



Linux: Understanding Process-Level Power Consumption

Aurélien Bourdon, Adel Nouredine, Romain Rouvoy, Lionel Seinturier

► **To cite this version:**

Aurélien Bourdon, Adel Nouredine, Romain Rouvoy, Lionel Seinturier. Linux: Understanding Process-Level Power Consumption. Green Computing Middleware (GCM'2011), Dec 2011, Lisbonne, Portugal. 2011. <hal-00641706>

HAL Id: hal-00641706

<https://hal.inria.fr/hal-00641706>

Submitted on 13 Dec 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Linux: Understanding Process-Level Power Consumption

Aurélien Bourdon

Adel Noureddine

Romain Rouvoy

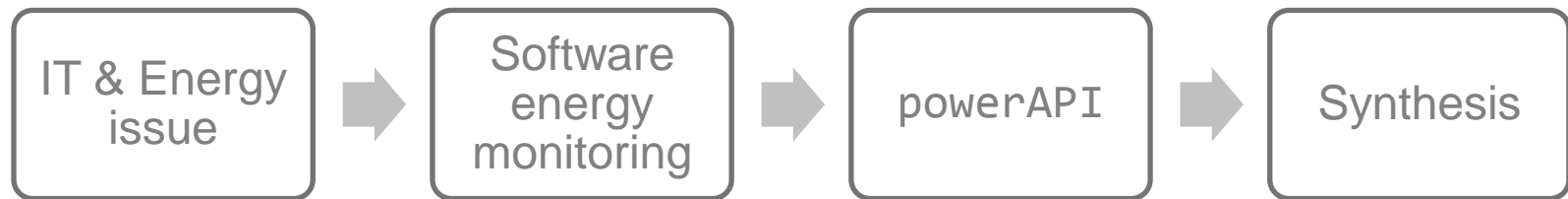
Lionel Seinturier

firstname.lastname@inria.fr

Green Computing Middleware'11

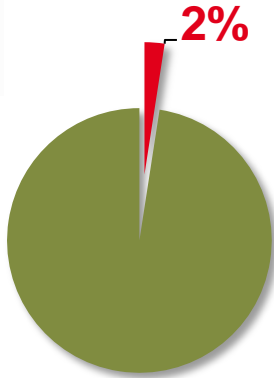
December 12th 2011

Outline



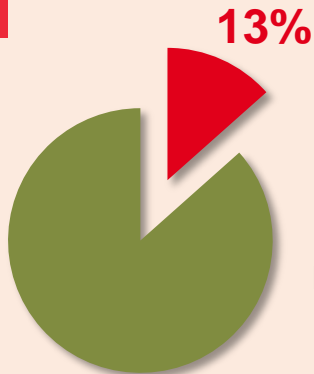
Information Technologies & Energy issue

IT & Energy



World wide carbon emission

Gartner group report



French electricity consumption

French ministries. IT & Sustainable Development report

8_x



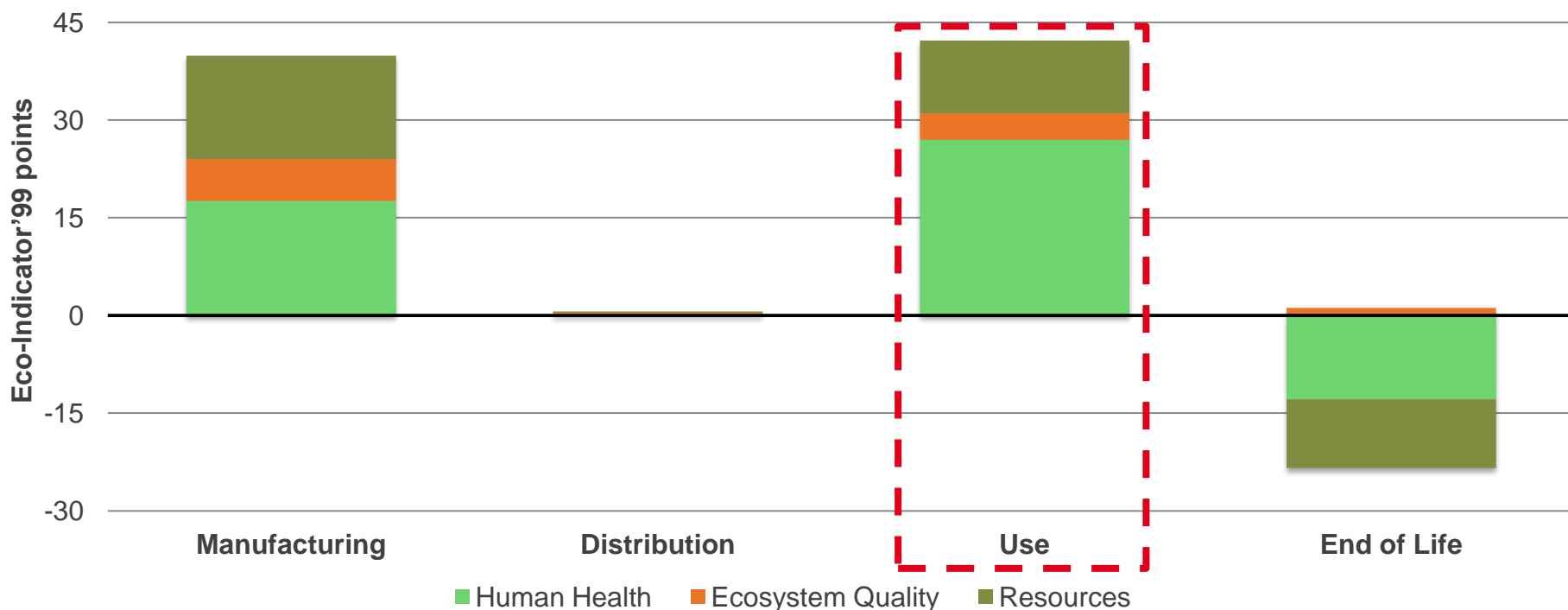
8_x



10% increase by year, over the past 10 years

IT: Where is the energy consumed?

Life Cycle Assessment (LCA) for a common PC



Duan & al. Life cycle assessment study of a Chinese desktop personal computer

IT & Energy: Towards a software concern

Hardware optimizations... but not software

Energy intelligence software is... out of software

Software is not energy aware by design

Software has to be energy aware

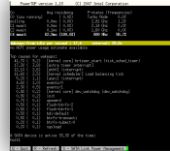
Tools which could help developers to visualize energy consumption

