Vrije Universiteit Amsterdam

Universiteit van Amsterdam

Master Thesis

End-to-End Power Model for the Compute Continuum

Author: David Freina (2757359)

1st supervisor:	Dr. Tiziano De Matteis
daily supervisor:	Matthijs Jansen, MSc
2nd reader:	Prof. Dr. Ir. Alexandru Iosup

A thesis submitted in fulfillment of the requirements for the joint UvA-VU Master of Science degree in Computer Science

August 6, 2024

Abstract

In this thesis we explore power measurement and modeling within the compute continuum. Specifically, we examine the power consumption of devices in the compute continuum, spanning from user-facing endpoint devices, like security cameras or smartphones, to cloud infrastructures. In addition, we investigate the role of different network standards, that is, WiFi and Ethernet, in influencing power consumption. We consider the constraints and the practical hurdles coming with the possibility to combine various devices and the variety of components within the devices itself, when creating an overarching model to measure the power consumed by these devices.

A command-line application prototype is developed to analyze power usage across different compute continuum configurations, employing diverse measurement and modeling techniques. The prototype we developed is adaptable to different compute continuum configurations, by having different measurement and modeling techniques. Furthermore, by providing real-time data reporting and timestamps, the prototype allows calculations of energy use of the measured devices as well. The prototype's performance is validated through experiments simulating different configurations of the compute continuum. Results demonstrate that the prototype accurately measures power usage and adapts to various configurations. Additionally, the results highlight the trade-offs between computational power and electrical power.

This study contributes to understanding power usage in extensive computational activities and promotes sustainability in the tech industry by offering tools and methodologies to optimize energy consumption across the compute continuum. All source code, including artifacts for reproducibility, is available at `https://github.com/davidfreina/VU-Thesis-24`.

Contents

List of Figures **iii**

List of Tables **v**

Acronyms **vii**

1 Introduction **1**

1.1 Devices in the Compute Continuum 2

1.2 Energy Consumption in the Compute Continuum 4

1.3 Research Questions and Methodology 5

1.4 Societal Relevance 9

1.5 Thesis Outline 9

2 Background & Related Work **11**

2.1 Power Measurement and Modeling 11

3 Power Consumers in the Compute Continuum **15**

3.1 Classification 15

3.2 Identifying Power Consuming Components 17

3.3 Identifying Measurement & Modeling Methodologies 20

3.4 Conclusion & Future Work 24

4 Designing the Model **27**

4.1 Design Requirements 27

4.2 Measurement Methodologies 29

4.3 Model Design 31

4.4 Conclusion & Future Work 32

5 Implementing a Prototype **35**
5.1 Device Selection 35
5.2 Prototype Design 37
5.3 Model Selection 37
5.4 Implementing Intel RAPL Measurements 40
5.5 Limitations & Future Work 40

6 Evaluation **43**
6.1 Experiment Design & Setup 43
6.2 Experiment Results 45
6.3 Evaluating Intel RAPL 47
6.4 Prototype Compliance with Design Requirements 48
6.5 Limitations & Future Work 51

7 Conclusion **53**
7.1 Contributions 53
7.2 Future Directions 54

References **55**

Appendix **67**

List of Figures

1.1 Reference architecture of the compute continuum 2
1.2 End-to-end model problems 3
1.3 Thesis Outline 9

3.1 Overview of the compute continuum with endpoints, edge servers and cloud infrastructure. 16
3.2 Energy impact of components in the layers of the compute continuum. . . . 16

4.1 Overview of the model 29
4.2 End-to-End power model design 31

5.1 Peak power usage of components for device selection. 36
5.2 Design of the Prototype 37

6.1 Experiment Design 44
6.2 Power usage between experiment runs with 30fps and 5fps. 45
6.3 Energy usage of cloud, edge, and endpoint nodes 47
6.4 Comparison of Kepler and Scaphandre 47
6.5 Validation of Scaphandre, perf, and our custom Intel RAPL power measurement implementation. 49
6.6 Overview of the model and the respective design requirements. 49

LIST OF FIGURES

List of Tables

3.1 Overview of energy using components of endpoint devices in the compute continuum. 18
3.2 Overview of energy using components of edge devices in the compute continuum. 19
3.3 Overview of energy using components of cloud devices in the compute continuum. 20

4.1 Design requirements for the model 28

5.1 Selection of devices for the prototype 37

LIST OF TABLES

Acronyms

AI Artifical Intelligence. 9

APM Average Power Management. 11

FPS frames per second. 39, 44, 46

HW-based hardware-based. 4, 6, 20, 32, 43

ICT Information and communications technology. 1

IoT Internet of Things. 1, 9, 17, 18, 30

MSR model-specific register. 20, 30, 31, 40

PSU Power supply unit. 30

RAPL Running Average Power Limit. 11

RFID Radio Frequency Identification. 17

SW-based software-based. 4, 6, 20, 32

VM virtual machine. 40, 44

1

Introduction

The continuous evolution of computing technologies has propelled modern society into the era of the compute continuum, where the boundaries between endpoints, such as mobile phones, Internet of Things (IoT) devices and smart devices, edge, and cloud computing are increasingly blurred. The concept of a compute continuum refers to a fully integrated computing architecture that spans from endpoints across edge servers to centralized cloud infrastructure as seen in Figure 1.1. While the model only refers to a network of devices it is used with specialized techniques to enhance resource allocation and data processing tailored to specific application requirements including speed, data volume, user interaction, and security. This approach also addresses common trade-offs between the devices in the continuum, such as balancing available resources with the devices' proximity to users, to achieve optimal performance and efficiency.

This integrated ecosystem offers unprecedented computational power and connectivity due to its complex, distributed infrastructure, which combines diverse hardware, software, and user needs. However, the inherent heterogeneity of the compute continuum poses a significant challenge regarding power consumption monitoring. The diversity results in numerous combinations of hard- and software, each with different power needs and profiles, complicating the process of standardizing power measurement across the ecosystem. This highlights the need for a power measurement solution that addresses the unique characteristics of the compute continuum. The urgency to address these challenges is underscored by growing environmental concerns and a push toward sustainability in the Information and communications technology (ICT) sector. The ICT sector is predicted to use 8% in the best-case or 51% in the worst-case scenario of the global energy by 2030 ([1]).

In this thesis we explore and address critical questions regarding power measurement and modeling within the compute continuum.

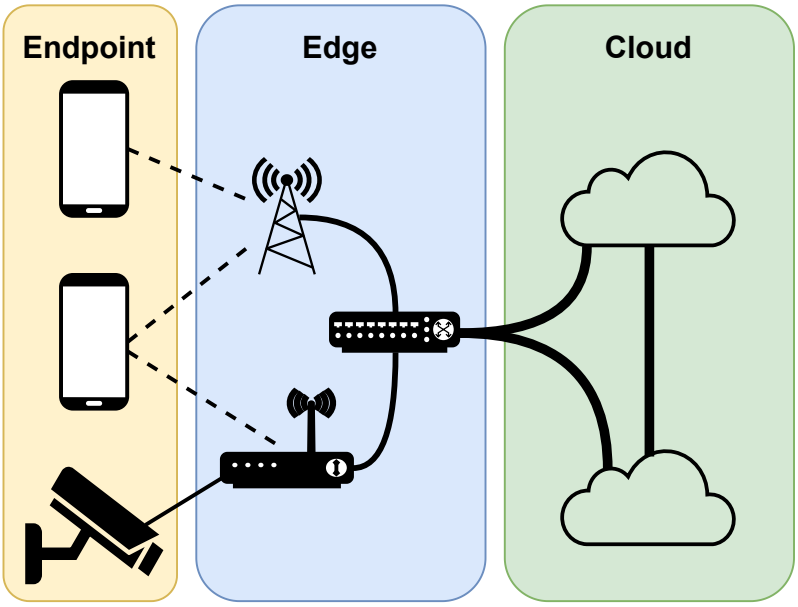


Figure 1.1: Reference architecture of the compute continuum ([2]).

1.1 Devices in the Compute Continuum

The computing continuum is a complex architecture divided into three main layers: endpoint, edge, and cloud. Every layer includes a range of devices, each with its specific features, functionalities, and energy demands.

1.1.1 Endpoint Devices

Endpoint devices are typically the user-facing end of the compute continuum. Examples of these include smartphones, security cameras, and IoT sensors. These devices are often equipped with various features, such as cameras or temperature sensors, as well as different communication modules such as WiFi, Ethernet, and cellular. Endpoint devices are generally characterized by portability and low power consumption, designed to operate with minimal power to extend battery life and reduce energy consumption ([2]).

1.1.2 Edge Devices

Edge devices serve as the intermediary layer in the compute continuum, often bridging the gap between endpoint devices and the cloud. However, they can also take over functions that would have been executed otherwise by the cloud or endpoints themselves. Common examples include Raspberry Pis or Nvidia Jetson platforms. These devices are capable of processing data closer to where it is generated, reducing latency and bandwidth usage.

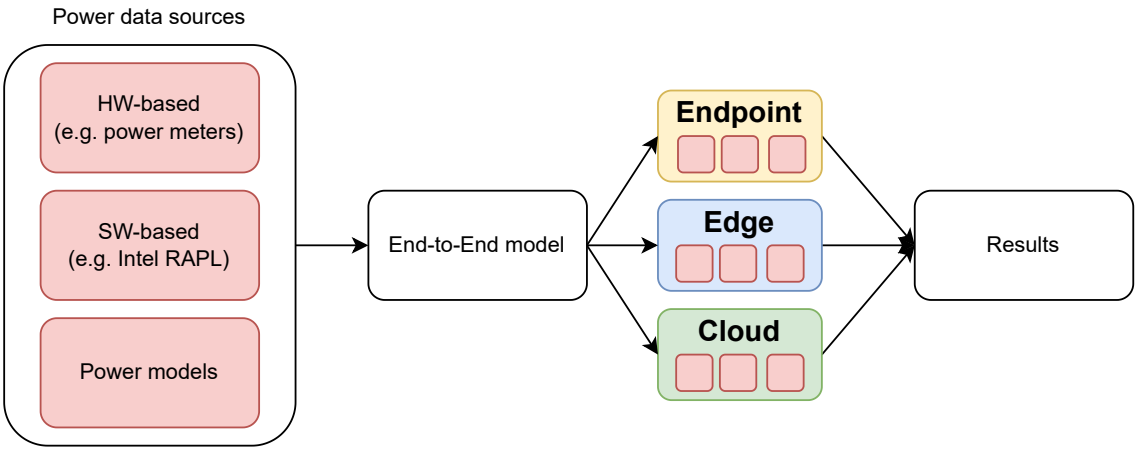


Figure 1.2: End-to-end model problems

Edge devices often have greater computational power than endpoint devices, but are generally still lower-power than cloud infrastructure. They can perform more complex tasks, such as preliminary data analysis and real-time processing, thus offloading some of the computational burdens from the cloud ([2]).

1.1.3 Cloud Devices

At the top of the compute continuum are cloud devices, primarily consisting of server deployments in data centers. These servers provide vast computational resources and storage capabilities that facilitate complex data processing and computing tasks. Cloud-grade servers are characterized by their high power, with CPUs that can consume more energy than e.g., multiple smartphones combined ([3]). Despite their significant power requirements, these devices offer unparalleled performance and scalability, supporting a wide range of applications, from big data analytics to machine learning and artificial intelligence ([2]).

In summary, the **devices within the compute continuum vary widely in terms of features, functions, energy requirements, and components**. From energy-efficient smartphones and IoT sensors at endpoints to high-performance servers in the cloud, each device plays a crucial role in the seamless operation of this layered computing architecture. Our investigation into devices across all layers aims to **identify their energy-consuming components**. This identification is essential because it forms the foundation of our end-to-end power model design (Figure 1.2), ensuring a comprehensive understanding of energy consumption throughout the compute continuum.

1.2 Energy Consumption in the Compute Continuum

Monitoring power consumption is essential for optimizing resource utilization, reducing costs, and minimizing environmental impact. It provides the foundational data needed to implement effective energy management strategies, making it a critical practice in modern computing environments.

Within the compute continuum networking forms the backbone of connectivity and data flow between devices and services, spanning from local networks to extensive cloud infrastructures. Therefore it is a significant consumer of power, as highlighted in previous studies ([1]). We review existing methodologies to measure this power consumption, and include it in our end-to-end model. In addition, we examine the energy consumption of different network standards, that is, WiFi and Ethernet, in influencing power consumption.

In parallel, we focus on the compute components of the devices that populate the compute continuum. These components present a diverse challenge to accurately model their power consumption due to the **sparse support for hardware-based (HW-based) measurement infrastructure and software-based (SW-based) measurement interfaces**.

SW-based SW-based measurements interfaces obtain their measurement data through the energy measurement infrastructure already available built into the hardware [4], [5]).

HW-based HW-based energy measurement systems, in contrast to SW-based approaches, require the use of additional hardware to gather measurement data from computer systems. Within this category, a distinction can be made between systems that utilize commodity hardware and those that require specialized hardware, such as purpose-built measurement boards or FPGA-based technologies ([6]).

Due to this variability in power measurement infrastructure, the development of power models that operate exclusively on accessible data without the need for specific hardware support has been a relevant topic in academia for years (e.g., [7] (2003), [8] (2005), [9] (2010), [10] (2014), [11] (2018)).

Given these constraints and the practical hurdles of comprehensive power measurements, we design our model to use current power models found in the literature when other measurement infrastructures are not available.

1.3 Research Questions and Methodology

From the problems highlighted above, we design a diagram (Figure 1.2) that highlights the areas that this thesis will focus on. Furthermore, this helps us to formulate the ensuing research question: **How do we design an end-to-end power model for the compute continuum?**

Our primary focus is on determining the power usage of the network and compute components of the devices in the compute continuum. But, a full end-to-end model should also include e.g., the networking infrastructure between the layers of the compute continuum as well as additional power consumption not directly related to the devices (e.g., cooling and lighting in data centers). Additionally, devices have components not related to compute and networking like storage, memory, or fans which should also be included. However, due to sparse literature around power measurements of these additional components in the compute continuum we opt to explore only the compute and network parts.

To measure these parts we will use HW/SW-based measurements or interfaces where possible. We make this decision based on the examinations in our literature study [6] where we find that HW-based measurements provide the most accuracy closely followed by SW-based approaches. However, if obtaining measurement data using HW-based/SW-based measurement interfaces is infeasible (e.g., due to a lack of interfaces) we turn to power models. Power models provide us an estimate of power consumption based on some hardware utilization metric.

Answering this question will significantly improve our understanding of energy utilization in the compute continuum by helping us understand where and how energy is consumed with different deployment scenarios. We divide the main research question into several sub-questions to facilitate clear and precise responses.

1.3.1 Research Questions

RQ-1. **What are the power consuming components in the compute continuum?**

Problem Description The compute continuum is composed of three layers with each layer being constructed of multiple, sometimes different, devices. Furthermore, each of these devices is made with different components (e.g, CPUs, GPUs). Therefore, in order for us to analyze the power consumption in the compute continuum the first objective addressed in this thesis is the identification and classification of components responsible

for power consumption within the compute continuum. Understanding the specific power consumers in the network and compute infrastructure of the compute continuum is crucial to developing strategies to measure and model power usage and enhance the sustainability of these systems.

Methodology To address this issue, a top-down design approach is utilized, employing quantitative research methods such as statistical modeling and extensive surveys (**M1**). These techniques are designed to precisely quantify and categorize the power consumption in different components of the computing continuum. Special emphasis is placed on integrating data regarding power usage into a detailed power model.

