A Hardware Testbed for Renewable-Aware Resource Management at the Edge

Bachelor's Thesis

Marvin Steinke, 401089

18. September 2022

Technische Universität Berlin
Faculty IV – Electrical Engineering and Computer Science
Department of Telecommunication Systems
Distributed and Operating Systems

First Reviewer:	Prof. Dr. habil. Odej Kao
Second Reviewer:	Prof. Dr. Dr. h.c. Sahin Albayrak

Eidestattliche Erklärung / Statutory Declaration

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

I hereby declare that the thesis submitted is my own, unaided work, completed without any unpermitted external help. Only the sources and resources listed were used. The independent and unaided completion of the thesis is affirmed by affidavit.

Berlin, September 18, 2022

Marvin Steinke

Zusammenfassung

Abstract

Contents

Chapter 1

Introduction

Edge computing is a promising, emerging paradigm in the area of distributed systems and while still primarily being a theoretical concept, the rise of new domains like Internet of Things (IoT) establishes numerous areas of application [1]. Because of the decentralized nature of edge computing, new devices at the edge of the network are essential for many of these approaches. In some cases though, edge devices do not have access to the electrical grid and require on-site energy generation. Some examples include portable weather stations, smart watering and metering in autonomous farming or numerous sensors in smart city designs. Large-scale data centers already profit from integration of on-site renewable energy generation and are able to achieve significant cost reductions while also reducing their greenhouse gas emissions [2]. In its current state however, renewable energy generation is rather volatile and unable to supply sufficient uninterrupted power on its own [3], resulting in a problem for edge devices without a connection to the electrical grid. If devices under these circumstances aim to operate self-sufficiently and maximize uptime, their resources need to be managed dynamically, relative to current on-site renewable energy production.

To allow for dynamic resource management, matching the system's energy consumption to the on-site production, numerous approaches for power management pose a viable option. Power management, in this case energy proportional computing, can generally be pursued on a hardware- and software-level [4]. On a hardware-level, dynamic voltage and frequency scaling (DVFS), the dynamic adjustment of both voltage and frequency to reduce dynamic power consumption, is vital for this intention as the CPU traditionally consumes the most power in a system.

Recent research in the area of energy-aware resource management in edge/fog computing utilizing DVFS for power management mainly relies on simulations to predict a real-world outcome [5, 6, 7]. Most simulators make the assumption, that computational load can be adjusted in a way that energy consumption perfectly matches the energy production [8, 9, 10, 11]. While the reasoning for using simulations as opposed to real hardware testbeds may be justified and opportune for most research projects, it remains unclear how close these assumptions are to reality and how this may change the accuracy of the predictions and consequently research outcomes.

1.1 Testbed Requirements

This bachelor's thesis proposes a hardware testbed for renewable-aware resource management in edge computing, capable of dynamically adjusting its computational load relative to the on-site renewable energy production. In order to examine the assumption of energy-aware simulators, that computational load is adjustable so that energy consumption is matched to production, this thesis proposes to compare simulations derived from real-world data to this physical hardware testbed. To provide the necessary data to compare the simulations to, a testbed with the following properties is required.

1. Renewable energy is produced and consumed respectively.
2. Excess energy produced, can be stored and retrieved if the production fails to provide sufficient power.
3. $$\frac{P_{\text{load}}}{P_{\text{idle}}} \geq 2.$$
   Adjusting the consumption to current production is more meaningful, in regard to energy savings, if the quotient between the power consumption under load and idle is as high as possible. In order to work properly with the data provided, the lower bound 2 was chosen.
4. The current energy production and resources' power draw can be measured in Volts and Amperes respectively.
5. Resources can be managed dynamically. This is necessary for adjusting the consumption depending on the current production of renewable energy.

1.2 Thesis Outline

This thesis is divided into 6 chapters.