Discover new green development methodologies

Software energy monitoring

State-of-the-art

PowerTop



JouleMeter

pTop



EnergyChecker

There are not reusable libraries

Not easy to support platform interoperability

Development is over

Manual calibration

Requires external device

Our solution: powerAPI

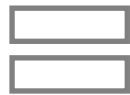
Architecture overview

Energy module

=

Process(es) energy profiler through a specific hardware component

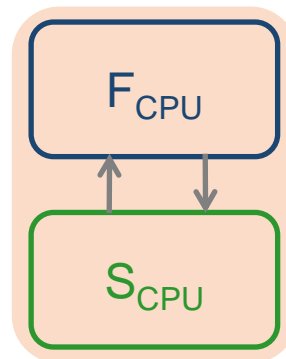
Energy module



Formula

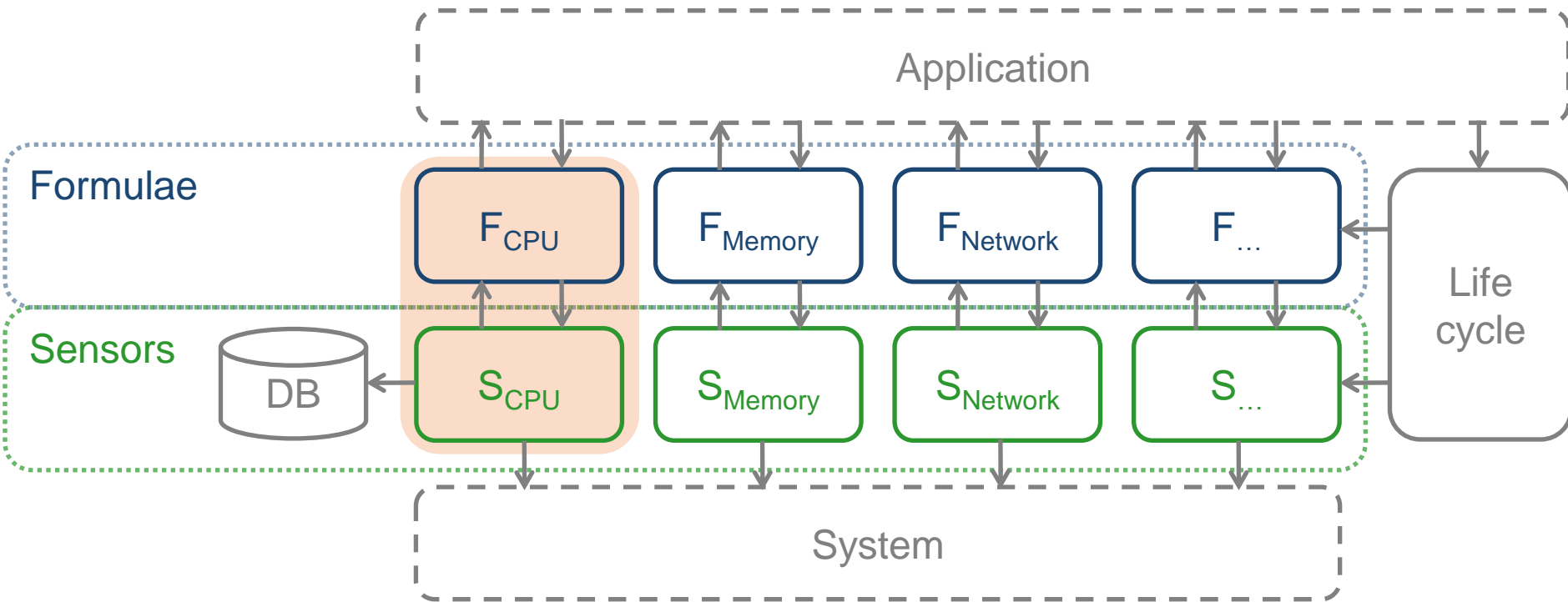


Sensor



As many sensors as there are specific environments

Architecture overview



Architecture benefits

Modular approach, easy to extend

Interoperability

Adaptation to the execution platform, auto-calibration

Adaptation to the application needs

How does it work?

CPU case study, Linux system

How does it work? CPU case study



$$P_{CPU}^{PID}(d) = P_{CPU}(d) \times U_{CPU}^{PID}(d)$$

Process CPU
power

Global CPU
power

Process CPU
usage

Global CPU Power

Process CPU Usage

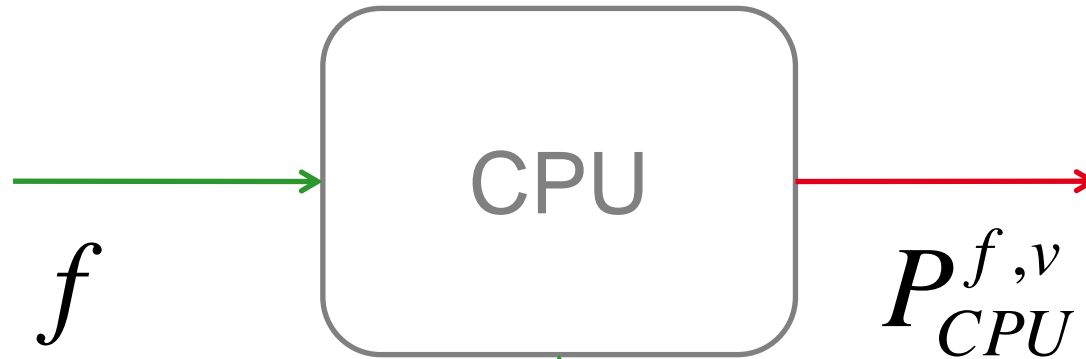
Static part

Dynamic part

Process CPU time

Global CPU time

How does it work? CPU case study



$$\underbrace{C}_{\text{Static part}} \times \underbrace{\quad \times \quad}_{\text{Dynamic part}} \times v^2 = \quad [1]$$

[1] Ge & al. Improvement of Power-Performance Efficiency for High-End Computing

Global CPU Power 

Process CPU Usage

Static part

Dynamic part

Process CPU time

Global CPU time

How does it work? CPU case study

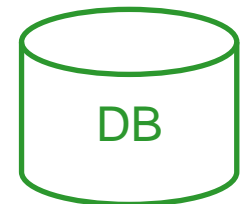
Static part is made up of several constants hard to find out...