Contribution The contribution achieved by answering this question lies in the identification of the components required to design a comprehensive power model. The research focuses on these crucial elements with the goal of gaining in-depth understanding of their power usage to enable a focused examination of potential strategies for power measurement. The identification of these components provides an important stage in the development of an all-encompassing power model for the compute continuum that can steer more eco-friendly computing methods throughout the spectrum.

RQ-2. **How can the power use of compute continuum components be accurately measured or modeled?**

Problem Description Based on the insights gained from RQ-1 the core challenge addressed in this phase is the identification of power measurement interfaces and power models applicable to the components. Because the availability of HW-based and SW-based is not clear for every component we first have to identify their availability. Furthermore, we can find multiple power models for every component which creates a confusion about which power model works best for any given component.

Therefore, this identification is essential to evaluate the capabilities to measure power usage of the components identified in RQ-1. The choice of power models is essential to enable the end-to-end power model to accurately include all the components in the various systems of the compute continuum.

Methodology The methodology involves two main steps. Firstly, the abstracted components are organized into a theoretical model structure. This abstraction is crucial in constructing a model that is both comprehensive and understandable, allowing for easier

manipulation and modification in the subsequent phases of prototyping. Secondly, the architecture of the model is constructed. (**M2**) This stage requires decisions on the model's configuration. Decisions made during this stage determine how accurate that model will depict real-world power consumption patterns and can scale and adapt to new insights.

Contribution The design phase of this research contributes significantly by developing a power consumption model that encompasses the components of the network and compute infrastructure of the compute continuum. In addition, the process includes evaluating the support for hardware-based energy measurement within the components, enhancing the accuracy and reliability of the model, where applicable. This structured approach not only advances theoretical understanding, but also provides a solid foundation for practical implementations aimed at optimizing power use across the computing continuum.

RQ-3. **How do we design an end-to-end power model for compute continuum devices?**

Problem Description Designing the end-to-end power model presents a significant challenge. Given the vast amount of different devices and components that are identified in RQ-1 and could be used within the compute continuum it is important to design a model that is as adaptable to all possible configurations. The design needs to reflect the insights gained from answering RQ-2 and combine the model's and power usage measurements that are identified.

Methodology To overcome this problem, design and abstraction (**M2**) will be employed. Based on the insights from RQ-2 and on a set of design requirements an overarching model combining the different, previously identified, power measurement and modeling techniques is designed.

Contribution The abstract power model can serve as the foundation of other implementations for specific configurations. It simplifies the different layers of networking and computation, from endpoint devices to cloud infrastructure, into a more manageable format. It can serve as a base for implementing various power models and evaluating them against each other or to compare different configurations of the compute continuum with one another. Furthermore, it is the first time (known to the authors) that a power model for multiple components and devices across different infrastructures is designed.

RQ-4. **How can we prototype the end-to-end power model to estimate the power consumption in the compute continuum?**

Problem Description The central challenge addressed here is the implementation of the design outlined in RQ-3. This involves developing a tool that can accurately measure and analyze the power of different deployment configurations in the compute continuum. However, due to the large design space identified in the previous research questions it is impossible to implement a prototype for all possible configurations. However, we implement a prototype that is working for one configuration of devices and demonstrate the conceptual idea behind the end-to-end power model.

Methodology The approach focuses on the prototyping (**M2**) and experimental validation (**M3**) phase of the tool. A command-line application is developed, enabling the analysis of power usage of deployments in the compute continuum. The tool is designed to provide real-time estimates of power consumption. It will undergo testing against workload-level benchmarks. This process is designed to evaluate the tool, ensuring that it can reliably measure power consumption under various operational conditions.

Contribution The prototyping and validation phase is crucial as it will reveal any discrepancies between the model's implementation and design, pinpointing areas that may require adjustments. Furthermore, in line with open science principles (**M4**), the tool is developed as open-source software. This openness promotes community contributions and ongoing enhancement, increasing the tool's adaptability and utility. This transparent and collaborative approach advances research in energy-efficient computing and fosters an environment of continuous innovation, contributing significantly to the field's development.

1.3.2 Methodologies

One (or more) of the following methodologies are employed above:

M1 Quantitative research (statistical modeling, simulations, comprehensive surveys) [12], [13];

M2 Design, abstraction, prototyping [14]–[16];

M3 Experimental research, designing appropriate micro and workload-level benchmarks, quantifying a running system prototype [17], [18];

M4 Open-science, open-source software, community building, peer-reviewed scientific publications, reproducible experiments [19]–[22].

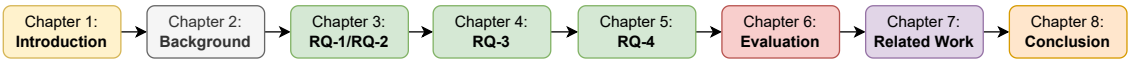


Figure 1.3: Thesis Outline

1.4 Societal Relevance

This thesis addresses the critical and socially relevant issue of energy consumption in the compute continuum. This is a concern that resonates deeply with ongoing global efforts to mitigate climate change and reduce the environmental footprint of the digital age. In response to advances in technologies such as the IoT, Artifical Intelligence (AI), and big data analytics, the demand for computing power increases. The research carried out here is pivotal, as it seeks to develop models and methodologies to measure and optimize energy use on various networking and computing devices. By providing tools to determine the energy usage of components in the compute continuum, this thesis contributes to obtain the knowledge of potential carbon emissions associated with extensive computational activities, and therefore promotes sustainability in the tech industry. Through these contributions, the thesis not only addresses a key technological challenge, but also aligns with societal values and priorities, underscoring the importance of energy-conscious innovations in securing a sustainable future for technology-driven societies.

1.5 Thesis Outline

This thesis is organized as follows (Figure 1.3). First, in the Background & Related Work (Chapter 2), we provide a review of related work, situating our research within the broader context of power measurement and modeling.

Subsequently, Chapter 3 identifies and analyzes the power consumers across different layers of the compute continuum, consisting of endpoint, edge, and cloud devices. This chapter lays the groundwork for understanding the specific power consumers critical for developing our power model.

Chapter 4 discusses the design of our end-to-end power model. We outline the design requirements and describe the various measurement methodologies employed to accurately capture power consumption across the compute continuum.

In Chapter 5, we present the implementation of our prototype. This includes the selection of power models for different components, as well as the implementation of our own Intel RAPL measurements.

Chapter 6 covers the experimental evaluation of our prototype. We detail the experiment design and setup, present the results, and validate our Intel RAPL implementation against other tools. We also assess the compliance of our prototype with the design requirements outlined in Chapter 4.

Finally, Chapter 7 concludes the thesis by summing up the answers to the research questions formulated in the introduction. We summarize our findings, discuss their implications, and suggest directions for future research. Artifacts for reproducing the working environment, including the prototype and experimental setup, are provided in the appendix.

2

Background & Related Work

The exploration of energy consumption within parts of the compute continuum has garnered substantial interest, with numerous studies contributing to the understanding of power usage across the various layers. However, to the best of our knowledge there is no methodology or tool that evaluates a end-to-end power model throughout all the layers of the compute continuum.

The following sections review key contributions, methodologies, and findings that support the research presented in this thesis.

2.1 Power Measurement and Modeling

Several methodologies for power measurement have been explored, including both hardware and software-based approaches. Hardware-based solutions, such as PowerMon/2 [23] and WattProf [24], provide detailed power consumption data at the component level but often face scalability and deployment challenges due to their intrusive nature. Conversely, software-based solutions like Intel RAPL [5] and AMD's Average Power Management (APM) [25] offer accessible and less intrusive means to measure power consumption, leveraging model-specific registers (MSRs) to report energy usage directly from the CPU [26].

Power modeling for the different devices in the layers of the compute continuum has also been studied extensively. The following subsections list various studies conducted to identify power models for different aspects of these devices.

2.1.1 Endpoint Devices

A study conducted by Jung *et al.* [27] (2012) provide an extensive look into power modeling for smartphones. In this study they provide models for CPU, GPS, LCD display, 3G, WiFi including all the required coefficients required to implement these models. They use the models from this study to provide a Android-based energy metering system called AppScope [28]. They follow-up this study in 2017 and additionally provide models for GPU, 2D graphics accelerator, video processor, LTE, OLED display, camera, and audio interface [29]. However, in their second study they do not provide the coefficient values, which makes it much harder to reimplement their models.

A more recent study conducted in 2019 [30] explores the accuracy of artifical neural networks for predicting mobile GPU power data based on performance counters. They are able to show impressive results with a mean relative error of 4.4%. In comparison with a linear regression model they achieve 3.3x more accurate results.

2.1.2 Edge Devices

For edge devices our main focus point are SBCs (single-board computers) (e.g., Raspberry Pi, Nvidia Jetson). Kaup *et al.* [10] (2014) present PowerPi, a power consumption model for the first generation Raspberry Pi. They include CPU, Ethernet, and WiFi consumption in their model. In 2018 [31] they follow up on their study and analyze multiple different SBCs (Raspberry Pi B, Pi 2 B, Pi 3 B, Cubietruck, Odroid C1 & C2) in order to understand how their energy efficiency has progressed throughout the years. They present all of the equations needed to reimplement their models.

Ardito *et al.* [11] create power models for a Raspberry Pi 2 B similar to first study by Kaup *et al.* [10]. In their study they also create a CPU and Ethernet model with the main difference being, that their evaluation shows that it is better to construct a split model rather than a full model for the NIC. They decide on this split model based on the observation of a jump in power usage between 40 and 50Mb/s.

2.1.3 Cloud Devices

In our prototype, CPU power measurement relies on Intel RAPL, a methodology previously explored by Khan *et al.* [32] and Hackenberg *et al.* [26]. These studies delve into the accuracy and granularity of data provided by Intel RAPL. Additionally, other research has validated the accuracy of Intel RAPL's memory power measurement data [33], [34].

Regarding Ethernet energy usage, Reviriego *et al.* [35] and Christensen *et al.* [36] conduct studies that emphasize the energy-saving potential of Energy Efficient Ethernet. To establish this, they first analyze the energy consumption of traditional Ethernet, offering comprehensive data on this aspect.

3

Power Consumers in the Compute Continuum

The compute continuum is a computing architecture that consists of three layers: endpoint, edge, and cloud (Figure 3.1)([2]). Every layer in itself can be composed of different devices, ranging from power- and resource-limited IoT sensors as endpoints to cloud infrastructures with abundance of available resources. To connect the layers of the compute continuum different network standards (like cellular, WiFi, and Ethernet) can be employed.

In accordance with RQ-1, the first step in conceptualizing an end-to-end power model for the compute continuum is to identify the core components that are using energy. This identification is necessary due to the diverse nature of devices and components that can be employed across all of the layers. Therefore, in order to identify these components we first must identify devices across all layers in the model. After identifying these devices, we can analyze their composition of various components and assess what affects their energy consumption (Figure 3.2).

The following sections do not aim to provide a complete overview of all possible devices and sources for energy usage in the compute continuum. They should rather serve as an overview and help to enable further examinations.

3.1 Classification

To establish a standardized notation for classifying the power consumption of various components within the compute continuum, a literature review is conducted. The goal is to identify the maximum power values of different components, which are then used to create a classification system.

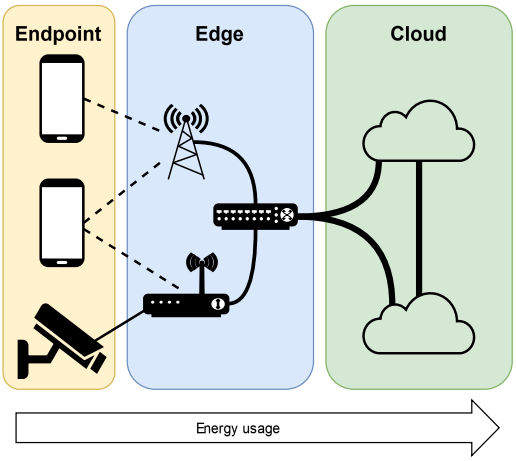


Figure 3.1: Overview of the compute continuum with endpoints, edge servers and cloud infrastructure ([2]).

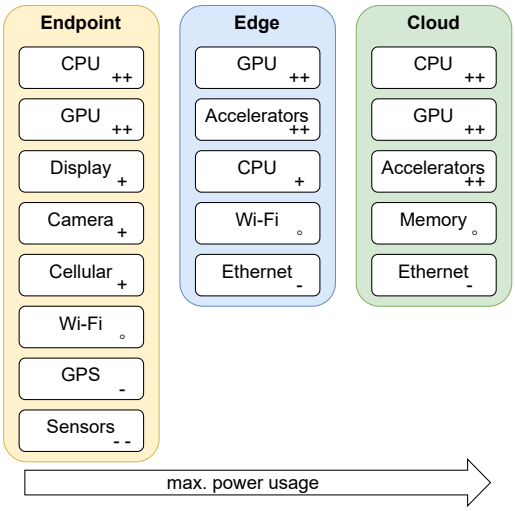


Figure 3.2: Energy impact of components in the layers of the compute continuum.

First, a review of existing literature is performed to gather data on the power consumption of components such as CPUs, GPUs, network interfaces, memory modules, and sensors across endpoint, edge, and cloud devices. Sources included peer-reviewed journals, technical reports, and technical specification documents that provide empirical data on maximum power usage under different operating conditions. The collected data is normalized to a common unit to facilitate comparison. Power consumption values are converted to watts (W) where necessary, ensuring that the power usage of different components could be compared directly, irrespective of their original measurement units.

Next, based on the normalized data, power consumption thresholds are defined to categorize the components. The thresholds are determined using statistical methods to ensure that the classification was both meaningful and reflective of real-world variations. The categories are defined as follows:

- Components with significantly high power consumption, such as high-performance GPUs, are classified as "++";
- Components with above-average power consumption, such as displays or cameras, are classified as "+";
- Components with average power consumption, such as wireless network interfaces and memory modules, are classified as "0";

- Components with below-average power consumption, such as wired network interfaces, are classified as "-";
- Components with significantly low power consumption, such as ultra-low power components, are classified as "–"

It is important to note that the classification are not comparable between different layers of the compute continuum, because the maximum power consumption values would be to far apart (e.g., a cloud-grade CPU $>$ 100W compared to a smartphone CPU with $<$ 10W).

Each component within the compute continuum was then classified according to the established categories. This classification was based on the maximum power values identified during the literature review. For instance, high-performance GPUs, which were found to consume power in the range of hundreds of watts, were classified as "++", while low-power sensors, consuming power in the milliwatt range, were classified as "–".

3.2 Identifying Power Consuming Components

In this section we are examining the power consuming components of the different devices in the layers of the compute continuum (RQ-1). To identify these power-consuming components, a literature survey is conducted. Keywords related to the different layers in the compute continuum are derived from prior studies and literature surveys. Peer-reviewed publications are queried using these keywords to gather data on energy consumption across various devices and components.

Additionally, the SPEC-RG reference architecture for the compute continuum provided a baseline for identifying devices of the compute continuum. The classification of energy-consuming components is based on their frequency of usage and the availability of measurement data in the literature.

3.2.1 Endpoint Devices

Endpoint devices represent the final stage of processing and connecting to users. They can be limited in terms of resources and energy and can be location-independent (e.g., smartphones) ([2]). Endpoint devices are devices such as mobile phones, sensors, Radio Frequency Identification (RFID) tags, and various other IoT devices that form the user-facing endpoint of the compute continuum. These devices are crucial for collecting and transmitting data ([37]).

Component	Energy impact
CPU	++
GPU	++
Display	+
Camera	+
Communication modules	
Cellular	++
Wi-Fi	○
GPS	-
Sensors	- -

Table 3.1: Overview of energy using components of endpoint devices in the compute continuum ([27]–[30], [38]–[41]).

