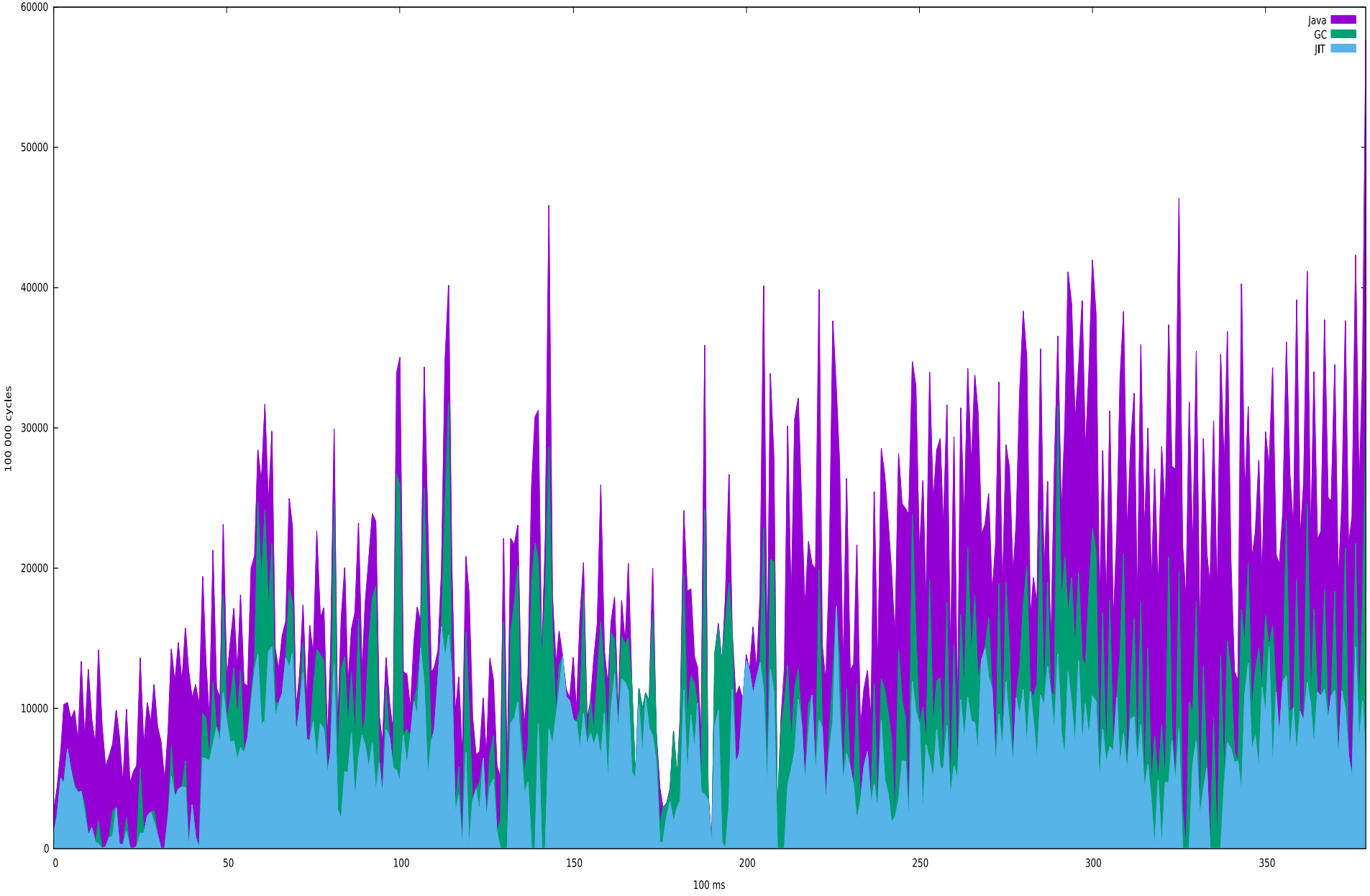




spring-boot-rest-example
Commit
N°7c9ae49
Total average energy
359.2 Mw
master
Number tests
8
Build
N°13
Execution time
15617 ms
should create hotel
Iterations
50
Energie Moyenne
Durée moyenne
Tendance
should delete existing hotel
Iterations
50
Energie Moyenne
Durée moyenne
Tendance