That's why we used constructor's Thermal Dissipation Power value

$$P_{CPU}^{f_{TDP}, v_{TDP}} = TDP \times 0.7$$

Rivoire & al. JouleSort: A Balanced Energy-Efficiency Benchmark

$$C = \frac{P_{CPU}^{f_{TDP}, v_{TDP}}}{f_{TDP} \times v_{TDP}^2}$$



Global CPU Power



Static part



Dynamic part

Process CPU Usage

Process CPU time

Global CPU time

How does it work? CPU case study

We need to have

All frequencies used by CPU during the d period

Correspondance table between a frequency and its voltage

Global CPU Power



Static part ✓

Dynamic part



Process CPU Usage

Process CPU time

Global CPU time

How does it work? CPU case study

Frequencies used by CPU during d period

cpu-freq-utils tool

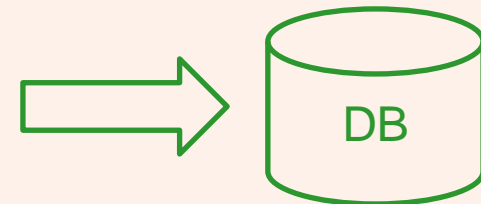
```
/sys/devices/system/cpu/  
|_ cpu[0-n]/  
|_ cpufreq/time_in_state
```

| | |
|-----------|----------|
| 2,8 GHz | 70722 ms |
| 1,599 GHz | 4477 ms |
| ... | ... |

Correspondance table between frequency and voltage

Constructor's data

| | |
|-----------|-------|
| 2,8 GHz | 1.5 V |
| 1,599 GHz | 0.8 V |
| ... | ... |



Global CPU Power



Static part ✓

Dynamic part



Process CPU Usage

Process CPU time

Global CPU time

How does it work? CPU case study



$$P_{CPU}(d)$$

Global CPU
power

×

$$U_{CPU}^{PID}(d)$$

Process CPU
usage

Global CPU Power ✓

Static part ✓

Dynamic part ✓

Process CPU Usage

Process CPU time

Global CPU time

How does it work? CPU case study



$$U_{CPU}^{PID}(d) = t_{CPU}^{PID}(d) / t_{CPU}(d)$$

Process CPU
usage

Process CPU
time

Global CPU
time

Global CPU Power ✓

Process CPU Usage ⚠

Static part ✓

Dynamic part ✓

Process CPU time

Global CPU time

How does it work? CPU case study

Process CPU time

Global CPU time

procfs virtual file system

/proc/[pid]/stat file

/proc/stat file

Sum of all kind of times (usertime, systemtime, blockingiotime...)

Global CPU Power ✓

Static part ✓

Dynamic part ✓

Process CPU Usage ⚠

Process CPU time ⚠

Global CPU time ⚠

How does it work? CPU case study



$$P_{CPU}^{PID}(d) = P_{CPU}(d) \times U_{CPU}^{PID}(d)$$

Process CPU
power

Global CPU
power

Process CPU
usage

Global CPU Power ✓

Process CPU Usage ✓

Static part ✓

Dynamic part ✓

Process CPU time ✓

Global CPU time ✓

Does it work?