The energy consuming elements within these endpoint components are presented in Table 3.1. We identified sensors, communication modules (like Wi-Fi and cellular chips), processing units (CPUs, GPUs), cameras, and any related displays or user interface features. Other components like Bluetooth modules, memory, or storage devices are omitted due to the lack of data in the available literature regarding their energy usage. The components identified frequently use energy for tasks such as data acquisition and processing or data transmission, which are crucial to their functionality in the compute continuum ([37]).

Multiple studies ([27]–[30], [38], [41]) detail the energy consumption of various components in smartphones. They are able to highlight significant differences in energy usage between the various components. Furthermore, multiple studies ([39], [40], [42], [43]) analyze various IoT devices like Arduino's and low-power sensor ASICs. We use the measurement data of these studies to classify the energy impact of the different components in Table 3.1.

3.2.2 Edge Devices

Edge devices refer to computing units located near the periphery of the network, close to data sources and endpoint devices. They are a fundamental component of the compute continuum, and their use aims to bring computation and data storage closer to the location where it is needed to improve response times and save bandwidth. These devices include edge servers (e.g., Raspberry Pi, Nvidia Jetson) and networking devices that handle preliminary data processing. Their location helps to reduce the need to transmit data back to cloud infrastructure. ([44]–[47]) In terms of design edge and cloud components are the

Component	Energy impact
GPU	++
Accelerators	++
CPU	+
Communication modules	
Wi-Fi	○
Ethernet	-

Table 3.2: Overview of energy using components of edge devices in the compute continuum ([3], [10], [31], [44], [49]–[51]).

same. Therefore, the case for edge devices comes from the aforementioned locality advantage ([2]). With the difference being that edge devices should be able to offload between the cloud and edge as well as different edge devices ([48]).

The energy consuming parts of edge devices are presented in Table 3.2.

We identified their processing units (CPUs, GPUs, Accelerators), and communication modules (Wi-Fi, Ethernet). Other components that are present in the edge devices like storage hardware or memory are omitted due to the lack of data in the available literature regarding their energy usage. GPUs, which are used mainly for their high throughput and parallelization, are particularly known for their high power consumption ([3], [50]). Similarly, specialized hardware accelerators, while offering high performance, also consume significant energy ([3]). In addition, communication and memory components also contribute to the overall energy consumption of edge devices although not as extensively as the processing components ([10], [46], [47], [52]).

3.2.3 Cloud

In the compute continuum, the cloud serves as both a central coordinator and an offload destination, managing resources and scheduling workloads across endpoints, edge, and cloud for effective task allocation. It offers substantial computing and storage capacity for resource-demanding applications that edge devices cannot handle. Typical cloud components include high-performance servers, storage solutions, and sophisticated networking infrastructure, ensuring scalable and reliable computation and data management [2].

Cloud components, such as CPUs, GPUs, memory and storage systems, and cooling systems, are energy-intensive due to their high computational tasks [58]. However, due to missing data about many of these components we limited our research on the components in Table 3.3.

Component	Energy impact
CPU	++
GPU	++
Accelerators	++
Memory	∘
Ethernet	-

Table 3.3: Overview of energy using components of cloud devices in the compute continuum ([3], [23], [24], [53]–[57]).

3.3 Identifying Measurement & Modeling Methodologies

3.3.1 Power Measurement

This section provides a detailed overview of available power measurement methodologies. The methodologies are categorized into two primary types: SW-based and HW-based measurement systems.

3.3.1.1 Software-Based Energy Measurement Systems

SW-based measurements do not rely on additional hardware but instead utilize the energy measurement infrastructure built into the hardware. These systems use model-specific registers (MSRs) and are present in hardware manufactured by e.g., Intel and AMD. MSRs provide detailed energy usage data directly from the processor. Tools like Intel's Running Average Power Limit (RAPL) and AMD's Average Power Management (APM) access this data to estimate energy usage. NVIDIA's Management Library (NVML) offers similar capabilities for GPUs [6].

3.3.1.2 Hardware-Based Energy Measurement Systems

HW-based systems require additional hardware to gather measurement data and can be divided into two subcategories: commodity hardware and specialized hardware. Systems built on commodity hardware include external power meters and smart Power Distribution Units (PDUs), which are cost-effective and widely used but generally lack the ability to provide fine-grained, component-specific data. Improvements over time have enhanced their accuracy and measurement frequency. Specialized hardware has the advantage of employing custom-built hardware to offer superior measurement capabilities [6]. Examples include:

- PowerMon and PowerMon2 ([23]): These provide in-band monitoring at the component level with high measurement frequencies.
- PowerInsight ([59]): Offers 15 measurement channels and out-of-band data collection.
- WattProf ([24]): Features a monitoring board as a PCIe expansion card with up to 128 measurement channels and high-frequency data collection, using a Xilinx Spartan FPGA for precise monitoring.

3.3.2 Power Modeling

In case of unavailability of the aforementioned measurement systems we can use power modeling as a fallback. Power models generally use one or multiple resource metrics to estimate the power consumption of a component. The sections that follow will highlight the different metrics used by various power models found in the literature. We choose those power models to highlight that power consumption data for one component may not only be modeled with one (or more) specific metrics but rather that it is possible to use different metrics to get to the same result.

3.3.2.1 Endpoint Devices

Sensors According to Martinez *et al.* [43] a model for the energy usage of data acquisition from sensors depends on the category of sensing. They classify the monitoring of sensors as "regular" or "event-driven". For regular sensors the energy per acquired sample as well as the number of samples dedicated the amount of energy used. For event-driven ones the probability of an event occurring is additionally taken into account.

CPU According to the energy consumption values presented by Yoon *et al.* [28], the CPU's power usage is highly dependent on its utilization. The power usage spikes during tasks that require a big amount of processing, while consuming less than a quarter of this spike during idle. Additionally, in their follow-up study Yoon *et al.* [29] clearly demonstrate that the CPU power consumption is also highly dependent on the core count. Furthermore, Jung *et al.* [27] shows that the CPU power usage is also proportional to the CPU frequency (higher frequency = higher power usage).

GPU As evidenced by Yoon *et al.* [29] the GPU energy usage is mainly influenced by its utilization. Furthermore, they show that the GPU clock frequency also plays a vital role in the amount of energy used.

Display Furthermore, the display is identified as a major energy consumer, which has a particularly high energy usage impact when applications keep the screen on for extended periods ([28], [29], [41]).

Camera The camera captures and transfers image data from an image sensor with a specified resolution, affecting its power consumption. The size of the transferred pixels depends on the resolution of the image frame and the number of frames per second (FPS). The power model for the camera is calculated using number of pixels per second, which are acquired from the image sensor and determined of the frame rate and resolution ([29]).

Communication Modules The WiFi module shows variable energy consumption that correlates with packet transmission rates. This is evidenced by the data shown in [29], [42] and the model presented in [9].

The components that most impact energy (CPU, GPU) of modern smartphones remain under a peak power usage of 10W ([3], [29]). Nevertheless, endpoint devices significantly impact the energy consumption of the compute continuum due to their abundance and constant activity. The energy usage is driven not only by the computational needs, but also by the regular data transmission over the network, a process that also requires considerable energy, especially when using cellular networks ([37]).

3.3.2.2 Edge Devices

CPU Similar to the findings in the previous section the energy usage of the CPU is highly dependent on its utilization, frequency and core count. This is further evidenced by the findings of Kaup *et al.* [10] and Halawa *et al.* [50].

GPU Similar to the CPU Rungsuptaweekoon *et al.* [49] demonstrate that the GPU power consumption is proportional to the utilization and frequency. They conduct their experiments on two different Nvidia Jetson devices. Furthermore, they use different performance profiles and variation of default and max. clock to produce their results. Comparable results are presented by Hanafy *et al.* [44] with a similar Nvidia Jetson platform and varying neural network models.

Accelerators Accelerators, especially TPUs, break the trend previously established by CPUs & GPUs for having a power consumption highly proportional to their utilization. Jouppi *et al.* [60] evaluate that the power consumption of a TPU at 10% utilization is

already 88% of the power consumption it reaches at 100% for a compute-bound task. A non-compute bound task is even worse with 94% of the power consumption at 10% utilization.

Communication Modules Kaup *et al.* [31] create WiFi models for various edge devices based on their bandwidth. Similar to this they and Ardito *et al.* [11] provide models for Ethernet based communication modules also using the bandwidth.

Given their significant power consumption, incorporating edge devices into the model is crucial to accurately assess and manage the total power consumption of the compute continuum. This is particularly important in scenarios where such devices are deployed, like in multi-tenant workloads where many devices perform a significant amount of local processing. ([2])

3.3.2.3 Cloud

CPU Due to their similar nature to CPUs in endpoint and edge devices the power usage of CPUs in cloud devices is also dependent on their utilization, core count and frequency. These dependencies are highlighted by Zhang *et al.* [55].

GPU While studying energy efficient interference on edge devices Hanafy *et al.* [44] provide valuable insights on the power consumption of GPUs in cloud systems as well. They show that the energy use of a GPU is proportional to its utilization and that the power consumption has a linear relationship to the frequency of the GPU.

Accelerator The same accelerators technologies can be used for cloud devices and edge devices. Therefore, the same points as above apply.

Memory We are unable to find any literature on the power consumption of modern memory-storage sandwich architectures. However, Desrochers *et al.* [33] are able to present findings that show a clear correlation between last level cache misses and memory power usage. They obtain this data using Intel RAPL on a Haswell system with DDR3 memory. This pattern is clearly visible in multiple different benchmarks that stress various parts of the system.

Ethernet Contrary to previous discussed approaches, network power consumption can also be modeled using packets per second (pps). Exemplary data for such a approach can be found in [35].

3.4 Conclusion & Future Work

In this chapter, we address two critical research questions concerning the power consumption within the compute continuum. These questions are fundamental to developing a comprehensive understanding necessary for creating an end-to-end power model.

RQ-1: What are the power-consuming components in the compute continuum?

Through a detailed analysis, we identify the components responsible for power consumption across the compute continuum. Our findings reveal that components such as CPUs, GPUs, communication modules, and various sensors are significant contributors to power usage in endpoint, edge, and cloud devices. This identification process is essential as it lays the groundwork for constructing a detailed end-to-end power model. The comprehensive classification of these components provides a necessary foundation for further analysis and model development, ensuring that all major power consumers are accounted for in the subsequent phases of this research.

RQ-2: How can the power use of compute continuum components be accurately measured or modeled?

Building on the insights gained from answering the first research question (RQ-1), we explore various methodologies for measuring and modeling the power consumption of these identified components. We first examine two different measurement technologies: software-based and hardware-based measurements. Because they are not always readily available we also look at power models that can be used as a fallback mechanic. For the power models we identify the relevant resource metrics for each component in each layer of the compute continuum. Furthermore, we show that the power usage of one component cannot only be modeled with one metric but often times it is possible to use different metrics to obtain the same result.

3.4.1 Future Work

Many other components are required to provide a full end-to-end view of the compute continuum. These include but are not limited to: memory systems with high operating frequencies, cooling systems for optimal data center temperatures, storage devices like hard disks and SSDs, and networking equipment. However, due to missing data in the literature we are unable to identify how significant these components are in regard to their energy impact of the compute continuum.

In conclusion, we successfully address both research questions (RQ-1 & RQ-2). By identifying the power-consuming components and establishing robust methodologies for their

measurement and modeling, we have set a solid foundation for the subsequent development of a comprehensive end-to-end power model. The insights gained here will be pivotal in guiding the design and implementation phases discussed in the following chapters

4

Designing the Model

The following sections outline the design requirements and measurement methodologies essential for developing a end-to-end power model for the compute continuum. We define "end-to-end" as encompassing the devices and components (see Chapter 3) directly involved in executing workflows within the compute continuum. This definition explicitly excludes additional hardware, such as data center cooling systems, which do not participate directly in the workflow execution.

Identifying and understanding the requirements and methodologies outlined in the following sections is crucial for making informed decisions about the most suitable approaches for power modeling across diverse deployment scenarios.

4.1 Design Requirements

In this subsection, we outline the set of requirements for the model. This process enables us to understand why specific research directions are pursued, facilitating an informed decision on the most suitable methodology for end-to-end power modeling.

Table 4.1 presents the design requirements and Figure 4.1 shows an overview of the different parts that are relevant for the model. In the subsequent paragraphs, we will discuss the rationale behind these requirements.

Adaptability (DR-1) In the previous Chapter 3 we discuss the diversity of the compute continuum and its devices. To accurately model the power and energy from end-to-end it is necessary to ensure that the model is adaptable to various configurations. Each component outlined in Table 3.1, Table 3.2, and Table 3.3 must be monitored to achieve a

ID	Description
DR-1	The model must be adaptable to different deployment scenarios, with a variety of devices and components being present
DR-2	The model must have sane default values that can be supplemented in the event of missing data
DR-3	The model must use direct measurements where available and fall back to predictive measurements
DR-4	The model must measure power
DR-5	The model must measure in a fixed time interval or give time information about the measurements to allow calculations of energy usage
DR-6	The model must be context aware to model only the components present in the given configuration
DR-7	The model must report the monitoring data in real-time

Table 4.1: Design requirements for the model

end-to-end power model of the compute continuum.

Stability (DR-2) Additionally, it must be ensured that the data required by the predictive power models is available and otherwise sane defaults must be used. These defaults should be obtained either by testing and finding them as described for each model or alternatively (if applicable) values from the literature must be used.

Accuracy (DR-3) To achieve as accurate results as possible the model must use measurements available directly, like RAPL or HW-based measurements, or alternatively fall back to predictive models if the measurement data cannot be gathered directly.

Power Measurement Capability (DR-4) Measuring power instead of energy is a necessary requirement for the model because energy is the integral of power over a period of time. Therefore, it is much harder to measure energy because it requires a additional information (execution time) in advance. However, when measuring power and providing information about time it is trivial to calculate the energy usage.

Timing (DR-5) For continuous or long-term monitoring the model must either work in a fixed time interval or give measurement times to the user to allow calculations of the used energy instead of power.

[1] (Colors indicate affiliation to requirement)

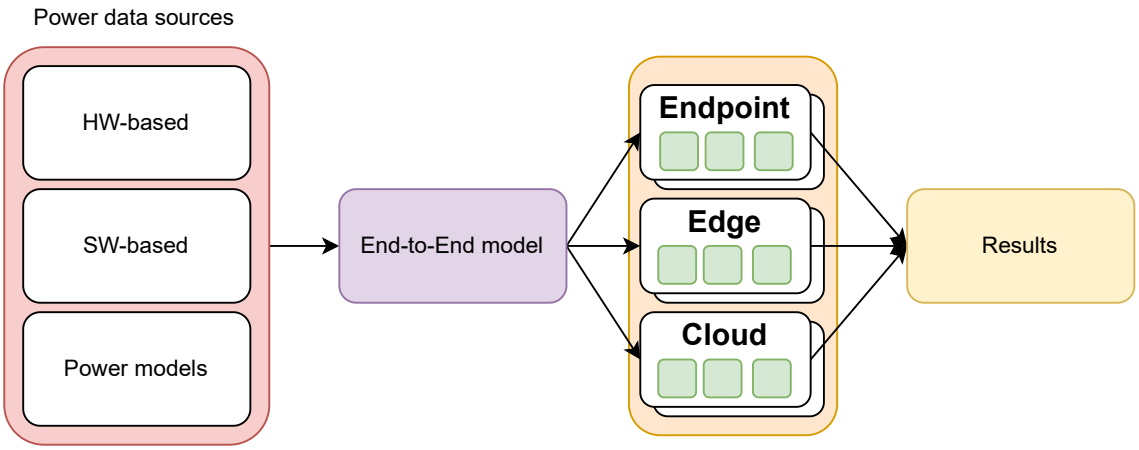


Figure 4.1: Overview of the model[1]

Context Awareness (DR-6) The model should be able to selectively adapt to the components present in the system(s) it is monitoring to prevent unnecessary overhead and possible wrong measurements.

