# Reducing energy consumption of fault tolerance algorithms

Mehdi Diouri, Olivier Glück, Laurent Lefèvre, Franck Cappello
A.-C. Orgerie, M. Dias, J.-P. Gelas
INRIA – Université de Lyon – LIP

The Sixth Workshop of the INRIA-Illinois Joint Laboratory on Petascale Computing
November 21-23, 2011, Urbana, USA


ENS DE LYON

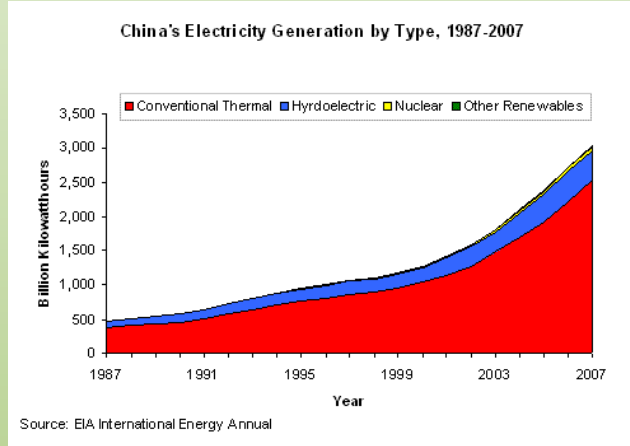
Lyon 1

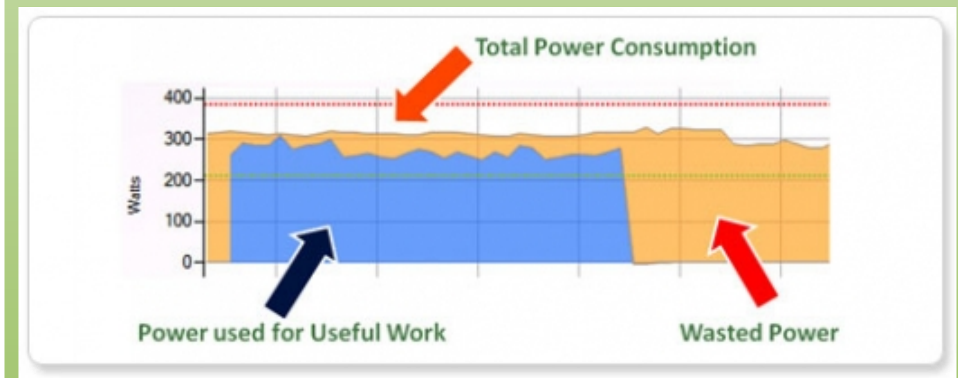
# Outline

- **Towards energy efficient large scale distributed systems**
- Reducing energy consumption of fault tolerance algorithms (Joint Very Early Results)

# Power demand and Green IT explosion

- IT – 2-5% of $CO_2$ emissions (the same as aviation)
- Green IT: reducing electrical consumption of IT equipments - $CO_2$ impact depends on countries
- Focus on usage: fighting un-used/over-provisioned plugged resources
- Problem: grey energy (ecoinfo.org)
- Green IT scientific events





# Explosion of initiatives

For each domain

- Data centers/HPC: Green500 (8Gflops/W –> 40 Gflops/W), EU CoC
- Grids: The Green Grid (metrics) / Open Grid Forum
- Storage: SNIA (ON/Off disks, metrics)
- Networks: Green Touch (x1000 factor) / EEE (LPI)

# Energy: 1st challenge for large scale systems (datacenter, grids, clouds, internet)?

- Future exascale platforms: systems from 20 to 120MW (current 4-10MW)
- Ratio performance/energy:
  - when TOP500 becomes Green500: the energy aware TOP 500
  - current systems are around 1 Gflops/watt
  - exascale forecasts with today's technology: around 8 Gflops/watt but DARPA has fixed an objective of 40 Gflops/watt (threshold set to 25 MW for one exascale system)

# Green500 ranking

- Supported by Virginia Tech (US)
- *http://green500.org (june 2011 ranking)*

Green500

| Green500 Rank | MFLOPS/W | Site* | Computer* | Total Power (kW) |
|---|---|---|---|---|
| 1 | 2097.19 | IBM Thomas J. Watson Research Center | NNSA/SC Blue Gene/Q Prototype 2 | 40.95 |
| 2 | 1684.20 | IBM Thomas J. Watson Research Center | NNSA/SC Blue Gene/Q Prototype 1 | 38.80 |
| 3 | 1375.88 | Nagasaki University | DEGIMA Cluster, Intel i5, ATI Radeon GPU, Infiniband QDR | 34.24 |
| 4 | 958.35 | GSIC Center, Tokyo Institute of Technology | HP ProLiant SL390s G7 Xeon 6C X5670, Nvidia GPU, Linux/Windows | 1243.80 |
| 5 | 891.88 | CINECA / SCS - SuperComputing Solution | iDataPlex DX360M3, Xeon 2.4, nVidia GPU, Infiniband | 160.00 |
| 6 | 824.56 | RIKEN Advanced Institute for Computational Science (AICS) | K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect | 9898.56 |
| 7 | 773.38 | Forschungszentrum Juelich (FZJ) | QPACE SFB TR Cluster, PowerXCell 8i, 3.2 GHz, 3D-Torus | 57.54 |
| 8 | 773.38 | Universitaet Regensburg | QPACE SFB TR Cluster, PowerXCell 8i, 3.2 GHz, 3D-Torus | 57.54 |
| 9 | 773.38 | Universitaet Wuppertal | QPACE SFB TR Cluster, PowerXCell 8i, 3.2 GHz, 3D-Torus | 57.54 |
| 10 | 718.13 | Universitaet Frankfurt | Supermicro Cluster, QC Opteron 2.1 GHz, ATI Radeon GPU, Infiniband | 416.78 |

* Performance data obtained from publicly available sources including TOP500

# Green500 vs. TOP500 (1)

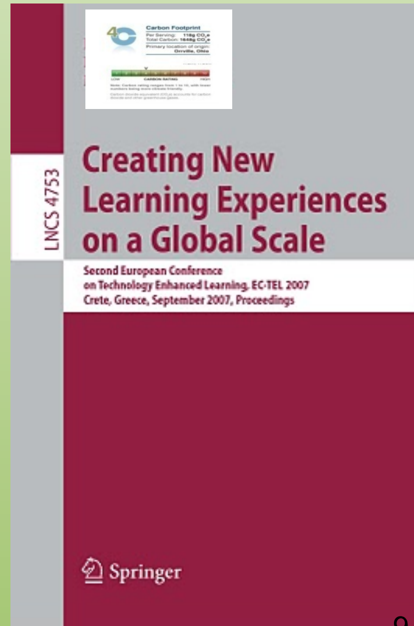
| Green500 rank | Site | Manufacturer | Computer | Country | mf/watt | Power (kW) | Top500 rank |
|---|---|---|---|---|---|---|---|
| 1 | IBM Thomas J. Watson Research Center | IBM | NNSA/SC Blue Gene/Q Prototype 2 | United States | 2097 | 40 | 109 |
| 2 | IBM Thomas J. Watson Research Center | IBM | NNSA/SC Blue Gene/Q Prototype 1 | United States | 1684 | 38 | 165 |
| 3 | Nagasaki University | Self-made | DEGIMA Cluster, Intel i5, ATI Radeon GPU, Infiniband QDR | Japan | 1375 | 34 | 430 |
| 4 | GSIC Center, Tokyo Institute of Technology | NEC/HP | HP ProLiant SL390s G7 Xeon 6C X5670, Nvidia GPU, Linux/Windows (TSUBAME) | Japan | 958 | 1243 | 5 |
| 5 | CINECA / SCS - SuperComputing Solution | IBM | iDataPlex DX360M3, Xeon 2.4, nVidia GPU, Infiniband | Italy | 891 | 160 | 54 |
| 6 | RIKEN Advanced Institute for Computational Science (AICS) | Fujitsu | K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect | Japan | 824 | 9898 | 1 |
| 7 | Forschungszentrum Juelich (FZJ) | IBM | QPACE SFB TR Cluster, PowerXCell 8i, 3.2 GHz, 3D-Torus | Germany | 773 | 57 | 406 |
| 8 | Universitaet Regensburg | IBM | QPACE SFB TR Cluster, PowerXCell 8i, 3.2 GHz, 3D-Torus | Germany | 773 | 57 | 407 |
| 9 | Universitaet Wuppertal | IBM | QPACE SFB TR Cluster, PowerXCell 8i, 3.2 GHz, 3D-Torus | Germany | 773 | 57 | 408 |
| 10 | Universitaet Frankfurt | Clustervision/ Supermicro | Supermicro Cluster, QC Opteron 2.1 GHz, ATI Radeon GPU, Infiniband | Germany | 718 | 416 | 22 |
| 11 | Georgia Institute of Technology | Hewlett-Packard | HP ProLiant SL390s G7 Xeon 6C X5660 2.8Ghz, nVidia Fermi, Infiniband QDR | United States | 677 | 94 | 169 |
| 12 | National Institute for Environmental Studies | NSSOL / SGI Japan | Asterism ID318, Intel Xeon E5530, NVIDIA C2050, Infiniband | Japan | 650 | 115 | 126 |
| 13 | National Supercomputing Center in Tianjin | NUDT | NUDT TH MPP, X5670 2.93Ghz 6C, NVIDIA GPU, FT-1000 8C (TIANHE) | China | 635 | 4040 | 2 |

# Green 500vs.TOP 500 (2)

| Green500 rank | Site | Manufacturer | Computer | Country | mf/watt | Power (kW) | Top500 rank |
|---|---|---|---|---|---|---|---|
| 6 | RIKEN Advanced Institute for Computational Science (AICS) | Fujitsu | K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect | Japan | 824 | 9898 | 1 |
| 96 | DOE/SC/Oak Ridge National Laboratory | Cray Inc. | Cray XT5-HE Opteron 6-core 2.6 GHz | United States | 253 | 6950 | 3 |
| 495 | Government | Cray Inc. | Cray XT5 QC 2.4 GHz | United States | 34 | 4812 | 47 |
| 111 | Commissariat a l'Energie Atomique (CEA) | Bull SA | Bull bullx super-node S6010/S6030 | France | 228 | 4590 | 9 |
| 353 | Sandia National Laboratories / National Renewable Energy Laboratory | Oracle | Sun Blade x6275, Xeon X55xx 2.93 Ghz, Infiniband | United States | 99 | 4343 | 16 |
| 88 | NASA/Ames Research Center/NAS | SGI | SGI Altix ICE 8200EX/8400EX, Xeon HT QC 3.0/Xeon 5570/5670 2.93 Ghz, Infiniband | United States | 265 | 4102 | 7 |
| 13 | National Supercomputing Center in Tianjin | NUDT | NUDT TH MPP, X5670 2.93Ghz 6C, NVIDIA GPU, FT-1000 8C (TIANHE) | China | 635 | 4040 | 2 |
| 78 | DOE/NNSA/LANL/SNL | Cray Inc. | Cray XE6 8-core 2.4 GHz | United States | 278 | 3980 | 6 |
| 73 | National Institute for Computational Sciences/ University of Tennessee | Cray Inc. | Cray XT5-HE Opteron Six Core 2.6 GHz | United States | 297 | 3090 | 11 |
| 42 | DOE/SC/LBNL/NERSC | Cray Inc. | Cray XE6 12-core 2.1 GHz | United States | 362 | 2910 | 8 |
| 409 | Government | Hewlett-Packard | Cluster Platform 3000 BL2x220, L54xx 2.5 Ghz, Infiniband | France | 63 | 2806 | 41 |
| 496 | Financial Services Company (G) | Hewlett-Packard | Cluster Platform 3000 DL165, Opteron 2.1 GHz 12C, GigE | United States | 33 | 2598 | 104 |
| 16 | National Supercomputing Centre in Shenzhen (NSCS) | Dawning | Dawning TC3600 Blade, Intel X5650, NVidia Tesla C2050 GPU | China | 492 | 2580 | 4 |
| 383 | Sandia National Laboratories | Cray Inc. | Cray XT3/XT4 | United States | 81 | 2506 | 36 |
| 500 | NNSA/Sandia National Laboratories | Dell | PowerEdge 1850, 3.6 GHz, Infiniband | United States | 21 | 2481 | 252 |

# Energy: 1st challenge for large scale systems (datacenter, grids, clouds, internet)?

- How to build such systems and make them energy sustainable/responsible?
  - Hardware can help (component by component)
  - Software must be adapted to be scalable but also more energy efficient
  - Usage must be energy aware






# Bad usage example

- Bad usage (some French statistics ADEME / ENERTECH) about machines usage in companies:
  - CPU: 4004 h of running per year = 17.8 h per working day
  - Screen: 2510 h per year = 11.2 h per working day

- But users effectively use their machines: 686 h per year = 3 h a day
- Computers are switch on ¾ time for nothing!

# Towards Energy Aware Large Scale Systems: open questions

**How to decrease the energy consumption of large scale systems without impacting the performances?**

- How to understand and to analyze the usage and energy consumption of large scale platforms?
- How to monitor lively such usage from pico to large scale views?
- How to design energy aware software frameworks?
- How to help users to express theirs Green concerns and to express tradeoffs between performance and energy efficiency?

and then propose an Energy-Aware Reservation Infrastructure

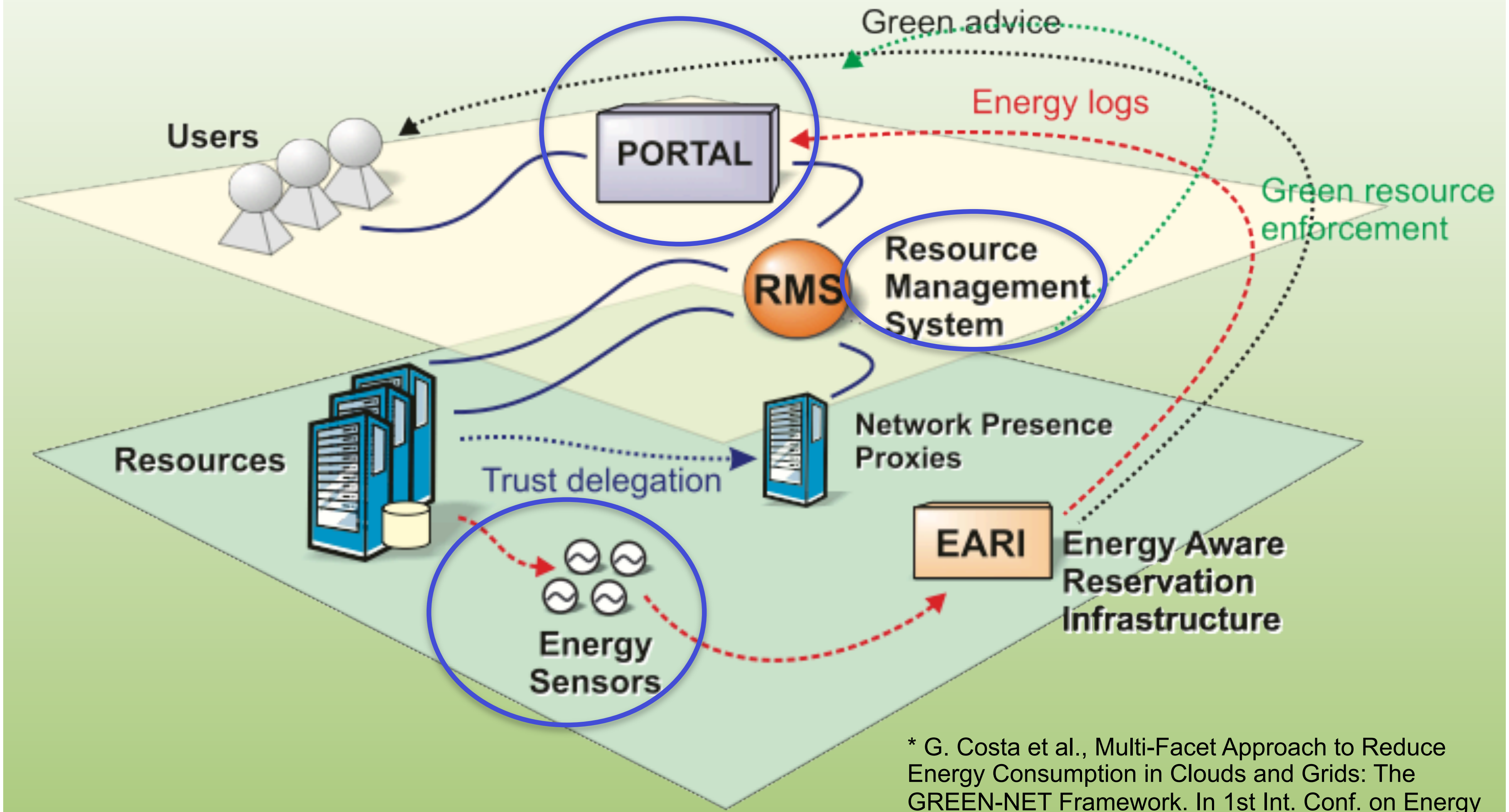
# Green-IT leverages

- **Shutdown**: reducing the amount of powered unused resources
- **Slowdown**: adapting the speed of resources to real usage
- **Optimizing**: improving hardware and software for energy reduction purpose
- **Coordinating:** using large scale approaches to enhance green leverages (task placement, synchronization aspects)

