Zurich University of Applied Sciences
Department School of Engineering
Institute of Computer Science

SPECIALIZATION PROJECT 2

Title

Author:
Caspar Wackerle

Supervisors:
Prof. Dr. Thomas Bohnert
Christof Marti

Submitted on
July 31, 2025

Study program:
Computer Science, M.Sc.

Imprint

Project: Specialization Project 2
Title: Title
Author: Caspar Wackerle
Date: July 31, 2025
Keywords: energy efficiency, cloud, kubernetes

Study program:
Computer Science, M.Sc.
Zurich University of Applied Sciences

Supervisor 1:
Prof. Dr. Thomas Bohnert
Zurich University of Applied Sciences
Email: thomas.michael.bohnert@zhaw.ch
Web: Link

Supervisor 2:
Christof Marti
Zurich University of Applied Sciences
Email: christof.marti@zhaw.ch
Web: Link

Abstract

Abstract

The accompanying source code for this thesis, including all deployment and automation scripts, is available in the **PowerStack**[1] repository on GitHub.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Context

1.1 Introduction and Context

1.1.1 Cloud Computing and its impact on the global energy challenge

Global energy consumption is rising at an alarming pace, driven in part by the accelerating digital transformation of society. A significant share of this growth comes from data centers, which form the physical backbone of cloud computing. While the cloud offers substantial efficiency gains through resource sharing and dynamic scaling, its aggregate energy footprint is growing rapidly. While data center accounted for around 1.5% (around 415 TWh) of the worlds electricity consumption in 2024, they are set to more than double by 2030[2]. That is slightly more than Japans's current electricity consumption today.

This increase is fueled by the rising demand for compute-heavy workloads such as artificial intelligence, large-scale data processing, and real-time services. Meanwhile, traditional drivers of efficiency—such as Moore's law and Dennard scaling—are slowing down[3][4]. Improvements in data center infrastructure, like cooling and power delivery, have helped reduce energy intensity per operation[5], but these gains are approaching diminishing returns. As a result, total data center energy use is expected to grow faster than before, even as efficiency per unit of compute continues to improve more slowly[6].

1.1.2 Rise of the Container

Containers have become a core abstraction in modern computing, enabling lightweight, fast, and scalable deployment of applications. Compared to virtual machines, containers impose less overhead, start faster, and support finer-grained resource control. As such, they are widely used in microservice architectures and cloud-native environments[7].

This trend is amplified by the growing popularity of Container-as-a-Service (CaaS) platforms, where containerized workloads are scheduled and managed at high density on shared infrastructure. Kubernetes has become the de facto orchestration tool for managing such workloads at scale. While containers are inherently more energy-efficient than virtual machines in many scenarios[8], their widespread use presents a new challenge: understanding and attributing their energy consumption accurately.

1.1.3 Container Energy Consumption Measurement Challenges

Energy consumption in containerized systems is inherently hard to measure due to the abstraction layers involved. Tools like RAPL (Running Average Power Limit) expose component-level energy metrics on modern Intel and AMD CPUs, but this information is not accessible from within containers or virtual machines. In public cloud environments, such telemetry is either not exposed or aggregated at coarse granularity, making direct measurement infeasible.

Containers further complicate attribution: because they share the kernel and hardware resources, it is difficult to isolate the energy impact of one container from another. Only indirect metrics—such as CPU time, memory usage, or performance counters—are available, and even these may be incomplete or noisy depending on system configuration and workload behavior. Various tools exist that attempt to model container power usage based on these inputs, but rarely are their produced metrics transistent and verified.

1.1.4 Problem Definition

The growing importance of containers in cloud environments, combined with the difficulty of directly measuring their energy usage, motivates this work. In particular, this thesis investigates the questions:

Which metrics and models allow for reliable power estimation and attribution in containerized systems?

How can existing container energy consumption estimation tools be validated?

To answer this, the study explores methods of measuring server energy consumption, analyzes container workload metrics, and evaluates modeling techniques that aim to bridge the gap between raw energy data and container-level attribution. The focus is on bare-metal Kubernetes environments, where full system observability allows for deeper analysis and model validation, serving as a foundation for future energy-aware cloud architectures.