Real-Time Monitoring (DR-7) For the system to be easy to use it should report real-time power values to the user. This enables users to quickly evaluate the power usage of their system.

4.2 Measurement Methodologies

In this section we discuss different methodologies that can be used to measure data from the components of the devices in the compute continuum. Due to the diversity of device and deployment configuration it will not be feasible to decide on one measurement methodology that will be used throughout the end-to-end model. Choosing the right measurement methodology will rather be a component-by-component based decision process and it will highly depend on the requirements of each methodology. The selection of methodologies is based on the taxonomy presented in [61] as well as the taxonomy from our literature survey [6].

4.2.1 Hardware Measurement

While commodity hardware power meters can be used to measure a entire device, specialized hardware is required to enable detailed measurements of specific components. Projects

like PowerMon/2 [23], WattProf [24] or PowerInsight [59] provide component-level granularity power measurements.

External devices are compatible with various bare-metal machines but the compatibility for endpoint and edge devices is not necessarily given. However, where compatible, hardware-based data collection methods provide high accuracy on account of deployment difficulties due to their intrusive nature. Additionally, poor scalability is a challenge, as retrofitting power metering devices on every device in a compute continuum deployment can be difficult depending on the number of applicable devices ([6], [61]).

To use such devices in the end-to-end power model the following requirements must be satisfied:

- **Availability**: While it is possible to retrofit internal measurement devices, they do not come pre-deployed with any device.
- **Compatibility**: The interal measurement devices work based on the ATX specification for Power supply units (PSUs). Therefore, such a solution is not applicable for devices without a compatible power supply (IoT sensors, phones, etc...).
- **Interface**: The measurement devices need to provide an interface to allow obtaining the measurement data on the system itself to be used in the end-to-end power model.

4.2.2 Direct Energy Interface Measurement

The support of manufacturers for integrated energy measurement methodologies starts to appear in 2011 when both Intel and AMD include new measurement capabilities in their respective CPUs [26]. Those measurement interfaces are available to the kernel through MSRs that provide information on the energy usage of the processor. Intel's Running Average Power Limit (RAPL) [5] and AMD's Average Power Management (APM) are two vendor-specific tools that provide the energy usage of the processor to be accessed through a kernel module. These two tools derive their estimates by accessing the data stored in MSRs [26]. A third manufacturer, NVIDIA, also introduces API commands one year later in their NVIDIA Management Library (NVML) [62] that allow users to obtain energy and power measurement data.

Measuring power or energy using direct interfaces is dependent on the availability of MSRs. Therefore, it is up to the vendor of any given component if this measurement methodology is feasible. Furthermore, accessing the interfaces of a kernel module can require administrative access on a machine.

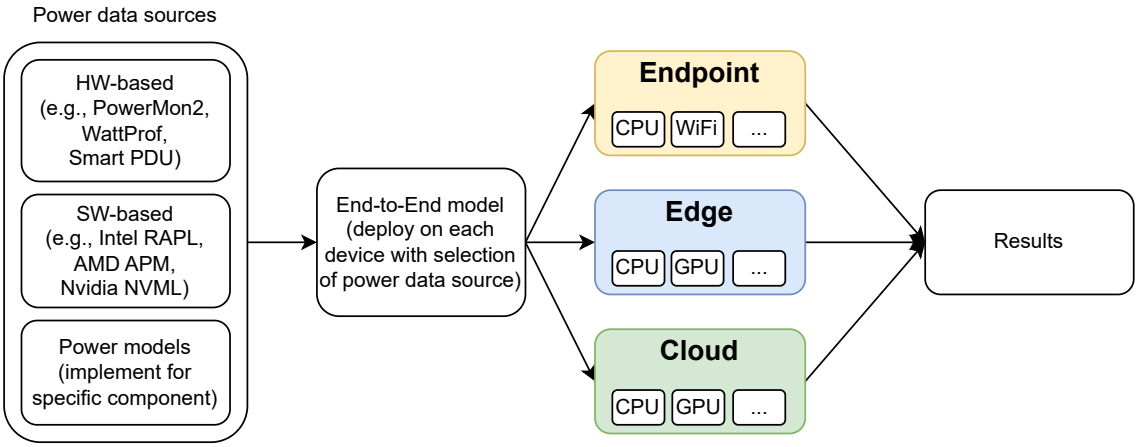


Figure 4.2: End-to-End power model design

Therefore, to use direct energy interface measurements in the end-to-end model a component needs satisfy the following requirements:

- **Availability**: The component must include MSRs that report the power or energy usage.
- **Interface**: A kernel module needs to be available that exposes the MSRs to the user.

4.2.3 Power Models

Last but not least, we power modeling based on the resource usage or utilization of a component can be used to obtain power data of a system. We analyze the resources that influence the power usage of various components in the previous Chapter 3. While power models should be considered least accurate compared to the previous methodologies they still help to provide invaluable insights when other methodologies are not available. However, many power models are able to estimate the power consumption of a component with low deviation of better measurement methodologies.

4.3 Model Design

We design the end-to-end power model to monitor and measure power consumption across the compute continuum, from endpoints and edge devices to cloud infrastructure. The model's design, as illustrated in Figure 4.2, integrates the previously discussed measurement and modeling methodologies to ensure a proper model of the compute continuum.

Deployment on Each Device The model is designed to be deployed on each device within a given compute continuum configuration. The deployment process involves selecting the appropriate power data source for each device based on its capabilities.

Power Data Sources The model uses a combination of HW-based and SW-based power data sources to gather power consumption data. The diagram (Figure 4.2) includes the same exemplary HW- and SW-based measurement interfaces as mentioned before (Section 4.2). If these power data sources are not available the models uses power models from the literature to substitute missing power data.

Power Models Where direct measurements from HW-based or SW-based sources are not available or feasible, the model employs predictive power models tailored for specific components. These models use resource utilization metrics (such as CPU usage, network bandwidth, and GPU load) to estimate power consumption. The different resource utilization metrics are previously discussed in Section 3.3.2.

Results The gathered data is processed to provide real-time power consumption metrics and detailed insights into the power usage of each device.

4.4 Conclusion & Future Work

In this chapter we demonstrate the process of designing an end-to-end power model for the compute continuum. The chapter begins by establishing the fundamental requirements and constraints, ensuring the model's relevance and adaptability to various configurations within the compute continuum. We integrate HW-based and SW-based power measurement techniques, in our model to provide accurate power consumption estimates. Furthermore, we add power models in the end-to-end model if HW- or SW-based power measurements are infeasible to provide power data.

The model design emphasizes modularity, allowing it to be tailored to specific devices and deployment scenarios. This adaptability is crucial for reflecting the diverse nature of the compute continuum. By incorporating insights from previous research questions, particularly the identification and classification of power-consuming components (RQ-1), the model achieves a good representation of power usage in the compute continuum.

4.4.1 Future Work

Future work for the power model should focus on expanding its scope to include additional components and layers are not yet integrated. While the current model provides a robust framework for incorporating missing components of the devices in the compute continuum, it lacks support for parts of the compute continuum that span exist between layers. Specifically, future development should aim to incorporate networking infrastructure between layers, as well as additional hardware such as cooling systems and lighting infrastructure in data centers. Including these elements will enhance the model's comprehensiveness and accuracy, allowing for a more detailed and holistic understanding of power consumption across the entire compute continuum.

In conclusion, this chapter successfully addresses the research question (RQ-3) regarding the design of an end-to-end power model for the compute continuum.

5

Implementing a Prototype

This section answers RQ-4: How can we prototype this model to estimate the power consumption in the compute continuum? Implementing the model introduced in the previous section is necessary to validate it. However, due to the large model design space it is only feasible to implement a prototype. This prototype covers a subset of all possible components.

To implement the prototype we select one device for each layer of the compute continuum (endpoint, edge and cloud). Section 5.1 presents the selection of devices.

We only implement the components relevant to the workflow we will be validating the prototype against in Chapter 6. Therefore, Table 5.1 only lists a subset of all the components previously identified in Chapter 3 and shown in Figure 5.1.

Furthermore, to comply with the design requirements outlined in Chapter 4, we also implement a Intel RAPL measurement for the cloud-grade server. This Intel RAPL measurement implementation is explained in detail in Section 5.4.

5.1 Device Selection

The decision to use a smartphone is driven by the increasing amount of modern smartphones and the identification as endpoint devices in the compute continuum. Smartphones are equipped with powerful cameras and processing units, making them ideal for tasks requiring image acquisition and initial processing. Additionally, they are portable and widely available, making them a practical choice for a variety of applications in both research and industry settings. Last but not least, they are powered by batteries which makes them a valuable target for power consumption analysis.

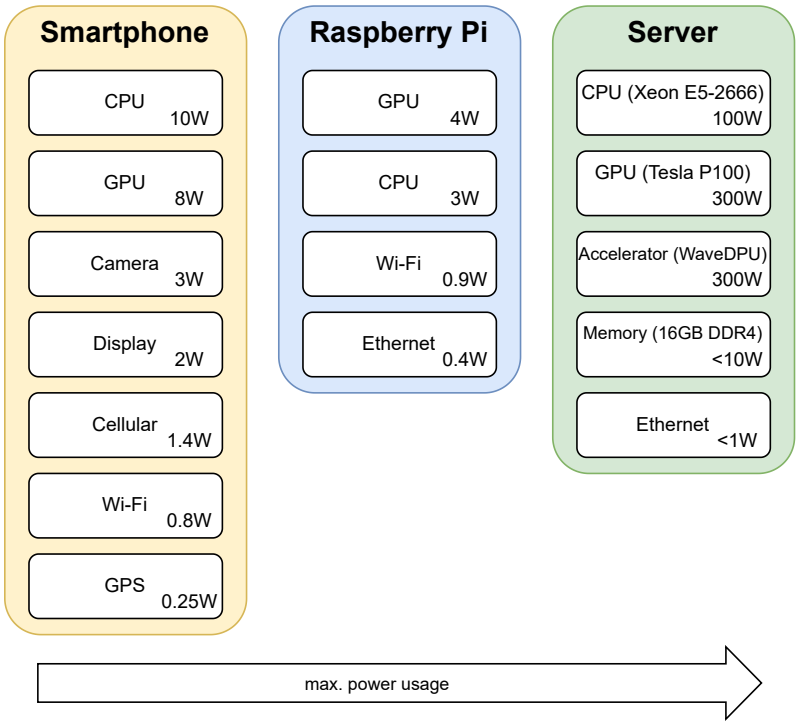


Figure 5.1: Peak power usage of components for device selection.

Furthermore, we choose the Raspberry Pi as an edge server due to its balance between popularity, performance, energy efficiency, and cost-effectiveness. The Raspberry Pi is widely used in edge computing scenarios because it provides sufficient computational power to handle various tasks while maintaining low power consumption, which is crucial for sustainable and scalable edge computing solutions [31]. On the other hand, a cloud-grade server is included to represent the high-performance end of the spectrum, where extensive computational resources and scalability are available. This allows us to evaluate the power model's performance across different layers of the compute continuum, from resource-constrained edge devices to powerful cloud servers.

We considered alternative configurations such as using dedicated industrial cameras for the publisher or more powerful edge devices like Nvidia Jetson boards. However, a lack of available hardware as well as the fact that the chosen setup provides a good balance between accessibility, performance, and relevance to common real-world scenarios leads us to decide on the selected components.

This configuration fits well within our prototype as it reflects common deployment scenarios in edge and cloud computing. It provides a realistic and practical framework for evaluating the power models under different conditions, ensuring that the findings are

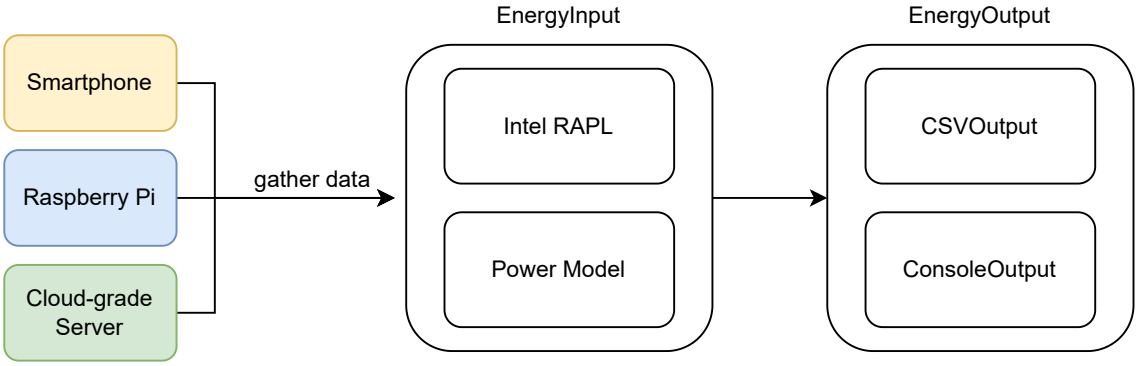


Figure 5.2: Design of the Prototype

applicable to a broad range of applications in the compute continuum.

Component	Smartphone (End-point)	Raspberry Pi (Edge)	Cloud-grade server (Cloud)
CPU	Snapdragon 600 @ 1GHz	ARM Cortex-A53 @ 1Ghz	Xeon Silver 4210R @ 2.4Ghz
Network	Wi-Fi	Ethernet	Ethernet
Camera	1280x720	n.a.	n.a.

Table 5.1: Selection of devices for the prototype

5.2 Prototype Design

Figure 5.2 illustrates the overarching design of our prototype. The components *EnergyInput* and *EnergyOutput* are implemented as abstract base classes using the *abc* package in Python. This design choice ensures that the prototype remains modular and facilitates easy extensibility for incorporating additional components or models in the future. By creating subclasses of the *EnergyInput* class, various power models or other power measurement interfaces can be seamlessly integrated. Similarly, different output formats can be added by subclassing the *EnergyOutput* class, maintaining the flexibility and scalability of the prototype.

5.3 Model Selection

During the research to identify components and their energy impact for Chapter 3 we are additionally able to find many different power models for various devices. We then choose

power models based on the device selection in Table 5.1. Our main focus when selecting the power models is if they are evaluated with external HW-based measurements. High compliance with these measurements ensures the quality and accuracy of the models.

5.3.1 CPU

In general CPU models are dependent on three factors: utilization, core count, and frequency [29]. To simplify this for our prototype implementation we have fixed the frequency of the endpoint and edge devices at 1GHz. This removes the frequency as a variable from the models.

Based on the findings in [29] we can obtained the following power model for the CPU of the endpoint device:

$$P_{CPU} = \sum_{c=0}^{m} (\beta_{core}^{c} * u_{core}^{c} + P_{idle}^{c}) + P_{uncore} \tag{5.1}$$

where P_{CPU} denotes the overall CPU power usage, β_{core}^{c} the power coefficient per core, u_{core}^{c} the utilization per core, P_{idle}^{c} the idle power usage per core, and P_{uncore} the base power usage.

Based on the findings in [31] we can obtained the following power model for the CPU of the edge device:

$$P_{CPU} = P_{base} + \beta_{Pi} * u \tag{5.2}$$

where P_{CPU} denotes the overall CPU power usage, P_{base} the base power usage, β_{Pi} the power coefficient, u the utilization.

We do not select a power model for the cloud device because it is measured using Intel RAPL in our prototype.

5.3.2 Network

Modeling network power usage can be done using bandwidth- or packet-based utilization models [10], [29], [31], [35]. When considering wireless communication equipment like WiFi or cellular there are other factors impacting the energy usage such as signal strength or transmission mode [29]. Even though the endpoint device in our test is supposed to use WiFi, we will only simulate this due to the virtualized nature of our testing equipment.

