Parallel Programming and Heterogeneous Computing

E1 - Energy-Aware Computing

Sven Köhler, Lukas Wenzel, Max Plauth, and Andreas Polze

Operating Systems and Middleware Group

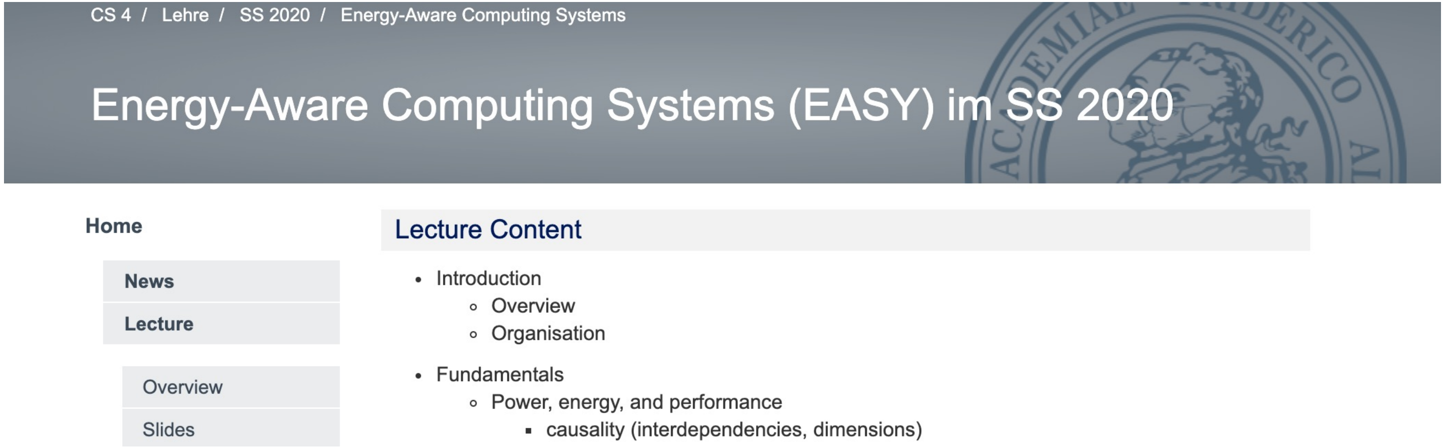
Hasso Plattner Institut

Digital Engineering · Universität Potsdam

About this Lecture

This unit is a highly condensed version of the *Energy-Aware Computing Systems (EASY)* lecture by Prof. Dr.-Ing. Timo Hönig (RUB, formerly FAU).

If you are interested in more content, check out the FAU-CS4 website or convince us to offer an entire semester-spanning lecture, here at HPI.


CS 4 / Lehre / SS 2020 / Energy-Aware Computing Systems

Energy-Aware Computing Systems (EASY) im SS 2020

Home
- News
- Lecture
  - Overview
  - Slides

Lecture Content

- Introduction
  - Overview
  - Organisation
- Fundamentals
  - Power, energy, and performance
    - causality (interdependencies, dimensions)

https://www4.cs.fau.de/Lehre/SS20/V_EASY/

1 Background

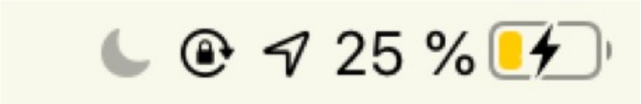
Our Computing Systems Use Massive Amounts of Energy

SUPERCOMPUTER FUGAKU - SUPERCOMPUTER FUGAKU, A64FX 48C 2.2GHZ, TOFU INTERCONNECT D [1]

Site:	RIKEN Center for Computational Science
System URL:	https://www.r-ccs.riken.jp/en/fugaku/project
Manufacturer:	Fujitsu
Cores:	7,630,848
Memory:	5,087,232 GB
Processor:	A64FX 48C 2.2GHz
Theoretical Peak (Rpeak)	537,212 TFlop/s
Nmax	21,288,960
Power Consumption	
Power:	29,899.23 kW (Optimized: **26248.36** kW)
Power Measurement Level:	2

[2]

