The Role of Software Operation in Ensuring Energy-Optimal Designs in Modern Computing

Prof. Samuel Xavier-de-Souza

Laboratory of Parallel Architectures for Signal Processing

November 26, 2024

Agenda

Agenda

Universidade Federal do Rio Grande do Norte

- **Brazil - Rio Grande do Norte - Natal**



- **UFRN:** 66-year old university. 40k students, 2.5k profs.
- **IMD/UFRN:** IT Institute, graduate programs, technology park ∼150 startup and companies
- **Lab. of Parallel Architectures for Signal Processing**[1]
  - **Staff**: 5 Profs, 9 postdocs, visiting associate researchers
  - **Students**: ∼20 grads, ∼10 undergrads
  - **Collaboration**: many universities and companies around the globe

[1] http://lapps.imd.ufrn.br

Laboratory of Parallel Architectures for Signal Processing

- **Basic research:** high-performance computing, numerical algorithms, information theory, analysis of cyclostacionary processes, data analytics, and machine learning.
- **Applied research:** high-performance geophysics, fault-tolerant computing for aerospace, parallel GNSS receivers, **energy-efficient parallel software**, energy-efficient communications, parallel scalability profiling tools, computational load balancing, block recursive matrix inversion, **software-performance and software-energy models**, correntropy, automatic classification of modulations, and channel and source encodings.

Agenda

The need for energy-efficient computing systems

Information & Communication Technology

Consume about 10% of Earth's energy resources.

- Estimated to consume 1/5 of the energy produced on the planet by 2030.
- The whole aviation industry consumes only about 50% of that.
- **Energy**: important at different scales for different reasons:
  - **Large systems**: financial & environmental costs
    e.g. Datacentres, Supercomputers, corporate IT infrastructure, etc.
  - **Small systems**: autonomy & size, environmental cost
    e.g. Wearables, embedded & mobile, wireless comms & computing, etc.

Present and future necessity:

More energy-efficient communications & computing systems.

The need for energy-efficient computing systems

- This talk focus on **Energy-efficient Computing.**
- Nevertheless, a lot on communications is becoming computing.

The convergence of communications towards software:

- Software-defined networking & software-defined datacenter;
- Software-defined radio & cognitive radio;
- more complex source & channel encodings;
- data compression, data & traffic analysis;
- and so on.

Agenda

Hardware & software energy-optimal designs

Ways to improve energy efficiency in computing systems:

Hardware optimisation

- ILP, multiple issues, out-of-order, hardware threads;
- presenting **diminishing returns**.

Hardware specialisation

- ASICs, FPGA, accelerators;
- **less flexibility**.

Hardware-software co-design

- specialized solution;
- **higher** non-recurring engineering costs.

Software closer to hardware

- less abstractions → **harder to program**;
- performance-driven rather than **energy-driven** tools.

Hardware & software energy-optimal designs

Living beings are very energy efficient.

Why computing isn't?

- Hardware designs are often **extensively optimised** for
  - Power, performance, or cost.
- **Optimising for energy** requires design **trade-offs**
  - between power and performance.
- Energy-optimised **programmable** hardware (CPUs):
  - **ultimately controlled by software.**

Hardware & software energy-optimal designs

Living beings are very energy efficient.

Why computing isn't?

- Hardware designs are often **extensively optimised** for
  - Power, performance, or cost.
- **Optimising for energy** requires design **trade-offs**
  - between power and performance.
- Energy-optimised **programmable** hardware (CPUs):
  - **ultimately controlled by software.**

Problem 1: **poor software development**.

Waste of **hardware optimization** efforts.

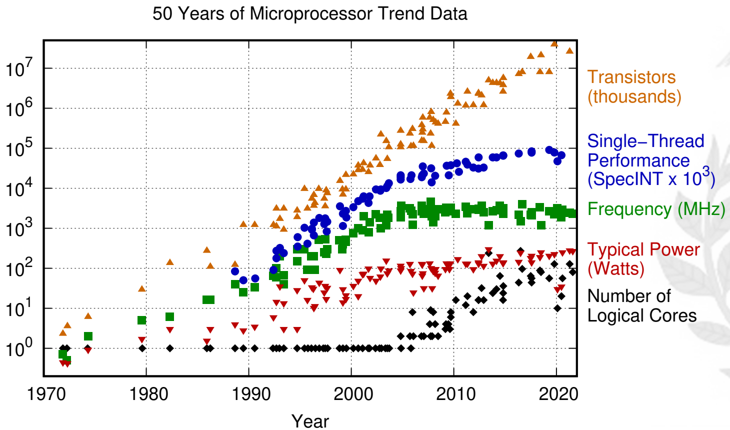
Hardware & software energy-optimal designs