:42
:43
:44
:45
:46
:47
:48
:49
:50
:51
:52
aeProcessEvents (1)
readQueryFromClient (1 - 2)
sdscatlen (1 - 2)
sdsnewlen (1 - 2)
sendReplyToCl
flushAppendOnlyFile (1 - 2)
aofChildWriteDi Data (2)
aofRewriteBu erAppend (2)
je_arena_ralloc (2)
je_huge_ralloc (2)
je_pages_purge (2)
aofRewriteBu erWrite (2)
writeToClient (2)


Java
GC
JIT
60000
50000
40000
30000
20000
10000
0
100 000 cycles
0
50
100
150
200
250
300
350
100 ms

	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

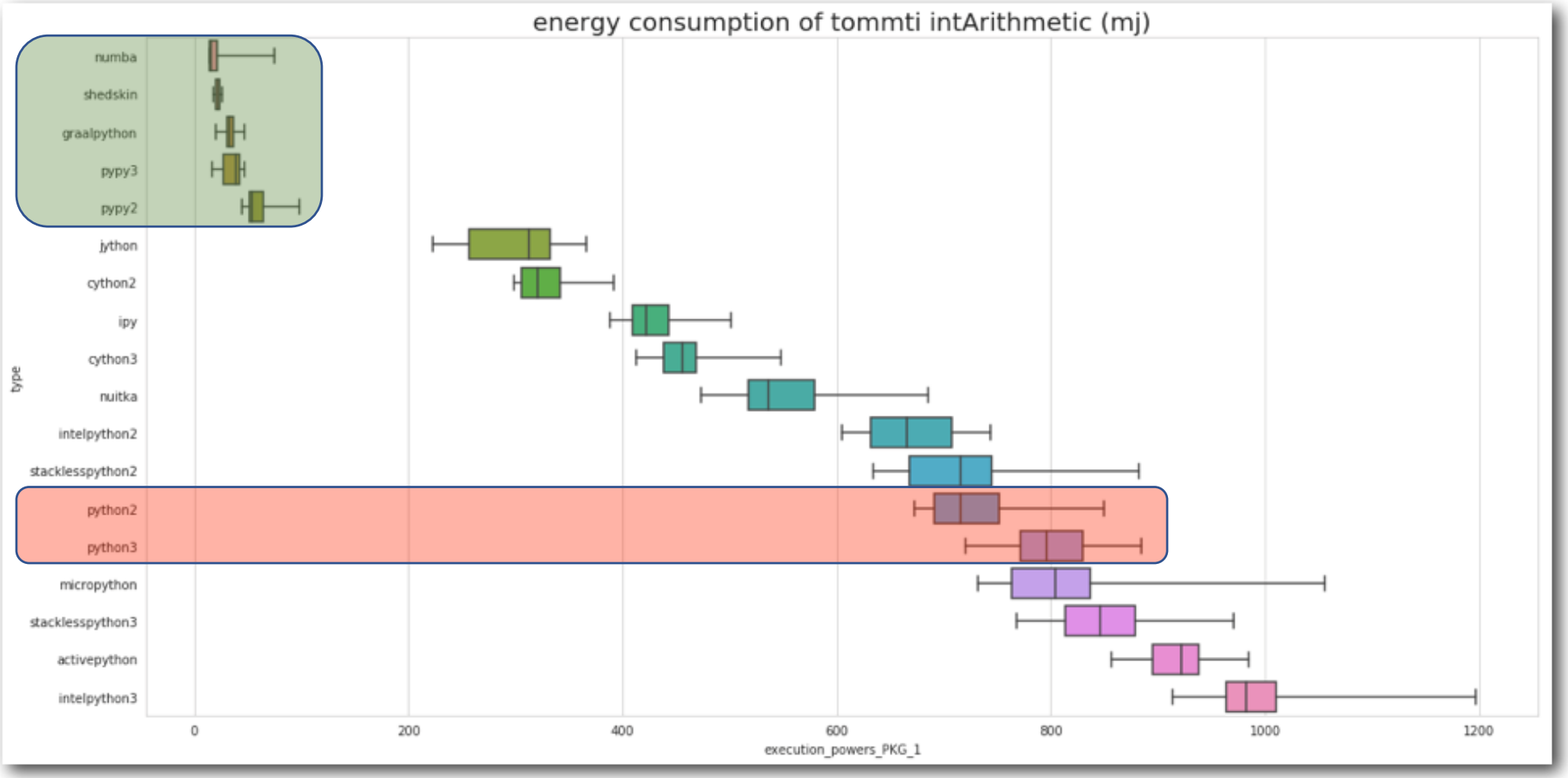
"Only **four languages maintain the same energy and time rank** (OCaml, Haskel, Racket, and Python), while the remainder are completely shuffled."

