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

Knowing the energy consumed by a container on a server is the essential elemenent to a container-level energy efficiency assessment of both the container itself, as well as the environment surrounding it. An accurate energy consumption estimation is therefore required to validate and improve any potential energy efficiency improvements of a container enviromnent, from kubernetes system components (e.g. Kubernetes Schedulers) to the containers themselves.

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:

Question 1: Which metrics and models allow for reliable container-level power estimation?

Question 2: How should a software-based container energy consumption estimation tool be implemented?

Question 3: How can existng container energy consumption estimation tools be validated?

To answer these questions, this 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

This thesis is part of the Master's program in Computer Science at the Zurich University of Applied Sciences (ZHAW) and represents the second of two specialization projects ("VTs"). The preceding project (VT1) focused on the practical implementation of a test environment for energy efficiency research in Kubernetes clusters. This thesis (VT2) is meant to explore theoretical and methodolocical aspects of container energy consumption measurements in detail.

Furhtermore, this thesis builds upon prior works focused on performance optimization and energy measurement. EVA1 covered topics such as operating system tools, statistics, and eBPF, while EVA2 explored energy measurement in computer systems, covering hardware, firmware, and software aspects. These foundational topics provide the basis for the current thesis but will not be revisited in detail.

1.1.6 Use of AI Tools

During the writing of this thesis, *ChatGPT*[9] (Version 4o, OpenAI, 2025) was used as an auxiliary tool to enhance efficiency in documentation and technical writing. Specifically, it assisted in:

- Structuring and improving documentation clarity.
- Beautifying and formatting smaller code snippets.
- Assisting in LaTeX syntax corrections and debugging.

All AI-generated content was critically reviewed, edited, and adapted to fit the specific context of this thesis. **ChatGPT was not used for literature research, conceptual development, methodology design, or analytical reasoning.** The core ideas, analysis, and implementation details were developed independently.

1.1.7 Project Repository

All code, configurations, and automation scripts developed for this thesis are publicly available in the PowerStack[1] repository on GitHub. The repository contains Ansible playbooks for automated deployment, Kubernetes configurations, monitoring stack setups, and benchmarking scripts. This allows for full reproducibility of the test environment and facilitates further research or adaptation for similar projects.

Chapter 2

State of the Art and Related Research

2.1 Energy consumption measurement and efficiency on data center level

Energy consumption and efficiency on a data center level has been well-studied to the point where various Literature reviews were published[10][11]. The bigger part of this research is focused on the data center infrasturture (cooling and power), and with good reason, as the data center infrastructure is responsible for a large part of the energy consumption. While a large number of coarse-, medium- and fine-grained metrics for data center energy consumption exist, most data center operators have focused on improving coarse-grained mertrics (especially the *Power Utilization Effectiveness, PUE*) with improvements to infrastructure. This has resulted in a PUE of 1.1 or lower in some cases[5]. Meanwhile, server energy efficiency has substantially improved, especially for parial load and idle power[12]. This has allowed data center operators to improve energy efficiency by simply installing more efficient cooling and power systems and servers. Fine-grained metrics such as server component utilization rates or speed were generally not used in the context of energy efficiency, but rather as performance metrics to ensure customer satisfaction.

2.2 Energy consumption measurement on a server level

As a result of the energy efficiency improvements of both data center infrastructure and server hardware mentioned in the previous section, a shift has started towards evaluating the actual server load energy efficiency. Efficiency gains on this level compound into further gains at the data center level. The method of resource-sharing of modern cloud computing (and especially the use of containers) have created great opportunities for server workload optimitation for energy efficiency, which in turn require power consumption measurements for evaluation. In the context of containers on multi-core processors, measuring the energy consumption of the entire server is insufficient, since it does not allow the attribution of consumped energy to specific containers or processes. While component-level power measurements provide finer measurements that could theoretically be modelled to display container energy consumption, they drastically raise the complexity for a number of reasons:

- Component-level energy consumption measurement without external tools is far from easy. While some components provide estimation models (e.g. Intel RAPL or Nvidia SMI), others can only be estimated using their performance metrics. This will invariably lead to large measurement uncertainties, especially with the component hardware differences between generations and manufacturers.
- The problem of attributing measured or estimated energy consumption to individual containers is in itself non-trivial: It not only requires a fine-grained time synchronization of energy consumption and used container resources due to the fast-switching nature most server components during any sort of multi-tasking.
- A deep understanding of dynamic or static energy consumption is required: Depending on the energy consumption attribution model, a container might not only account the energy it actively used, but potentially also account for a fraction of the energy consumed for any shared overhead such as shared hardware components, or system resources (such as the Kubernetes system architecure). This idea can be further extended: containers could potentially be penalized for any unused server resources, as these unused capacity still consume energy. These different attribution models lead to a larger debate about the goals of the measurements.
- Any server-level power models used to estimate the relation of individual component energy consumption suffers from the varity of different server configurations due to server specialization, such as Storage-, GPU-, or Memory-optimized servers.
-

The following sections of this chapter aim to present the current state-of-the-art in the various fields of research of the problem domains listed above.

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx here, a sentence will introduce the subsections:

- Hardware components: CPU / RAM / SoC / GPU / ...
-

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Finally, section xxxxxxxxxxxxxxxxxxxxxxxxxxxx aims to give an overview of the currently existing implementations of software-based container-level energy consumption estimation.

2.3 Overview of Power Data collection

In a systematic review cloud servers power models, Lin et al[13] categorize power collection methods into 4 categories:

TABLE 2.1: Comparison of power collection methods for cloud servers

Key	Value	Description	Deployment Difficulty	Data Granularity	Data Credibility
Based on instruments	Installation of extra devices	Bare-metal machines	Easy	Machine Level	Very high
Based on dedicated aquisition system	Specialized systems	Specified models of machines	Difficult	Machine or component-level	High
Based on software monitoring	Build-in power models	Bare-metal and virtual servers	Moderate	Machine, component, or VM level	Fair
Based on simulation	System simulation	Machine, component or VM level	Easy	Machine, component, or VM level	Low

2.3.1 Instrument-based power data aquisition

Instrument-based Data collection aquisition produces the hightest data credibility at a low granularity: These devices, installed externally (measuring the power supplied to the PDU) or internally (measuring the power flow between the PDU and motherboard) have been the source of information for a number of studies. The approach to simply measure electric power at convenient hardware locations using dedicated equipment can of course be extended to provide additional granularity: For example, Desrocher et al[14] custom-created a DIMM extender custom-fitted with Hall-sensor resistors and a linux measurement utility to measure power consumed by a DIMM memory module at 1kHz sampling rate using a *WattsAppPro?* power meter and a *Measurment Computing USB.1208FS-Plus* data aquisition board.

This of course highlights a fundamental truth of instrument-based data collection: While it is possible to implement a measuring solution that provides high-granular and high-sampling rate power data, it is paired with an immense effort since solutions like this are not provided off-the-shelf. Unsuprisingly, this is most valuable for benchmarking or validation (Desrochers used this setup to validate Intel RAPL DRAM power estimations on three different systems). However, this methodology is (currently) unsuitable for deployment to data center servers due to its bad scalability and prohibitive costs. Hence, the primary role of instrument-based power data aquisiton is as a benchmarking and validation tool for research and development.

2.3.2 Dedicated Aquisition systems

BMC Devices / IPMI implementations

Some manufacturers have developed specialized power data aquisition systems for their own server products. The baseboard management controller (BMC) is a typical dedicated aquisition system usually integrated with the motherboard, usually as part of the intelligent platform management interface (IPMI)[13]. It can be connected to the system bus, sensors and a number of components to provide power and temparature information about the CPU, memory, LAN port, fan, and the BMC itself. Some comprehensive management systems such as Dell iDRAC or Lenovo xClarity have been further developed to provide high-quality, fine-grained power data due to their close interoperation between system software and underlying hardware. IPMI-implementations are well-established in servers, mainly due to their adminstration capabilities.