Based on the findings in [29] we can obtained the following power model for the WiFi interface of the endpoint device:

$$P_{WiFi} = \begin{cases} P_{LT} * u_{TRX} + \beta_{LT_{TRX}} & \text{if } u_{trx} \leq \text{ Threshold} \\ P_{HT} * u_{TRX} + \beta_{HT_{TRX}} & \text{else} \end{cases} \tag{5.3}$$

where P_{WiFi} denotes the overall WiFi power usage, P_{HT}/P_{LT} the high/low transmission mode base power, u_{TRX} the utilization in packets per second, and $\beta_{LT_{TRX}}/\beta_{HT_{TRX}}$ the beta coefficient for high/low power.

Based on the findings in [11] we can obtained the following power model for the NIC of the edge device:

$$P_{NIC} = P_{up} + P_{down} + P_{idle} \tag{5.4}$$

where P_{NIC} denotes the overall NIC power usage, P_{up}/P_{down} the upload/download power usage, and P_{idle} the NIC idle power usage. P_{up}/P_{down} can be modeled using the following equation:

$$P_{up,down} = \begin{cases} \beta_{up,down_{LP}} * u_{up,down} & \text{if } r \leq \text{ Threshold} \\ \beta_{up,down_{HP}} * u_{up,down} & \text{else} \end{cases} \tag{5.5}$$

where $\beta_{up,down_{LP}}/\beta_{up,down_{HP}}$ are the power coefficients for low/high power upload/download, and $u_{up,down}$ is the upload/download utilization in Mbps.

Last but not least we use the data presented in [35] to obtain a model for the NIC of the cloud device:

$$P_{NIC} = (u_{up} + u_{down}) * P_{packet} + P_{idle} \tag{5.6}$$

where P_{NIC} denotes the overall NIC power usage, u_{up}/u_{down} the upload/download utilization in packets per second, P_{packet} the power per packet and P_{idle} the NIC idle power usage.

5.3.3 Camera

The camera power usage is dependent on the resolution and FPS with which the camera is capturing images [29]. The following model is used in the prototype:

$$P_{Camera} = \beta_{camera} * (res_w * res_h * fps) + P_{idle} \tag{5.7}$$

where P_{Camera} denotes the overall camera power usage, β_{camera} is the power coefficient for the camera, res_w/res_h is the height/width of the image in pixels, fps are the frames per second, and P_{idle} is the idle power usage.

In this section, we selected power models for various devices, focusing on CPU, network, and camera power usage. For CPUs, we derived models for endpoint and edge devices, excluding cloud devices that will be measured with Intel RAPL. Network and camera power models were also established based on utilization metrics and power coefficients. Next, we will discuss how Intel RAPL measurements are utilized for power assessment in cloud devices.

5.4 Implementing Intel RAPL Measurements

Intel RAPL offers several methods for measuring the energy consumption of a system, including the use of MSR and a sysfs interface known as *powercap*. In our implementation, we utilize the sysfs interface due to its accessibility and ease of integration.

Intel RAPL provides detailed energy consumption data for each CPU socket within a system, organized into distinct zones. Additionally, for each socket sub-zones are available to report energy usage of other parts, such as memory consumption. The sysfs interface includes a file named "energy_uj," which serves as a continuously increasing counter that reports the energy usage of a CPU or socket. This file records energy consumption in micro-joules (µJ).

In our specific setup, only the total package power is reported by Intel RAPL. However, newer CPU versions support more granular measurement domains, such as per-core measurements. By regularly reading the "energy_uj" file, we can determine the energy consumption over time. Since 1 joule (J) equals 1 watt-second (Ws), measuring the energy consumption every second allows us to obtain power values.

To achieve finer granularity in our measurements, we estimate the per-application power usage by examining the ratio between the overall CPU usage and the CPU usage of our specific application. This ratio is then used to calculate the relative power consumption of our application, providing us with detailed insights into its energy efficiency.

5.5 Limitations & Future Work

Virtualization of Devices Due to unavailability of hardware the devices used in the experiments with the prototype have to be virtualized. The continuum framework [63] is used to create the virtual machines (VMs) representing the actual devices. We are using QEMU's time share feature to limit the available processor resources to match the hardware listed in Table 5.1. Furthermore, the usage of the camera is completely simulated and reports as active as long as the *publisher*-process of the image-classification benchmark is running on the endpoint node.

Extend Prototype for More Components Currently the scope of the protoype is narrowly fixed to the devices we outline at the start of this chapter. However, like we discussed before the compute continuum consists of a large variation of devices and components. This variation provides a perfect basis for future work on this prototype by extending its current implementation with power models for new devices.

Furthermore, there are still missing components (e.g., memory, disk, fans, ACs) that would need to be covered for the prototype to provide a real end-to-end image of the workflow. We omitted these components due to missing data (fans) or lack of good power models in the literature (memory, disk). However, we already provide a example implementation that is able to measure the power consumption of memory on platforms with Intel RAPL support.

Extend Prototype to Support HW-Based Measurements Currently the prototype does not contain any interfacing capabilities with regards to external, HW-based measurement equipment due to our lack of such equipment. However, considering the modular design of the prototype it should be trivial to implement a EnergyInput subclass that is able to interface with HW-based measurement equipment.

Extend Prototype to Support More SW-Based Interfaces Currently our prototype only supports Intel RAPL for power measurement data. However, similar interfaces exist for hardware from Nvidia and AMD. Due to a lack of hardware form those vendors we are unable to include these interfaces in our prototype.

6

Evaluation

After answering RQ-4 in the previous section by providing a prototype implementation it is important to validate this implementation. We evaluate the end-to-end power model with other software tools. The first part that we evaluate is our implementation of the Intel RAPL measurements. There are other tools that utilize Intel RAPL for power/energy measurements. We evaluate our implementations against this other tools to validate if the measurements are reported correctly. Secondly, we also evaluate how many of the design requirements we identified in Chapter 4 are implemented.

However, due to a lack of hardware (Section 5.5) and hardware-based measurement infrastructure we leave HW-based verification to future work.

6.1 Experiment Design & Setup

In order to evaluate our prototype we need a workflow for the compute continuum to use as an experiment. We decide to use the image-classification experiment that is included in the continuum framework [63].

6.1.1 Design

Figure 6.1 is a diagram describing this experiment. The image-classification experiment consists of three main components: *publisher*, *subscriber*, and *broker*.

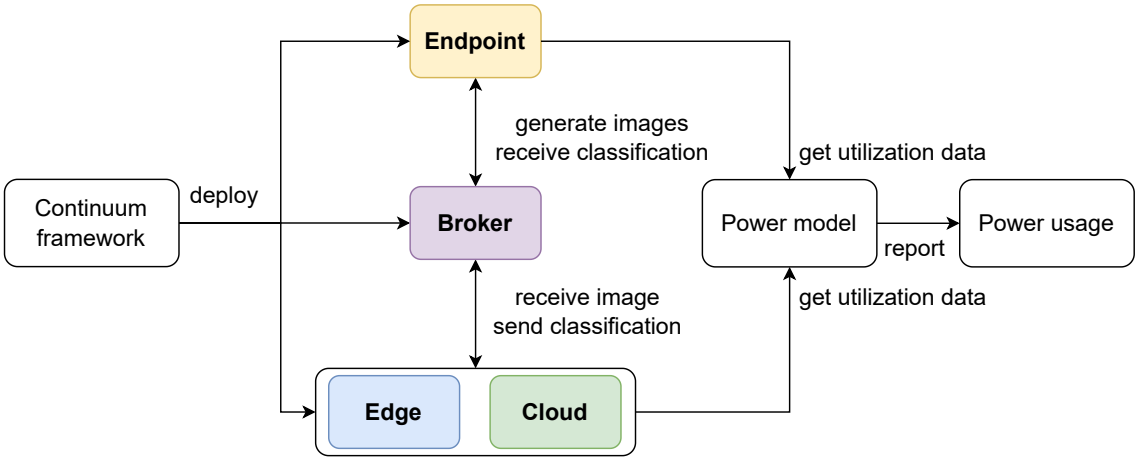


Figure 6.1: Experiment Design

- Publisher: MQTT client that sends images to the broker
- Subscriber: MQTT client that receives and classifies images from the publisher using a TensorFlow model
- Broker: MQTT broker that acts as the relay between the publisher and subscriber.

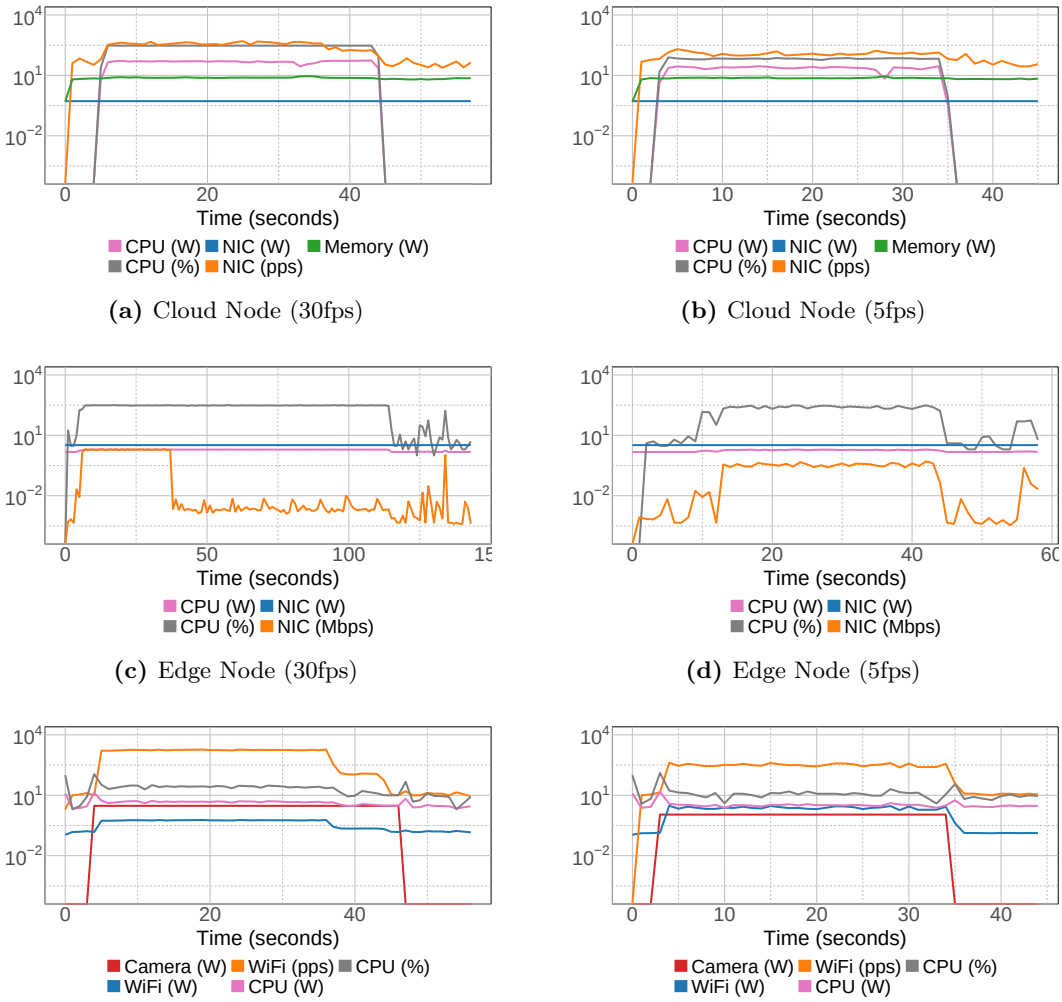
In our case, the publisher is considered to be a smartphone that captures the images which are then sent to the subscriber. The subscriber can either be a Raspberry Pi serving as an edge server or a cloud-grade server.

The implications of this setup are significant for understanding the power consumption across the compute continuum. By using a smartphone as the publisher, we can explore the impact of mobile device power usage in data acquisition. The comparison between the Raspberry Pi and cloud server as subscribers allows us to assess how shifting computational tasks from the cloud to the edge affects overall power consumption and performance.

6.1.2 Setup

As mentioned in Section 5.5 we are using VMs instead of real devices for the endpoint and edge. These VMs are deployed by the continuum framework including the necessary files to run the experiment. However, the cloud server is a bare-metal server, so we are able to test our Intel RAPL measurement implementation.

The experiment is executed with different configurations for the endpoint and edge/cloud devices. For the endpoint device the configuration of the experiment is changed to generate 5 frames per second (FPS) and 30 FPS. We change the configuration to validate that



(a) Cloud Node (30fps)

(b) Cloud Node (5fps)

(c) Edge Node (30fps)

(d) Edge Node (5fps)

(e) Endpoint Node (30fps)

(f) Endpoint Node (5fps)

Figure 6.2: Power usage between experiment runs with 30fps and 5fps.

our model is reacting to changes of the components or their utilization. Both of these experiment runs are then repeated for a deployment using a edge and a cloud subscriber.

6.2 Experiment Results

Figure 6.2 presents the results of the different experiment executions in terms of power while Figure 6.3 is a comparison of the amount of energy used by the different executions.

6.2.1 Cloud

To obtain the results for the cloud node (Figure 6.2a and Figure 6.2b) we use Intel RAPL for CPU and memory and the Equation (5.6) for the NIC. We observe that there is almost no increase in NIC power between 30FPS and 5FPS executions due to the low utilization of the network link. Contrary to this, the CPU power use increases significantly due to the much higher amount of work it has to do. This increase is also clearly observed in the energy comparison in Figure 6.3 where the cloud is using roughly double the energy when comparing the 5FPS and 30FPS execution. This increase between the two executions is expected due to the $\tilde{3}$x higher CPU utilization (Figure 6.2b & Figure 6.2a) which results form the much higher processing required for 30FPS compared to 5FPS.

6.2.2 Edge

The results for the edge node (Figure 6.2c and Figure 6.2d) are obtained using the Equation (5.2) for the CPU and Equation (5.4) & Equation (5.5) for the NIC. Contrary to the cloud, there are no significant increases in power use to be observed between the two different executions. This is due to the already maxed out utilization of the CPU on the 5FPS execution. However, the important observation we have here is that the 30FPS execution takes significantly longer (2.5x execution time) than the 5FPS execution. While there is no increase in power visible Figure 6.3 clearly shows that the energy consumption has increase by 2.5 times. This is exactly according to our expectation after observing the 2.5 times longer execution time with the same amount of power usage.

6.2.3 Endpoint

For the endpoint (Figure 6.2e and Figure 6.2f) three different models are employed: CPU (Equation (5.1)), WiFi (Equation (5.3)), Camera (Equation (5.7)). While the WiFi and CPU power usage increase slightly the most prominent observation is the roughly 3x hike in camera power usage. This leads to a doubling in energy consumption (Figure 6.3) for which the camera is almost solely responsible.

Finally, Figure 6.3 allows one further conclusion namely that there is a clear trade-off between computational power and electrical power. Apart from looking at the results of our experimental results there is one component in our prototype that allows further validation, namely our Intel RAPL implementation.