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

Mon *talk* en 180 secondes

- Explosion de la **consommation énergétique des TIC**
  - Multiplication des usages des services numériques
- Matériel **améliore son efficience** en continu
  - Bonne monture ne fait pas bon cavalier
- **Logiciels** sont plus que jamais **prépondérants**
  - *Everything is software-defined*
- **Énergie ≈ performance** (temps)
  - *Relationship: it's complicated*
- Nécessité de **travailler sur toutes les couches**
  - Chaque couche = logiciel optimisable

Et après ??

SPIRALS

Un thème, des milliers de sujets

- Quel est le **PUE** d'un programme ?
  - Peut-on quantifier le gâchis de ressources en cours d'execution?
- Comment tirer partie de l'**hétérogénité matérielle** ?
  - Comment exploiter les composants tiers (GPU, FPGA) du cloud ?
- Comment mieux **conseiller les développeurs** ?
  - Comment enricher l'analyse statique à partir d'analyses dynamiques ?
- Quid de l'**impact de la maintenance** logicielle ?
  - Quelle énergie (globale) est dépensée pour corriger un *bug* ?
- Quid d'autres **environnements d'exécution** ?
  - Android / iOS, IoT, ROS…
- …

Rejoignez la G-Team

Lionel **Seinturier**

Université de Lille

Adel **Noureddine**

Inria

Maxime **Colmant**

ADEME

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

Guillaume **Fieni**

Université de Lille

Chakib **Belgaid**

DAVIDSON CONSULTING

Zakaria **Ournani**

orange™

Aurélien **Bourdon**

Inria

Loïc **Huertas**