- **Single-core** systems used to rule the world
  - Parallel systems **historically neglected** due to Amdahl's law (1967)
  - Exponential growth in single-core **performance**:
    - **Semiconductor industry:** Moore's law → higher operating frequencies
    - **Hardware design:** Better ILP, caches, out-of-order, hardware threads
- Only possible because **heat** was kept at acceptable and controlled levels
  - Power density driven up by
    - Higher operating frequencies
    - Moore's law: smaller transistors
  - Power dissipation (and heat) controlled by
    - Advances in cooling systems; increase in overall system power.
    - Moore's Law: smaller transistors → less power dissipation per transistor.

Software development in the single-core era:

Optimize software for **energy** by targeting **performance**.

Hardware & software energy-optimal designs



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
Plot and data collected for 2010–2019 by K. Rupp

Hardware & software energy-optimal designs

In mid-2000s, the industry realized single-core was impractical

By 2030s, one cm² of silicon → one cm² of the sun's surface.

The **Multi-core Era** was born

- Nominal hardware performance **growth sustained**.
  - More processing cores → more **nominal performance**.
- **Software** becomes **main responsible** for transforming
  - **nominal** performance into **actual** performance.

Software development in the multi-core era:

Parallel computing becomes a **necessity** rather than an alternative.

Agenda

Software operation to re-ensure energy-optimal designs

Software operation is a fairly new concept

For decades, the energy/performance optimum was {single-core, maximum-frequency}

- **Operation space** is much larger in the **multi-core era**
  - Number of cores,
  - Number of threads per core,
  - Operating frequency of the cores (single- or multi-domain),
  - Type of core (heterogeneous computing).
  - Accelerators (GPUs, FPGAs, etc.)

Energy-optimal software operation in the **multi-core era:**

requires a careful choice of the operating configuration (OS + user)

Software operation to re-ensure energy-optimal designs

Total number of possible configurations - 2-cluster HMP

$$T_{\mathrm{Conf}} = C_{\mathrm{b}} \times N_{\mathrm{F_b}} \times C_{\mathrm{L}} \times N_{\mathrm{F_L}} - (N_{\mathrm{F_b}} \times N_{\mathrm{F_L}})$$

- C_{b} and C_{L} $\rightarrow$ Possible counts of *big* and *LITTLE* cores
- $N_{\mathrm{F_b}}$ and $N_{\mathrm{F_L}}$ $\rightarrow$ Possible *big* and *LITTLE* frequencies

Typical heterogeneous processor: Samsung Exynos 7420

- 4 A53 *LITTLE* cores (12 frequencies)
- 4 A57 *big* cores (14 frequencies)
- Operation space $\rightarrow 5 \times 14 \times 5 \times 12 - (12 \times 14) = 4032$ points

Software operation to re-ensure energy-optimal designs

Energy-efficient computing systems

Energy-efficient **hardware design**, AND

Energy-efficient **software development**, AND

Energy-efficient **software operation**.

- Assuming parallel hardware is energy-optimal
- Assuming software development is parallel and energy-optimal
- How not to waste these optimizations with sub-optimal software operation?

Software operation to re-ensure energy-optimal designs

Energy-efficient computing systems

Energy-efficient **hardware design**, AND

Energy-efficient **software development**, AND

Energy-efficient **software operation**.

- Assuming parallel hardware is energy-optimal
- Assuming software development is parallel and energy-optimal
- How not to waste these optimizations with sub-optimal software operation?

Problem 2: **poor software operation**.

Waste of **hardware optimisation** efforts

Software operation to re-ensure energy-optimal designs

Energy-efficient computing systems

Energy-efficient **hardware design**, AND

Energy-efficient **software development**, AND

Energy-efficient **software operation**.