1.1.5 Context of this thesis

Chapter 2

Related work

2.1 Energy consumption of computer systems

2.2

2.3 Estimating Container energy consumption

Chapter 3

Related work

3.0.1 Intel RAPL

Intel Running Average Power Level (RAPL) is a Power Monitoring Counter (PMC)-based feature introduced by Intel in their Sandy Bridge processors and provides a way to monitor and control the energy consumption of various components within their processor package[9]. The processor is divided into different power domains or "planes", representing specific components. These domains typically include CPU cores, integrated graphics and DRAM. RAPL provides hardware counters and interfaces to read the energy consumption (and set power limits) for each domain. The energy consumption is measured in terms of processor-specific "energy units" (e.g. 61μJ for Haswell and Skylake processors). These counters are updated approximately every millisecond. An adaptation of RAPL for AMD processors uses largely the same mechansms and the same MSR interface[10], although it provides less information than Intel's RAPL[11], providing no DRAM energy consumption.

Intel RAPL has been used extensively in research to measure energy consumption[12] despite some objections about its accuracy, which will be discussed sections 3.0.1 and 3.0.1. The general concencus is that RAPL is *good enough* for most scientific work in the field of server energy consumption and efficiency. As Raffin et at[13] point out, it is mostly used *like a black box without deep knowledge of its behavior*, resulting in implementation mistakes. For this reason, the next section 3.0.1 presents an overview of the RAPL fundamentals. Finally, section 3.0.2 discusses the currently available RAPL-based tools.

How RAPL works, and how it's used

[14], [13]

Validation

Since its inception, RAPL has been subject of various validation studies, with the general concensus that it's accuracy could be considered "good enough"[13]. Notable works are Hackenberg et al, that in 2013 found RAPL accurate but missing timestamps[15], and in 2015 noticed a major improvement to RAPL accuracy, after Intel switched from a modeling approach to actual measurements for their Haswell architecture[16]. Desrochers et al concluded in a 2016 RAPL DRAM validation study[17] that DRAM power measurement was reasonably accurate, especially on server-grade CPUs. They also found measurement quality to drop when measuring and idling system.

More recently, Schöne et al found RAPL in the Alder Lake architecture to be generally consistent with external measurements, but missing DRAM measurements and exhibiting lower accuracy in low power scenarios[14]. This is noteworthy because Alder Lake is Intel's first heterogeneous processor, combining two different core architectures from the Core and Atom families (commonly referred to as P-Cores and E-cores) to improve performance and energy efficiency. While this heterogenity can improve performance and energy efficiency, it also increases complexit of scheduling decisions and power saving mechanisms. This complexity adds to the already complex architecture, featuring per-core Dynamic Voltage and frequency Scaling (DVFS), Idle states and Power Limiting / Thermal Proection.

Limitations and issues

Several limitations of RAPL were noticed in various research works. Since RAPL is continually improved by Intel as new Processors are released, some of these issues have since been improved or entirely solved.

- **Register overflow:** The 32-bit register can experience an overflow error[13, 18]. This can be mitigated by sampling more frequently than the register takes to overflow. This interval can be calculated using the following equation:

$$t_{\text{overflow}} = \frac{2^{32} \cdot E_u}{P} \tag{3.1}$$

  Here, E_u is the energy unit used (61μJ for haswell), and P is the power consumption. On a Haswell processor consuming 84W, an overflow would occur every 52 minutes.

- **DRAM Accuracy:** DRAM Accuracy can only reliably be used for the Haswell architecture[17, 18], and may still exibit a constant power offset.

- **Unpredictable Timings:** While the Intel documentation states that the RAPL time unit is 0.976ms, the actual intervals may vary. This is an issue since the measurements do not come with timestamps, making precise measurements difficult[18]. Several coping mechanisms have been used to mitigate this, notably *busypolling* (busypolling the counter for updates, significantly compromizing overhead in terms of time and energy[19]), *supersampling* (lowering the sampling interval, enlarging overhead and occasionaly creating duplicates that need to be filtered[18]), or *high frequency sampling* (*lowering* the sampling rate when the resulting data is still sufficient[20]).