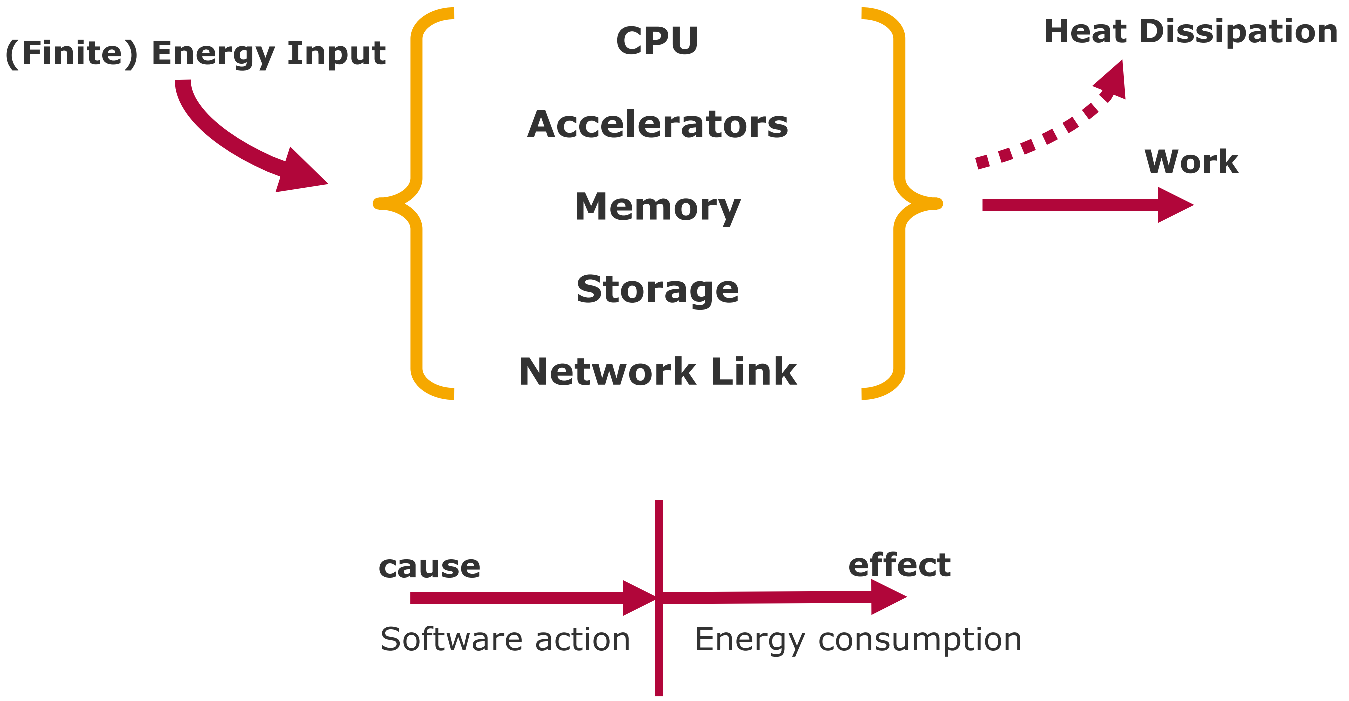

Energy Research & Social Science
Volume 38, April 2018, Pages 128-137
Original research article
Digitalisation, energy and data demand: The impact of Internet traffic on overall and peak electricity consumption
Janine Morley, Kelly Widdicks, Mike Hazas
Show more
https://doi.org/10.1016/j.erss.2018.01.018
Get rights and content
Under a Creative Commons license
open access
Abstract
Over the last decade, concerns have been raised about increases in the electricity used by information technologies, other consumer electronic devices, data centres



ParProg21 E1
Energy-Aware Computing
Sven Köhler

[1] Fugaku Supercomputer, Top500 List, Acc. 2021-06-22. https://www.top500.org/system/179807
[2] Morley, J., Widdicks, K., & Hazas, M. (2018). Digitalisation, energy and data demand: The impact of Internet traffic on overall and peak electricity consumption. Energy Research & Social Science, 38, 128-137.

What Consumes Energy?



Energy vs. Power

The energy demand E that is required to execute an operation is the integral over the system's power demand from start (t_s) to end (t_e) of the operation.

$$E = \int_{t_s}^{t_e} P(t)\,dt$$

Energy E (unit: J or Ws) is the ability to do work.

E is a suitable metric for:

- your battery life
- your electricity bill
- your carbon footprint

Power P (unit: W or J/s) is the rate of doing work.

P is a suitable metric for:

- power supply constraints (peak power)
- prediction of heat dissipation (cooling facilities)

Reducing the energy demand requires to reduce the run-time or the power demand.

Power Demand of Computing Circuits

$$P_{total} \sim \overbrace{N \cdot C \cdot F \cdot V^2}^{\text{dynamic}} + \overbrace{N \cdot R_{leak}^{-1} \cdot V^2}^{\text{static}}$$

Number of Active Transistors	Capacitance	Clock Frequency	Supply Voltage	Leak Resistance
depends on circuit design and usage (if power-gating is present)	fixed by circuit design	limits given by circuit design	limits given by circuit design	fixed by circuit design
		configurable (impacts performance)	relates to clock frequency	

Reducing the power demand requires to shut off transistors or reduce the clock frequency.

2

Energy Management

Dynamic Voltage and Frequency Scaling

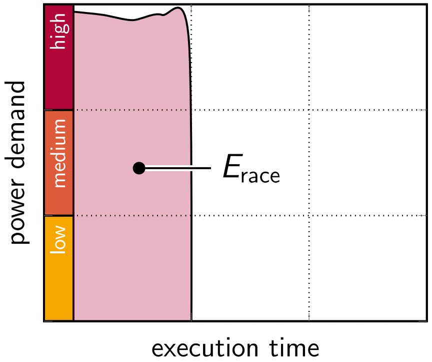
Modern compute architectures allow developers to actively regulate voltage and clock frequency at a fine time granularity (tens of milliseconds).

Examples:

- Intel CPUs: RAPL using e.g., `powergov` or direct control register access
- IBM POWER CPUs: EnergyScale via CIM or HMC
- ARM: Plenty of tools and libraries, usually by SOC/board vendor
- NVidia GPUs: `nvidia-smi` or NVidia Management Library
- AMD GPUs: In the Linux sysfs at `/sys/class/drm/…/pp_od_clk_voltage`

Proper power-gating is tricky. Without, your **idling** core **is wasting energy**. Thus, **minimize** the **idle time!** Put your cores to sleep, when you can.