- Assuming parallel hardware is energy-optimal
- Assuming software development is parallel and energy-optimal
- How not to waste these optimizations with sub-optimal software operation?

Problem 2: **poor software operation**.

Waste of **hardware optimisation** efforts & **software optimisation** efforts.

Software operation to re-ensure energy-optimal designs

Solution: first, make energy a primary goal, then either

- Look for the overall best energy configuration
- Look for the best energy configuration for a given performance

E.g. symmetric multi-core: for a given performance

- Reduce power by reducing frequency and voltage
- Compensate slowdown by increasing core count → Amdahl's law: limited gains

- What is the energy-optimal frequency and core count?
- What about heterogeneous multi-processing?

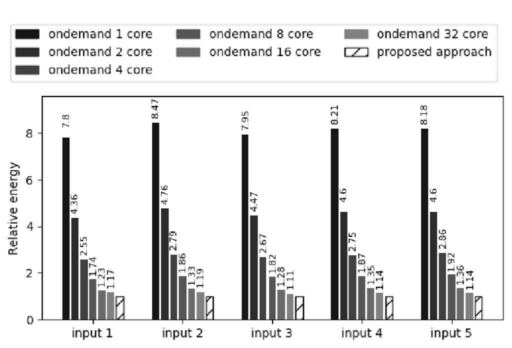
Need for energy models

The cost of modeling

- The cost of building models is relevant
  - Build model off-line
  - Use model to reduce operation space (Pareto points)
  - Avoid non-structured models to reduce sampling needs
  - Combine off- and online approaches to reduce overhead

Our approach: focus on the software operation

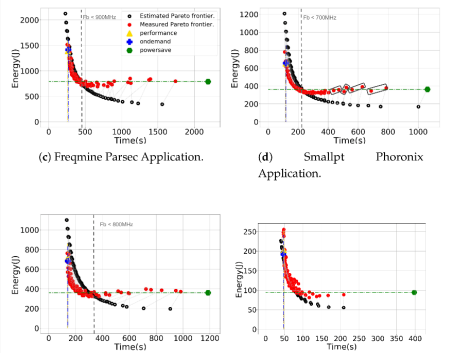
High-Performance Computing



(a) Fluidanimate

https://doi.org/10.1109/HPCS48598.2019.9188110

Embedded/Mobile Systems



(c) Freqmine Parsec Application.

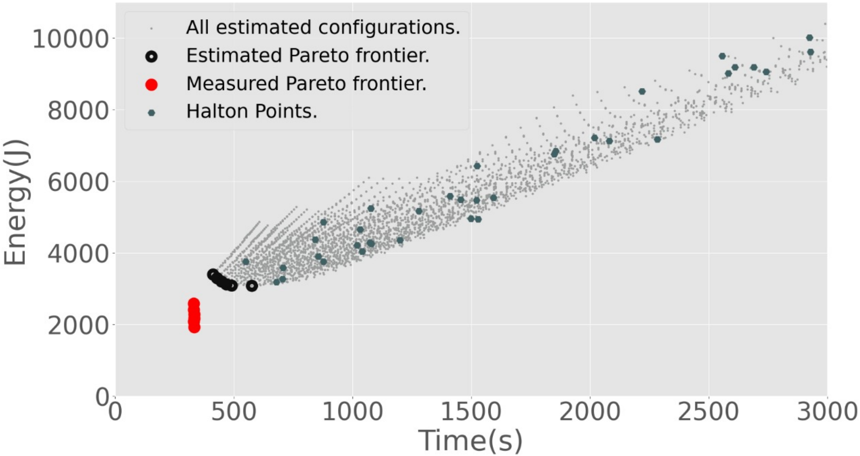
(d) Smallpt Phoronix Application.

(e) x264 Phoronix Application.

(f) kmeans Rodinia Application.