Chapter 2 introduces the fundamental concepts discussed in subsequential chapters. First, edge and fog computing paradigms and their necessity for future information and communications technology (ICT) are presented. Second, the respective use cases of testbeds, emulations and simulations in edge and fog computing environments are outlined. Third, energy proportional computing and its importance for our use case is described. Fourth, dynamic and energy-aware resource management techniques are presented and DVFS, as the main technique for our physical testbed, along with its implementation in the Linux kernel, is presented.

Chapter 3 first reviews related research in the area of energy-aware resource management. The utilization of simulations in all presented papers leads to the described problem which is subsequently outlined. Then, these simulators are presented and aforementioned problem is demonstrated.

Chapter 4 presents the hardware testbed. First, all hardware components used and their conformity with the requirements listed in section 1.1 are presented. Second, the assembly of the individual components and their relations to each other are described. Third, the setup of the software, necessary to ensure functionality of the components, data transmission and operability, is presented. Fourth, the implementation of energy-aware resource management for this testbed is demonstrated.

Chapter 5 evaluates the testbed by assessing the accuracy of its capability to adjusting its computational load relative to the on-site energy generation. An error analysis is conducted and reviewed in regards to collected data by the testbed. The subsequent consequences for conducted research in the area of energy-aware resource management are discussed.

Chapter 6 concludes the thesis by summarizing the main points and contributions.

Chapter 2

Background

This chapter elaborates on relevant concepts revolving around energy-aware resource management in edge and fog computing. Section 2.1 introduces the motivation and explains the necessity for edge and fog computing paradigms. Section 2.2 focuses on the development process of edge and fog infrastructure and outlines the essential use of testbeds, simulations and emulations. Section 2.3 presents main concepts of energy proportional computing. Since the renewable energy source of the hardware testbed is not able to supply uninterrupted power, section 2.4.1 outlines specific techniques used in energy-aware resource management. Dynamic voltage and frequency scaling will be the primary technique used in the hardware testbed and is therefore elaborated on further in section 2.4.2.

2.1 Edge and Fog Computing Paradigms

In 2012 Xia et al. defines the Internet of Things as the networked interconnection of everyday objects, which are often equipped with ubiquitous intelligence [12]. Since then, IoT devices have spread to all aspects of our everyday lives. Cisco estimates that by 2023 there will be 14.7 billion machine-to-machine connections, more than a 140% increase to 2018 [13]. The rise of IoT results in a significant increase of devices connected to the internet and may, with its current rate of development, soon grow to a point, that our current network structure cannot manage. In specific, major challenges that are posed by traditional cloud-centric IoT architectures include the excess of bandwidth availability due to the increasingly large and high-frequent rate of data being produced by IoT devices, and high end-to-end latency along with unstable, intermittent network connectivity due to potentially large physical distance [14].

As an alternative to cloud-centric architectures, computing resources can be brought closer to end-devices. Using these resources, even partially, for data processing would reduce the amount of data sent to the cloud. With the physical proximity of provided services, latency for end-devices is highly lowered, cloud servers are relieved due to less network traffic and network throughput is consequently raised, favoring both end-devices and cloud servers. With these alternative paradigms, fog and edge computing are introduced.

The terms *fog* and *edge* computing are often used inconsistently throughout current literature. The research field is still rather young and allows for multiple interpretations. In the context of this thesis, the distinction from Iorga et al., from the National Institute of Standards and Technology, will be used. Iorga et al. define fog computing as a layered model for enabling ubiquitous access to a shared continuum of scalable computing resource, whereas edge computing describes the layer of end-devices that are used to do limited local computing or sensor metering [15]. A concurrent example for this interpretation of the terminology is presented by Atos' BullSequana edge computing servers, utilized for visual quality control in production lines. Possible defects are detected locally and may subsequently be transmitted to a nearby fog server for further processing [16, 17].

2.2 Testbeds, Emulations and Simulations in Edge and Fog Computing Environments

Opposed to simulators or emulators, hardware testbeds are subject to real-life environments and therefore able to accommodate more realistic experiments. While some elaborate approaches are already in use, none of these are deployed on a large scale yet and are therefore, for the most part, unfit for research revolving around fog and edge computing. Deploying a large scale physical testbed for research or commercial purposes is not only very cost intensive, but also highly time consuming.