Validation

Validation



Dell Precision T3400



Ubuntu 11.10 generic
(kernel version 3.0.4)



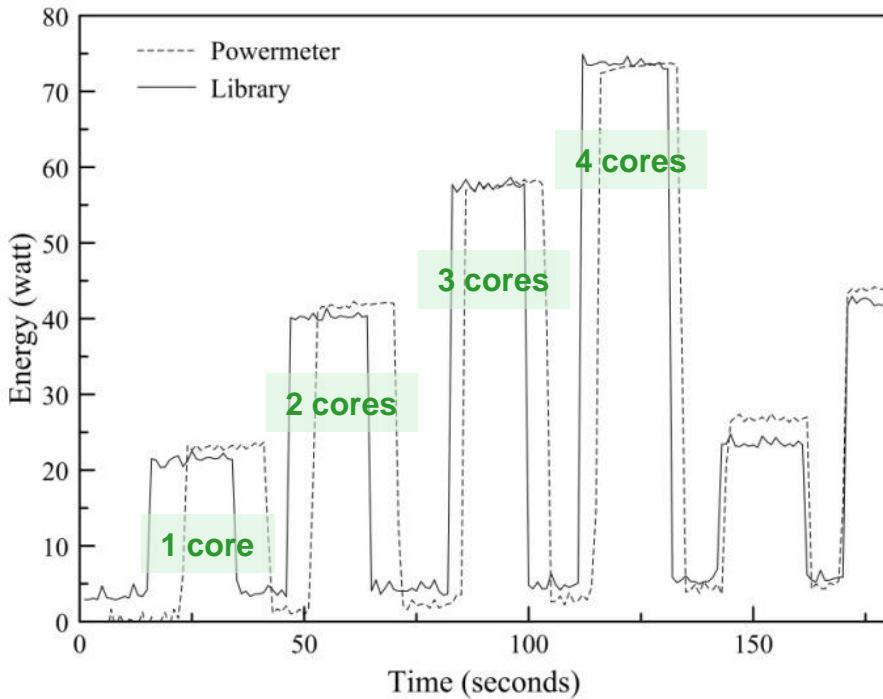
Intel Core 2 Quad Q6600
(2.4 GHz)