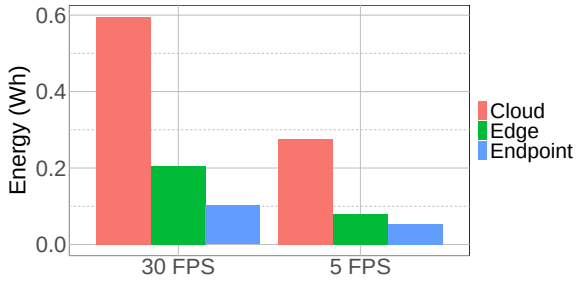
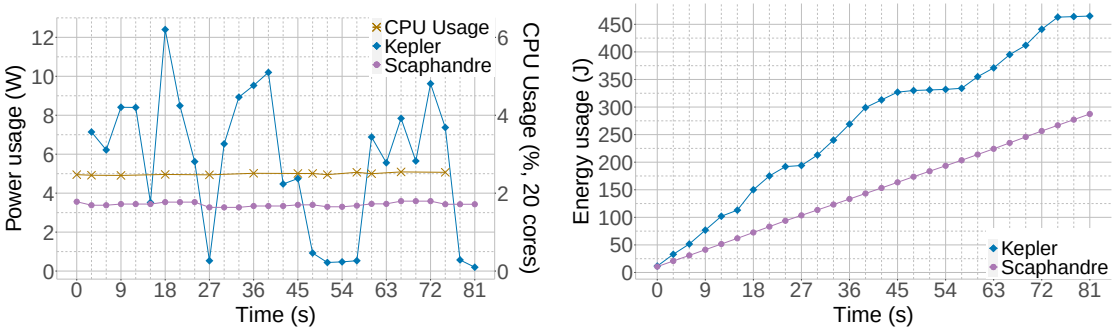


Figure 6.3: Energy usage of cloud, edge, and endpoint nodes



(a) Reported power usage

(b) Reported energy usage

Figure 6.4: Comparison of Kepler and Scaphandre

6.3 Evaluating Intel RAPL

Before we can evaluate our own Intel RAPL implementation we first need to identify the tools we want to evaluate it against. After identifying these tools we can evalute that our own Intel RAPL implementation provides accurate and trustworthy results.

6.3.1 Evaluation of Other Intel RAPL tools

From a comprehensive literature review conducted prior to this thesis [6], we have identified two software-based power meters, Scaphandre and Kepler, as the state-of-the-art tools for CPU energy measurements. To establish a validation baseline for assessing our own Intel RAPL measurement implementation, it is essential to compare our implementation with other state-of-the-art tools.

We evaluate these software-based power meters, which are both using the RAPL technology, in a bare-metal Kubernetes environment. The Kubernetes environment is used because Kepler is a acronym for "Kubernetes-based Efficient Power Level Exporter" and therefore it is limited to measure energy in such an environment.

Both applications measure a micro-benchmark that involves a simple Fibonacci sequence computation (Listing 1) five times, adhering to the same experimental setup. They utilize the sysfs energy_uj metric from the powercap framework for collecting data. During a one-minute test period, each benchmark consistently utilizes approximately 100% CPU power on a single core.

The evaluation reveals that Scaphandre produces more consistent results, maintaining stable energy usage readings throughout the testing period. In contrast, Kepler displays significant fluctuations and a broader range of deviations in its measurements (Figure 6.4a). Additionally, the energy consumption reported by Kepler exceeds that of Scaphandre by more than 1.5 times (Figure 6.4b). Despite employing the same methodological approach, the reasons for Kepler's substantial inconsistencies and higher reported energy usage remain unexplained.

6.3.2 Evaluation of Our Intel RAPL Implementation

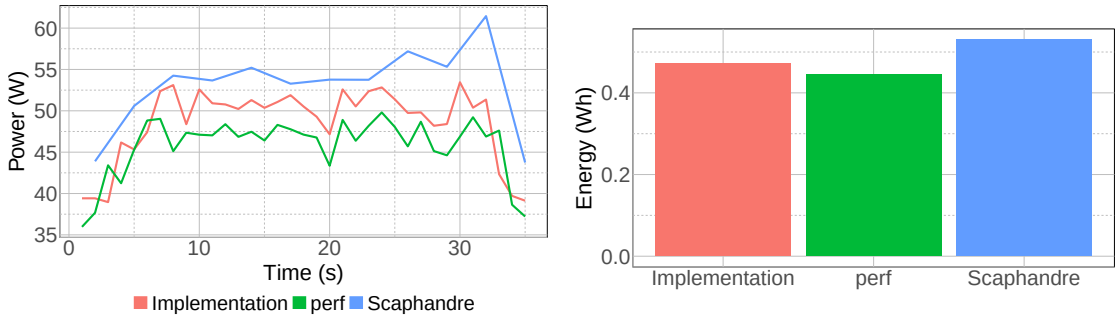
Based on the finding of the previous Section 6.3.1 we select Scaphandre to evaluate our Intel RAPL power measurement implementation against. Furthermore, we also use *perf* because it is a well-known tool included in the Linux kernel.

Figure 6.5a shows the power usage of the tools over one run of the image-classification experiment. Even though all tools use the Intel RAPL sysfs interface exposed by powercap the results differ. The root mean square error (RMSE) between our custom implementation and perf is 5.59 while the RMSE between our implementation and Scaphandre is 3.75.

Even after careful reviewing of the Scaphandre source code it remains unclear to us why the results of Scaphandre and our own implementation would differ so much. The only possible explanation we could come up with is that Scaphandre is probing the Intel RAPL sysfs interface every three seconds and calculates the power values it reports from these measurements. Contrary to this our implementation probes the interface every second. In this scenario a timing difference between measurements of 10% or 0.1 seconds would already amount to a difference of roughly 5W.

6.4 Prototype Compliance with Design Requirements

In this section, we evaluate the compliance of our prototype with the requirements in Chapter 4. We assess each requirement based on how the prototype addresses the stated criteria. For a comprehensive overview of how each requirement corresponds to specific parts of the design, we refer to Figure 6.6 once more.



(a) Power **(b)** Energy

Figure 6.5: Validation of Scaphandre, perf, and our custom Intel RAPL power measurement implementation.

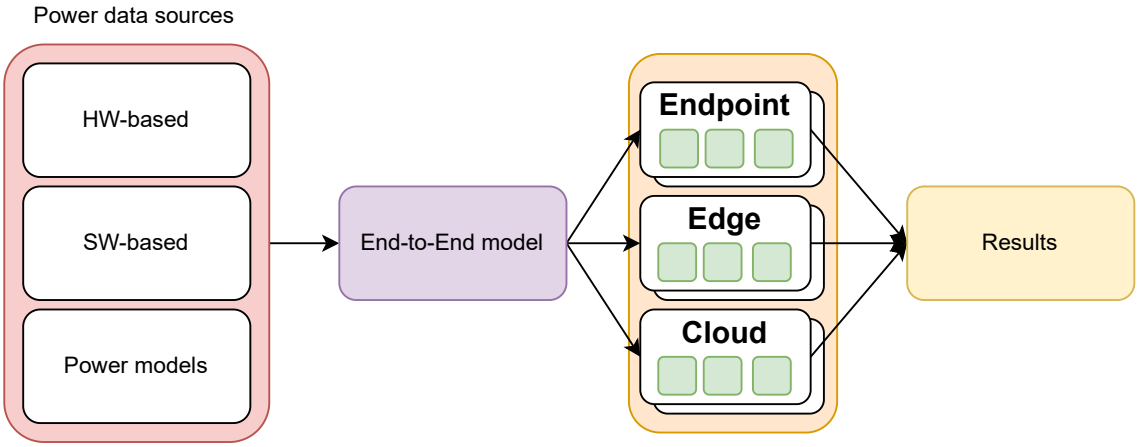


Figure 6.6: Overview of the model and the respective design requirements.

Adaptability (DR-1) Our prototype meets this requirement by being successfully deployed in endpoint, edge and cloud environments. Due to the modular approach, employing subclasses of one abstract base class (*EnergyInput*, as discussed in Section 5.2), we are able to implement power models for different components. For each deployment we can then select the fitting power models to ensure proper estimations for the given deployment. This flexibility highlights the prototype's ability to function in various scenarios, achieving the goal of adaptable deployment.

We can see the adaptability of the prototype evidenced in Figure 6.2. The graphs show that our prototype measures different components based on the deployment.

Stability (DR-2) Although we supply particular initial values for the models implemented in our prototype, these values are customized for the specific models and compo-

nents used during testing. To provide these initial values we again make use of the benefits of having a abstract base class that can be implemented in subclasses that provide specific initial values as constructor parameters. However, this customization indicates that distinct models and components will require unique initial values, making it challenging to meet this requirement in a more generalized way.

Accuracy (DR-3) The prototype relies on direct measurements whenever possible and opts for model-based estimations in other cases. Because hardware-based measurement tools were unavailable during the development of the prototype, they are excluded. Alternatively, we use software-based interfaces (i.e., Intel RAPL) for cloud deployments on bare metal when these interfaces are available.

Power Measurement Capability (DR-4) As per this requirement, the model is required to measure power. Our prototype derives power values from Intel RAPL, which reports in microjoules. This is possible because 1 Joule is equal to 1 W*s. Therefore, we can derive the power usage from the energy usage if we have time between measurements. All other models included in the prototype provide power values in watts, ensuring adherence to this requirement.

Timing (DR-5) The prototype takes measurements every second and records a timestamp for each. This method allows for precise energy usage calculations, thus meeting the requirement. We can clearly see this measurement resolution in Figure 6.2c, where the NIC utilization fluctuates frequently.

Context Awareness (DR-6) Currently, the prototype requires manual declaration of the components and their measurement parameters. It does not automatically collect these details on the basis of system specifications. This shows that the requirement is partially met and that there is potential for improving automatic context-awareness.

Real-Time Monitoring (DR-7) The prototype meets this requirement by including a console output feature that reports measured power values in real time. This real-time reporting capability ensures that monitoring data is available instantly, thereby meeting the requirement.

6.5 Limitations & Future Work

HW-Based Evaluation By measuring the power or energy used by the different components in the system with HW-based measurement interface (e.g., WattProf[24]) we would be able to evaluate the end-to-end model properly. Even though SW-based measurement interfaces are provided by (some) vendors our results (Figure 6.4 and Figure 6.5) show that they are trivial to use and can lead to varying results. The comparison of Scaphandre and Kepler (Section 6.3.1) as well as the comparison of our RAPL implementation compared to Scaphandre and perf (Section 6.3.2) clearly indicates a need for external validation of these tools.

Compliance with Design Requirements In summary, our prototype exhibits substantial compliance with the majority of requirements, demonstrating adaptability (DR-1), specific measurement techniques (DR-3), accurate power measurement (DR-4), fixed interval measurements (DR-5), and real-time data reporting (DR-7). However, there are areas, such as generalizing initial values (DR-2) and improving context awareness (DR-6), where further development could enhance compliance.

7

Conclusion

This thesis presents a comprehensive study into the energy consumption of devices within the compute continuum, addressing a critical and socially relevant issue in the context of advancing global efforts to mitigate climate change and reduce the environmental footprint of digital technologies. The research has been driven by four key research questions, each contributing to the development and validation of an end-to-end power model that encompasses endpoint, edge, and cloud devices.

7.1 Contributions

RQ-1: What are the primary energy consuming components in the compute continuum? By answering RQ-1 we now know what we components of the devices in the compute continuum we have to measure. We systematically identify and classify components responsible for power consumption across different layers of the compute continuum. Therefore, this thesis provides a foundational understanding necessary for developing accurate power models. The analysis highlighted the significant energy impact of various components, such as CPUs, GPUs, and communication modules, in endpoint, edge, and cloud devices.

RQ-2: How can the power use of these components be accurately measured or modeled? The research introduced robust methodologies for measuring and modeling the power consumption of identified components. Leveraging quantitative research methods and extensive surveys, the study formulated theoretical model structures and configurations that reflect real-world power consumption patterns. This approach ensures the adaptability and scalability of the models to accommodate new data and insights.

RQ-3: How can a end-to-end power model for devices in the compute continuum be designed? A comprehensive power model was designed, integrating insights from the identification and measurement phases. This model simplifies the complexity of various devices and components within the compute continuum, providing a manageable framework for implementing and comparing different power models. The design phase also evaluated the support for hardware-based energy measurement, enhancing the model's accuracy and reliability.

RQ-4: How can we prototype this model to estimate the power consumption in the compute continuum? A prototype tool was developed to estimate power consumption across different deployment configurations within the compute continuum. The prototype's implementation demonstrated the conceptual feasibility of the end-to-end power model, undergoing rigorous testing against workload-level benchmarks. The open-source nature of the tool promotes community contributions, fostering continuous innovation and enhancement in energy-efficient computing.

7.2 Future Directions

The findings and contributions of this thesis pave the way for several avenues of future research. Key directions include:

Expanding the Prototype Future work can focus on expanding the prototype to cover a broader range of configurations and devices within the compute continuum. Enhancing the tool's capabilities to provide more granular power consumption estimates will further its applicability and utility in real-world scenarios.

Integration with Emerging Technologies As new technologies such as advanced IoT devices, AI, and big data analytics continue to evolve, integrating these advancements into the power model will be essential. Continuous updating and validation of the model will ensure its relevance and accuracy in measuring the energy impact of cutting-edge technologies.

Policy and Societal Impact The research highlights the societal relevance of energy-efficient computing. Future studies could explore the policy implications of the findings, providing recommendations for regulatory frameworks (e.g., EU Energy Efficiency Directive [64]) and industry standards that promote sustainable practices in the tech industry.

In conclusion, this thesis has made significant strides in understanding and modeling energy consumption within the compute continuum. By providing a robust framework and practical tools, it contributes to the ongoing efforts to achieve energy-efficient computing, aligning with global sustainability goals and societal priorities. The open-science approach adopted in this research ensures that the contributions will continue to evolve, driving further innovation and development in the field of energy-efficient technology.

References

[1] A. Andrae and T. Edler, "On global electricity usage of communication technology: Trends to 2030," *Challenges*, vol. 6, no. 1, pp. 117–157, Apr. 30, 2015, ISSN: 2078-1547. DOI: 10.3390/challe6010117. [Online]. Available: http://www.mdpi.com/2078-1547/6/1/117 (1, 4).

[2] M. Jansen, A. Al-Dulaimy, A. V. Papadopoulos, A. Trivedi, and A. Iosup, *The SPEC-RG reference architecture for the compute continuum*, Mar. 2, 2023. arXiv: 2207.04159[cs]. [Online]. Available: http://arxiv.org/abs/2207.04159 (visited on 04/22/2024) (2, 3, 15–17, 19, 23).

[3] A. Reuther, P. Michaleas, M. Jones, V. Gadepally, S. Samsi, and J. Kepner, "Survey and benchmarking of machine learning accelerators," in *2019 IEEE High Performance Extreme Computing Conference (HPEC)*, Waltham, MA, USA: IEEE, Sep. 2019, pp. 1–9, ISBN: 978-1-72815-020-8. DOI: 10.1109/HPEC.2019.8916327. [Online]. Available: https://ieeexplore.ieee.org/document/8916327/ (visited on 06/06/2024) (3, 19, 20, 22).

[4] A. Cabrera, F. Almeida, J. Arteaga, and V. Blanco, "Measuring energy consumption using EML (energy measurement library)," en, *Computer Science - Research and Development*, vol. 30, no. 2, pp. 135–143, May 2015, ISSN: 1865-2034, 1865-2042. DOI: 10.1007/s00450-014-0269-5. [Online]. Available: http://link.springer.com/10.1007/s00450-014-0269-5 (visited on 11/22/2023) (4).

[5] H. David, E. Gorbatov, U. R. Hanebutte, R. Khanna, and C. Le, "RAPL: Memory power estimation and capping," en, in *Proceedings of the 16th ACM/IEEE international symposium on Low power electronics and design*, Austin Texas USA: ACM, Aug. 2010, pp. 189–194, ISBN: 978-1-4503-0146-6. DOI: 10.1145/1840845.1840883. [Online]. Available: https://dl.acm.org/doi/10.1145/1840845.1840883 (visited on 11/23/2023) (4, 11, 30).