# The ERIDIS approach

- Energy-efficient Reservation Infrastructure for large-scale Distributed Systems
- Collecting and exposing usage, energy profiling of applications and infrastructures: to see where energy savings are possible
  - we have analyzed two years of logs on Grid5000 platform
  - fully monitored Lyon site by energy sensors (wattmeters)
- Expressing and proposing Green Policies: to deal with tradeoffs between performance and energy
  - Computing the energy consumption of each reservation: for the user in order to increase his energy-awareness, for the resource manager
  - Enforcing green leverages: shutdown or adapt performance
  - Predicting usage of infrastructures (for instance to not switch off resources that are required in a near future)
  - Aggregating some resource reservations to avoid frequent on/off
- Experimental validation by a replay on the Grid5000 traces

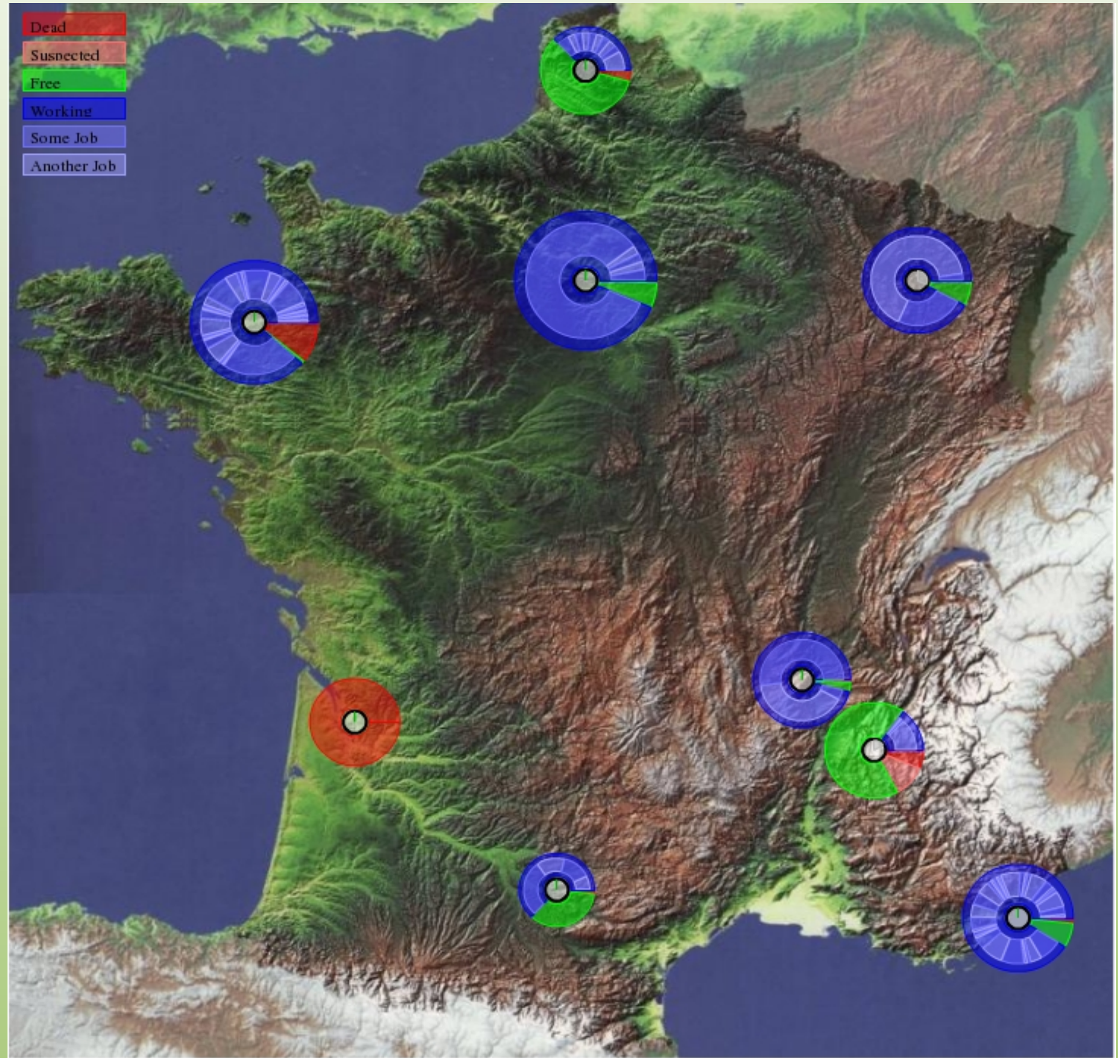
# The ERIDIS Framework for Grids




* G. Costa et al., Multi-Facet Approach to Reduce Energy Consumption in Clouds and Grids: The GREEN-NET Framework. In 1st Int. Conf. on Energy Efficient Computing and Networking, Passau, 2010.

# Collecting and exposing platform usage and energy consumption

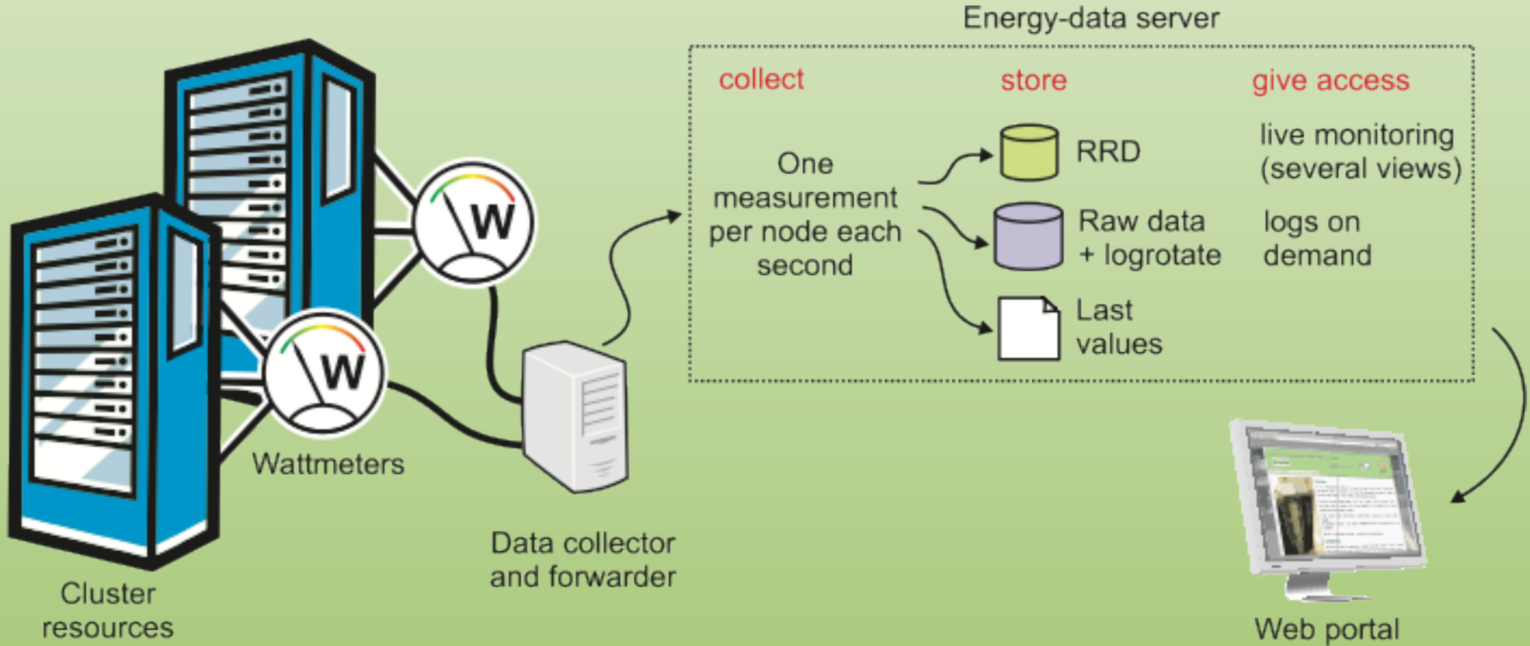
- Grid'5000
  - French experimental testbed
  - 5000 cores (8000 now)
  - 9 sites (10 now)



# The Green Grid5000

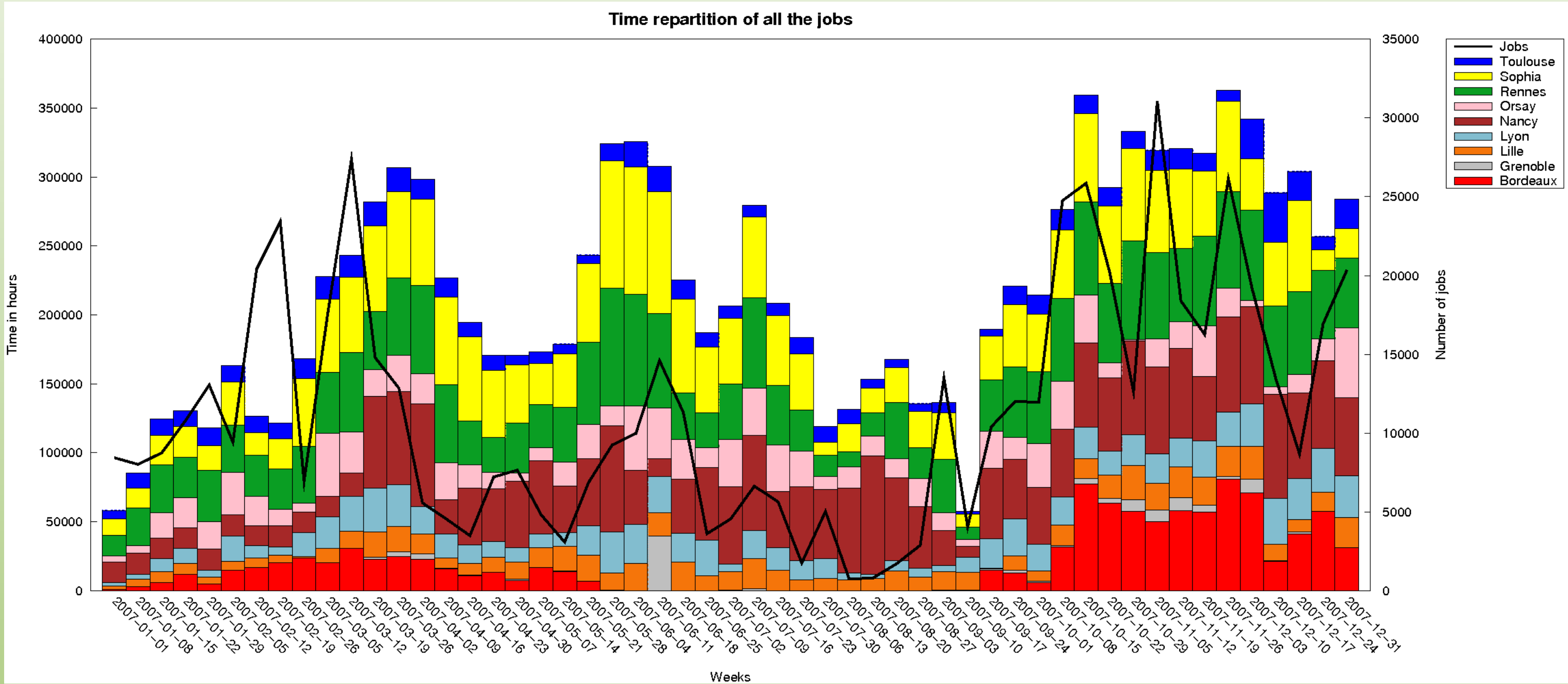
Energy sensors (one power measurement each second)

- 6 or 48 ports wattmeters boxes / PDUs
- Deployed on three sites of Grid5000 (Lyon, Grenoble, Toulouse)
- Library for **interfacing** with **energy sensors**
- **Client-side** applications to **obtain** and **store** the energy consumption data
- One year of energy logs for a 150-node platform = 70 GB