Inria

Arthur **D'Azémar**

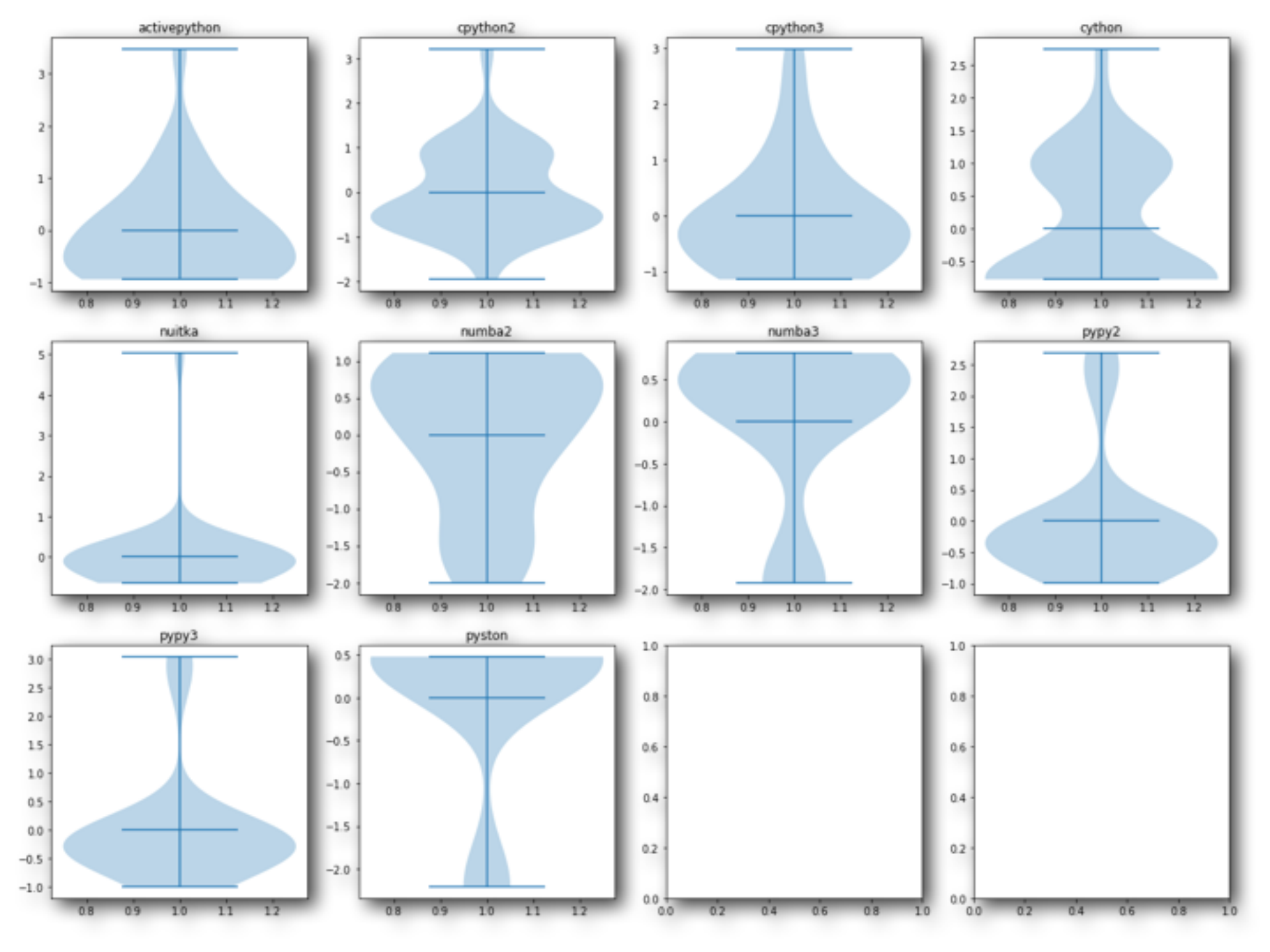
Inria

Jordan **Bouchoucha**

Université de Lille

Pascal **Felber**, *Bo* **Zhang**, *Aurélien* **Havet**, *Mascha* **Kurpicz**, *Valerio* **Schiavoni**, *Anita* **Sobe**, *Christof* **Fetzer,** *Yahya* **Al-Dhuraibi**, *Fawaz* **Paraiso**, *Georges-Aaron* **Randrianaina,** *Antoine* **Huyghes**…

1. **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)
2. **WattsKit: Software-Defined Power Monitoring of Distributed Systems**. M. Colmant, P. Felber, R. Rouvoy, L. Seinturier: *CCGrid'17*: 514-523
3. **GENPACK: A Generational Scheduler for Cloud Data Centers**. A. Havet, A. Schiavoni, P. Felber, M. Colmant, R. Rouvoy, C. Fetzer: *IC2E'17*: 95-104
4. **CLOUDGC: Recycling Idle Virtual Machines in the Cloud**. B. Zhang, Y. Al-Dhuraibi, R. Rouvoy, F. Paraiso, L. Seinturier: *IC2E'17*: 105-115
5. **Process-level power estimation in VM-based systems**. M. Colmant, M. Kurpicz, P. Felber, L. Huertas, R. Rouvoy, A. Sobe: *EuroSys'15*: 1-14
6. **Unit testing of energy consumption of software libraries**. A. Noureddine, R. Rouvoy, L. Seinturier: *SAC'14*: 1200-1205
7. **A preliminary study of the impact of software engineering on GreenIT**. A. Noureddine, A. Bourdon, R. Rouvoy, L. Seinturier: *GREENS'12*: 21-27
8. **Runtime monitoring of software energy hotspots**. A. Noureddine, A. Bourdon, R. Rouvoy, L. Seinturier: *ASE'12*: 160-169


activepython
cpython2
cpython3
cython
nuitka
numba2
numba3
pypy2
pypy3
pyston