Emulators serve as a more abstract alternative. Instead of physically setting up and initializing hardware and network for the desired system, the components are imitated on a computer, while keeping all functionality of the emulated system. This enables multiple components to be emulated on a single system, making them independent from their physical location. While experiments utilizing emulators are saving space and costs to a certain extent, execution time is bound by the emulated hardware, making them very inefficient for many scenarios, where gathering data takes a lot of time.

Simulators on the other hand [18]

2.3 Energy Proportional Computing

2.4 Dynamic Resource Management

2.4.1 Energy-Aware Resource Management

2.4.2 Dynamic Voltage and Frequency Scaling

Chapter 3

Related Work

3.1 Energy-Aware Resource Management

3.2 Energy-Aware Simulators

Chapter 4

Testbed

This chapter presents the hardware testbed. The assembled, setup system is capable of dynamically adjusting its computational load relative to the on-site energy production by photo-voltaic (PV) modules. Section 4.1 lists all main physical components. Section 4.2 shows the assembly of all components and explains their interrelations. Subsequently section 4.3 shows the software setup, necessary to ensure functionality of the components, data transmission and operability. Section 4.4 presents the implementation of Energy-Aware Resource Management for this testbed and demonstrates the conformity with the requirements from section 1.1.

4.1 Hardware Components

The testbed is mainly composed of the following hardware. As the compute node, the single-board computer *Raspberry Pi 3b+* serves as a viable choice due to its low energy consumption, cost effectiveness and wide range of hardware applications.[1] Renewable energy is produced by four PV modules, each generating 330 mA at 6 V as stated by its manufacturer. Excess energy produced by the PV modules is stored in a 3.7 V, 6600 mAh lithium-ion polymer (LiPo) battery as backup energy source in case of suboptimal solar conditions. Ultimately a component, connecting all components listed above, is needed to measure the energy production of the PV modules and the energy consumption of the compute node. *SwitchDoc Labs* developed *SunControl*, an inexpensive solar power controller board, among multiple other things capable of these tasks [19] and is therefore ideal for this testbed.

[1] `https://raspberrypi.com`

4.2 Hardware Assembly

4.3 Software Setup

As the testbed is meant to be self-sufficient, remote communication is most suitable for monitoring and operations. A Secure Shell (SSH) connection to the Raspberry Pi is easily configured alongside the installation process of Raspberry Pi OS Lite via their imager software.[2] To enable functionality of SunControl, installation of SwitchDoc's Python driver code libraries are necessary. The official code from 2017 was written in Python 2.7.[3] Since Python 2 is deprecated since 2020 and vital official libraries are no longer supported,[4] the codebase needed to be refactored and ported from Python 2.7 to 3.7.[5] To allow SunControl to communicate with the Raspberry Pi, Inter-Integrated Circuit (I^2C) support for the ARM core and Linux kernel need to be enabled.[6]

4.3.1 Energy Preservation

The Raspberry Pi has many components, some of which might not be needed for the specific use-cases of the testbed, but still make up a large portion of overall energy consumption. The Bluetooth and HDMI module and the USB ports can be disabled to preserve energy.[6,7] Figure 4.1 displays the idle power usage of the Raspberry Pi 3b+ over the span of two minutes with the three components enabled and disabled respectively. Enabling the components yields in an average energy consumption of 2.1 W, while disabling them results in an average consumption of just 0.93 W, preserving 1.17 W in total. Since these components serve no purpose in this version of the testbed, they are disabled in order to preserve energy.