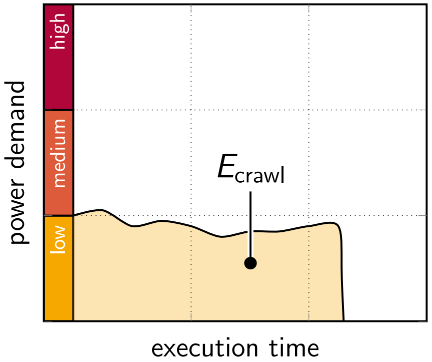
Race or Crawl to Sleep?



race-to-sleep

Maximize sleep time using a blocking management method after finishing pending work.

Suits especially compute-intensive processes



crawl-to-sleep

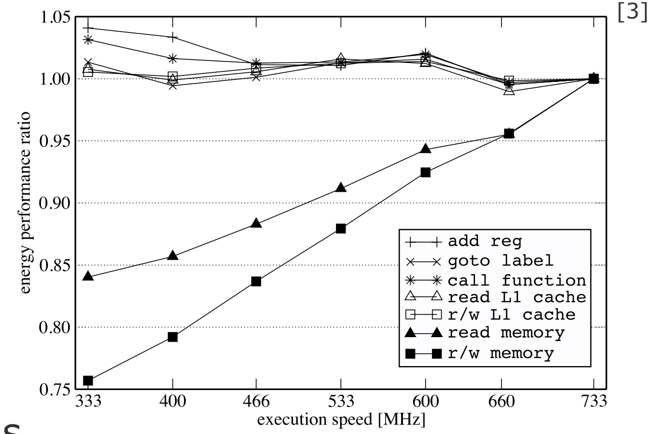
Configure system at minimum voltage and clock rate, aiming for low average/peak power.

Suits especially I/O- or memory-bound processes

Data Processing And Computing

A **naïve approach** to energy-aware computing:

Run memory-bound and CPU-bound threads with low and high clock speed, respectively.



Problems of this approach:

- dynamic characteristics of workloads
- simple system model (#cores, interlocked voltages, cache size)
- input-dependent, variable size of working set
- costs for frequency switching

[3] Weissel, A., Bellosa, F. Process Cruise Control: Event-Driven Clock Scaling for Dynamic Power Management. In: Proceedings of the International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES'02) ACM, 2002, S. 238–246

Memory-aware Scheduling (Combining) I

Observation:

Contention between cores due to resource demand (caches, memory) leads to run-time penalties (depending on thread characteristics).

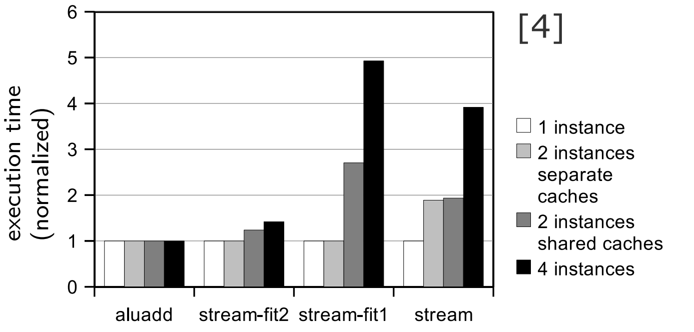


Figure 1. Normalized runtime of microbenchmarks running on the Core2 Quad

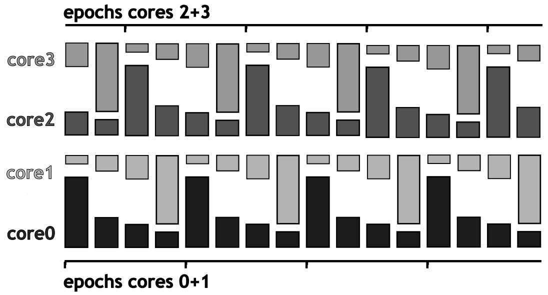


Figure 4. Sorted scheduling. Bars correspond to memory intensity.

Proposed strategy:
Combine and co-locate compute-bound and memory-bound threads to reduce contention (Gang scheduling[5])

[4] Merkel, A., Bellosa, F.: Memory-aware Scheduling for Energy Efficiency on Multicore Processors. In: Proceedings of the Workshop on Power Aware Computing and Systems (HotPower'08), 2008, S. 123–130
[5] Ousterhout, J. K. et. al.: Scheduling Techniques for Concurrent Systems. In: Proceedings of the 1982 International Conference on Distributed Computing Systems (ICDCS'82) Bd. 82, 1982, S. 22–30

Memory-aware Scheduling (Combining) II

Implementation:

- group CPU cores into pairs of two
- Run threads with complementary resource demands on each pair
- Scale to **lowest** frequency if **no** compute-bound threads are ready (only memory-bound threads ready)
- Scale to **highest** frequency if **at least one** compute-bound thread is ready

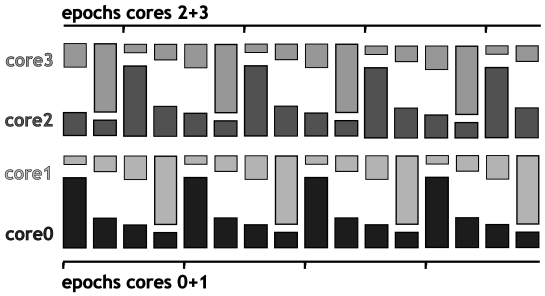


Figure 4. Sorted scheduling. Bars correspond to memory intensity.

Limitations and Considerations:

- inferences with kernel scheduling strategy (risks priority inversion)
- scheduling policy only effective for specific working set sizes
- memory hierarchy and cache sizes must be considered

Access and Execute (Sequencing)

Sequenced execution the extend phases of homogenous operations.

