Estimating processes' power usage
using only a wattmeter (and a solver)

Valentin LORENTZ, Laurent LEFÈVRE, Gilles FEDAK — valentin.lorentz@ens-lyon.org
ÉNS de Lyon – Inria – Université de Lyon – LIP (UMR CNRS, INRIA, ENSL, UCBL)

Introduction

We propose an automatic solution to estimate the energy consumption of applications running on a computer.

This solution combines a physical wattmeter (at the power plug) and a software framework (the solver) able to estimate automatically the power cost of a process or an application, based on a correlation between variations in the energy consumption and lifetimes of processes.

Policies

A simple catalog

A map: program $\mapsto$ average power. Problems:
- variability of the environment: instance of the hardware, DVFS, etc.
- variability of the input: size, nature of the content

Fine-grained catalog

A map: hardware event $\mapsto$ energy. Problems:
- requires specific hardware
- complex configuration
- different for each hardware instance

Policies based on a wattmeter

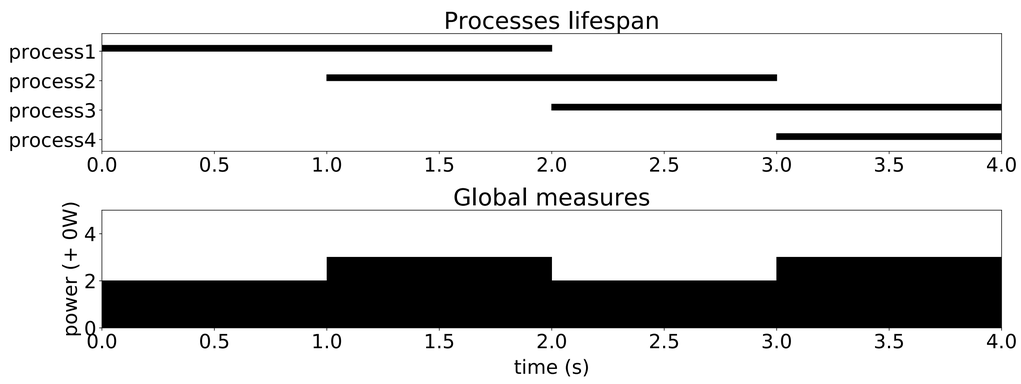
$$\mathcal{P}_{measured} - \mathcal{P}_{static} = \sum_i power(process_i)$$

Known variables: $\mathcal{P}_{measured}$ and $\mathcal{P}_{static}$.
Unknown variables: $power(process_i)$.
$\hookrightarrow$ Problem: 1 equation and n variables

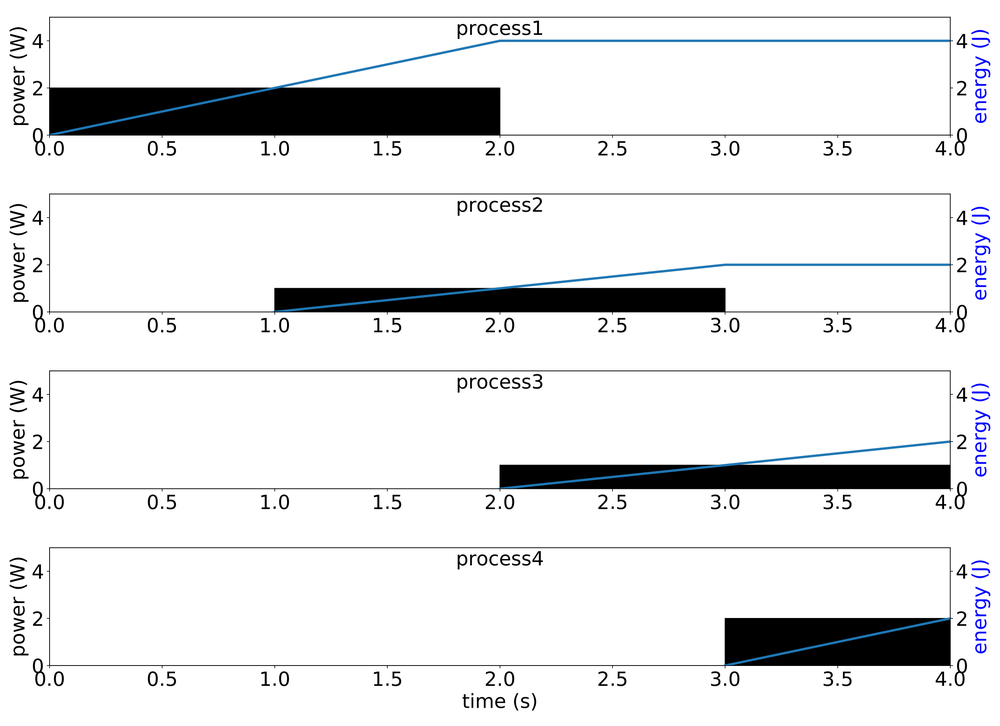
Solution: Add more equations.