[6] D. Freina, M. Jansen, and A. Trivedi, *A survey of energy measurement methodologies for computer systems*, Feb. 2024 (4, 5, 20, 29, 30, 47).

[7] C. Isci and M. Martonosi, "Runtime power monitoring in high-end processors: Methodology and empirical data," in *22nd Digital Avionics Systems Conference. Proceedings (Cat. No.03CH37449)*, San Diego, CA, USA: IEEE Comput. Soc, 2003, pp. 93–104, ISBN: 978-0-7695-2043-8. DOI: 10.1109/MICRO.2003.1253186. [Online]. Available: http://ieeexplore.ieee.org/document/1253186/ (visited on 11/23/2023) (4).

[8] Xizhou Feng, Rong Ge, and K. Cameron, "Power and energy profiling of scientific applications on distributed systems," in *19th IEEE International Parallel and Distributed Processing Symposium*, Denver, CO, USA: IEEE, 2005, pp. 34–34, ISBN: 978-0-7695-2312-5. DOI: 10.1109/IPDPS.2005.346. [Online]. Available: http://ieeexplore.ieee.org/document/1419856/ (visited on 12/18/2023) (4).

[9] L. Zhang, B. Tiwana, Z. Qian, *et al.*, "Accurate online power estimation and automatic battery behavior based power model generation for smartphones," in *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, Scottsdale Arizona USA: ACM, Oct. 24, 2010, pp. 105–114, ISBN: 978-1-60558-905-3. DOI: 10.1145/1878961.1878982. [Online]. Available: https://dl.acm.org/doi/10.1145/1878961.1878982 (visited on 06/04/2024) (4, 22).

[10] F. Kaup, P. Gottschling, and D. Hausheer, "PowerPi: Measuring and modeling the power consumption of the raspberry pi," in *39th Annual IEEE Conference on Local Computer Networks*, Edmonton, AB: IEEE, Sep. 2014, pp. 236–243, ISBN: 978-1-4799-3780-6. DOI: 10.1109/LCN.2014.6925777. [Online]. Available: http://ieeexplore.ieee.org/document/6925777/ (visited on 06/11/2024) (4, 12, 19, 22, 38).

[11] L. Ardito and M. Torchiano, "Creating and evaluating a software power model for linux single board computers," in *Proceedings of the 6th International Workshop on Green and Sustainable Software*, Gothenburg Sweden: ACM, May 27, 2018, pp. 1–8, ISBN: 978-1-4503-5732-6. DOI: 10.1145/3194078.3194079. [Online]. Available: https://dl.acm.org/doi/10.1145/3194078.3194079 (visited on 06/14/2024) (4, 12, 23, 39).

[12] Y. Levy and T. J. Ellis, "A systems approach to conduct an effective literature review in support of information systems research," *Informing Science: The International Journal of an Emerging Transdiscipline*, vol. 9, pp. 181–212, 2006, ISSN: 1547-9684, 1521-4672. DOI: 10.28945/479. [Online]. Available: https://www.informingscience.org/Publications/479 (visited on 04/22/2024) (8).

[13] N. A. Kheir, Ed., *Systems modeling and computer simulation*, 2nd ed, Electrical engineering and electronics 94, New York: M. Dekker, 1996, 729 pp., ISBN: 978-0-8247-9421-7 (8).

[14] A. Iosup, L. Versluis, A. Trivedi, *et al.*, "The AtLarge vision on the design of distributed systems and ecosystems," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, Dallas, TX, USA: IEEE, Jul. 2019, pp. 1765–1776, ISBN: 978-1-72812-519-0. DOI: 10.1109/ICDCS.2019.00175. [Online]. Available: https://ieeexplore.ieee.org/document/8885212/ (visited on 04/22/2024) (8).

[15] R. W. Hamming, *The art of doing science and engineering: learning to learn*, Fourth edition. San Francisco: Stripe Press, 2020, 403 pp., ISBN: 978-1-73226-517-2 (8).

[16] K. Peffers, T. Tuunanen, M. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, Dec. 1, 2007, ISSN: 0742-1222. DOI: 10.2753/MIS0742-1222240302. [Online]. Available: https://doi.org/10.2753/MIS0742-1222240302 (visited on 04/22/2024) (8).

[17] R. Jain, *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*. New York: Wiley, 1991, 685 pp., ISBN: 978-0-471-50336-1 (8).

[18] J. Ousterhout, "Always measure one level deeper," *Communications of the ACM*, vol. 61, no. 7, pp. 74–83, Jun. 25, 2018, ISSN: 0001-0782, 1557-7317. DOI: 10.1145/3213770. [Online]. Available: https://dl.acm.org/doi/10.1145/3213770 (visited on 04/22/2024) (8).

[19] S. Bezjak, A. Clyburne-Sherin, P. Conzett, *et al.*, *Open Science Training Handbook*. [object Object], Apr. 4, 2018. DOI: 10.5281/ZENODO.1212496. [Online]. Available: https://zenodo.org/record/1212496 (visited on 04/22/2024) (8).

[20] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, *et al.*, "The FAIR guiding principles for scientific data management and stewardship," *Scientific Data*, vol. 3, no. 1, p. 160 018, Mar. 15, 2016, ISSN: 2052-4463. DOI: 10.1038/sdata.2016.18. [Online]. Available: https://www.nature.com/articles/sdata201618 (visited on 04/22/2024) (8).

[21] E. D. Berger, S. M. Blackburn, M. Hauswirth, M. W. H. o. Aug 28, and 2019. "A checklist manifesto for empirical evaluation: A preemptive strike against a replication crisis in computer science," SIGPLAN Blog. (Aug. 28, 2019), [Online]. Available: https://blog.sigplan.org/2019/08/28/a-checklist-manifesto-for-empirical-evaluation-a-preemptive-strike-against-a-replication-crisis-in-computer-science/ (visited on 04/22/2024) (8).

[22] A. Uta, A. Custura, D. Duplyakin, *et al.*, "Is big data performance reproducible in modern cloud networks?" In *Proceedings of the 17th Usenix Conference on Networked Systems Design and Implementation*, ser. NSDI'20, USA: USENIX Association, Feb. 25, 2020, pp. 513–528, ISBN: 978-1-939133-13-7. (visited on 04/22/2024) (8).

[23] D. Bedard, M. Y. Lim, R. Fowler, and A. Porterfield, "PowerMon: Fine-grained and integrated power monitoring for commodity computer systems," in *Proceedings of the IEEE SoutheastCon 2010 (SoutheastCon)*, Concord, NC, USA: IEEE, Mar. 2010, pp. 479–484, ISBN: 978-1-4244-5854-7. DOI: 10.1109/SECON.2010.5453824. [Online]. Available: http://ieeexplore.ieee.org/document/5453824/ (visited on 11/21/2023) (11, 20, 21, 30).

[24] M. Rashti, G. Sabin, D. Vansickle, and B. Norris, "WattProf: A flexible platform for fine-grained HPC power profiling," in *2015 IEEE International Conference on Cluster Computing*, Chicago, IL, USA: IEEE, Sep. 2015, pp. 698–705, ISBN: 978-1-4673-6598-7. DOI: 10.1109/CLUSTER.2015.121. [Online]. Available: https://ieeexplore.ieee.org/document/7307670 (visited on 11/22/2023) (11, 20, 21, 30, 51).

[25] AMD, "BIOS and kernel developer's guide (BKDG) for AMD family 15h models 00h-0fh processors," 2013 (11).

[26] D. Hackenberg, T. Ilsche, R. Schone, D. Molka, M. Schmidt, and W. E. Nagel, "Power measurement techniques on standard compute nodes: A quantitative comparison," en, in *2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, Austin, TX, USA: IEEE, Apr. 2013, pp. 194–204. DOI: 10.1109/ISPASS.2013.6557170. [Online]. Available: http://ieeexplore.ieee.org/document/6557170/ (visited on 11/23/2023) (11, 12, 30).

[27] W. Jung, C. Kang, C. Yoon, D. Kim, and H. Cha, "DevScope: A nonintrusive and online power analysis tool for smartphone hardware components," in *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, Tampere Finland: ACM, Oct. 7, 2012, pp. 353–362, ISBN: 978-1-4503-1426-8. DOI: 10.1145/2380445.2380502. [Online]. Available: https://dl.acm.org/doi/10.1145/2380445.2380502 (visited on 06/05/2024) (12, 18, 21).

[28] C. Yoon, D. Kim, W. Jung, C. Kang, and H. Cha, "AppScope: Application energy metering framework for android smartphones using kernel activity monitoring," in *Proceedings of the 2012 USENIX conference on Annual Technical Conference*, ser. USENIX ATC'12, USA: USENIX Association, Jun. 13, 2012, p. 36. (visited on 07/14/2024) (12, 18, 21, 22).

[29] C. Yoon, S. Lee, Y. Choi, R. Ha, and H. Cha, "Accurate power modeling of modern mobile application processors," *Journal of Systems Architecture*, vol. 81, pp. 17–31, Nov. 1, 2017, ISSN: 1383-7621. DOI: 10.1016/j.sysarc.2017.10.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1383762117301947 (visited on 06/05/2024) (12, 18, 21, 22, 38, 39).

[30] N. Mammeri, M. Neu, S. Lal, and B. Juurlink, "Performance counters based power modeling of mobile GPUs using deep learning," in *2019 International Conference on High Performance Computing & Simulation (HPCS)*, Dublin, Ireland: IEEE, Jul. 2019, pp. 193–200, ISBN: 978-1-72814-484-9. DOI: 10.1109/HPCS48598.2019.9188139. [Online]. Available: https://ieeexplore.ieee.org/document/9188139/ (visited on 06/06/2024) (12, 18).

[31] F. Kaup, S. Hacker, E. Mentzendorff, C. Meurisch, and D. Hausheer, "The progress of the energy-efficiency of single-board computers," *Tech. Rep. NetSys-TR-2018-01*, 2018. [Online]. Available: https://www.netsys.ovgu.de/netsys_media/publications/NetSys_TR_2018_01-p-58.pdf (visited on 06/11/2024) (12, 19, 23, 36, 38).

[32] K. N. Khan, M. Hirki, T. Niemi, J. K. Nurminen, and Z. Ou, "RAPL in action: Experiences in using RAPL for power measurements," *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, vol. 3, no. 2, pp. 1–26, Jun. 30, 2018, ISSN: 2376-3639, 2376-3647. DOI: 10.1145/3177754. [Online]. Available: https://dl.acm.org/doi/10.1145/3177754 (visited on 06/17/2024) (12).

[33] S. Desrochers, C. Paradis, and V. M. Weaver, "A validation of DRAM RAPL power measurements," in *Proceedings of the Second International Symposium on Memory Systems*, Alexandria VA USA: ACM, Oct. 3, 2016, pp. 455–470, ISBN: 978-1-4503-4305-3. DOI: 10.1145/2989081.2989088. [Online]. Available: https://dl.acm.org/doi/10.1145/2989081.2989088 (visited on 01/25/2024) (12, 23).

[34] L. Alt, A. Kozhokanova, T. Ilsche, C. Terboven, and M. S. Mueller, "An experimental setup to evaluate RAPL energy counters for heterogeneous memory," in *Proceedings of the 15th ACM/SPEC International Conference on Performance Engineering*, London United Kingdom: ACM, May 7, 2024, pp. 71–82, ISBN: 9798400704444. DOI: 10.1145/3629526.3645052. [Online]. Available: https://dl.acm.org/doi/10.1145/3629526.3645052 (visited on 06/20/2024) (12).

[35] P. Reviriego, K. Christensen, J. Rabanillo, and J. A. Maestro, "An initial evaluation of energy efficient ethernet," *IEEE Communications Letters*, vol. 15, no. 5, pp. 578–580, May 2011, Conference Name: IEEE Communications Letters, ISSN: 1558-2558. DOI: 10.1109/LCOMM.2011.040111.102259. [Online]. Available: https:

//ieeexplore.ieee.org/abstract/document/5743052/authors#authors (visited on 06/04/2024) (13, 23, 38, 39).

[36] K. Christensen, P. Reviriego, B. Nordman, M. Bennett, M. Mostowfi, and J. Maestro, "IEEE 802.3az: The road to energy efficient ethernet," *IEEE Communications Magazine*, vol. 48, no. 11, pp. 50–56, Nov. 2010, ISSN: 0163-6804. DOI: 10.1109/MCOM.2010.5621967. [Online]. Available: http://ieeexplore.ieee.org/document/5621967/ (visited on 06/04/2024) (13).

[37] R. Arshad, S. Zahoor, M. A. Shah, A. Wahid, and H. Yu, "Green IoT: An investigation on energy saving practices for 2020 and beyond," *IEEE Access*, vol. 5, pp. 15 667–15 681, 2017, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2686092. [Online]. Available: http://ieeexplore.ieee.org/document/7997698/ (visited on 05/13/2024) (17, 18, 22).

[38] M. B. Kjærgaard, J. Langdal, T. Godsk, and T. Toftkjær, "EnTracked: Energy-efficient robust position tracking for mobile devices," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*, Kraków Poland: ACM, Jun. 22, 2009, pp. 221–234, ISBN: 978-1-60558-566-6. DOI: 10.1145/1555816.1555839. [Online]. Available: https://dl.acm.org/doi/10.1145/1555816.1555839 (visited on 06/05/2024) (18).

[39] L.-M. F. Burciu, R.-P. Fotescu, R. Constantinescu, B. Alexandrescu, and P. Svasta, "Energy consumption analysis of a network of sensors and energy consumption optimization methods," in *2023 IEEE 29th International Symposium for Design and Technology in Electronic Packaging (SIITME)*, ISSN: 2642-7036, Oct. 2023, pp. 219–222. DOI: 10.1109/SIITME59799.2023.10431388. [Online]. Available: https://ieeexplore-ieee-org.vu-nl.idm.oclc.org/abstract/document/10431388 (visited on 06/05/2024) (18).

[40] J. L. Soler-Fernández, O. Romera, A. Dieguez, J. D. Prades, and O. Alonso, "Ultra-low power readout electronics for wireless gas sensors in IoT," in *2023 30th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, Dec. 2023, pp. 1–4. DOI: 10.1109/ICECS58634.2023.10382924. [Online]. Available: https://ieeexplore-ieee-org.vu-nl.idm.oclc.org/document/10382924 (visited on 06/05/2024) (18).

[41] A. Shye, B. Scholbrock, and G. Memik, "Into the wild: Studying real user activity patterns to guide power optimizations for mobile architectures," in *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, New York New York: ACM, Dec. 12, 2009, pp. 168–178, ISBN: 978-1-60558-798-1. DOI: 10.1145/1669112.1669135. [Online]. Available: https://dl.acm.org/doi/10.1145/1669112.1669135 (visited on 06/04/2024) (18, 22).

[42] N. Jinaporn and P. Saengudomlert, "Impact of gateway placement and energy consumption for data processing on lifetime of IoT networks," in *2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, May 2021, pp. 90–93. DOI: 10.1109/ECTI-CON51831.2021.9454894. [Online]. Available: https://ieeexplore-ieee-org.vu-nl.idm.oclc.org/document/9454894 (visited on 06/05/2024) (18, 22).

[43] B. Martinez, M. Montón, I. Vilajosana, and J. D. Prades, "The power of models: Modeling power consumption for IoT devices," *IEEE Sensors Journal*, vol. 15, no. 10, pp. 5777–5789, Oct. 2015, Conference Name: IEEE Sensors Journal, ISSN: 1558-1748. DOI: 10.1109/JSEN.2015.2445094. [Online]. Available: https://ieeexplore-ieee-org.vu-nl.idm.oclc.org/abstract/document/7122861 (visited on 06/05/2024) (18, 21).