Reorder your instructions into two streams operations of the same kind

Access Phase

- prefetch data into caches, write intermediate results to memory
- run with low clock speed

Execute Phase

- Execute operations on data in hot caches (i.e., computations)
- run with high clock speed

Eliminates unnecessary CPU stalling and memory waits, but requires some compiler support and might cause additional synchronization efforts.

[6] Smith, J. E.: Decoupled Access/Execute Computer Architectures. In: Proceedings of the 9th Annual Symposium on Computer Architecture (ISCA’82), 1982, pp 112–119
[7] Koukos, K., Black-Schaffer, D., Spiliopoulos, V., Kaxiras, S.: Towards More Efficient Execution: A Decoupled Access-execute Approach. In: Proceedings of the 27th International ACM Conference on International Conference on Supercomputing (ICS’13), 2013

What else?

- Pick a more energy-efficient system (e.g., FPGA over CPU or high-efficiency cores like on ARM big.LITTLE machines)

- Optimize your algorithm!
- Optimize your implementation for performance, go to sleep
- But: Fast systems may use more energy than they save in time[8]

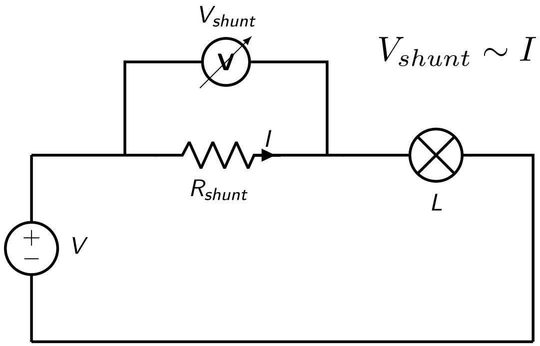
You will never know if your algorithm, implementation or management strategy is more energy efficient then another, unless you measure …

[8] Hönig, T., Janker, H., Eibel, C., Mihelic, O., & Kapitza, R. (2014). Proactive Energy-Aware Programming with PEEK. In 2014 Conference on Timely Results in Operating Systems (TRIOS 14).

Hasso Plattner Institut

3 Measuring Power and Energy

Measurement Methods



$$V_{shunt} \sim I$$

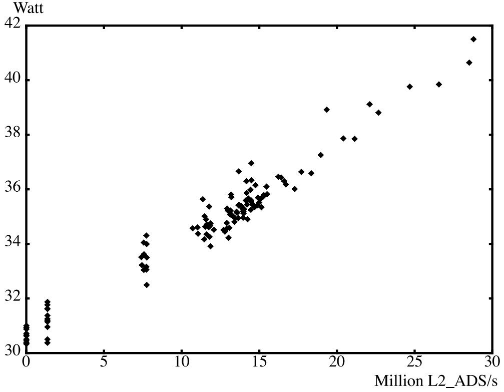


FIG. 3. Correlation of L2 Cache references and energy consumption

[9]

physical measurements

Direct or indirect physical method, like measuring the voltage drop across a resistor.

Quite accurate, little overhead, requires setup alteration

logical measurements

Based on a software power model, initially build upon physical measurements.

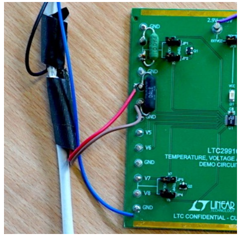
No additional circuits required, but model might be error-prone

[9] Bellosa, F.: The Benefits of Event-Driven Energy Accounting in Power-Sensitive Systems. In: Proceedings of the 2000 ACM SIGOPS European Workshop „Beyond the PC: New Challenges for the Operating System“ (EW ’00) ACM, 2000, S. 37–42

Measurement Facilities

External

i.e., standalone devices intercepting the supply between power source and measured device



On-Board

Part of the mainboard or SOC, often allow for distinction of separate power rails

IPMI, BMC,
Jetson counters

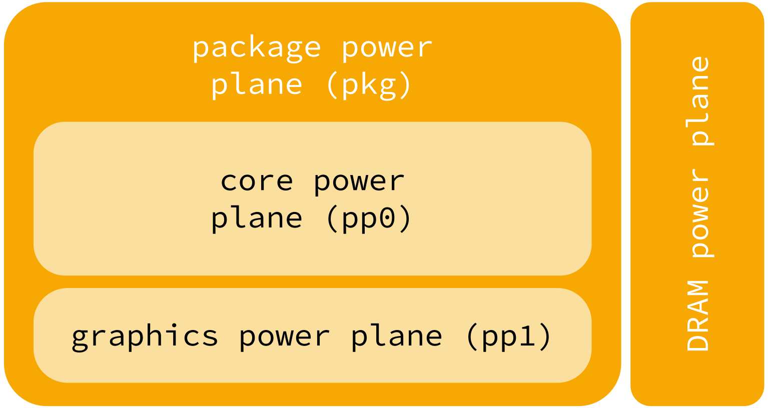
On-Chip

Integrated with the individual hardware platform, allows for most details

RAPL, PowerOCC
Apple M1 counters

Running Average Power Limit (RAPL)

- Available for Intel platforms, since Sandy Bridge
- Registers capture cumulative energy consumption (not power draw), at ~1 ms resolution (wrap around after ~60s)
- Accessible via control registers, Linux `sysfs`, or `perf_event_open`
- Semi-compatible AMD implementation since Ryzen Gen 3