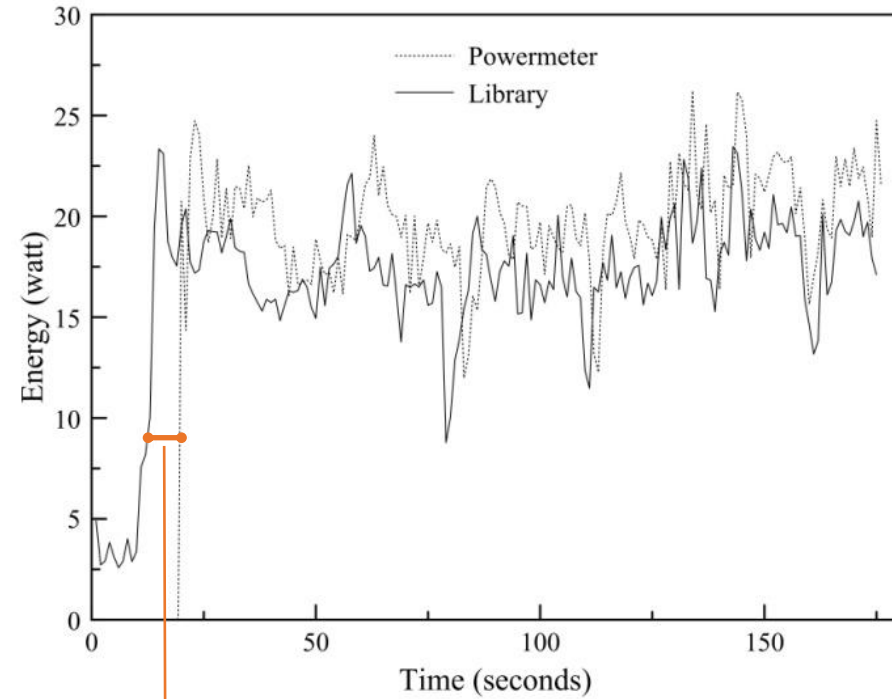
PowerSpy powermeter

Validation

Stress CPU stressor



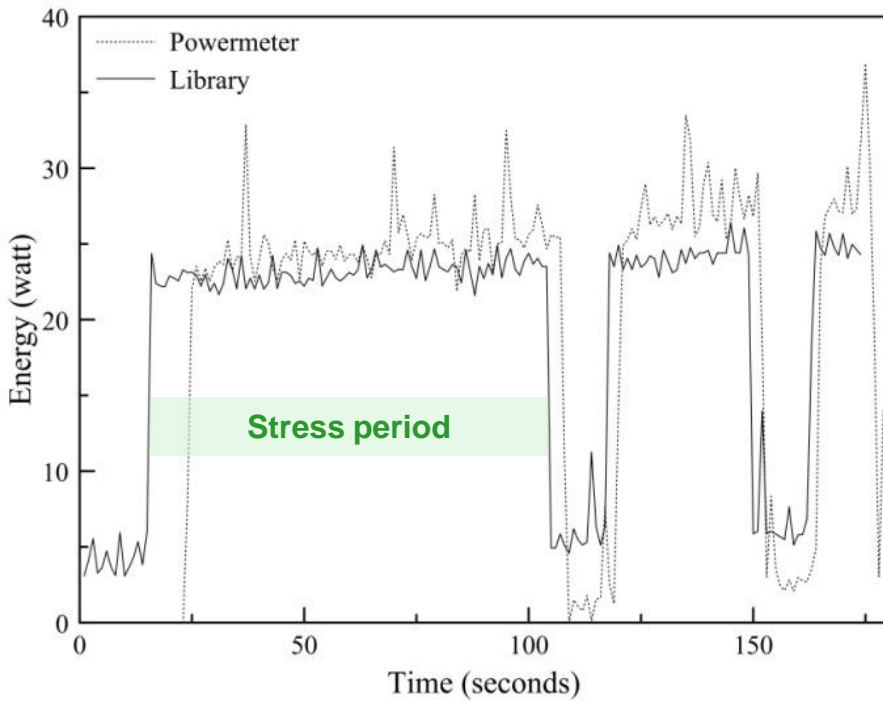
Mp1ayer movie player



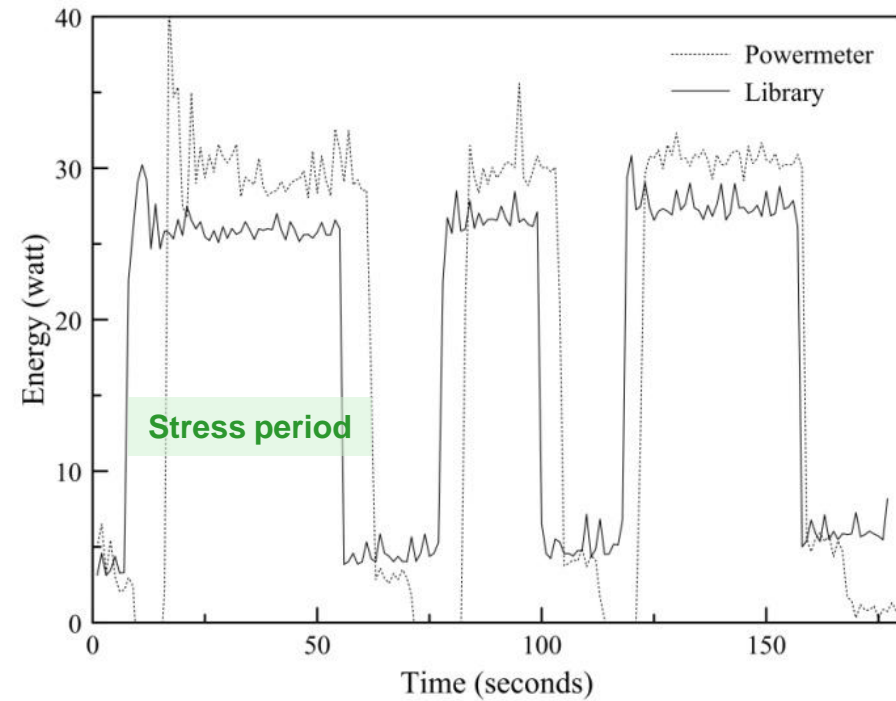
Time lag due to different refresh times between powermeter and powerAPI

Validation

Tomcat web server



Jetty web server



Conclusion & Perspectives

Synthesis

Need tools which could help developer to monitor software energy consumption

Some libraries already exist but integration and evolution are difficult

powerAPI = Process-level energy sensor library with a modular and adaptive architecture

We are working on new modules development (memory, disk...)