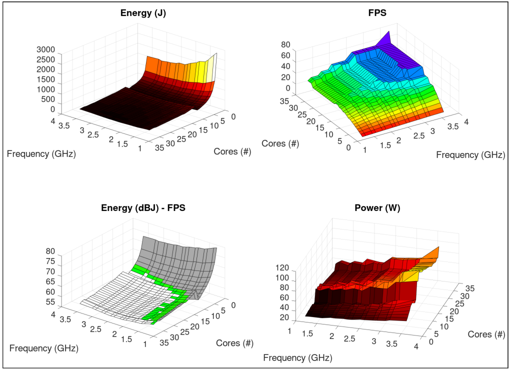
https://doi.org/10.3390/en13092409

Our approach: focus on the software operation



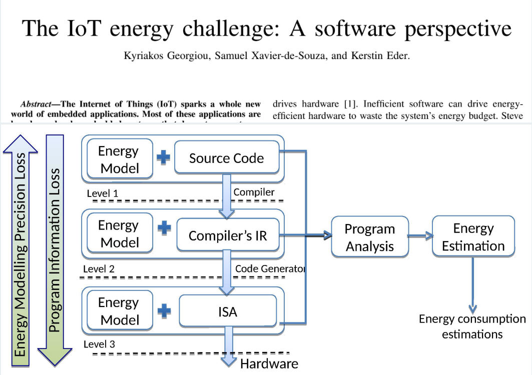
Our approach: focus on the software operation

Cloud/data centre



https://doi.org/10.3390/en13092162

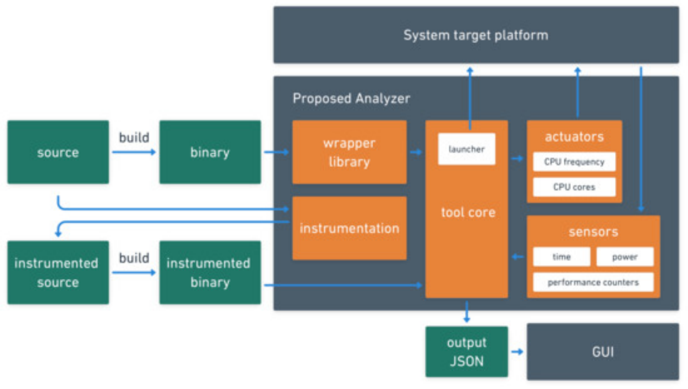
Internet of Things



https://doi.org/10.1109/LES.2017.2741419

Our approach: focus on the software operation

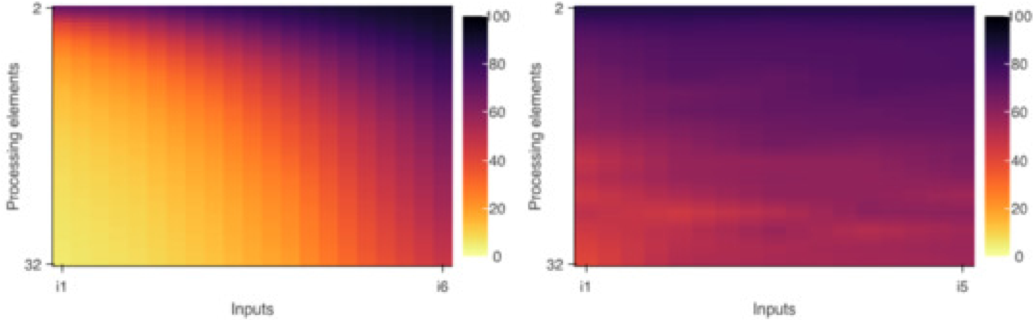
Analysis & visualization tools



https://doi.org/10.3390/electronics11050689

Our approach: focus on the software operation

Analysis & visualization tools



(a) (b)

Takeaways

- Increasing the energy efficiency of ICT is a present and future world necessity;
- Much of what we know and rely on computing changed in the mid-2000s;
- </> Sound parallel software development to re-ensure energy-optimal hardware designs;
- Software operation becomes a key to re-ensure optimizations are not wasted;
- To find optima, models are necessary to assess vast software operation space.

Takeaways

- Increasing the energy efficiency of ICT is a present and future world necessity;
- Much of what we know and rely on computing changed in the mid-2000s;
- </> Sound parallel software development to re-ensure energy-optimal hardware designs;
- Software operation becomes a key to re-ensure optimizations are not wasted;
- To find optima, models are necessary to assess vast software operation space.

A vision of the future:

- Every software release will have an associated performance & energy model;
- The OS will use it to make wise decisions about resource allocation and QoS;
- Software will be rated by its ability to consume the allocated resources efficiently.

Thank you!

http://dca.ufrn.br/~samuel