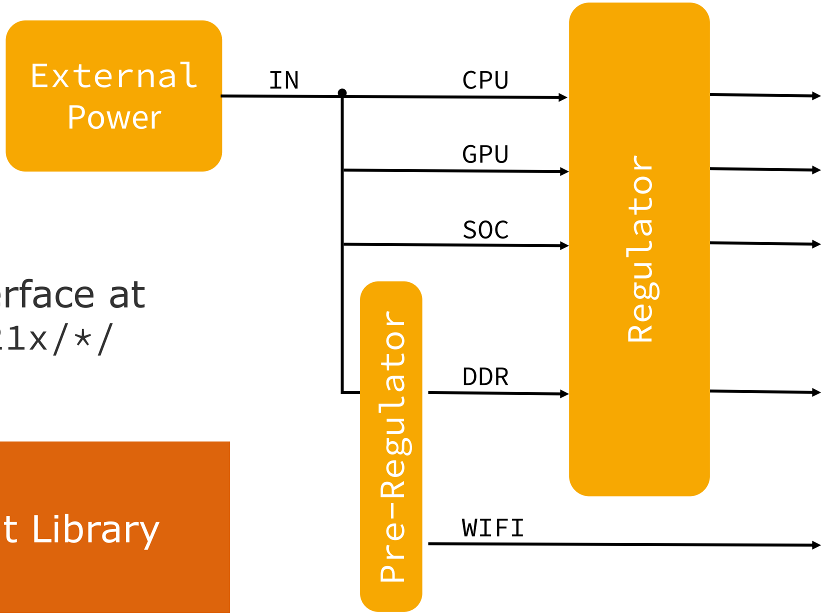


[10] Intel. 64 and IA-32 Architectures Software Developer's Manual (Volume 3). Acc. 2020-06-12
https://software.intel.com/content/dam/develop/public/us/en/documents/325384-sdm-vol-3abcd.pdf

NVidia Jetson TX2 Boards

Two triple-channel INA3221 power monitors:

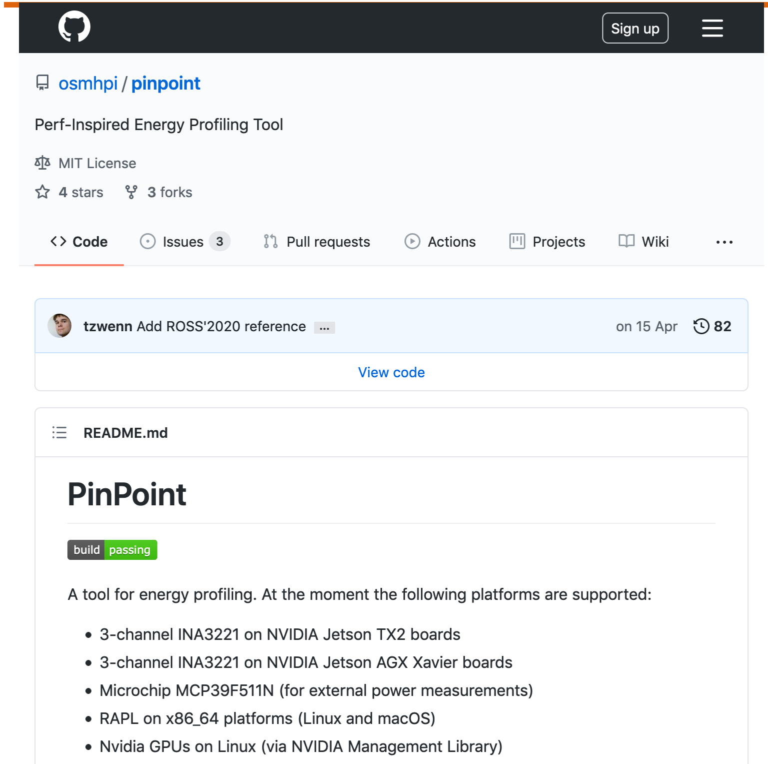
- report averaged power draw, voltage and current
- estimated 5% sample accuracy, 20 Hz sampling frequency
- I²C exposed via Linux sysfs-interface at `/sys/bus/i2c/drivers/ina3221x/*/iio_device/in_power`

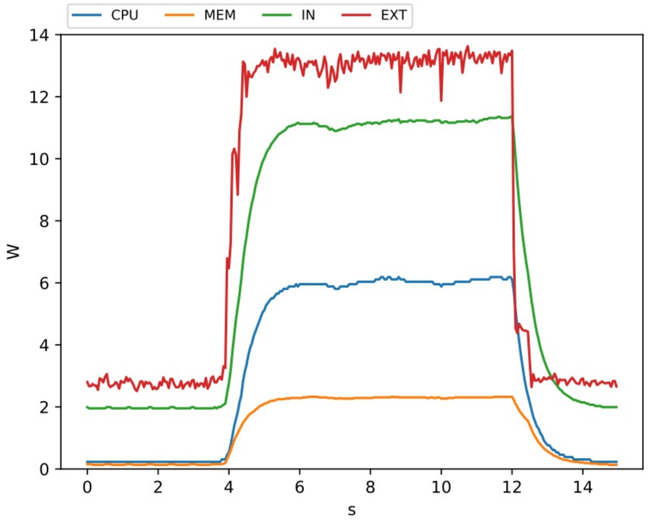


For all other NVidia GPUs:
Check out the NVidia Management Library (`nvml`) or the `nvidia-smi` tool.