Applications

Some applications already exist

Process-level energy consumption visualizations

Monitoring and profiling Java applications at runtime
(eSurgeon)

Thank you!

Linux: Understanding Process-Level Power Consumption

Aurélien Bourdon

Adel Noureddine

Romain Rouvoy

Lionel Seinturier

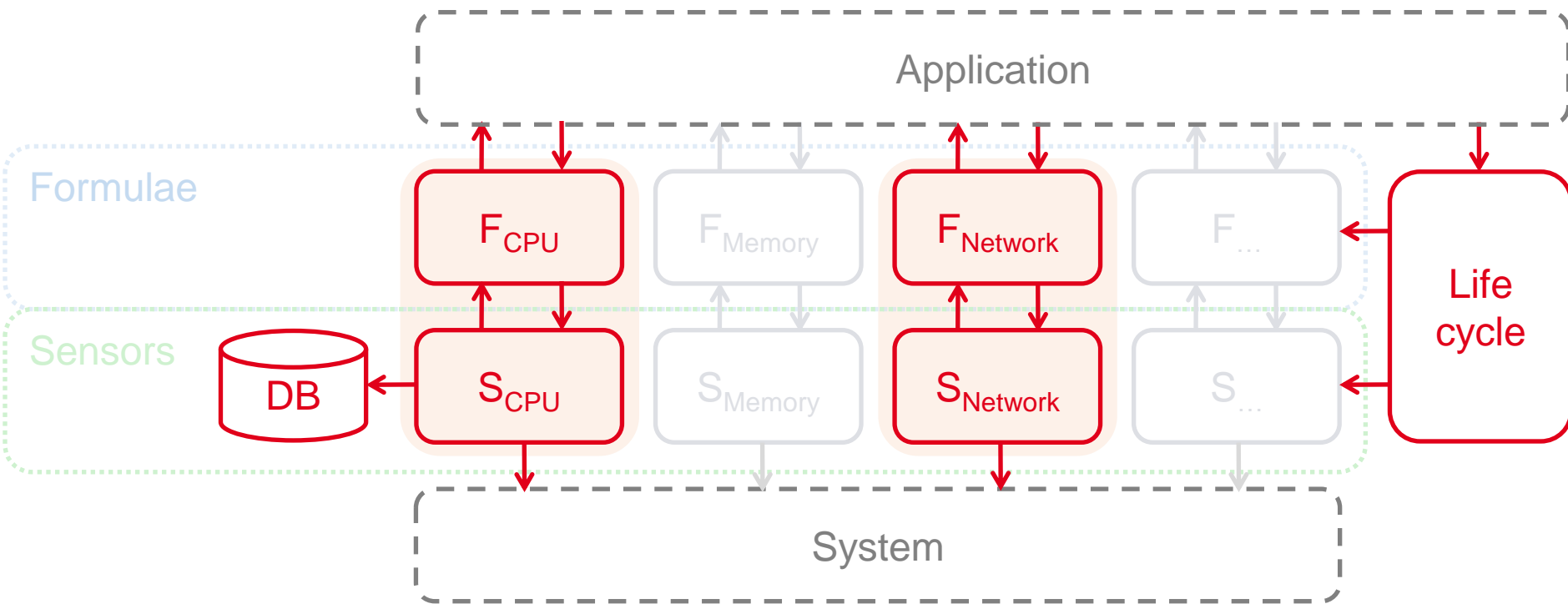
firstname.lastname@inria.fr

Green Computing Middleware'11

December 12th 2011

Backup slides

Architecture overview, current state



CPU power consumption formula

$$P_{CPU}^{f,v} = C \times f \times v^2$$

$$P_{CPU}^{f_{TDP},v_{TDP}} = TDP \times 0.7$$

$$TDP \times 0.7 = C \times f_{TDP} \times v_{TDP}^2$$

$$C = \frac{TDP \times 0.7}{f_{TDP} \times v_{TDP}^2}$$

Technical details

C++

Qt

SOF

CMake