How to estimate processes' power usage using only a wattmeter (and a solver)?

Valentin LORENTZ,
Laurent LEFÈVRE, Gilles FEDAK

COMPAS'2017

Motivation

How to know the power usage of each parts of an application without being intrusive?

Challenges

- Hardware heterogeneity
- Static and dynamic power

A simple catalog

A map: program $\mapsto$ energy (Joules) or average power (Watts)

Problems:

- Variability of the environment: instance of the hardware, DVFS, etc.
- Variability of the input: size, content

Fine-grained catalog

Relies on hardware counters (number of cache hits/misses, instructions, etc.)

Problems:

- Requires specific hardware
- Complex configuration
- Different for each hardware instance

Existing work

- hardware counters: PowerAPI, WattsKit, macOS' `top`
- monitor resource usage: pTop, PowerTop

Policies based on a wattmeter

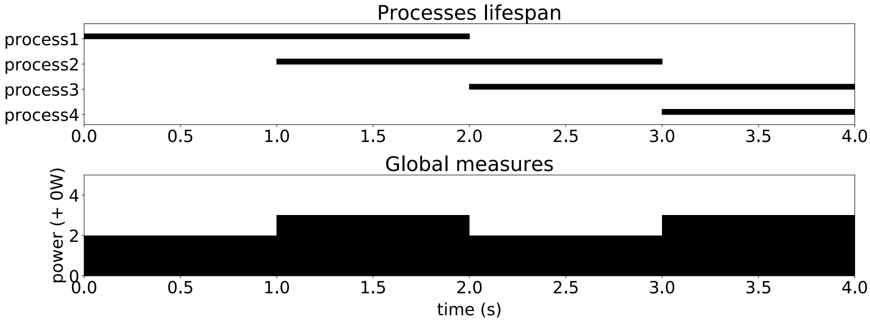
$$\mathcal{P}_{measured} - \mathcal{P}_{static} = \sum_i power(process_i)$$

We know $\mathcal{P}_{measured}$ and $\mathcal{P}_{static}$.
We want to know each of the $power(process_i)$.
$\hookrightarrow$ Problem: 1 linear equation and n variables

Solution: Add more equations. But how?

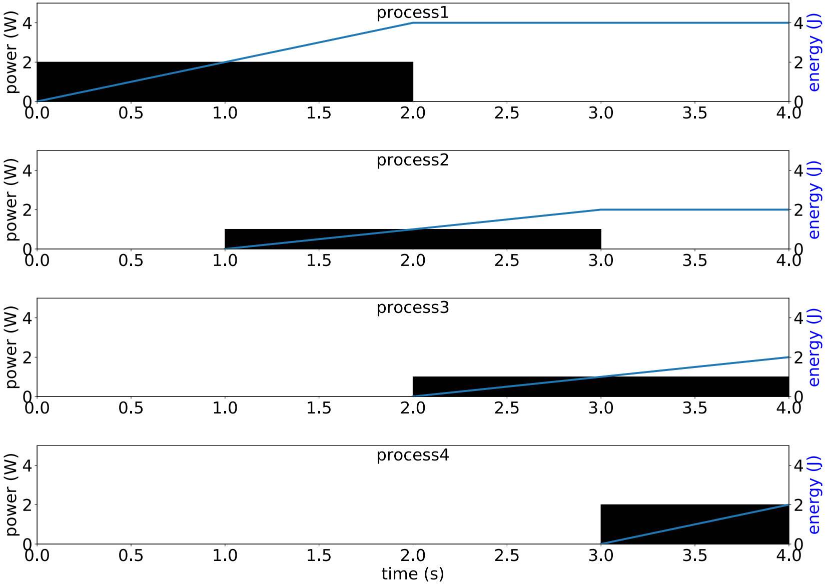
Key idea of the Frontier Policy

Example (input)



On this example, one can attribute a constant power usage to each process.