[11] NVIDIA. Jetson TX2 Series Thermal Design Guide. Acc. 2020-08-25.
http://developer.nvidia.com/embedded/dlc/jetson-tx2-series-thermal-design-guide

Shameless Self-Plug: PinPoint





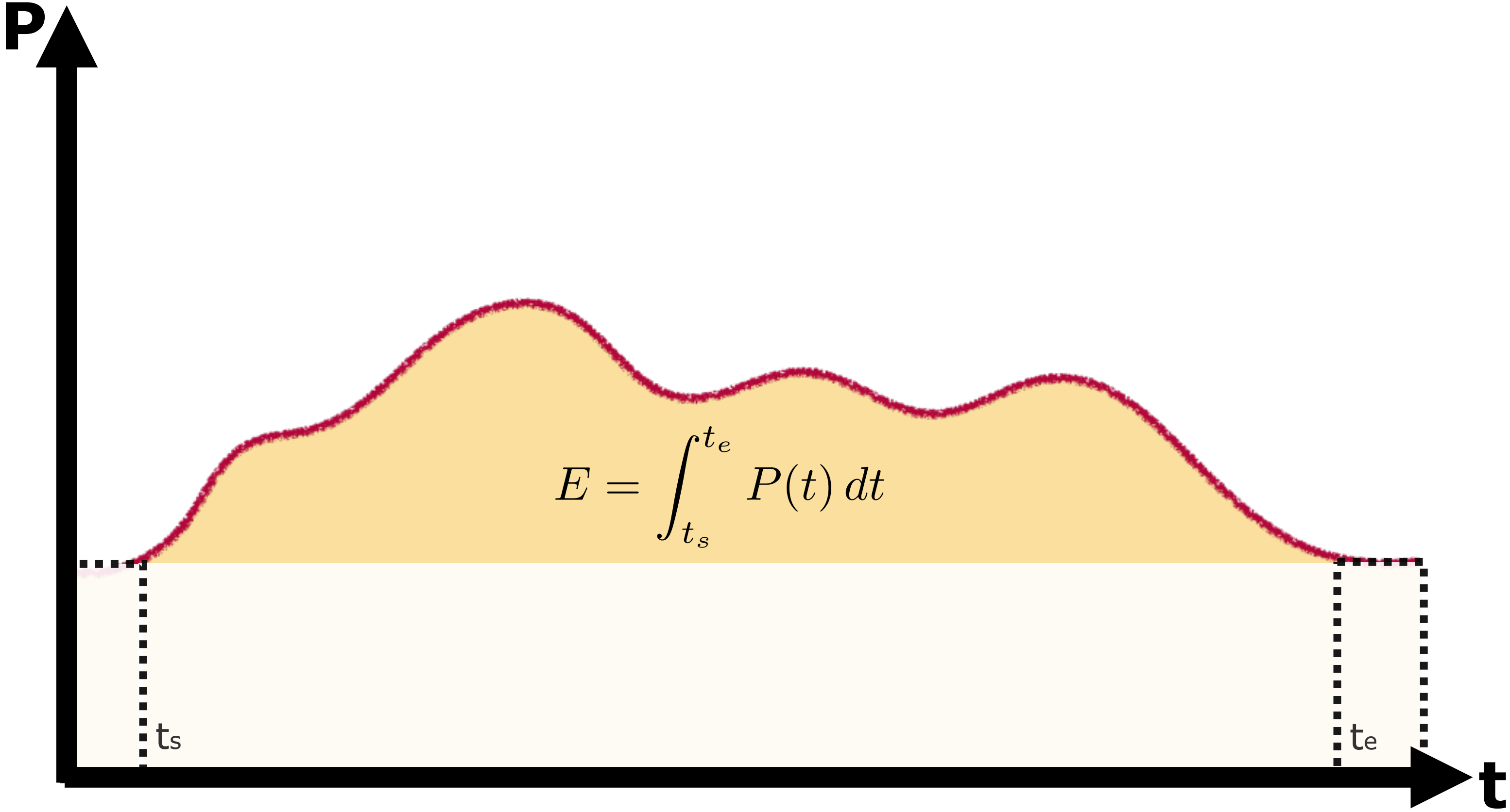
https://github.com/osmhpi/pinpoint

ParProg21 E1 Energy-Aware Computing

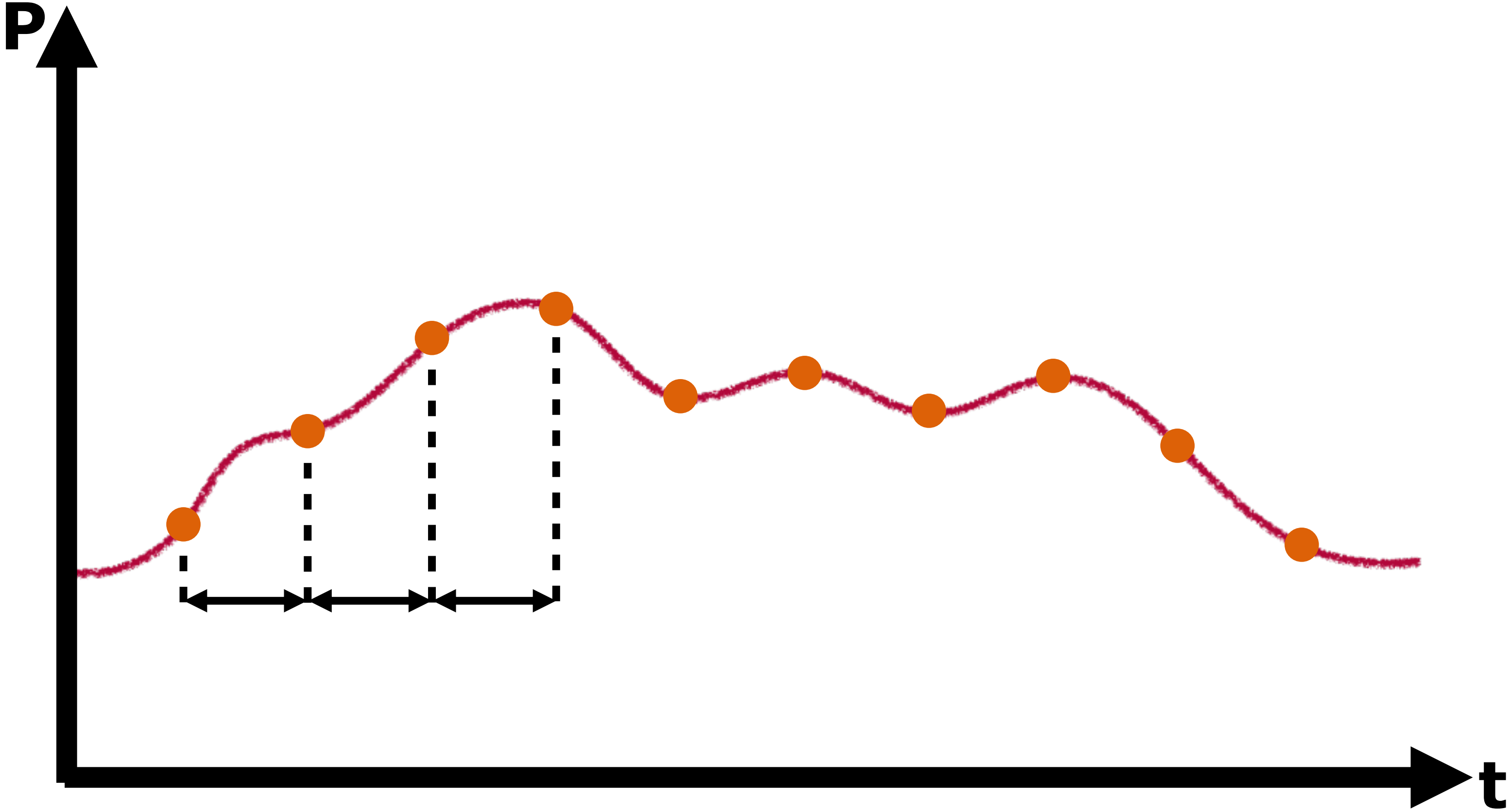
Sven Köhler

[12] Köhler, S., Herzog, B., Hönig, T., Wenzel, L., Plauth, M., Nolte, J., Polze, A., & Schröder-Preikschat, W. (2020, November). Pinpoint the Joules: Unifying Runtime-Support for Energy Measurements on Heterogeneous Systems. In *2020 IEEE/ACM International Workshop on Runtime and Operating Systems for Supercomputers (ROSS)* (pp. 31-40). IEEE.