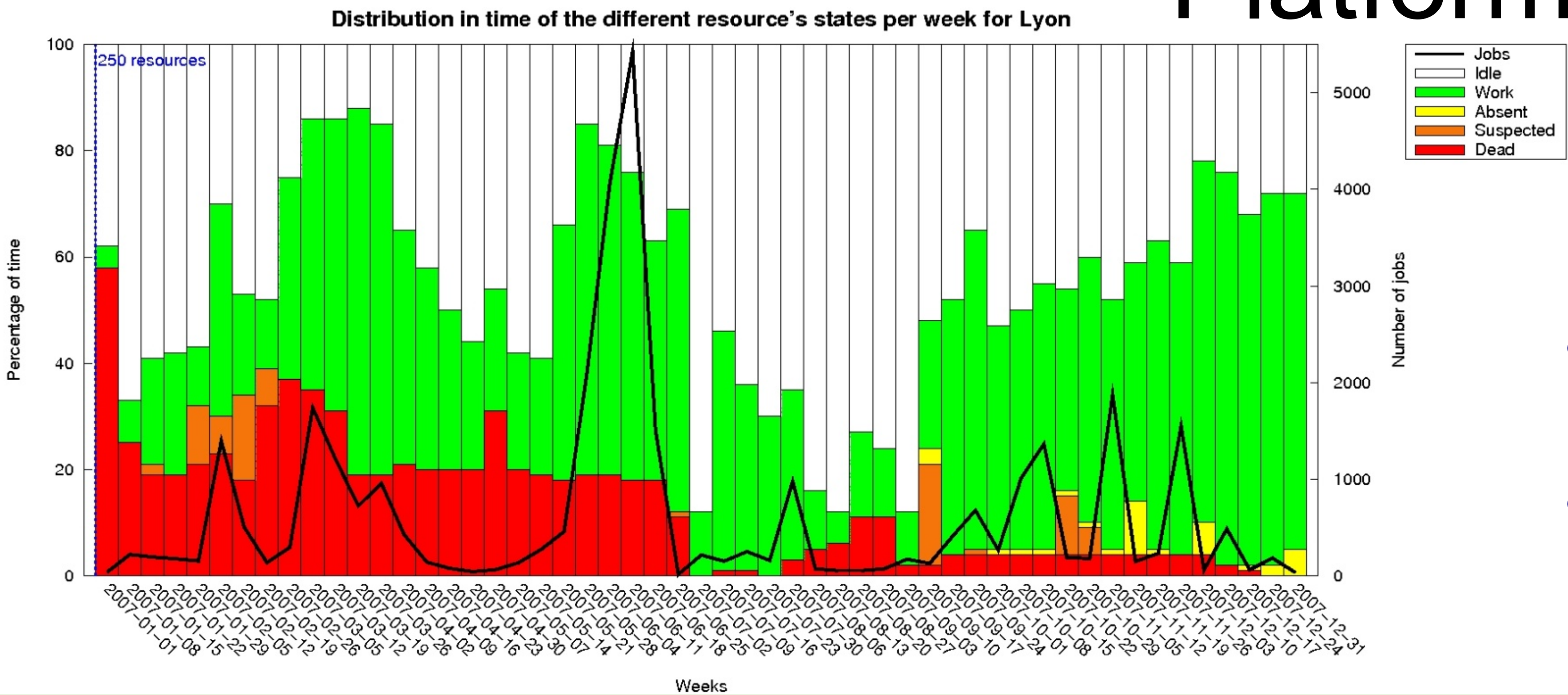

# Analysis of Grid5000 global usage

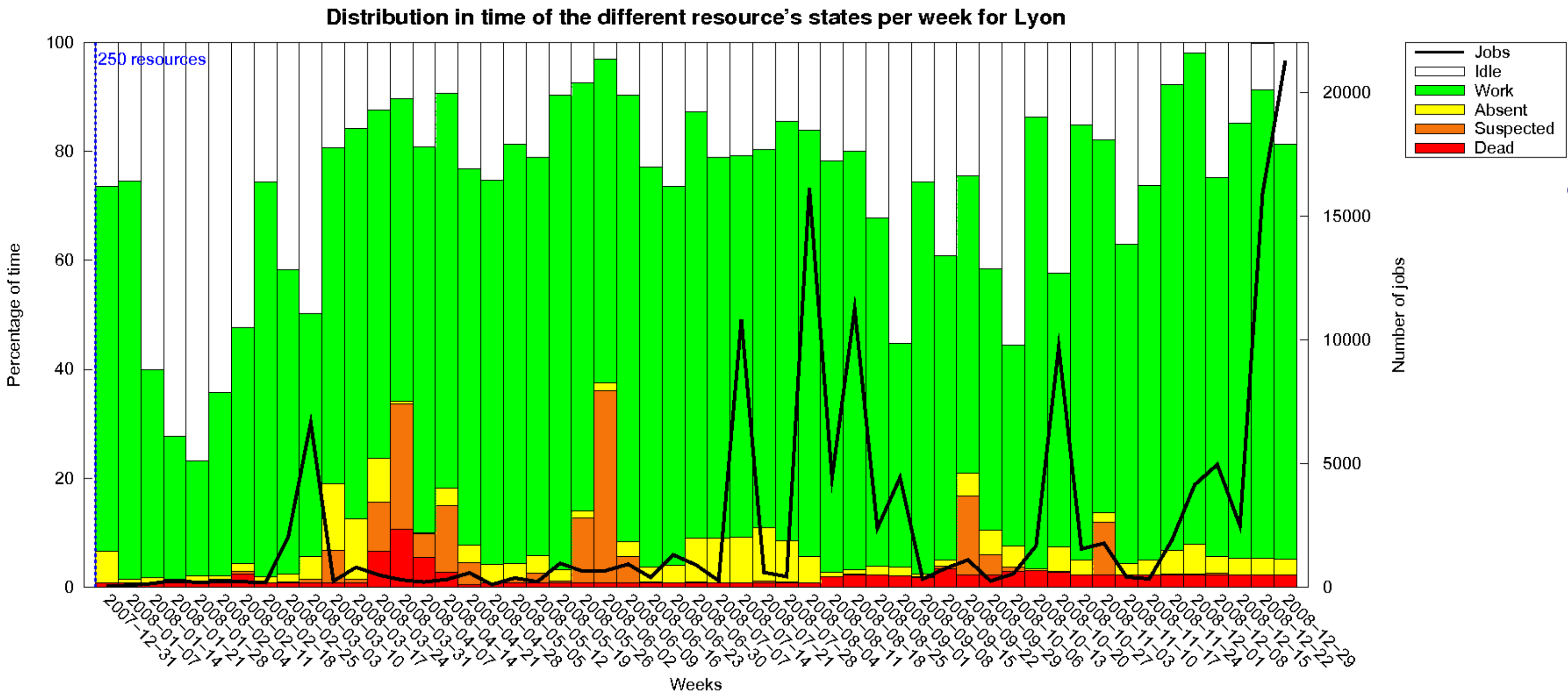
Using batch scheduler reservation system (OAR)
GBytes of reservation logs collected for 24 months period (here only 2007)




Some low average value but burst support - In operational Grids: 60% to 70% average usage - **Global usage is not enough -> need more precise views**

# Platform usage

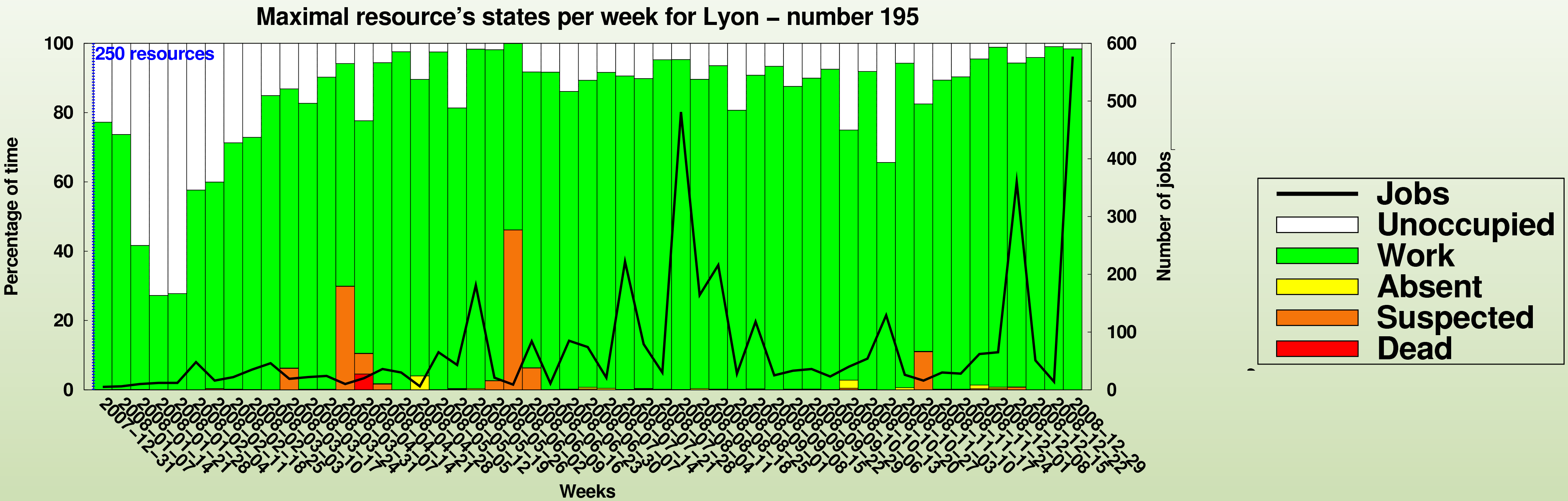


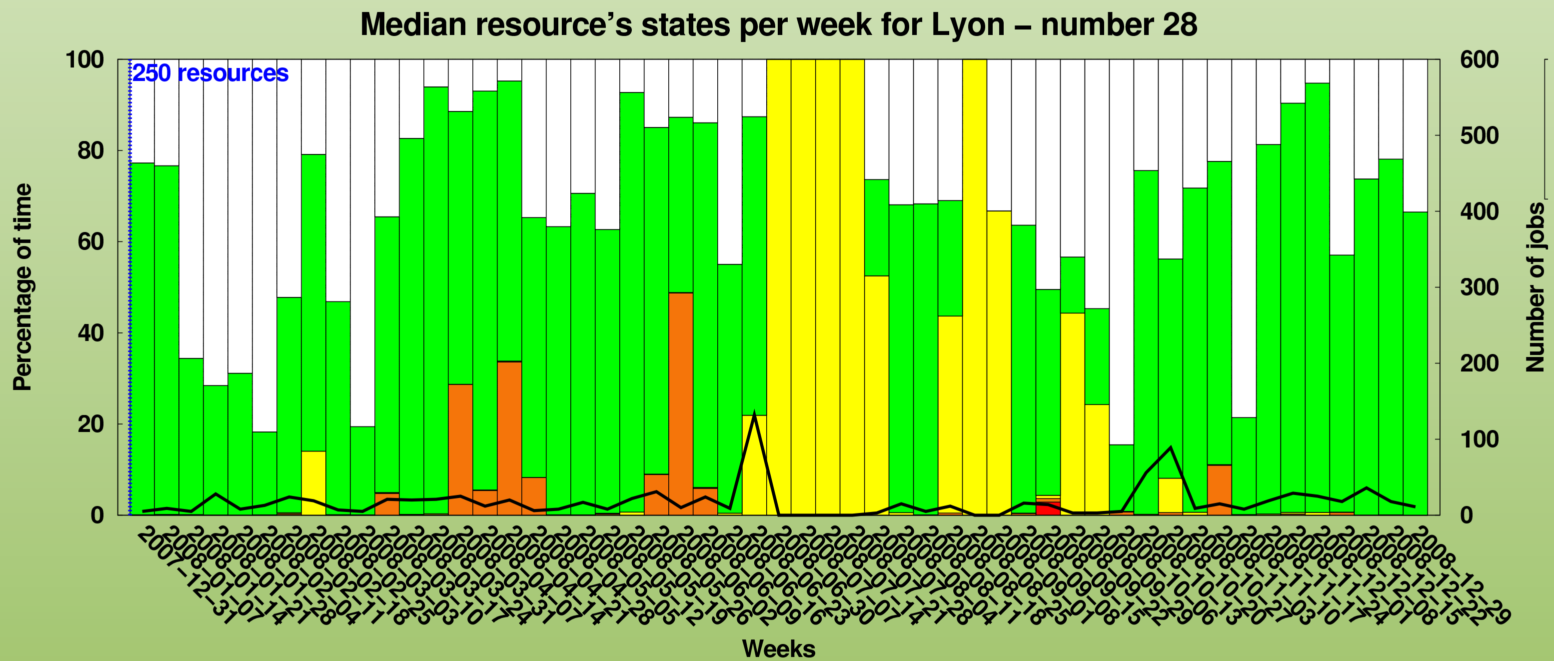




## Lessons Learned

- **Significant bursts -> significant gaps!**
- **A lot of small reservations (experimental platform)**
- **Significant energy usage while nodes are idle**
- **Benefit from gaps to propose energy savings models**

