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

Valentin Lorentz, Laurent Lefèvre, Gilles Fedak

COMPAS’2017
1 Context

2 Simple policies: Catalogs

3 The Frontier Policy

4 Experiments

5 Conclusion
1. Context

2. Simple policies: Catalogs

3. The Frontier Policy

4. Experiments

5. Conclusion
Motivation

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

- Hardware heterogeneity
- Static and dynamic power
1 Context

2 Simple policies: Catalogs

3 The Frontier Policy

4 Experiments

5 Conclusion
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
1. Context
2. Simple policies: Catalogs
3. The Frontier Policy
4. Experiments
5. Conclusion
Policies based on a wattmeter

\[ P_{\text{measured}} - P_{\text{static}} = \sum_{i} \text{power}(\text{process}_i) \]

We know \( P_{\text{measured}} \) and \( P_{\text{static}} \).
We want to know each of the \( \text{power}(\text{process}_i) \).
\[ \rightarrow \] Problem: 1 linear equation and \( n \) variables

Solution: Add more equations. But how?
Key idea of the Frontier Policy

Correlate power variations with processes’ start and stop.
On this example, one can attribute a constant power usage to each process.
Formally

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

$$
\begin{align*}
\sum_{p \in A_t} P_{p,t} &= P_t - P_{\text{static}} & \forall t, A_t \neq \emptyset \\
P_{p,t} &= 0\text{Watt} & \forall t, \forall p \notin A_t \\
P_{p,t} &\geq 0\text{Watt} & \forall t, \forall p \in A_t
\end{align*}
$$

and

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

2. Simple policies: Catalogs

3. The Frontier Policy

4. Experiments

5. Conclusion
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)

Valentin Lorentz

COMPAS'2017
Estimation of gzip (part 2)
Processing time increases rapidly... but we can easily decrease the number of measures.
1 Context

2 Simple policies: Catalogs

3 The Frontier Policy

4 Experiments

5 Conclusion
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:

\[
\begin{align*}
\sum_{p \in A_t} P_{p,t} &= P_t - P_{static} \quad \forall t, A_t \neq \emptyset \\
P_{p,t} &= 0 \text{Watt} \quad \forall t, \forall p \notin A_t \\
P_{p,t} &\geq 0 \text{Watt} \quad \forall t, \forall p \in A_t \\
\Delta P_{p,t} &\geq P_{p,t-1} - P_{p,t} \quad \forall t, \forall p \in A_{t-1} \cap A_t \\
\Delta P_{p,t} &\geq P_{p,t} - P_{p,t-1} \quad \forall t, \forall p \in A_{t-1} \cap A_t
\end{align*}
\]