[2] `https://raspberrypi.com/software/`
[3] `https://github.com/switchdoclabs/SDL_Pi_SunControl`
[4] `https://python.org/doc/sunset-python-2/`
[5] `https://github.com/marvin-steinke/SDL_Pi_SunControl`
[6] `https://github.com/marvin-steinke/bachelors_thesis/blob/master/src/config/config.txt`
[7] `https://github.com/marvin-steinke/bachelors_thesis/blob/master/src/config/rc.local`

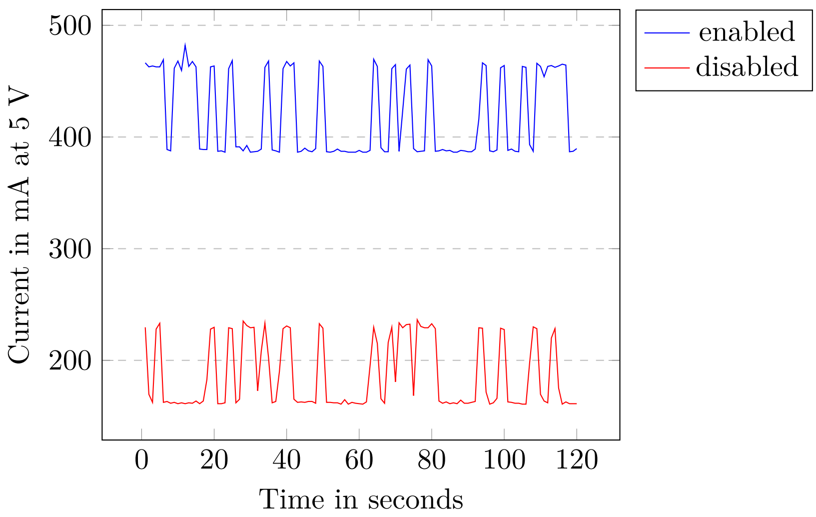

Figure 4.1: Idle power consumption with Bluetooth, HDMI and USB enabled and disabled

4.4 Implementation of Energy-Aware Resource Management

Because the CPU is the main consumer of energy in the Raspberry Pi, scaling the voltage and frequency of the CPU to control its consumption is the most viable approach to control total energy consumption.

4.4.1 DVFS versus `cpulimit`

4.4.2 Scaling the Output of the PV Modules

The PV modules utilized in this version of the testbed are also distributed by SwitchDoc Labs. They state that one panel is capable of generating a peak current of 330 mA at 6 V.[8] However, even while testing under strong sunlight, the modules were only capable of generating a third of the current that is stated. In figure 4.4 it can be observed that four of these PV modules would roughly only be capable of supplying the necessary current to the Raspberry Pi clocked at 400 MHz, the lowest possible clock frequency. With this current supplied, there is no margin for adjusting the energy consumption of the compute node to the production of the PV modules, as the clock speed could never surpass 400 MHz. Therefore to ensure an environment, in which the whole spectrum of energy consumed by the compute node at different frequencies can be supplied in theory, measured currents generated by the PV modules are multiplied by the factor three. Since the currents are still measured and reacted to in real time, the requirements for a testbed are still met, as presented in section 2.2.