- **Lower idle power accuracy:** When measuring an idling server, RAPL tends to be less accurate[14, 17].

- **Side-channel attacks:** While the update rate of RAPL is usually 1ms, it can get as low as 50 μs for the PP0 domain (processor cores) on desktop processors. This can be used to retrieve processed data in a side channel attack[14, 21]. To mitigate this issue while retaining RAPL functionality, Intel implements a filtering technique via the ENERGY_FILTERING_ENABLE[**Table 2-2**, 22] entry. This filter adds random noise to the reported values. While this does not affect the average power consumption, point measurement power consumption can be affected.

RAPL has several limitations.

- counter overflow of the 32 bit register non atomic register updates "lack of individual core-level measurements"?????? ->fixed later?? in virtualised environments like cloud instances, the RAPL readings may be intercepted or modified by the hypervisor, potentially affecting their accuracy https://projectexigence.eu/green-ict-digest/running-average-power-limit-rapl/ (bad source) - Not all measuremnt methods are equally accurate?? -> [13]

Methods of measurement

Good comparison -> [13] - MSR / perf-events + eBPF / perf-events / powercap

3.0.2 Tools

RAPL-based tools

[23] Rapid and accurate energy models through calibration with IPMI and RAPL [24] Scaphandre [25] JoularJX: jaba-based agent for power monitoring at the code level [26]: KEPLER [27]: "AI power meter": Library to measure energy usage of machine learning programs, uses RAPL for CPU and nvidia-smi for GPU [28] CodeCarbon: Python package, estimates GPU + CPU + RAM: uses pynvml, ram RATIO (3W for 8G) and RAPL [29]: powertop [30]: Green metrics tool: measuring energy and CO2 consumption of software through a software life cycle anslysis (SLCA): Metric providers: RAPL, IPMI, PSU, Docker, Temperature, CPU, ... (sone external devices) [31]: PowerAPI: Python framework for building software-defined power

Appendix A

Appendix Title

Bibliography

[1] Caspar Wackerle. *PowerStack: Automated Kubernetes Deployment for Energy Efficiency Analysis*. GitHub repository. 2025. URL: https://github.com/casparwackerle/PowerStack.

[2] International Energy Agency. *Energy and AI*. Licence: CC BY 4.0. Paris, 2025. URL: https://www.iea.org/reports/energy-and-ai.

[3] Ryan Smith. *Intel's CEO Says Moore's Law Is Slowing to a Three-Year Cadence — But It's Not Dead Yet*. Accessed: 2025-04-14. 2023. URL: https://www.tomshardware.com/tech-industry/semiconductors/intels-ceo-says-moores-law-is-slowing-to-a-three-year-cadence-but-its-not-dead-yet.

[4] Martin Keegan. *The End of Dennard Scaling*. Accessed: 2025-04-14. 2013. URL: https://cartesianproduct.wordpress.com/2013/04/15/the-end-of-dennard-scaling/.

[5] Uptime Institute. *Global PUEs – Are They Going Anywhere?* Accessed: 2025-04-14. 2023. URL: https://journal.uptimeinstitute.com/global-pues-are-they-going-anywhere/.

[6] Eric Masanet et al. "Recalibrating global data center energy-use estimates". In: *Science* 367.6481 (2020), pp. 984–986. DOI: 10.1126/science.aba3758. eprint: https://www.science.org/doi/pdf/10.1126/science.aba3758. URL: https://www.science.org/doi/abs/10.1126/science.aba3758.

[7] Amit M Potdar et al. *Performance Evaluation of Docker Container and Virtual Machine*. 2020. DOI: https://doi.org/10.1016/j.procs.2020.04.152.. URL: https://www.sciencedirect.com/science/article/pii/S1877050920311315.

[8] Roberto Morabito. *Power Consumption of Virtualization Technologies: An Empirical Investigation*. 2015. DOI: 10.1109/UCC.2015.93.

[9] Project Exigence. *Running Average Power Limit (RAPL)*. https://projectexigence.eu/green-ict-digest/running-average-power-limit-rapl/. Accessed April 2025. n.d.