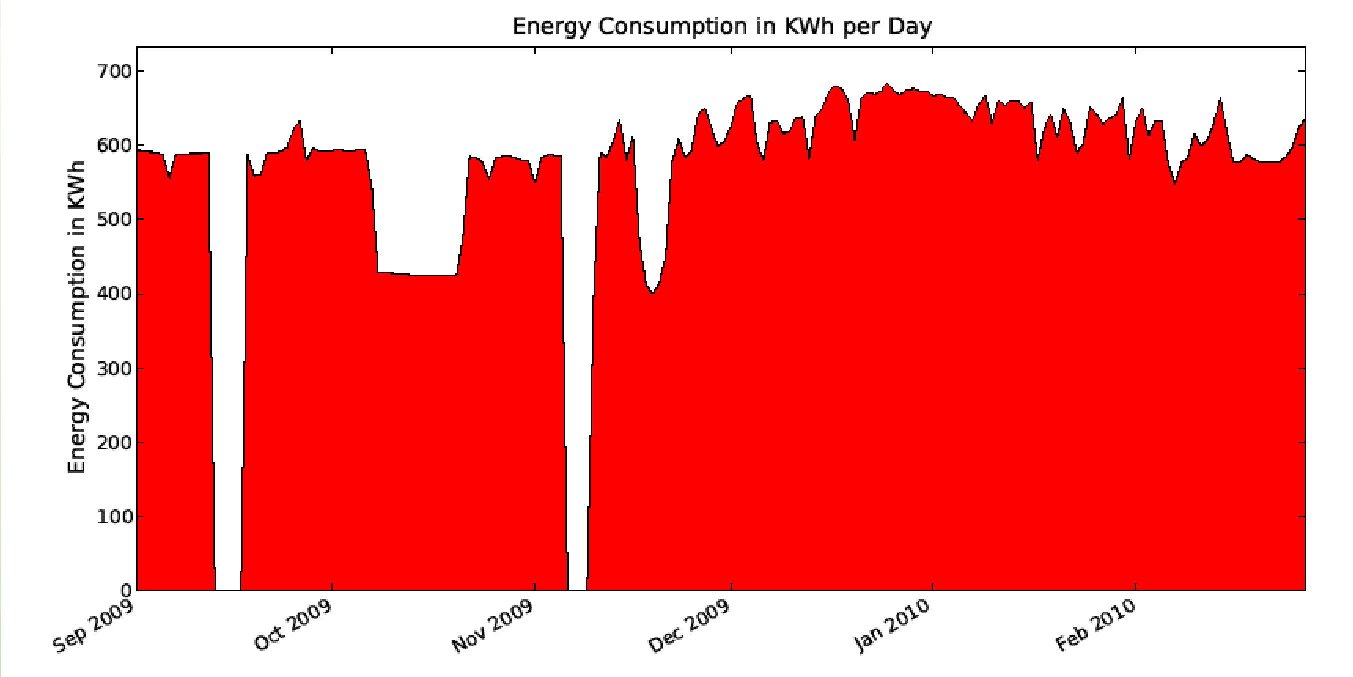
# Focus on nodes heterogeneity

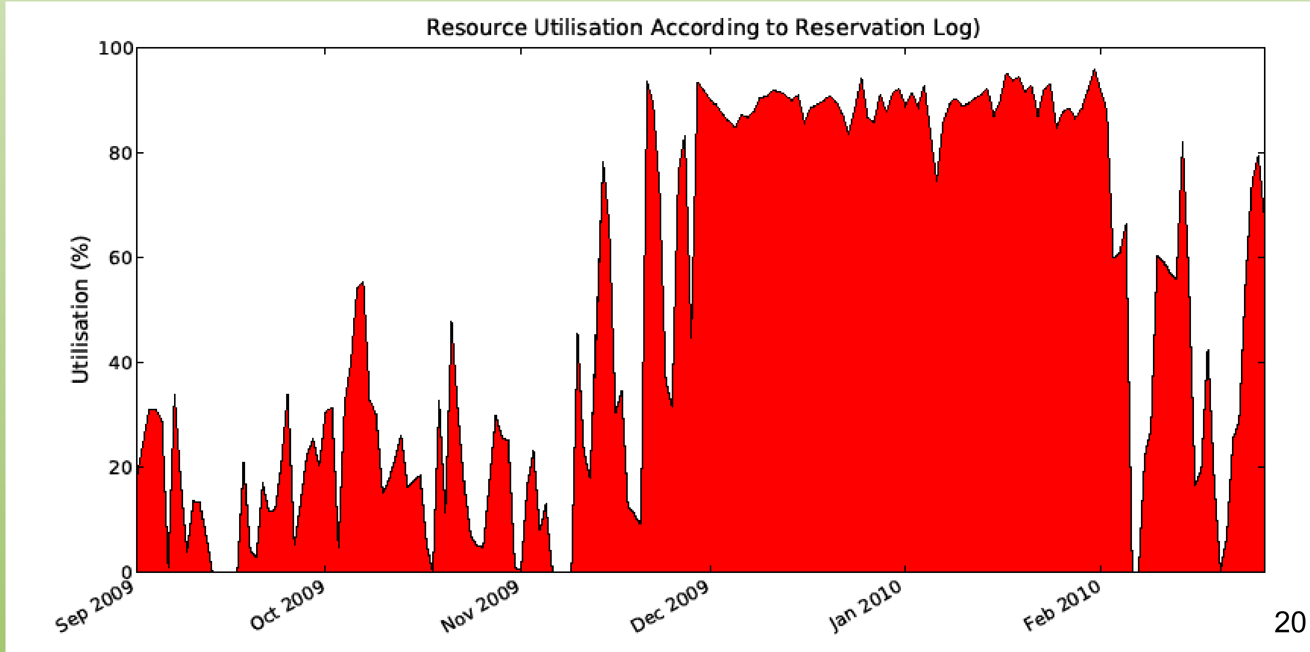


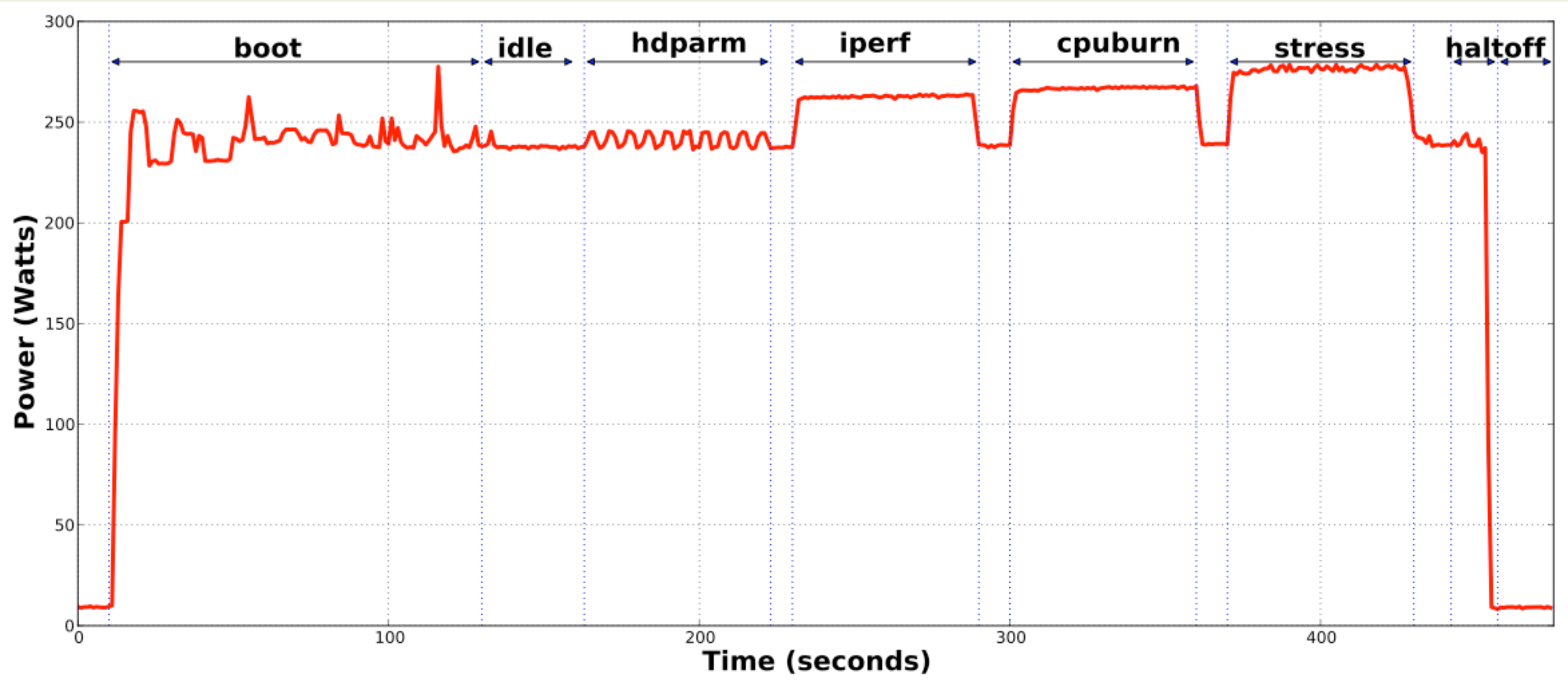
# Electrical consumption vs. usage

- One measurement per **second** for each equipment (150 nodes)
- Electrical consumption and resource usage are not directly correlated



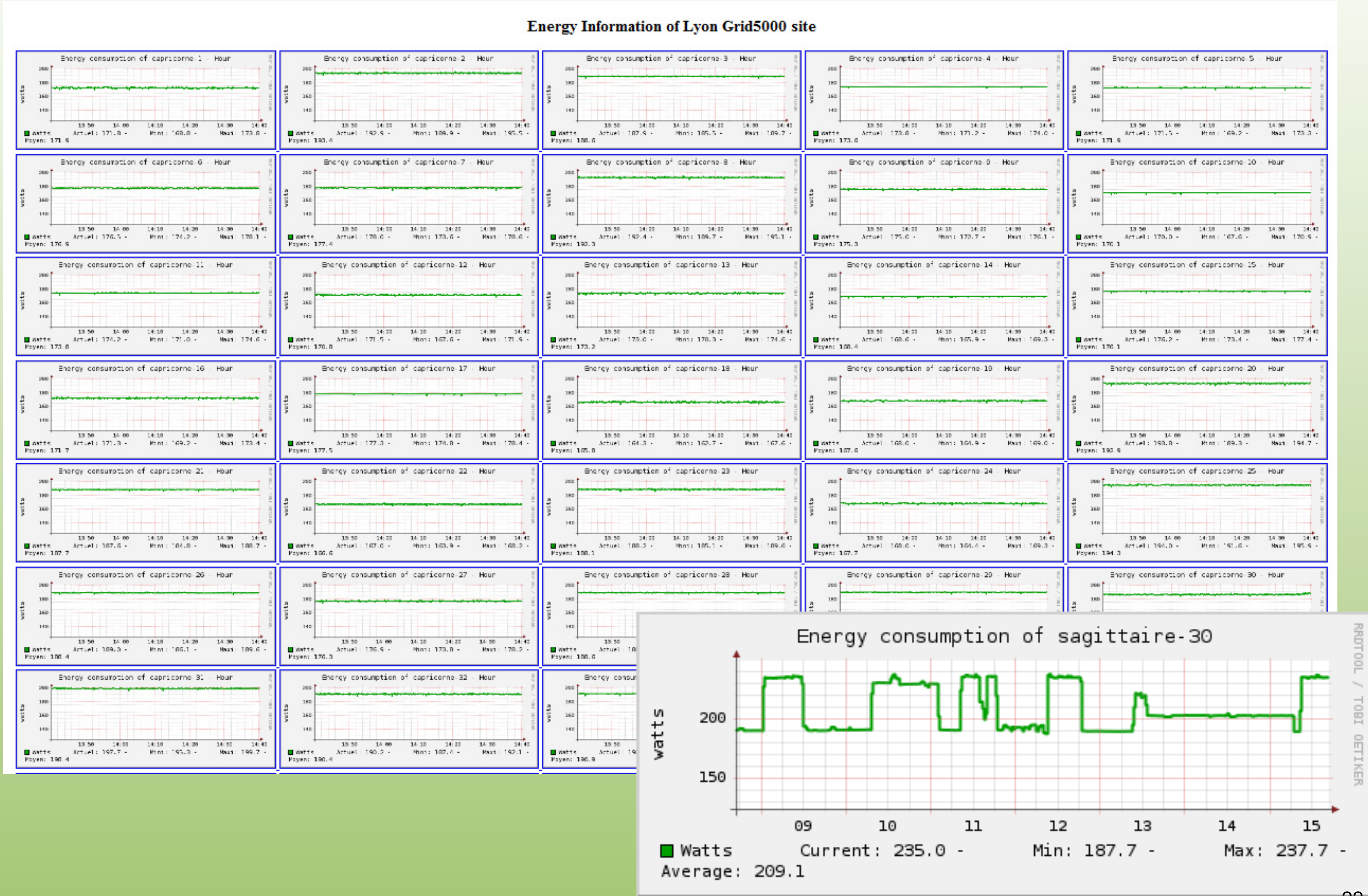



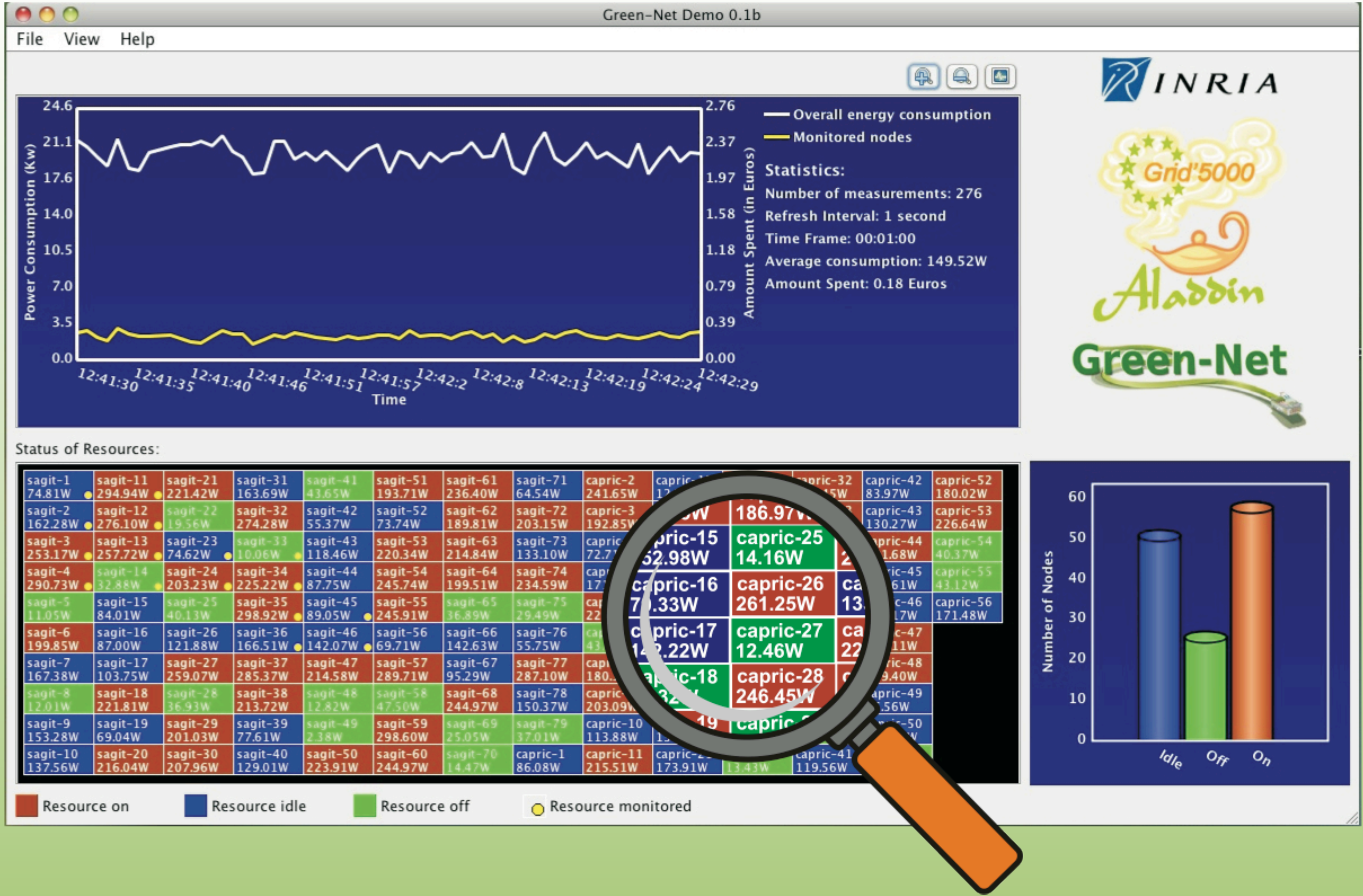
# Profiling the energy consumption of applications




It is important to point out that idle consumption of a node can account up to 90% of the power it consumes when performing a task!

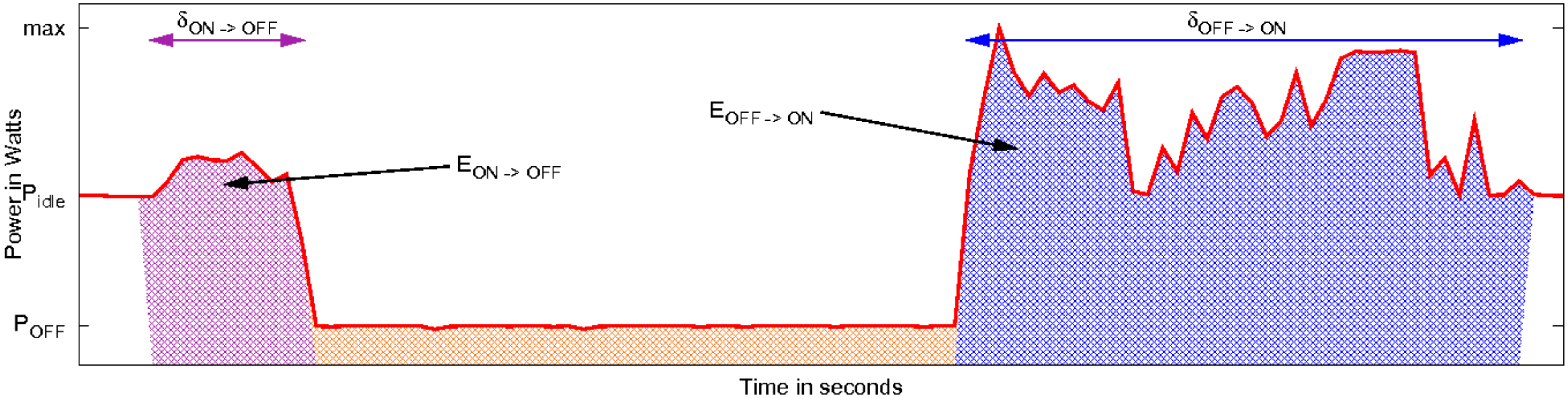
# Large scale energy exposing



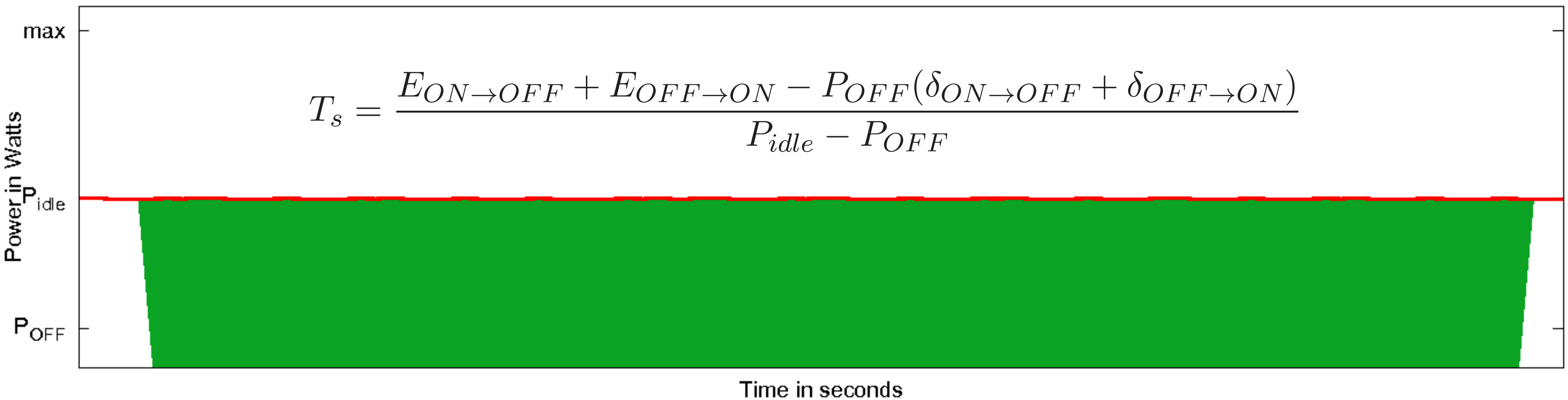

Green-Net Demo 0.1b
File View Help
Power Consumption (Kw)
24.6 21.1 17.6 14.0 10.5 7.0 3.5 0.0
2.76 2.37 1.97 1.58 1.18 0.79 0.39 0.00
Amount Spent (in Euros)
12:41:30 12:41:35 12:41:40 12:41:46 12:41:51 12:41:57 12:42:2 12:42:8 12:42:13 12:42:19 12:42:24 12:42:29
Time
Overall energy consumption
Monitored nodes
Statistics:
Number of measurements: 276
Refresh Interval: 1 second
Time Frame: 00:01:00
Average consumption: 149.52W
Amount Spent: 0.18 Euros
Status of Resources:
sagit-1 74.81W
sagit-2 162.28W
sagit-3 253.17W
sagit-4 290.73W
sagit-5 11.05W
sagit-6 199.85W
sagit-7 167.38W
sagit-8 12.01W
sagit-9 153.28W
sagit-10 137.56W
sagit-11 294.94W
sagit-12 276.10W
sagit-13 257.72W
sagit-14 32.88W
sagit-15 84.01W
sagit-16 87.00W
sagit-17 103.75W
sagit-18 221.81W
sagit-19 69.04W
sagit-20 216.04W
sagit-21 221.42W
sagit-22 19.56W
sagit-23 74.62W
sagit-24 203.23W
sagit-25 40.13W
sagit-26 121.88W
sagit-27 259.07W
sagit-28 36.93W
sagit-29 201.03W
sagit-30 207.96W
sagit-31 163.69W
sagit-32 274.28W
sagit-33 10.06W
sagit-34 225.22W
sagit-35 298.92W
sagit-36 166.51W
sagit-37 285.37W
sagit-38 213.72W
sagit-39 77.61W
sagit-40 129.01W
sagit-41 43.65W
sagit-42 55.37W
sagit-43 118.46W
sagit-44 87.75W
sagit-45 89.05W
sagit-46 142.07W
sagit-47 214.58W
sagit-48 12.82W
sagit-49 2.38W
sagit-50 223.91W
sagit-51 193.71W
sagit-52 73.74W
sagit-53 220.34W
sagit-54 245.74W
sagit-55 245.91W
sagit-56 69.71W
sagit-57 289.71W
sagit-58 47.50W
sagit-59 298.60W
sagit-60 244.97W
sagit-61 236.40W
sagit-62 189.81W
sagit-63 214.84W
sagit-64 199.51W
sagit-65 36.89W
sagit-66 142.63W
sagit-67 95.29W
sagit-68 244.97W
sagit-69 25.05W
sagit-70 14.47W
sagit-71 64.54W
sagit-72 203.15W
sagit-73 133.10W
sagit-74 234.59W
sagit-75 29.49W
sagit-76 55.75W
sagit-77 287.10W
sagit-78 150.37W
sagit-79 37.01W
capric-1 86.08W
capric-2 241.65W
capric-3 192.85W
capric-10 113.88W
capric-11 215.51W
capric-25 14.16W
capric-26 261.25W
capric-27 12.46W
capric-28 246.45W
capric-41 119.56W
capric-42 83.97W
capric-43 130.27W
capric-52 180.02W
capric-53 226.64W
capric-54 40.37W
capric-55 43.12W
capric-56 171.48W
Number of Nodes
60 50 40 30 20 10 0
Idle Off On
Resource on
Resource idle
Resource off
Resource monitored