[8] `https://switchdoc.com/2016/06/solar-panel-comparison-sunlight-test/`

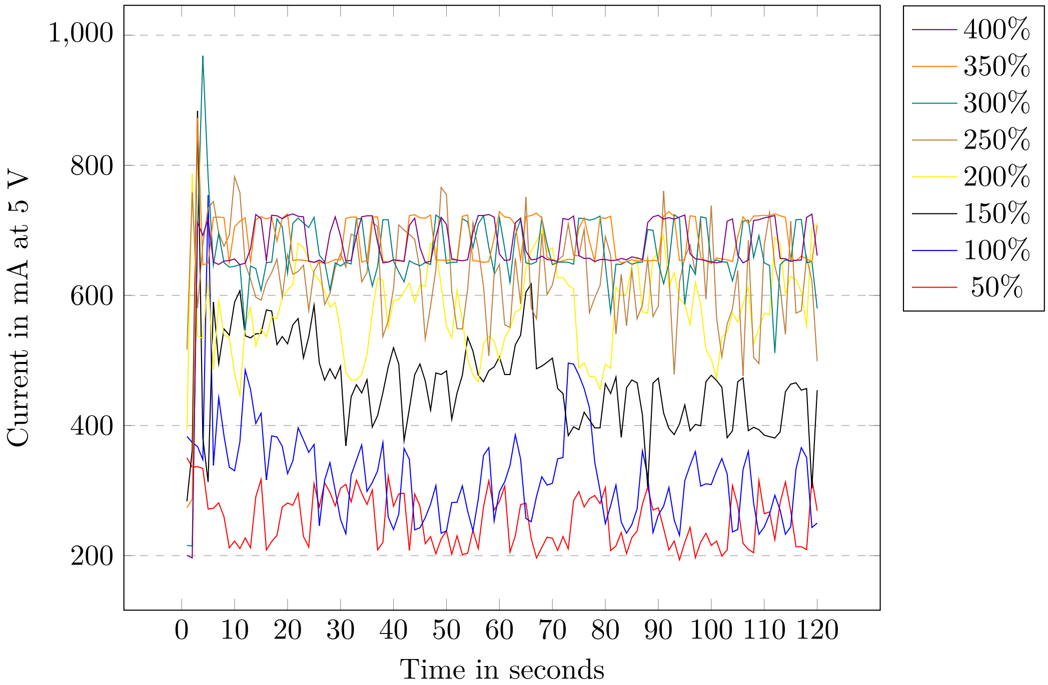

Figure 4.2: Power usage under load with `cpulimit`

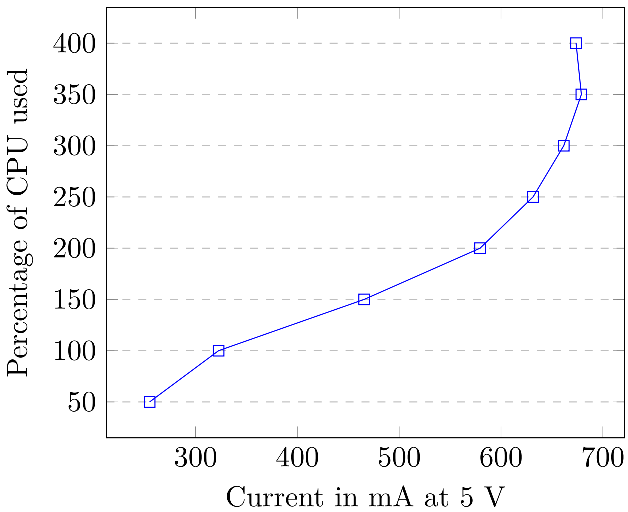

Figure 4.3: Mean power usage under load with `cpulimit`

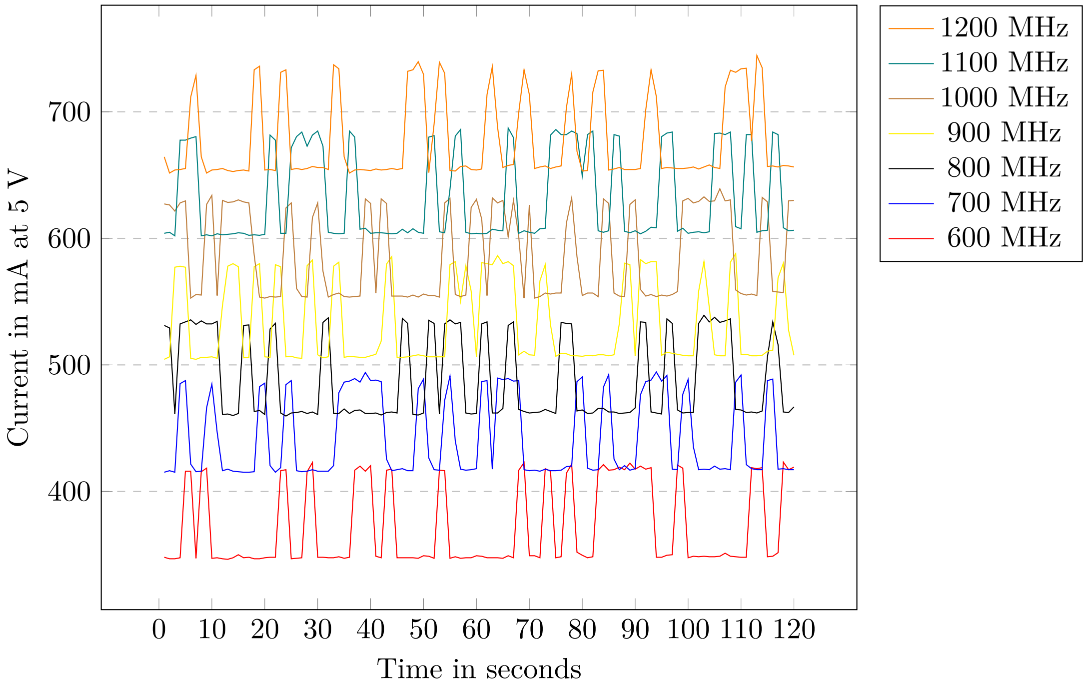

Figure 4.4: Power usage under load with different frequencies

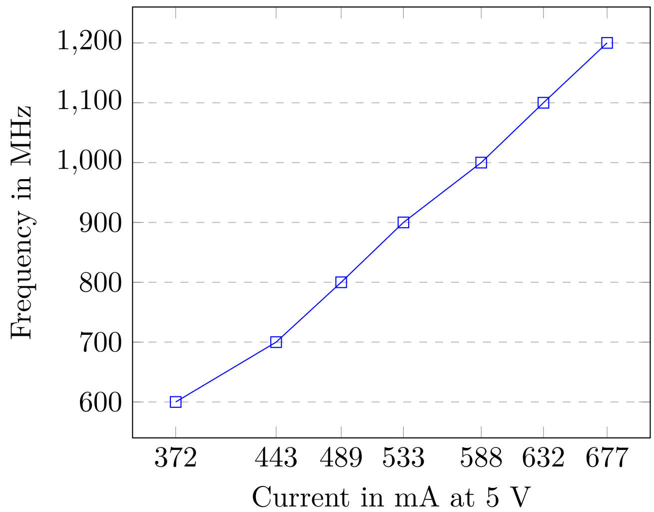

Figure 4.5: Mean power usage under load with different frequencies

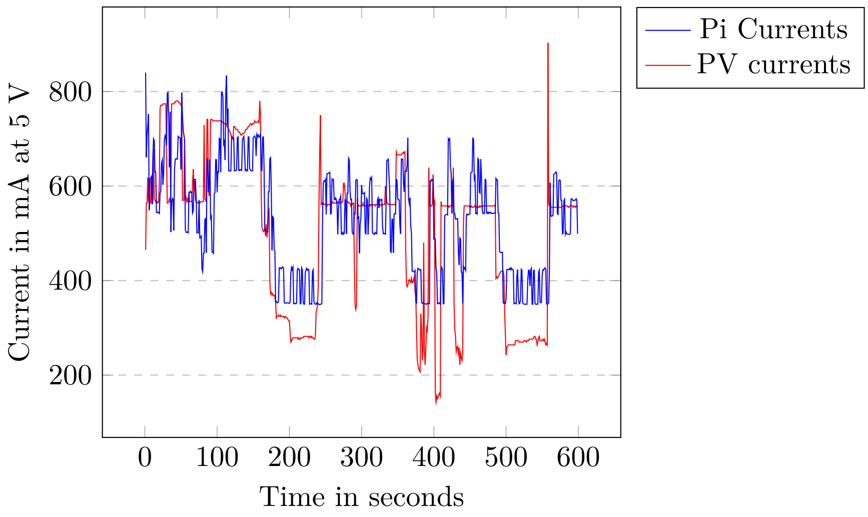

Figure 4.6: aware

4.4.3 The Approach

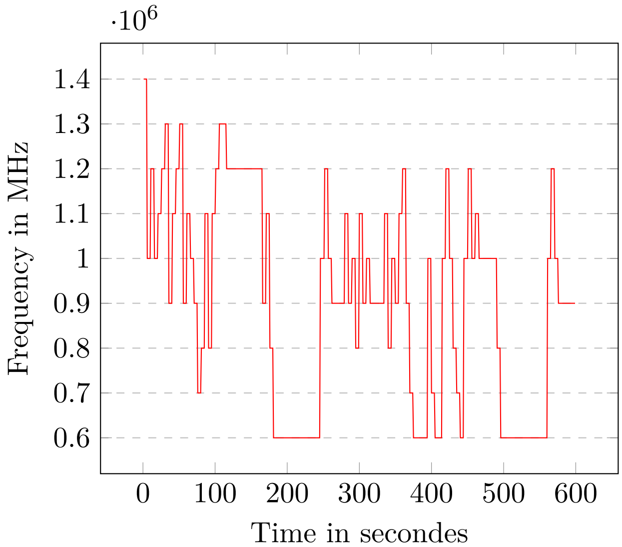

Figure 4.7: aware freqs

Chapter 5

Evaluation

Chapter 6

Conclusion

Bibliography

[1] G. Premsankar, M. Di Francesco, and T. Taleb, "Edge computing for the internet of things: A case study," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1275–1284, 2018.

[2] J. Tu, L. Lu, M. Chen, and R. K. Sitaraman, "Dynamic provisioning in next-generation data centers with on-site power production," in *e-Energy '13: Proceedings of the fourth international conference on Future energy systems*, p. 137–148, Association for Computing Machinery, 2013.