[44] W. A. Hanafy, T. Molom-Ochir, and R. Shenoy, "Design considerations for energy-efficient inference on edge devices," in *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*, Virtual Event Italy: ACM, Jun. 22, 2021, pp. 302–308, ISBN: 978-1-4503-8333-2. DOI: 10.1145/3447555.3465326. [Online]. Available: https://dl.acm.org/doi/10.1145/3447555.3465326 (visited on 05/13/2024) (18, 19, 22, 23).

[45] M. Daraghmeh, I. Al Ridhawi, M. Aloqaily, Y. Jararweh, and A. Agarwal, "A power management approach to reduce energy consumption for edge computing servers," in *2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC)*, Rome, Italy: IEEE, Jun. 2019, pp. 259–264, ISBN: 978-1-72811-796-6. DOI: 10.1109/FMEC.2019.8795328. [Online]. Available: https://ieeexplore.ieee.org/document/8795328/ (visited on 05/13/2024) (18).

[46] E. Ahmed and M. H. Rehmani, "Mobile edge computing: Opportunities, solutions, and challenges," *Future Generation Computer Systems*, vol. 70, pp. 59–63, May 2017, ISSN: 0167739X. DOI: 10.1016/j.future.2016.09.015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167739X16303260 (visited on 05/21/2024) (18, 19).

[47] J. Liu and X. Liu, "Energy-efficient allocation for multiple tasks in mobile edge computing," *Journal of Cloud Computing*, vol. 11, no. 1, p. 71, Oct. 27, 2022, ISSN: 2192-113X. DOI: 10.1186/s13677-022-00342-1. [Online]. Available: https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-022-00342-1 (visited on 05/21/2024) (18, 19).

[48] S. H. Mortazavi, M. Salehe, C. S. Gomes, C. Phillips, and E. De Lara, "Cloudpath: A multi-tier cloud computing framework," in *Proceedings of the Second ACM/IEEE*

Symposium on Edge Computing, San Jose California: ACM, Oct. 12, 2017, pp. 1–13, ISBN: 978-1-4503-5087-7. DOI: 10.1145/3132211.3134464. [Online]. Available: https://dl.acm.org/doi/10.1145/3132211.3134464 (visited on 06/11/2024) (19).

[49] K. Rungsuptaweekoon, V. Visoottiviseth, and R. Takano, "Evaluating the power efficiency of deep learning inference on embedded GPU systems," in *2017 2nd International Conference on Information Technology (INCIT)*, Nov. 2017, pp. 1–5. DOI: 10.1109/INCIT.2017.8257866. [Online]. Available: https://ieeexplore.ieee.org/document/8257866 (visited on 07/14/2024) (19, 22).

[50] H. Halawa, H. A. Abdelhafez, A. Boktor, and M. Ripeanu, "NVIDIA jetson platform characterization," in *Euro-Par 2017: Parallel Processing*, F. F. Rivera, T. F. Pena, and J. C. Cabaleiro, Eds., Cham: Springer International Publishing, 2017, pp. 92–105, ISBN: 978-3-319-64203-1. DOI: 10.1007/978-3-319-64203-1_7 (19, 22).

[51] H. A. Abdelhafez and M. Ripeanu, "Studying the impact of CPU and memory controller frequencies on power consumption of the jetson TX1," in *2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC)*, Rome, Italy: IEEE, Jun. 2019, pp. 105–112, ISBN: 978-1-72811-796-6. DOI: 10.1109/FMEC.2019.8795334. [Online]. Available: https://ieeexplore.ieee.org/document/8795334/ (visited on 06/11/2024) (19).

[52] C. Jiang, T. Fan, H. Gao, *et al.*, "Energy aware edge computing: A survey," *Computer Communications*, vol. 151, pp. 556–580, Feb. 2020, ISSN: 01403664. DOI: 10.1016/j.comcom.2020.01.004. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S014036641930831X (visited on 05/21/2024) (19).

[53] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *Proceedings of the 34th annual international symposium on Computer architecture*, ser. ISCA '07, New York, NY, USA: Association for Computing Machinery, Jun. 9, 2007, pp. 13–23, ISBN: 978-1-59593-706-3. DOI: 10.1145/1250662.1250665. [Online]. Available: https://dl.acm.org/doi/10.1145/1250662.1250665 (visited on 06/06/2024) (20).

[54] R. Basmadjian, N. Ali, F. Niedermeier, H. De Meer, and G. Giuliani, "A methodology to predict the power consumption of servers in data centres," in *Proceedings of the 2nd International Conference on Energy-Efficient Computing and Networking*, New York New York USA: ACM, May 31, 2011, pp. 1–10, ISBN: 978-1-4503-1313-1. DOI: 10.1145/2318716.2318718. [Online]. Available: https://dl.acm.org/doi/10.1145/2318716.2318718 (visited on 06/06/2024) (20).

[55] Z. Zhang and S. Fu, "Characterizing power and energy usage in cloud computing systems," in *2011 IEEE Third International Conference on Cloud Computing Technology and Science*, Athens, Greece: IEEE, Nov. 2011, pp. 146–153. DOI: 10.1109/CloudCom.2011.29. [Online]. Available: http://ieeexplore.ieee.org/document/6133138/ (visited on 05/21/2024) (20, 23).

[56] H. Nagasaka, N. Maruyama, A. Nukada, T. Endo, and S. Matsuoka, "Statistical power modeling of GPU kernels using performance counters," in *International Conference on Green Computing*, Chicago, IL, USA: IEEE, Aug. 2010, pp. 115–122, ISBN: 978-1-4244-7612-1. DOI: 10.1109/GREENCOMP.2010.5598315. [Online]. Available: http://ieeexplore.ieee.org/document/5598315/ (visited on 06/06/2024) (20).

[57] S. Song, C. Su, B. Rountree, and K. W. Cameron, "A simplified and accurate model of power-performance efficiency on emergent GPU architectures," in *2013 IEEE 27th International Symposium on Parallel and Distributed Processing*, Cambridge, MA, USA: IEEE, May 2013, pp. 673–686. DOI: 10.1109/IPDPS.2013.73. [Online]. Available: http://ieeexplore.ieee.org/document/6569853/ (visited on 06/06/2024) (20).

[58] Y. Li, A.-C. Orgerie, I. Rodero, B. L. Amersho, M. Parashar, and J.-M. Menaud, "End-to-end energy models for edge cloud-based IoT platforms: Application to data stream analysis in IoT," *Future Generation Computer Systems*, vol. 87, pp. 667–678, Oct. 2018, ISSN: 0167739X. DOI: 10.1016/j.future.2017.12.048. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167739X17314309 (visited on 05/21/2024) (19).

[59] J. H. Laros, P. Pokorny, and D. DeBonis, "PowerInsight - a commodity power measurement capability," in *2013 International Green Computing Conference Proceedings*, Arlington, VA, USA: IEEE, Jun. 2013, pp. 1–6, ISBN: 978-1-4799-0623-9. DOI: 10.1109/IGCC.2013.6604485. [Online]. Available: http://ieeexplore.ieee.org/document/6604485/ (visited on 11/21/2023) (21, 30).

[60] N. P. Jouppi, C. Young, N. Patil, *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *Proceedings of the 44th Annual International Symposium on Computer Architecture*, Toronto ON Canada: ACM, Jun. 24, 2017, pp. 1–12, ISBN: 978-1-4503-4892-8. DOI: 10.1145/3079856.3080246. [Online]. Available: https://dl.acm.org/doi/10.1145/3079856.3080246 (visited on 07/15/2024) (22).

[61] W. Lin, F. Shi, W. Wu, K. Li, G. Wu, and A.-A. Mohammed, "A taxonomy and survey of power models and power modeling for cloud servers," *ACM Computing Surveys*, vol. 53, no. 5, pp. 1–41, Sep. 30, 2021, ISSN: 0360-0300, 1557-7341. DOI:

10.1145/3406208. [Online]. Available: https://dl.acm.org/doi/10.1145/3406208 (visited on 05/21/2024) (29, 30).

[62] NVIDIA Corporation, *NVML API REFERENCE MANUAL*, https://developer.download.nvidia.com/assets/cuda/files/CUDADownloads/NVML/nvml.pdf, REFERENCE MANUAL, Accessed: 2024-01-24, 2012. (visited on 12/19/2023) (30).

[63] M. Jansen, L. Wagner, A. Trivedi, and A. Iosup, "Continuum: Automate infrastructure deployment and benchmarking in the compute continuum," in *Proceedings of the First FastContinuum Workshop, in conjuncrtion with ICPE, Coimbra, Portugal, April, 2023*, 2023. [Online]. Available: https://atlarge-research.com/pdfs/2023-fastcontinuum-continuum.pdf (40, 43).

[64] European Parliament, *Directive (EU) 2023/1791 of the European Parliament and of the Council of 13 September 2023 on energy efficiency and amending Regulation (EU) 2023/955 (recast)*, en (54).

Appendix

Artifact Description: Prototype End-to-End Model Implementation

Abstract

This artifact appendix describes how to setup the prototype implementation of the end-to-end model. Furthermore, it explains how to reproduce results as seen in the thesis. We describe how to obtain the required software, setup the same environment for experiments and execute these experiments. This setup consists of multiple parts: the prototype, continuum framework, benchmarks, and a set of R scripts to reproduce the plots presented in the thesis.

Artifact Check-list (Meta-information)

- **Program:** end-to-end-power-model (`https://github.com/davidfreina/VU-Thesis-24/tree/main/end-to-end-power-model`), continuum framework (`https://github.com/atlarge-research/continuum`)
- **Compilation:** Python3 (end-to-end-power-model, continuum framework), R (plotting)
- **Run-time environment:** Ubuntu 22.04.3 LTS, Python 3.10.12, root access required
- **Hardware:** Host system CPU with Intel RAPL support
- **Execution:** Approximate maximum runtime 3min
- **Metrics:** Power consumption
- **Output:** Console, CSV-file
- **Experiments:** Provided by continuum framework (`https://github.com/atlarge-research/continuum/tree/main/application/image_classification`
- **Publicly available?:** Yes
- **Code licenses (if publicly available)?:** MIT
- **Workflow framework used?:** Continuum framework

Description

How to access

The end-to-end-power-model and continuum framework can be obtained by cloning from Github:

```
$ git clone https://github.com/davidfreina/VU-Thesis-24.git
$ git clone https://github.com/atlarge-research/continuum.git
```

The end-to-end-power-model can be found in "VU-Thesis-24/end-to-end-power-model". The plotting tools can be found in "VU-Thesis-24/plotting".

Hardware dependencies

The CPU in the host system must have Intel RAPL compatibility.

Software dependencies

The software will only run on GNU/Linux and was specifically tested on Ubuntu 22.04.3 LTS. Furthermore, Python3 is required with the psutil package is required (refer to "VU-Thesis-24/end-to-end-power-model/requirements.txt". For the plots R in version 4.4.1 and the packages ggplot2, reshape2, scales, and dplyr are required.

Details about the setup of the continuum framework can be found in its repository (`https://github.com/atlarge-research/continuum/tree/main?tab=readme-ov-file#part-1-install-the-framework`).

Software and Hardware Configuration

All tests are run either bare-metal or on top of QEMU 6.2.0 with KVM enabled.

Host system

- Intel(R) Xeon(R) Silver 4210R CPU @ 2.40GHz with two sockets connected in NUMA mode.
- 256GB DDR4 RAM

VM's The virtual machines are deployed on the host system using the continuum framework. The required configurations can be found in "VU-Thesis-24/end-to-end-power-model/continuum-configurations"

Installation

Because all executable files are written in interpreted rather than compiled languages the installation steps are minimal. For the end-to-end-power-model the installation of the psutil dependency is required:

```
$ pip install -r VU-Thesis-24/end-to-end-power-model/requirements.txt
```

The installation of the continuum framework is more elaborate. Because it is only a dependency for us we refer to the installation and setup instruction on its own repository (`https://github.com/atlarge-research/continuum/tree/main?tab=readme-ov-file#part-1-install-the-framework`). However, we provide the configuration files used for the continuum framework in "VU-Thesis-24/end-to-end-power-model/continuum-configurations".

Experiment workflow

After setting up the execution environment using the continuum framework with the provided configuration files the image-classification workflow is executed automatically. When this execution is finished the continuum framework will provide the necessary commands to ssh into the deployed VMs.

Setup Edge (**NOTE:** This only applies if the edge.cfg file is deployed.) We SSH into the edge node and clone the end-to-end-power-model repository. The power measurement can be started using the following command:

```
$ python3 energy_monitor.py edge
```

Using a second SSH session, the subscriber for the image-classification experiment can also be started:

```
$ docker container run --rm --cpus=3 --memory=1000m --network
    =host --env MQTT_LOCAL_IP=192.168.210.3 --env MQTT_LOGS=
    True --env ENDPOINT_CONNECTED=1 --env CPU_THREADS=3 --name
     image-classification 192.168.1.101:5000/
    image_classification_subscriber
```

Setup Cloud (**NOTE:** This only applies if the cloud.cfg file is deployed.) We do not use the VM created by the continuum framework for the experiment execution due to missing support for Intel RAPL in virtualized enviroments. Therefore, we clone the repository to the host and start the power measurement:

```
$ sudo python3 energy_monitor.py cloud
```

(**NOTE:** sudo is required to access the sysfs interfaces for Intel RAPL measurements.)

Using a second SSH session, the subscriber for the image-classification experiment can also be started:

```
$ docker container run --rm --cpus=3 --memory=3500m --network
  =host --env MQTT_LOCAL_IP=192.168.210.3 --env MQTT_LOGS=
  True --env ENDPOINT_CONNECTED=1 --env CPU_THREADS=3 --name
   image-classification 192.168.1.101:5000/
  image_classification_subscriber
```

Setup Endpoint After setting up the power measurement on either the edge or cloud node we can setup the endpoint node. First we use the provided SSH command to connect to the VM and clone the Github repository containing the end-to-end-power-model. We can now already start the power measurements with the following command:

```
$ python3 energy_monitor.py endpoint
```

Using a second SSH session, the publisher for the image-classification experiment can also be started:

```
$ docker container run --rm --cpus=4 --network=host --env
  FREQUENCY=5 --env DURATION=30 --env MQTT_LOCAL_IP
  =192.168.210.4 --env MQTT_REMOTE_IP=192.168.210.3 --env
  MQTT_LOGS=True --env CLOUD_CONTROLLER_IP=192.168.210.2 --
  name cloud0_endpoint0 192.168.1.101:5000/
  image_classification_publisher
```

(**NOTE:** This command is used to produce the results for a 5FPS execution of the experiment (refer to Figure 6.2f). If the 30FPS experiment should be reproduced the FREQUENCY variable has to be change from 5 to 30.)

Gathering and Plotting Results After the docker commands have finished the power measurement process can be stopped (CTRL+C). The results are saved to CSV files in the same folder where the energy_monitor.py was executed. They are named after their respective command (edge.csv, cloud.csv, and endpoint.csv).

Depending on the execution (cloud/edge) the plotting script for Figure 6.2 can be found in "VU-Thesis-24/plotting/figure6.2/edge" or "VU-Thesis-24/plotting/figure6.2/cloud" respectively. The script requires the edge.csv/cloud.csv and endpoint.csv to be placed in the same folder as the script itself.

Additional Experiments

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int fib(int n) {
5      if (n == 0)
6          return 0;
7      if (n == 1)
8          return 1;
9      return fib(n-1) + fib(n-2);
10 }
11
12 int main(int argc, char const *argv[])
13 {
14     if (argc != 2)
15         printf("Usage: ./fib N\n");
16     printf("The argument supplied is %s\n", argv[1]);
17     printf("%d\n", fib(atoi(argv[1])));
18     return 0;
19 }
```

Listing 1: Micro-benchmark used to evaluate Scaphandre and Kepler