# The On/Off model

Consumption of the resource if it is switched off and on




Consumption of the resource if it stays idle

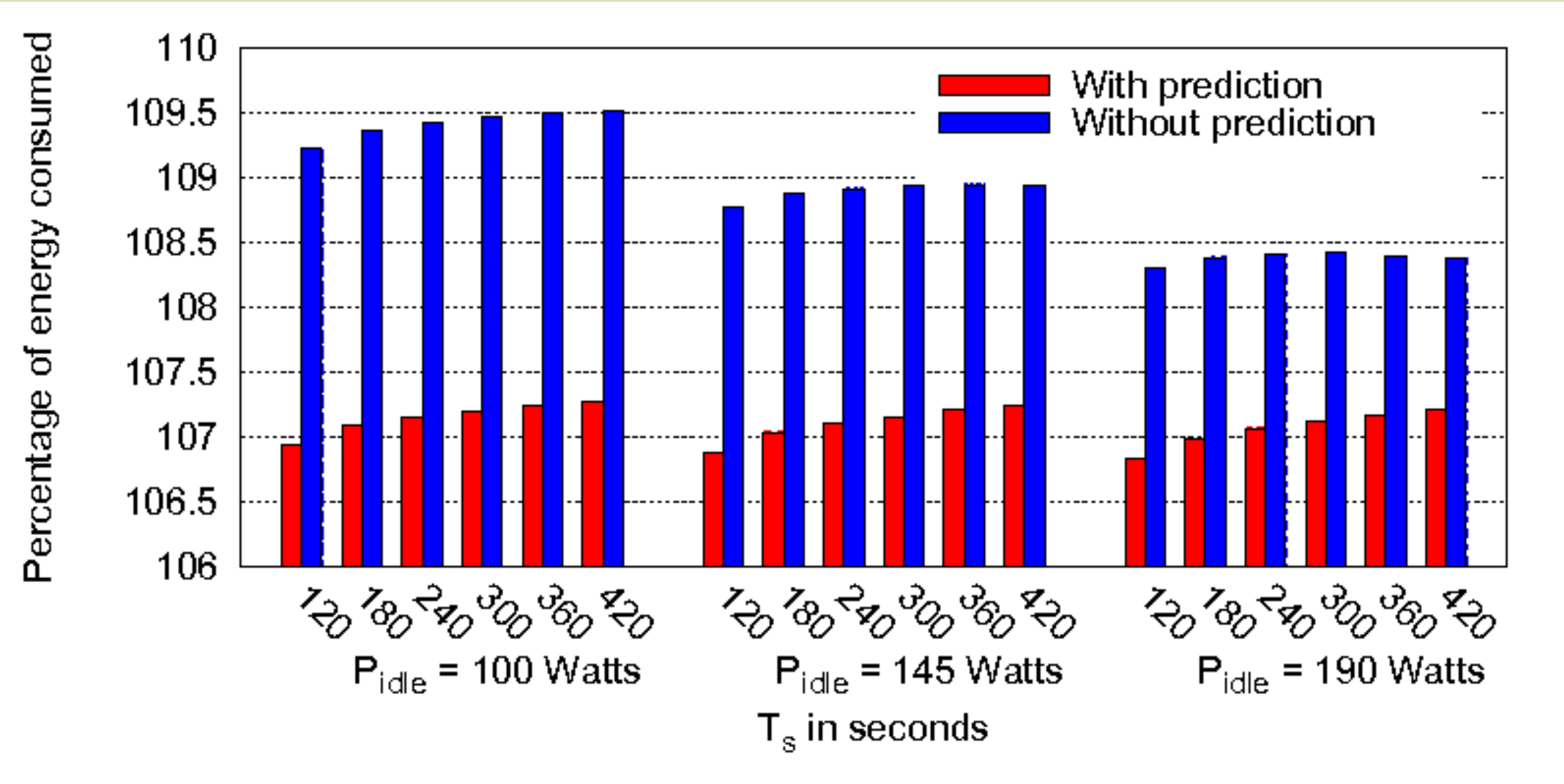



$$T_s = \frac{E_{ON\to OFF} + E_{OFF\to ON} - P_{OFF}(\delta_{ON\to OFF} + \delta_{OFF\to ON})}{P_{idle} - P_{OFF}}$$

On/Off is less energy consuming if the green area is higher than the On/Off area (inactivity > Ts)
And so, **we have to predict the next On** to anticipate imminent reservation not yet submitted

# Prediction evaluation based on replay

- Example: Bordeaux site (650 cores, 45K reservations, 45% usage)
- 100%: theoretical case (future perfectly known)
- Currently (always on): 185% energy

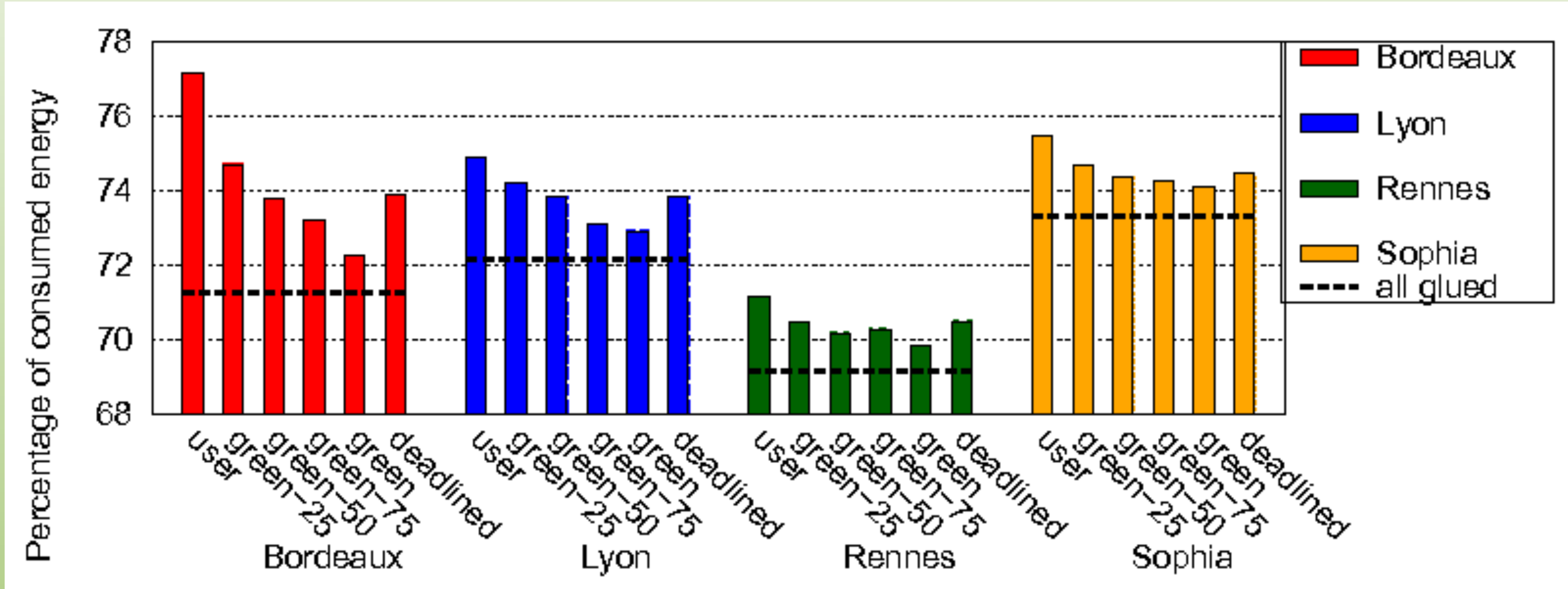


# Green policies

- **User**: requested date
- **25% green**: 25% of jobs follow Green advices – the rest follow user request
- **50% green**: 50% of jobs follow Green advices – the rest follow user request
- **75% green**: 75% of jobs follow Green advices – the rest follow user request
- **Fully green**: solution with uses the minimal amount of energy and follow Green advices
- **Deadlined**: fully green for 24h – then user policy

# Green policies replay results

- 100% = present energy consumption for 2007 year (all nodes are always powered on)
- All glued: unreachable theoretical limit, ideal case where we glue all reservations, putting one after the other and switching the resources for the rest of the time



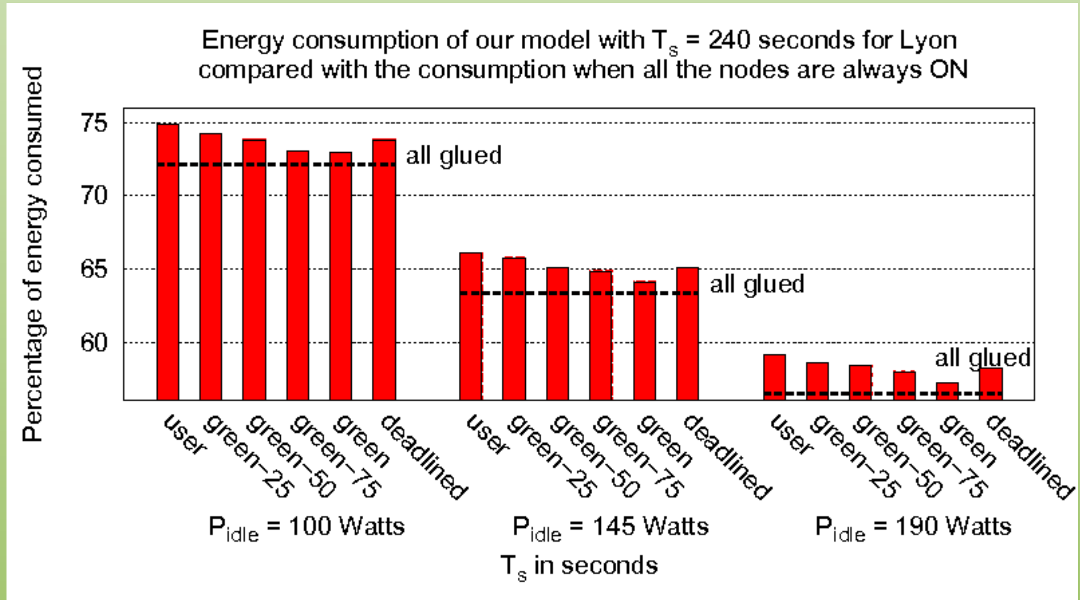
# Green policies evaluation and future energy savings

- Example of Lyon (322 cores, 33K reservations, 46% usage - 07)
- Current situation: always ON nodes (100 %)

- For Lyon site: saving of 73 800 kWh for 2007 period
- 1209159 kWh for the full Grid5000 platform (without air-cooling and network equipments) on a 12 month period

Saving of near 50%

Represents annual consumption of 600 inhabitants' village



# Summary

We have:

Analyzed a grid usage and energy consumption during one year: load and energy not directly correlated, burst usage, gaps, idle nodes are energy costly, we need to switch them off but not for a short time

- Having a configurable sensing infrastructure is mandatory for designing energy efficient software frameworks

Proposed an Energy-Aware Reservation Infrastructure

- Switching resources on/off only when necessary
- Predicting future reservations to not switch off resources that are required in a near future
- Aggregating some resource reservations to avoid frequent on/off

Experimental validation by a one year replay on the Grid5000 traces: with our infrastructure, saving 50% of energy is possible without modifying user performances

# Outline

- Towards energy efficient large scale distributed systems
- **Reducing energy consumption of fault tolerance algorithms (Joint Very Early Results)**

# Energy consumption and Fault Tolerance in HPC systems

- Energy consumption is a main issue at very large scale
- Resiliency is another major problem that we have to care about at Exascale
  - At Exascale: MTBF = few hours
  - Fault tolerance at Exascale is mandatory in order to reach applications termination
  - But resiliency has a price: more energy consumption!
- Need to adapt current fault tolerance protocols on an energy efficient way to scale over future supercomputers
- Need to apply green solutions over idle periods known in fault tolerance protocols

# Energy consumption vs fault tolerance

**<u>Our main goal:</u>**

**What?**
Decrease the total energy consumption due to fault tolerance

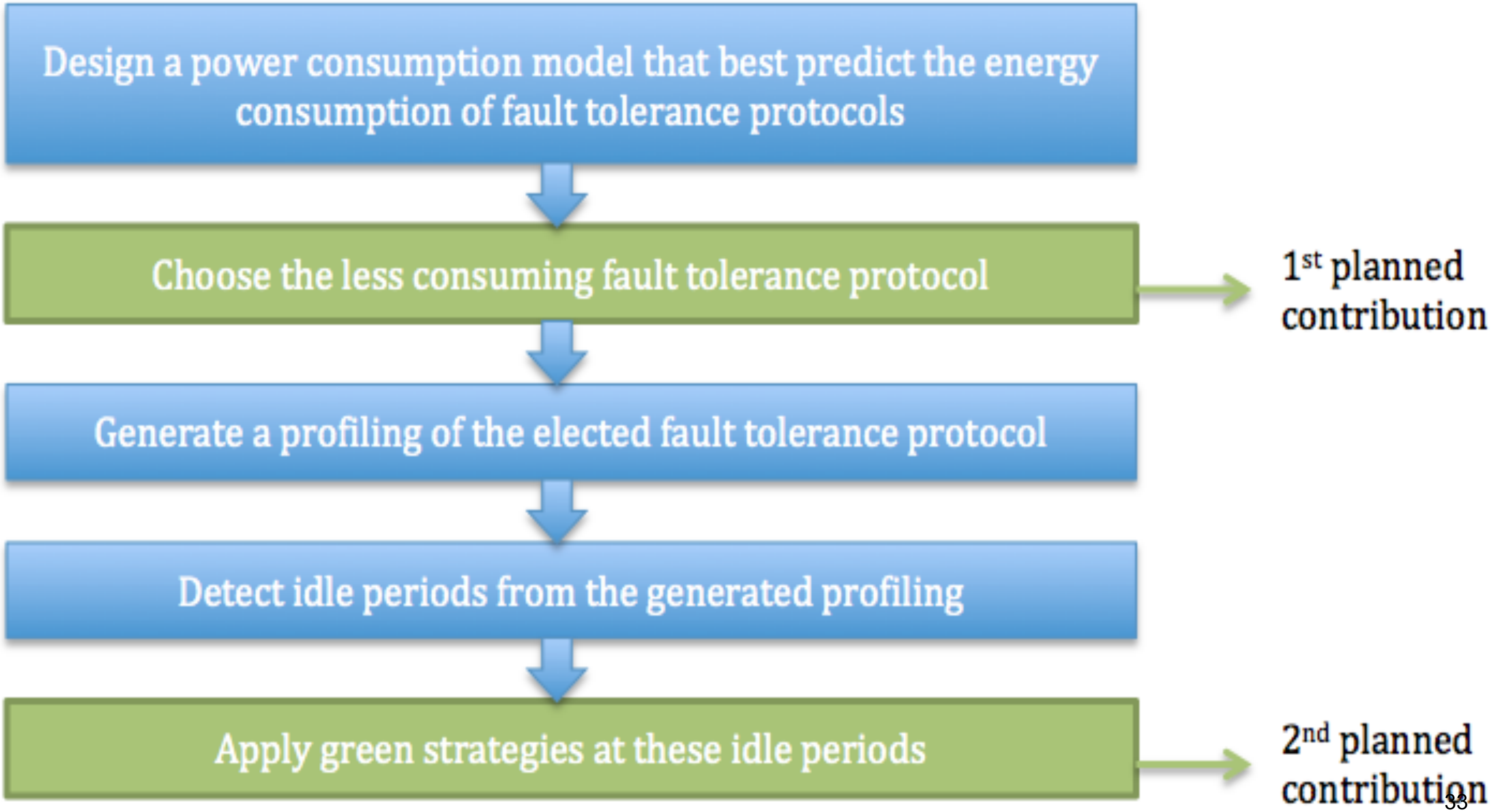
**Where ?**
For HPC applications running over very large scale distributed platforms
→ petascale supercomputers and beyond