Example (output)



Formally

Find values of $P_{p,t}$ such that:

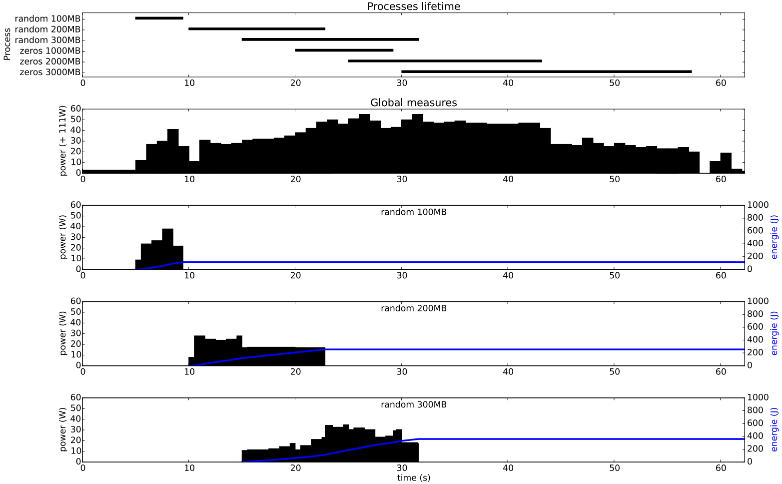
$$\begin{cases} \sum_{p \in A_t} P_{p,t} = \mathcal{P}_t - \mathcal{P}_{static} & \forall t, A_t \neq \emptyset \\ P_{p,t} = 0 Watt & \forall t, \forall p \notin A_t \\ P_{p,t} \geq 0 Watt & \forall t, \forall p \in A_t \end{cases}$$

and $\sum_t \sum_{p \in A_{t-1} \cap A_t} \left| P_{p,t-1} - P_{p,t} \right|$ is minimal

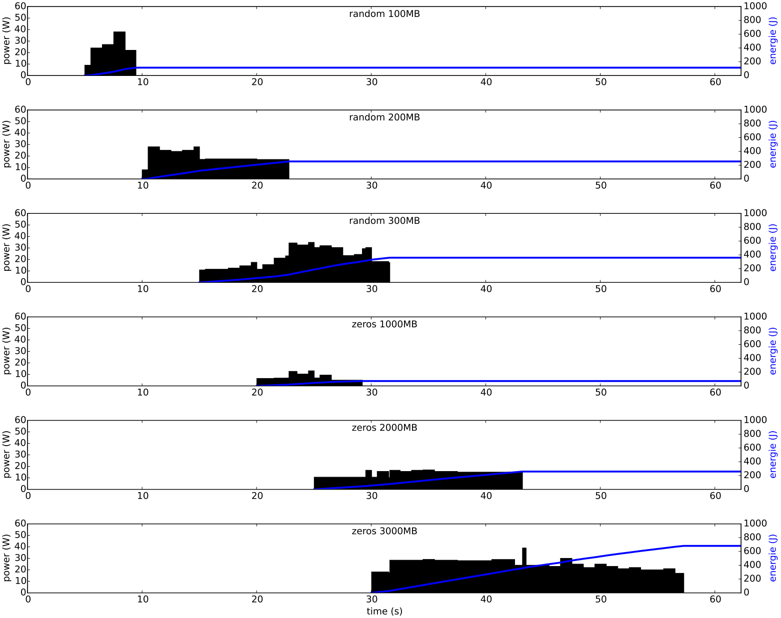
Experimental setting

- On Grid'5000, Taurus cluster.
- Running `gzip` on random files (100MB to 300MB) and zeroed files (1GB to 3GB).