A simple example of the Frontier Policy

Input



Resolution



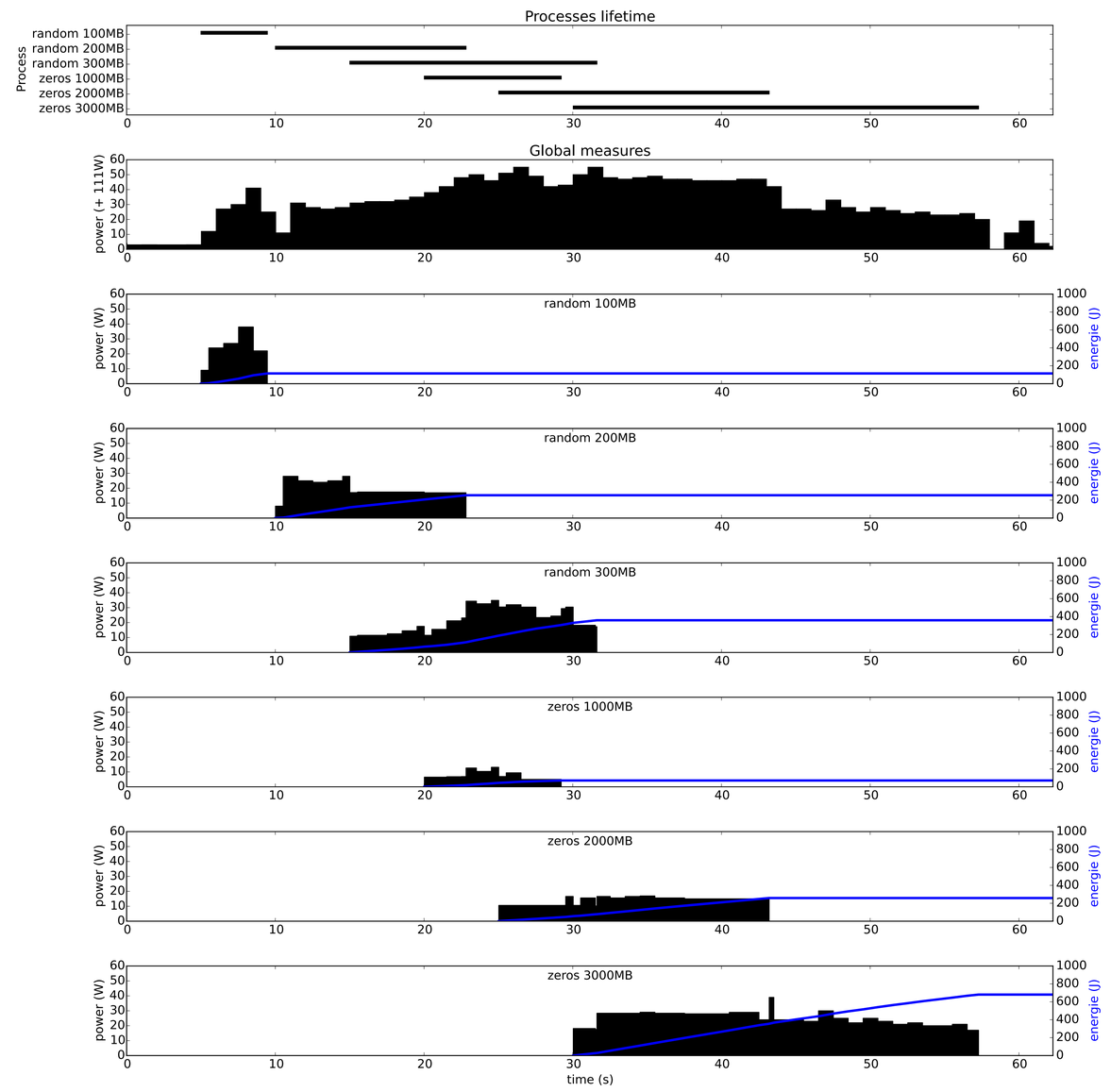
Formal definition of the Frontier Policy

A_t: set of processes alive at time t. $\mathcal{P}_t$: power used by the computer at t. $P_{p,t}$ (the unknown variables): power used by process p at t.

Minimize $\sum_t \sum_{p \in A_t \cap A_{t+1}} |P_{p,t+1} - P_{p,t}|$ such that:

$$\begin{cases} \sum_{p \in A_t} P_{p,t} = \mathcal{P}_t - \mathcal{P}_{static} & \forall t, A_t \neq \emptyset \\ P_{p,t} = 0Watt & \forall t, \forall p \notin A_t \\ P_{p,t} \geq 0Watt & \forall t, \forall p \in A_t \end{cases}$$

Example with `gzip`



Run time