**How ?**
1) By choosing the less energy consuming fault tolerance protocol

2) By applying green strategies whenever possible
→ partial or complete shutdown approach
→ partial or complete slowdown approach

# Energy consumption vs Fault tolerance

Our roadmap to reach our main goal:



# Energy consumption vs Fault tolerance

Currently, we are working on our first step:

→ Design a power consumption model for fault tolerance protocols to predict the energy consumption of **a given fault tolerance protocol for a given application on a given platform**

To do so:

1) Get some basic knowledge about how the most known fault tolerance protocols work

2) Explore how much time and energy they consume depending on the application and the distributed platforms features

   → By running some experiments

3) Draw some conclusions from the experimental results, that we should consider in our power consumption model

# Current fault tolerance protocols that we consider

I) Coordinated checkpointing protocol (provided in mpich2)

II) Two versions of Message Logging Protocol

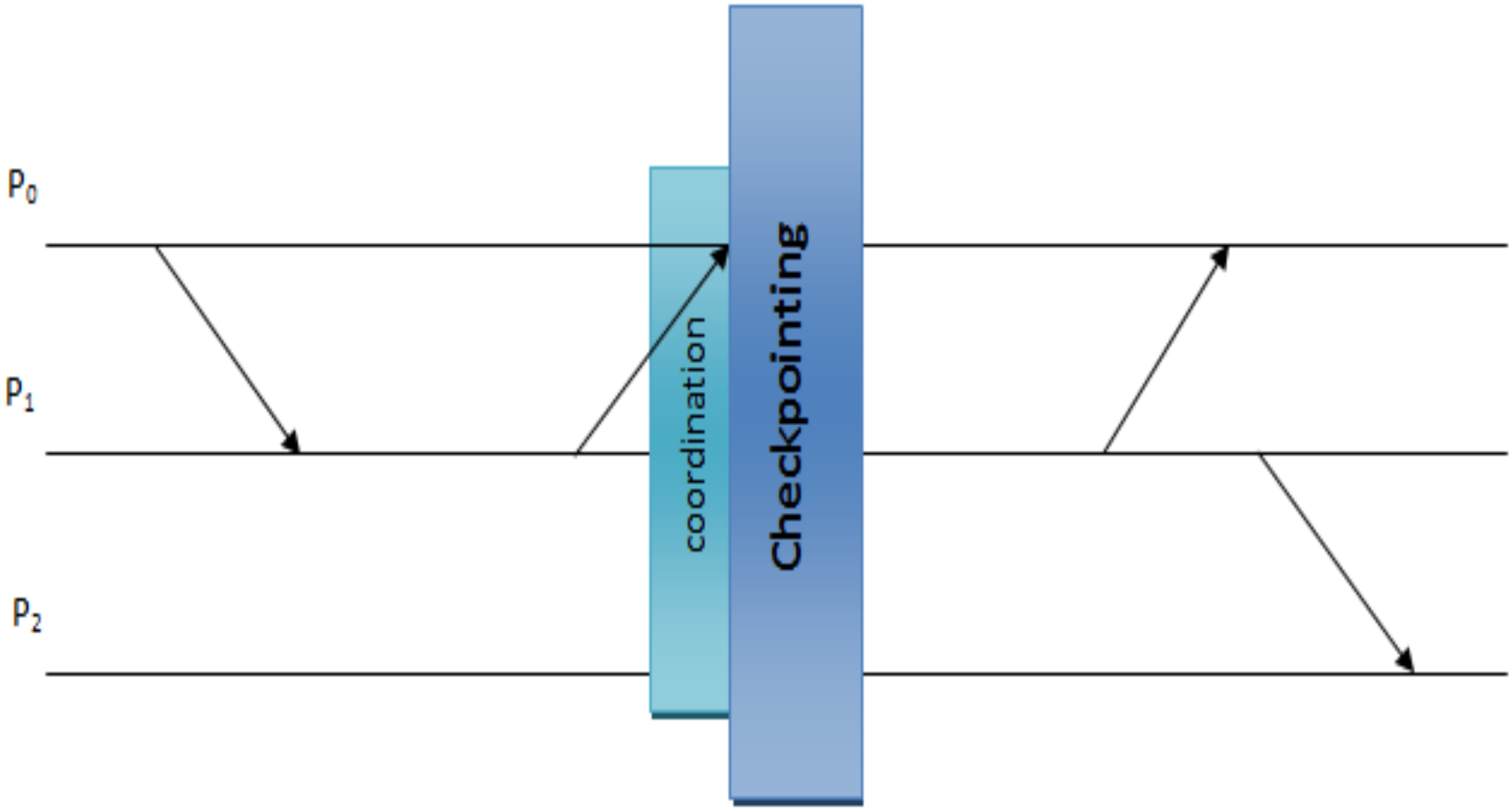
- 1st version: Logging all messages
- 2st version: Logging a subset of messages (Ropars et al. in IPDPS 2010)

III) Hybrid Protocol (Ropars et al. in IPDPS 2011)

- Defining clusters
- Coordinated checkpointing for nodes in the same cluster
- Message logging for nodes in different clusters

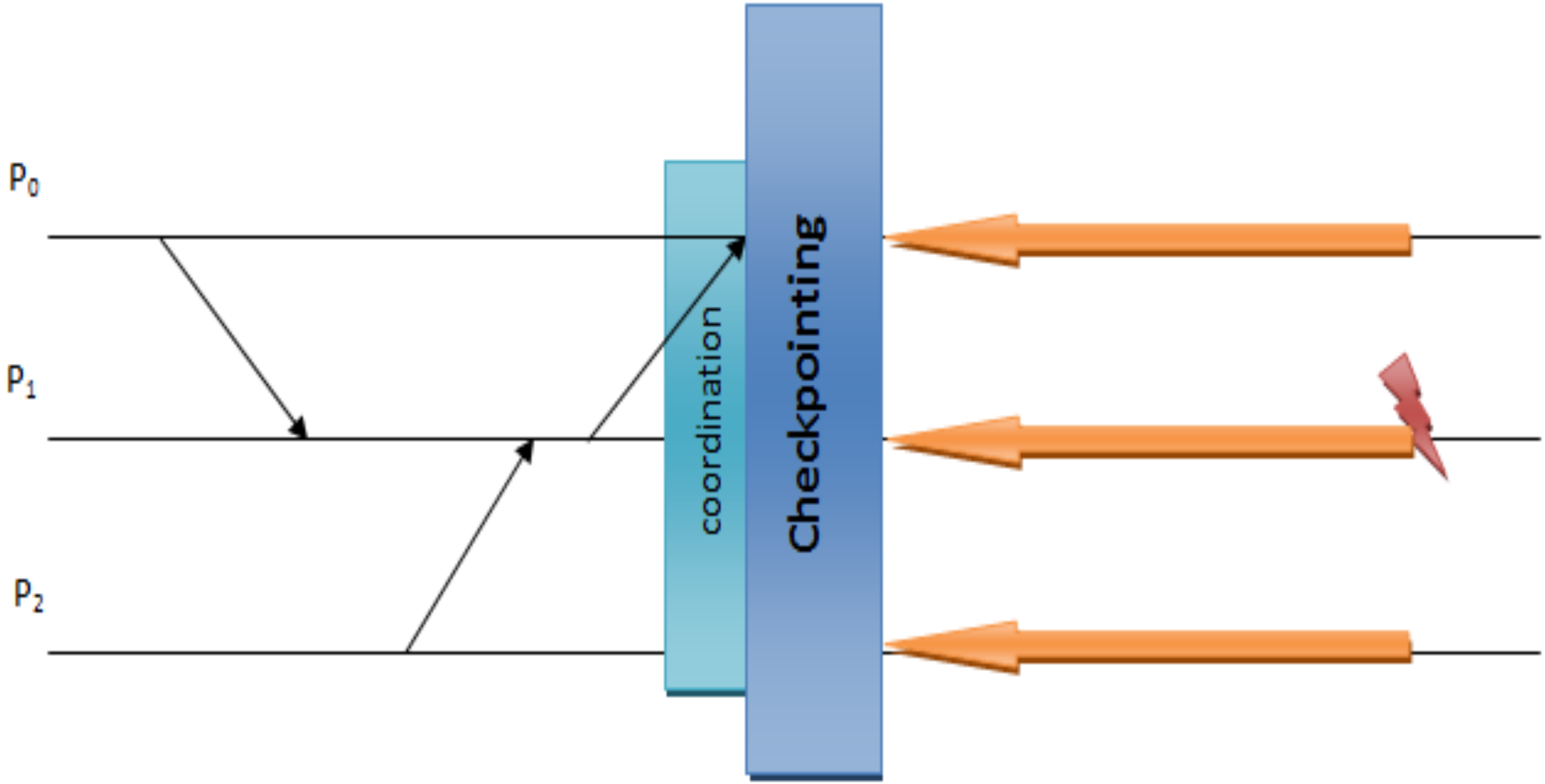
At this time, we only explored fault tolerant protocols on their normal operating stage (before the failure).

# Coordinated Checkpointing Protocol




Coordination in order to avoid orphan messages

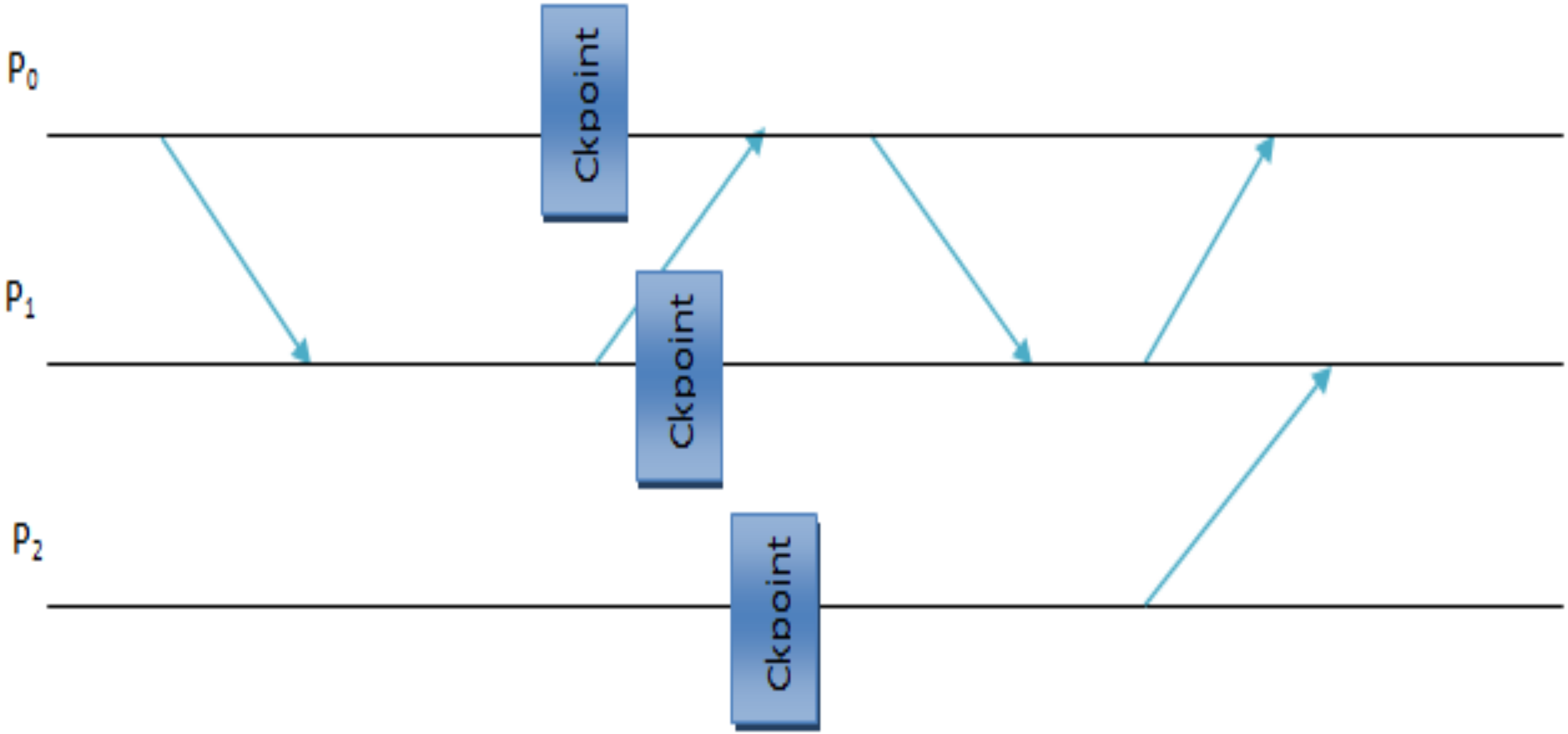
# Coordinated Checkpointing Protocol




When one failure, all the processes rollback to the last checkpointing

➔ It may be very energy and time consuming, especially at exascale

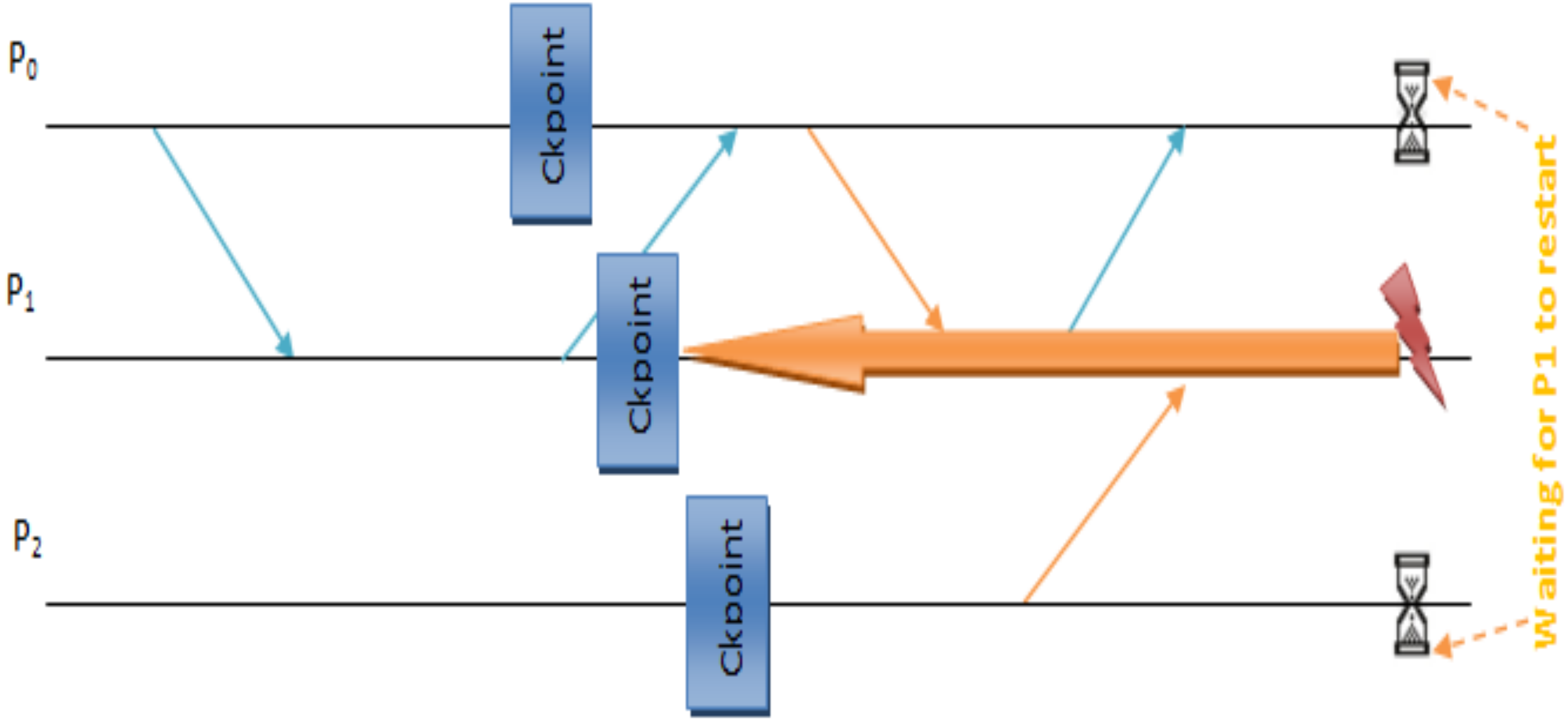
# Message Logging: 1st version




Checkpoints are taken in an uncoordinated way.