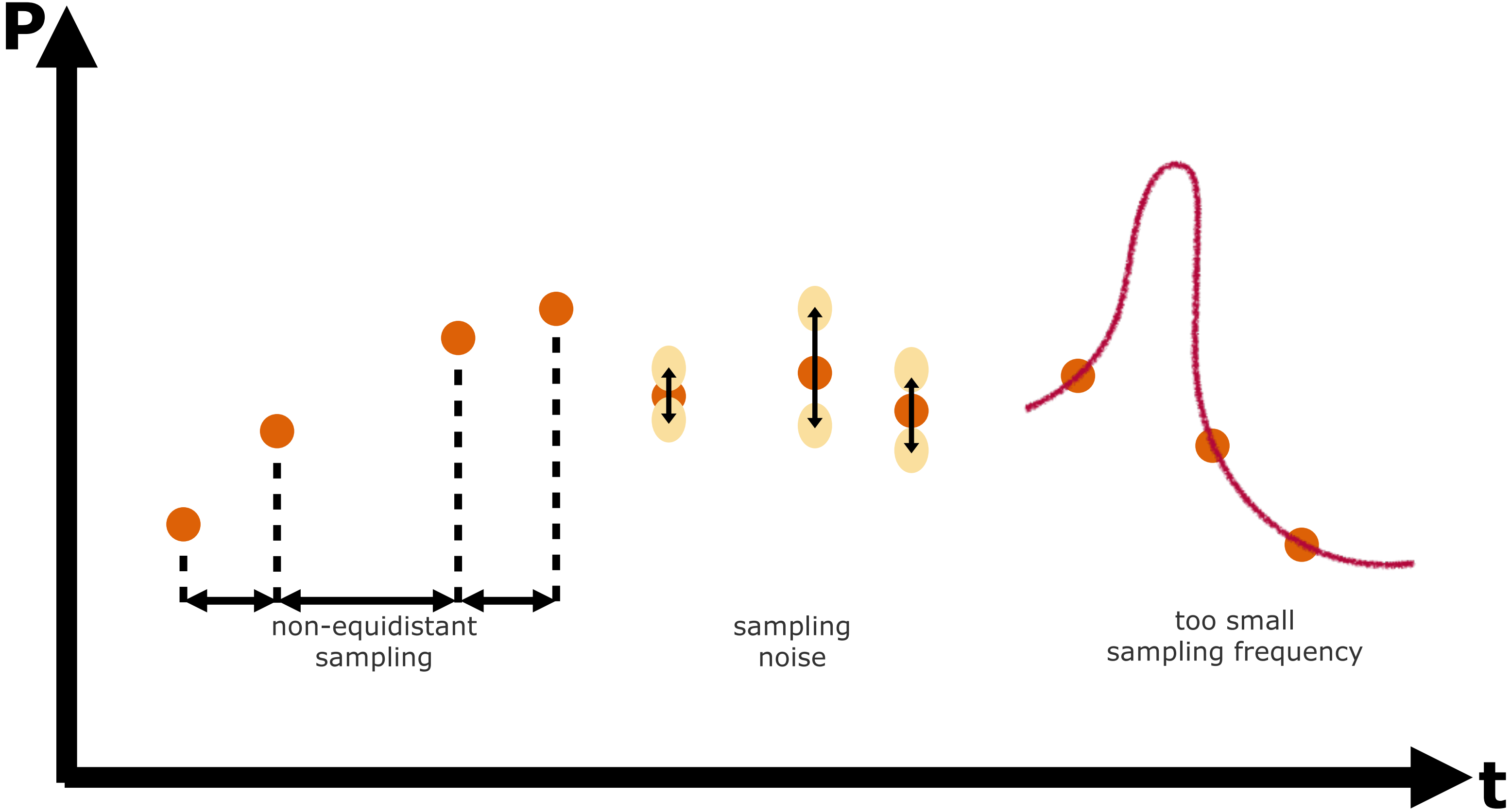
Be Careful What You Measure



Be Careful What You Measure



Be Careful What You Measure



ParProg21 E1
Energy-Aware
Computing

Sven Köhler

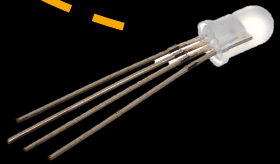
Extended and Composite Metrics

- Power and energy demand are insufficient metrics
- Other system characteristics (e.g., **performance** or **latency**) may **differ** strongly, even though **power** or **energy** characteristics are **the same**
- Extended metrics combine basic metrics (power/energy demand) with additional system properties like execution time:
  - **Power-Delay Product (PDP)**: $P_{avg} \cdot t$
    (approximates energy per switching event, good for fixed voltage)
  - **Energy-Delay Product (EDP)**: $E \cdot t \simeq P_{avg} \cdot t \cdot t$
    (equal weight for changes of energy demand and performance, but misleading metric for systems with dynamic voltage scaling[13])
  - **Energy-Delay-Squared Product (ED²P)**: $EDP \cdot t$
    (good for fixed micro-architecture with dynamic voltage scaling[14])

[13] Horowitz, M., Indermaur, T., Gonzalez, R.: Low-power digital design.In: Proceedings of 1994 IEEE Symposium on Low Power Electronics, 1994, S. 8–11
[14] Brooks, D. M., Bose, P., Schuster, S. E., Jacobson, H., Kudva, P. N., Buyuktosunoglu, A., Wellman, J., Zyuban, V., Gupta, M., Cook, P. W.: Power-aware microarchitecture: design and modeling challenges for next-generation microprocessors. In: IEEE Micro 20 (2000), Nov, Nr. 6, S. 26–44

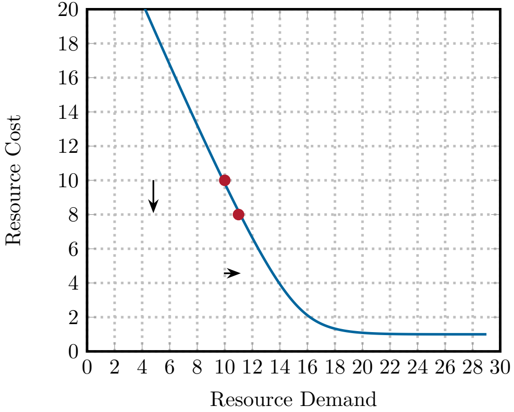
4 Closing Remarks

Rebound Effect

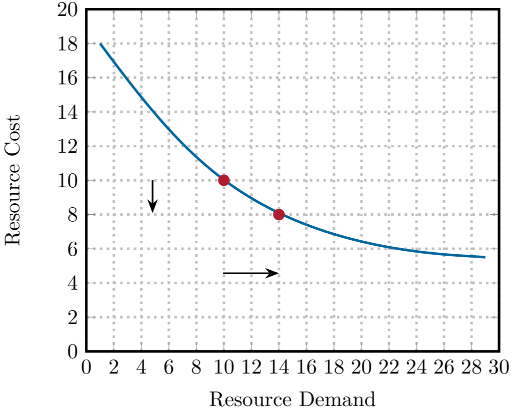


Jevons Paradox

Technological progress that increases the efficiency with which a resource is used tends to increase (rather than decrease) the rate of consumption of that resource.



increase efficiency by 20%
⇒ increase demand by 10%
(Jevons Paradox does **not apply**)



increase efficiency by 20%
⇒ increase demand by 40%
(Jevons Paradox **does apply**)

[15] Jevons, W. S. (1866). *The Coal Question; An Inquiry Concerning the Progress of the Nation, and the Probable Exhaustion of our Coal-Mines.* Macmillan & Co. London

And now for a break and
a glass of water*.

*or drink of your choice