In the context of container power consumption estimation, IPMI-implementations occupy an interesting role. In 2016, Kavanagh et al[15] found the accuracy of IMPI power data to be relatively inaccurate when compared with an external power meter, mainly due to the large measurement size of 120 to 180 seconds and the inaccurate assessment of the idle power. They concluded that IMPI power data was still useful when a longer averaging window was used, and the initial datapoints discounted. In a later study, they suggest combining the measurements of IPMI and Intel RAPL (which they find to underestimate the power consumption) for a reasonable approximation of true measurement[16].

Intel RAPL

2.4 Server Power models

In the absence of actual power data, power consumption models can be formulated that essentially map variables (such as CPU, Memory utilization) related to a server's state to its power consumption. Due to the strong correlation between CPU uzilization and server power, a great number of models use CPU metrics as the only indicator of server power. Fan et al[17] proposed a linear interpolation between idle power and full power, which they further refine into a non-linear form, with a parameter γ to be fitted to minimize mean square error. Similar research was done to further reduce error by introducting more complex non-linear models, such as Hsu and Poole[18], who studied the SPECpower_ssj2008-dataset of systems released between December 2007 and August 2010, and suggested the adaptation of two non-linear terms:

$$P_{\text{server}} = \alpha_0 + \alpha_1 u_{\text{cpu}} + \alpha_2 \left(u_{\text{cpu}}\right)^{\gamma_0} + \alpha_3 \left(1 - u_{\text{cpu}}\right)^{\gamma_1} \tag{2.1}$$

While models like these might work well when custom-fitted to specific, multi-purpose servers, they have since been surpassed by the more common approach of modelling server power is to consider it an assembly of its components, such as Song et al[19] propose as:

$$P_{\text{server}} = P_{\text{cpu}} + P_{\text{memory}} + P_{\text{disk}} + P_{\text{NIC}} + C \tag{2.2}$$

, where C denotes the server's base power, which includes the power consumption of other components (regarded as static). This approach can easily be extended to include various other components such as GPUs, FPGAs or other connected components.

....

In a systematic review cloud servers power models, Lin et al[13] state that the common way

2.5 Power data collection

see Lin et al for overview -> instruments / dedicated aquisition system / software monitoring and calculation / simulation

2.5.1 CPU

2.5.2 Memory

2.5.3 Storage

2.5.4 Networking

2.6 Container energy estimation based on hardware power estimation

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] OpenAI. *ChatGPT (Version 4o)*. Used for document generation and formatting. 2025. URL: https://chat.openai.com.

[10] Saiqin Long et al. "A review of energy efficiency evaluation technologies in cloud data centers". In: *Energy and Buildings* 260 (2022), p. 111848.

[11] Chaoqiang Jin et al. "A review of power consumption models of servers in data centers". In: *applied energy* 265 (2020), p. 114806.

[12] Hannes Tröpgen et al. "16 Years of SPEC Power: An Analysis of x86 Energy Efficiency Trends". In: *2024 IEEE International Conference on Cluster Computing Workshops (CLUSTER Workshops)*. IEEE. 2024, pp. 76–80.

[13] Weiwei Lin et al. "A taxonomy and survey of power models and power modeling for cloud servers". In: *ACM Computing Surveys (CSUR)* 53.5 (2020), pp. 1–41.

[14] 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.

[15] Richard Kavanagh, Django Armstrong, and Karim Djemame. "Accuracy of energy model calibration with IPMI". In: *2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*. IEEE. 2016, pp. 648–655.

[16] 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.

[17] Xiaobo Fan, Wolf-Dietrich Weber, and Luiz Andre Barroso. "Power provisioning for a warehouse-sized computer". In: *ACM SIGARCH computer architecture news* 35.2 (2007), pp. 13–23.

[18] Chung-Hsing Hsu and Stephen W Poole. "Power signature analysis of the SPECpower_ssj2008 benchmark". In: *(IEEE ISPASS) IEEE International Symposium on Performance Analysis of Systems and Software*. IEEE. 2011, pp. 227–236.

[19] Shuaiwen Leon Song, Kevin Barker, and Darren Kerbyson. "Unified performance and power modeling of scientific workloads". In: *Proceedings of the 1st International Workshop on Energy Efficient Supercomputing*. 2013, pp. 1–8.