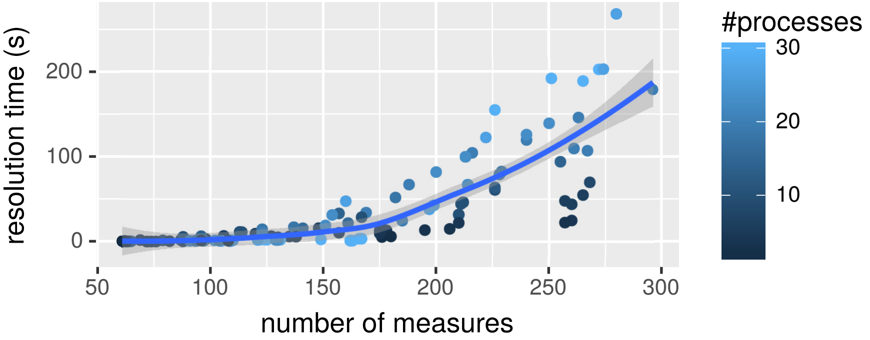
Estimation of gzip (part 1)



Estimation of gzip (part 2)



Execution time



Processing time increases rapidly... but we can easily decrease the number of measures.

Conclusion and future work

- Definition of a power attribution policy
- The Frontier policy: a zero-configuration non-intrusive attribution policy

- Correlate profiles of multiple executions of the same program
- Applications using this work: scheduling, ...

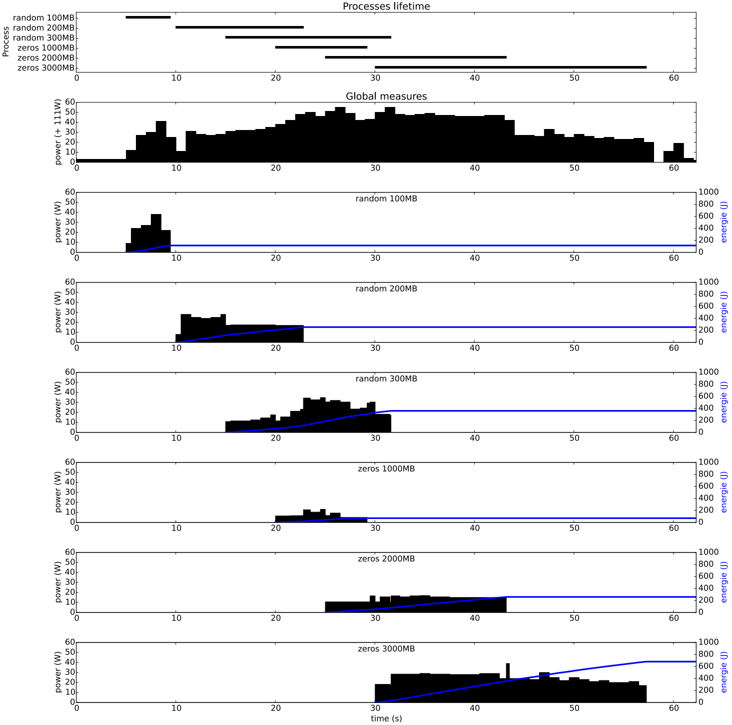
6. Linear program

7. Estimation of gzip

Linear program

Minimize $\sum_{t} \sum_{p \in A_t \cap A_{t+1}} \Delta_{p,t}$ such that:

$$\begin{cases} \sum_{p \in A_t} P_{p,t} = \mathcal{P}_t - \mathcal{P}_{static} & \forall t, A_t \neq \emptyset \\ P_{p,t} = 0 Watt & \forall t, \forall p \notin A_t \\ P_{p,t} \geq 0 Watt & \forall t, \forall p \in A_t \\ \Delta P_{p,t} \geq P_{p,t-1} - P_{p,t} & \forall t, \forall p \in A_{t-1} \cap A_t \\ \Delta P_{p,t} \geq P_{p,t} - P_{p,t-1} & \forall t, \forall p \in A_{t-1} \cap A_t \end{cases}$$


Processes lifetime
Process
random 100MB
random 200MB
random 300MB
zeros 1000MB
zeros 2000MB
zeros 3000MB
Global measures
power (+ 111W)
random 100MB
random 200MB
random 300MB
zeros 1000MB
zeros 2000MB
zeros 3000MB
power (W)
energie (J)
time (s)