No coordination but all sent messages are logged (in blue).

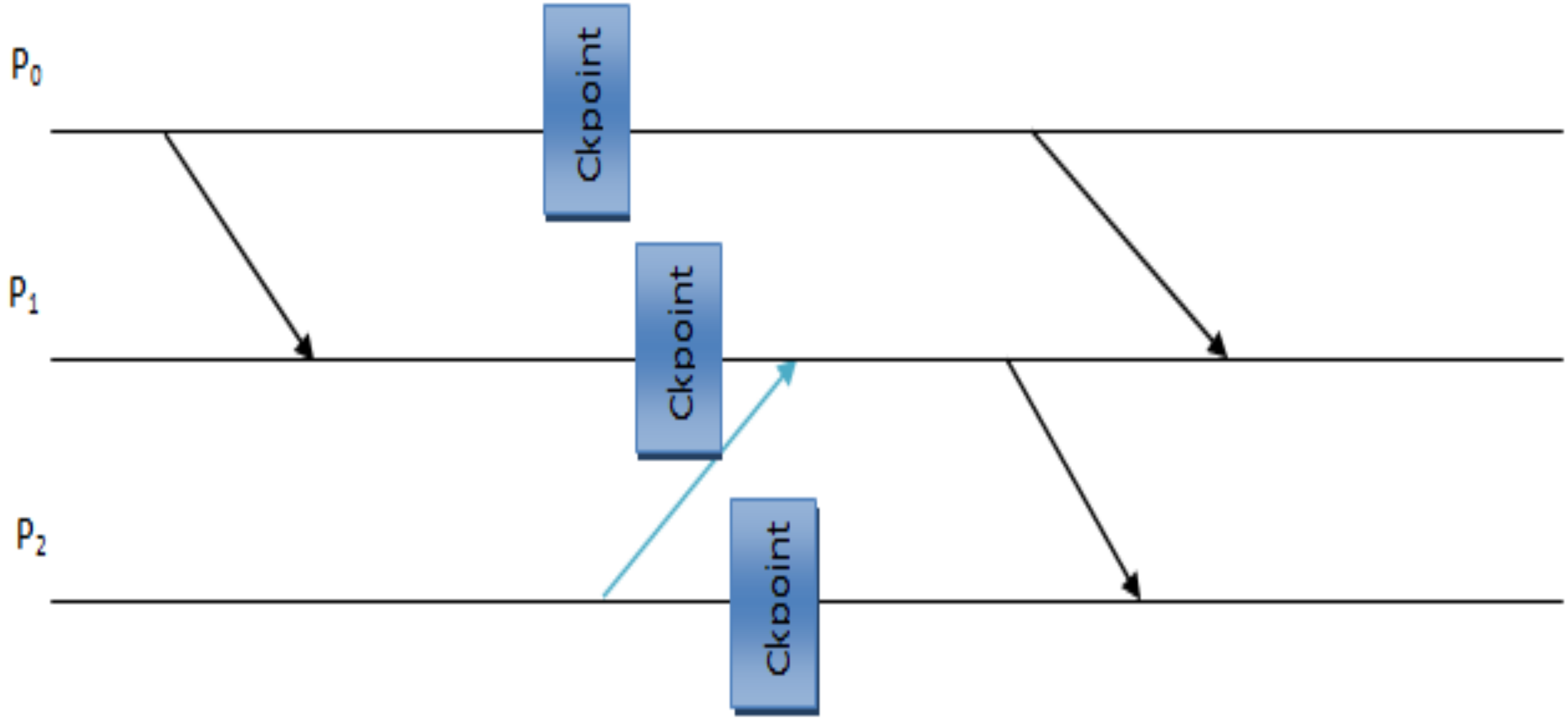
# Message Logging: 1st version




Only the process concerned by the failure rollback to its last checkpoint.

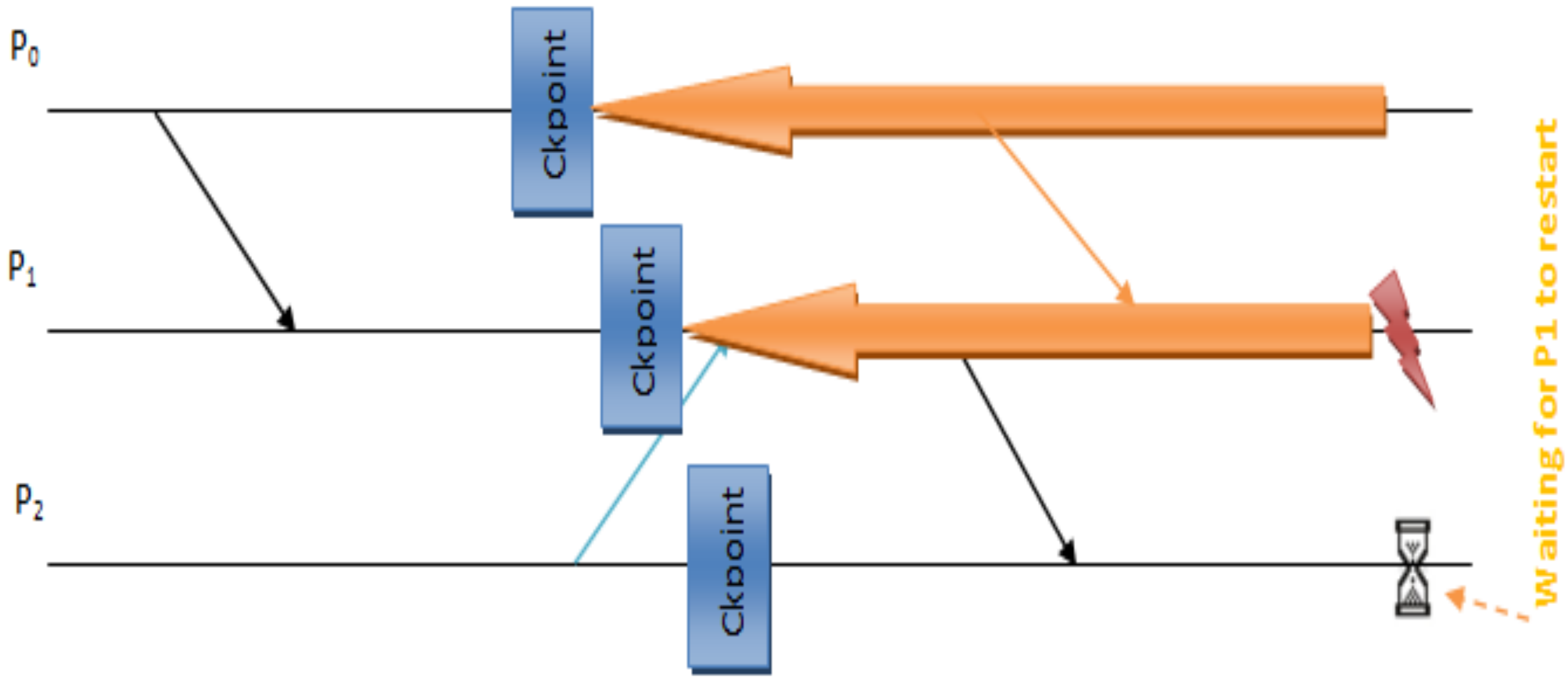
The other processes wait for P1 to restart and reach its state just before the failure ➔ we can imagine to switch them off meanwhile.

# Message Logging: 2nd version




Logged messages are such as the number of checkpoints at their reception is upper than the number of checkpoints at their emission.

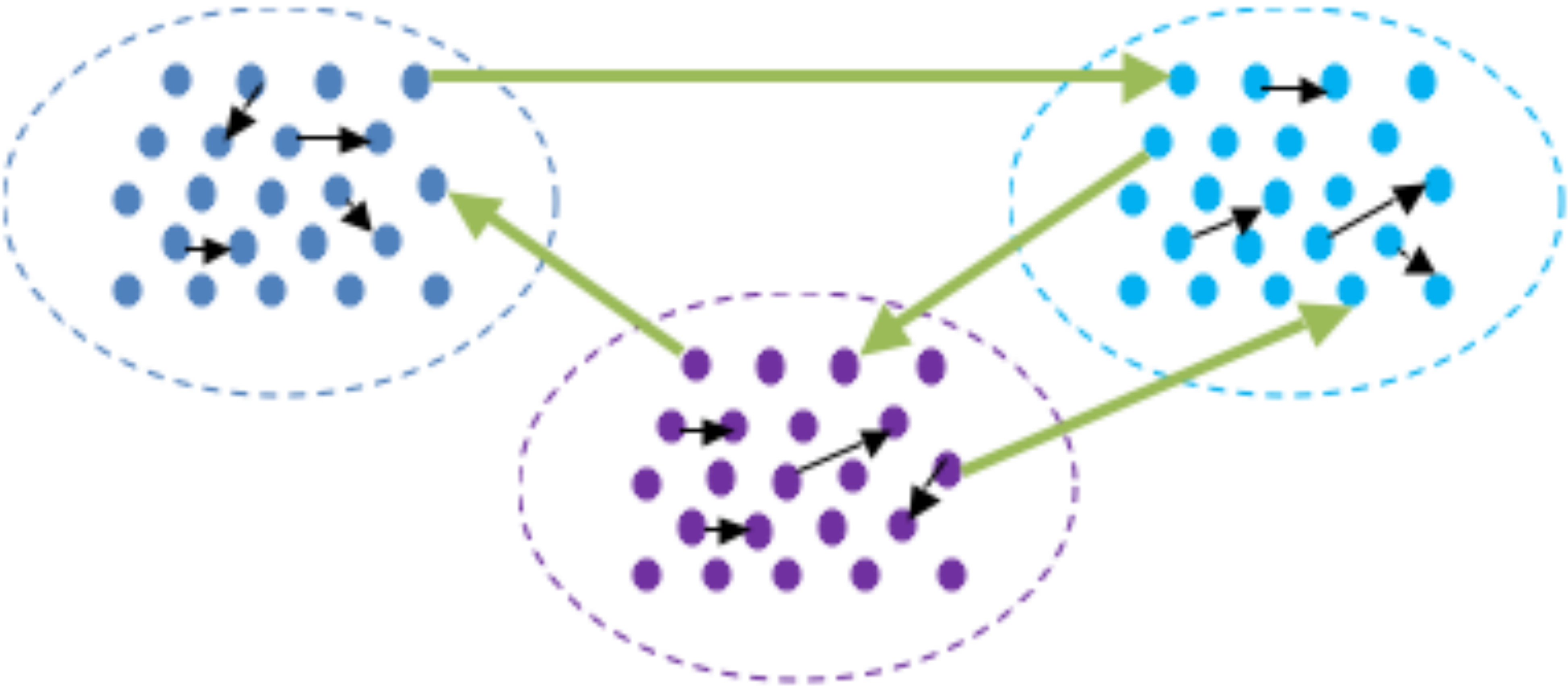
# Message Logging: 2nd version




Processes concerned by the P1 failure that have to rollback are those that have sent messages to the P1, which had not been logged since their last checkpoint.

The others are waiting for these processes to restart.

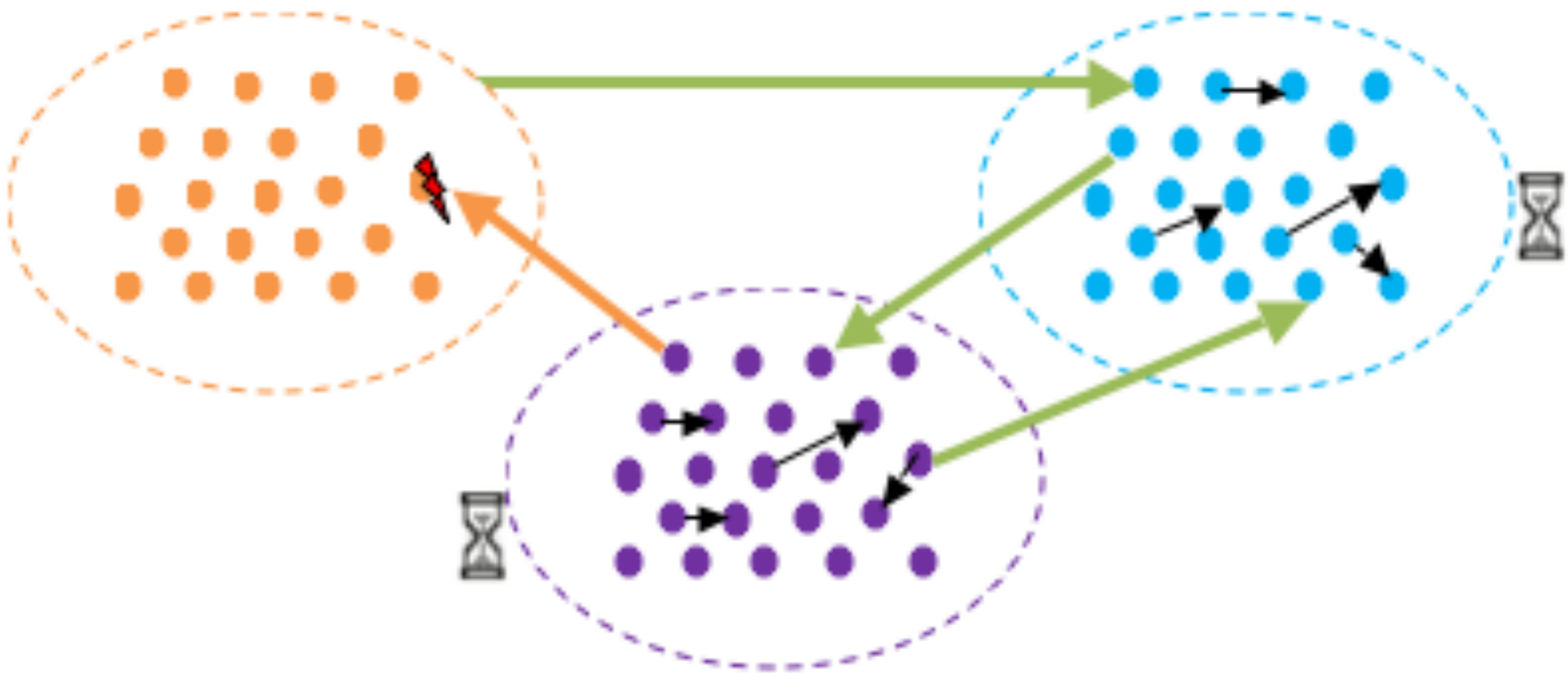
# Hybrid Protocol



In a same cluster: coordinated protocol (same color for nodes within a cluster to emphasize that they take theirs checkpoints at the same time).

All messages sent between nodes in different clusters (green arrows) are logged.

# Hybrid Protocol



All the processes in the same cluster rollback (in orange)
All other processes are waiting: we can imagine that we turn them off in order to save energy!

Messages received by this cluster since its last checkpoint are replayed (orange arrow)

# Description of the experimentations

Measuring execution time of CM1 (Cloud Model 1) over 256 cores on Grenoble cluster.

⇒ find the overhead (in execution time) of different fault tolerant protocols over a real application

⇒ determine the overhead (in execution time) of different phases experienced before the failure

- cost of one coordination
- cost of one checkpointing
- cost of one message logging on RAM or HDD
- …

Measuring execution time and energy consumption of BT Class C over 16 cores on Lyon cluster.