[10] AMD. *amd_energy: AMD Energy Driver*. Accessed: 2025-04-28. 2023. URL: https://github.com/amd/amd_energy.

[11] Robert Schöne et al. "Energy efficiency aspects of the AMD Zen 2 architecture". In: *2021 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE. 2021, pp. 562–571.

[12] Tom Kennes. "Measuring IT carbon footprint: What is the current status actually?" In: *arXiv preprint arXiv:2306.10049* (2023).

[13] Guillaume Raffin and Denis Trystram. "Dissecting the software-based measurement of CPU energy consumption: a comparative analysis". In: *IEEE Transactions on Parallel and Distributed Systems* (2024).

[14] Robert Schöne et al. "Energy efficiency features of the intel alder lake architecture". In: *Proceedings of the 15th ACM/SPEC International Conference on Performance Engineering*. 2024, pp. 95–106.

[15] Daniel Hackenberg et al. "Power measurement techniques on standard compute nodes: A quantitative comparison". In: *2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. IEEE. 2013, pp. 194–204.

[16] Daniel Hackenberg et al. "An energy efficiency feature survey of the intel haswell processor". In: *2015 IEEE international parallel and distributed processing symposium workshop*. IEEE. 2015, pp. 896–904.

[17] Spencer Desrochers, Chad Paradis, and Vincent M Weaver. "A validation of DRAM RAPL power measurements". In: *Proceedings of the Second International Symposium on Memory Systems*. 2016, pp. 455–470.

[18] Kashif Nizam Khan et al. "Rapl in action: Experiences in using rapl for power measurements". In: *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)* 3.2 (2018), pp. 1–26.

[19] Marcus Hähnel et al. "Measuring energy consumption for short code paths using RAPL". In: *ACM SIGMETRICS Performance Evaluation Review* 40.3 (2012), pp. 13–17.

[20] Harald Servat et al. "Detailed and simultaneous power and performance analysis". In: *Concurrency and Computation: Practice and Experience* 28.2 (2016), pp. 252–273.

[21] Moritz Lipp et al. "PLATYPUS: Software-based power side-channel attacks on x86". In: *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2021, pp. 355–371.

[22] Intel Corporation. *Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 4: Model-Specific Registers*. Tech. rep. 335592-081US. Accessed 2025-04-28. Intel Corporation, Sept. 2023. URL: https://cdrdv2.intel.com/v1/dl/getContent/671098.

[23] Richard Kavanagh and Karim Djemame. "Rapid and accurate energy models through calibration with IPMI and RAPL". In: *Concurrency and Computation: Practice and Experience* 31.13 (2019), e5124.

[24] Hubblo-org. *Scaphandre Documentation*. Accessed: 2025-04-28. 2024. URL: https://github.com/hubblo-org/scaphandre-documentation.

[25] JoularJX Contributors. *JoularJX: Energy profiling agent for Java applications*. Accessed: 2025-04-28. 2023. URL: https://github.com/joular/joularjx.

[26] Inc. Meta Platforms. *Kepler: Kubernetes-based power and energy estimation framework*. Accessed: 2025-04-28. 2023. URL: https://github.com/sustainable-computing-io/kepler.

[27] GreenAI-UPPA. *AI PowerMeter: A Tool to Estimate the Energy Consumption of AI Workloads*. Accessed: 2025-04-28. 2023. URL: https://greenai-uppa.github.io/AIPowerMeter/.

[28] MLCO2. *CodeCarbon: Track emissions from your computing*. Accessed: 2025-04-28. 2023. URL: https://github.com/mlco2/codecarbon.

[29] Intel Corporation. *PowerTOP: Linux tool to diagnose issues with power consumption and power management*. Accessed: 2025-04-28. 2023. URL: https://github.com/fenrus75/powertop.

[30] Arne Tarara. *Green Coding Documentation*. Accessed: 2025-04-28. 2023. URL: https://github.com/green-coding-solutions/green-metrics-tool.

[31] Guillaume Fieni et al. "PowerAPI: A Python framework for building software-defined power meters". In: *Journal of Open Source Software* 9.98 (2024), p. 6670.