=> find the energy consumed by fault tolerance protocols over a given application

# Description of the experimentations

We have run each application:

- Without fault tolerant protocol

- With coordinated protocol without taking checkpoint
  - 1, 10 coordinations

- With coordinated protocol by taking 1 checkpoint

- With message logging protocol (1st version) without checkpoint
  - Messages stored on RAM only, then on HDD only

- With message logging protocol (2st version) without checkpoint
  - Messages stored on RAM only, then on HDD

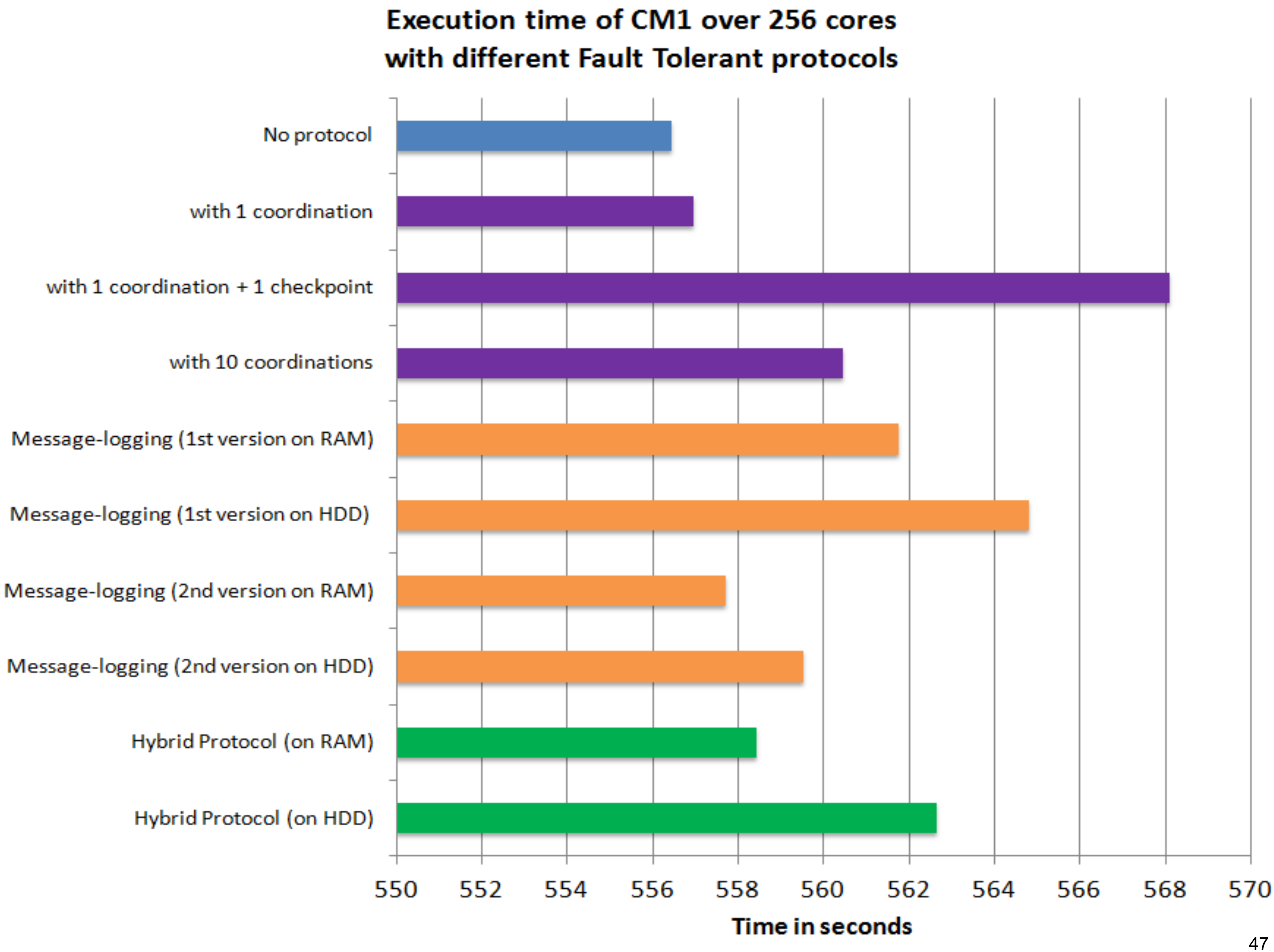
- With hybrid protocol without coordinated protocol inside clusters
  - Messages stored on RAM only, then on HDD
  - 8 clusters of 32 processes for CM1 (but not optimized clustering)

# Experimental setup

- For all the experiments:
  - Repeated 10 times
  - Obtained each measure:
    - By excluding the max and the min values,
    - And averaging the 8 remaining median values.

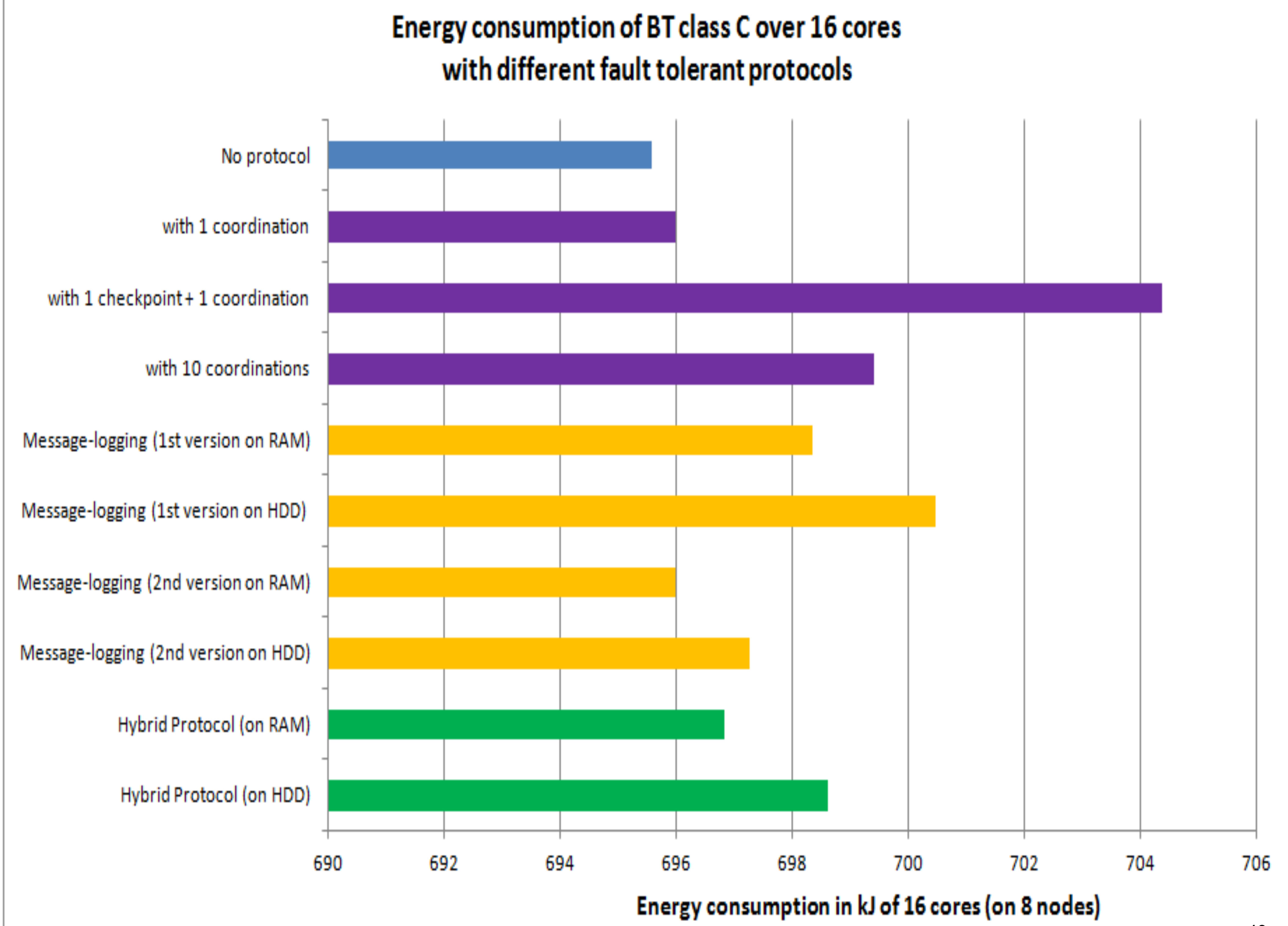
- On Lyon cluster, we have wattmeters that measure the power consumption of each node every second (1 measure each second) but we currently have less than 64 cores available (so, we have not yet energy consumption results with CM1)

➔ we calculate the energy consumption by summing power consumption over each experiment.

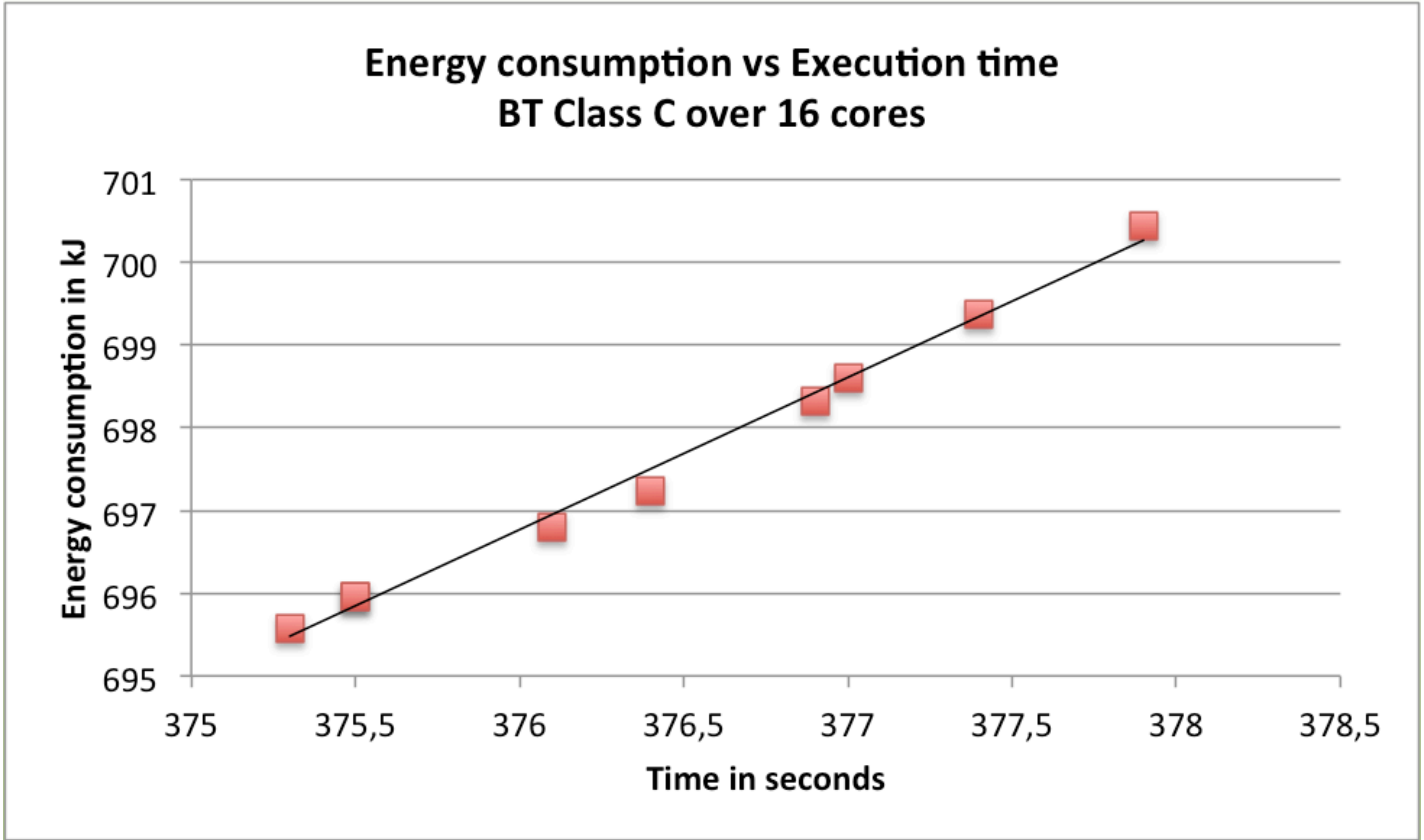


Execution time of CM1 over 256 cores
with different Fault Tolerant protocols
No protocol
with 1 coordination
with 1 coordination + 1 checkpoint
with 10 coordinations
Message-logging (1st version on RAM)
Message-logging (1st version on HDD)
Message-logging (2nd version on RAM)
Message-logging (2nd version on HDD)
Hybrid Protocol (on RAM)
Hybrid Protocol (on HDD)
550
552
554
556
558
560
562
564
566
568
570
Time in seconds

| | | | | |
|---|---|---|---|---|
| No protocol | 556 s | | | |
| With 1 coordination | 557 s | 0.52 s | **0.09 %** | execution time of 1 coordination over 256 cores |
| With 1 checkpoint + 1 coordination | 568 s | 11.14 s | **2.00 %** | execution time of 1 checkpoint |
| With 10 coordinations | 560 s | 0.40 s | **0.07 %** | execution time of 1 coordination (average of 10 coordinations) |
| Message-logging (1st version on RAM) | 562 s | 5.29 s | **0.95 %** | time to log all messages on RAM |
| Message-logging (1st version on HDD) | 565 s | 8.34 s | **1.49 %** | time to log all messages on HDD |
| Message-logging (2nd version on RAM) | 558 s | 1.26 s | **0.23 %** | time to log on RAM specified messages |
| Message-logging (2nd version on HDD) | 559 s | 3.07 s | **0.55 %** | time to log on HDD specified messages |
| Hybrid Protocol (on RAM) | 558 s | 1.99 s | **0.36 %** | time to log on RAM messages between clusters |
| Hybrid Protocol (on HDD) | 563 s | 6.18 s | **1.11 %** | time to log on HDD messages between clusters |


Energy consumption of BT class C over 16 cores
with different fault tolerant protocols
No protocol
with 1 coordination
with 1 checkpoint + 1 coordination
with 10 coordinations
Message-logging (1st version on RAM)
Message-logging (1st version on HDD)
Message-logging (2nd version on RAM)
Message-logging (2nd version on HDD)
Hybrid Protocol (on RAM)
Hybrid Protocol (on HDD)
690
692
694
696
698
700
702
704
706
Energy consumption in kJ of 16 cores (on 8 nodes)

| | | | | |
|---|---|---|---|---|
| No protocol | 695 kJ | | | |
| With 1 coordination | 696 kJ | **0.379 kJ** | **0.05** % | energy consumption of 1 coordination over 16 cores |
| With 1 checkpoint + 1 coordination | 704 kJ | **8.414 kJ** | **1.21 %** | energy consumption of 1 checkpoint |
| With 10 coordinations | 699 kJ | **0.343 kJ** | **0.05 %** | energy consumption of 1 coordination (average value) |
| Message logging (1st version on RAM) | 698 kJ | **2.746 kJ** | **0.40 %** | energy consumption for logging on RAM all messages |
| Message logging (1st version on HDD) | 700 kJ | **4.874 kJ** | **0.70 %** | energy consumption for logging on HDD all messages |
| Message logging (2nd version on RAM) | 696 kJ | **0.379 kJ** | **0.05 %** | energy consumption for logging on RAM specified messages |
| Message logging (2nd version on HDD) | 697 kJ | **1.667 kJ** | **0.24 %** | energy consumption for logging on HDD specified messages |
| Hybrid Protocol (on RAM) | 697 kJ | **1.232 kJ** | **0.18 %** | energy consumption for logging on RAM messages between clusters |
| Hybrid Protocol (on HDD) | 698 kJ | **3.028 kJ** | **0.44 %** | energy consumption for logging on HDD messages between clusters |




High correlation between energy consumption and execution time: Time is Energy !

# Future works

- Do experiments with other real applications at larger scale
  - SPECFEM3d and SPECFEM3D Globe
  - MILC
  - ENZO
  - …
- Design a power consumption model which predicts:
  - Energy consumption in order to choose the most energy efficient protocol
  - Idle periods in order to apply green strategies for reducing the energy consumption
    - By switching off nodes if idle periods are long enough
    - By slowing down nodes

# Akcnoledgments to F. Cappello and T. Ropars

# Questions?

Olivier Gluck
Olivier.Gluck@inria.fr