[3] G. Rostirolla, L. Grange, T. Minh-Thuyen, P. Stolf, J. Pierson, G. Da Costa, G. Baudic, M. Haddad, A. Kassab, J. Nicod, L. Philippe, V. Rehn-Sonigo, R. Roche, B. Celik, S. Caux, and J. Lecuivre, "A survey of challenges and solutions for the integration of renewable energy in datacenters," *Renewable and Sustainable Energy Reviews*, vol. 155, p. 111787, 2022.

[4] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *Computer*, vol. 40, no. 12, pp. 33–37, 2007.

[5] P. Wiesner, D. Scheinert, T. Wittkopp, L. Thamsen, and O. Kao, "Cucumber: Renewable-aware admission control for delay-tolerant cloud and edge workloads," in *Proceedings of the 28th International European Conference on Parallel and Distributed Computing*, Euro-Par, Springer, 2022.

[6] A. Karimiafshar, M. R. Hashemi, M. R. Heidarpour, and A. N. Toosi, "Effective utilization of renewable energy sources in fog computing environment via frequency and modulation level scaling," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 10912–10921, 2020.

[7] A. Toor, S. ul Islam, N. Sohail, A. Akhunzada, J. Boudjadar, H. A. Khattak, I. U. Din, and J. J. Rodrigues, "Energy and performance aware fog computing: A case of dvfs and green renewable energy," *Future Generation Computer Systems*, vol. 101, pp. 1112–1121, 2019.

[8] P. Wiesner and L. Thamsen, "LEAF: Simulating large energy-aware fog computing environments," in *2021 IEEE 5th International Conference on Fog and Edge Computing (ICFEC)*, 2021.

[9] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and experience*, vol. 41, no. 1, 2011.

[10] R. Mahmud, S. Pallewatta, M. Goudarzi, and R. Buyya, "Ifogsim2: An extended ifogsim simulator for mobility, clustering, and microservice management in edge and fog computing environments," *Journal of Systems and Software*, p. 111351, 2022.

[11] K. Keahey, J. Anderson, Z. Zhen, P. Riteau, P. Ruth, D. Stanzione, M. Cevik, J. Colleran, H. S. Gunawi, C. Hammock, J. Mambretti, A. Barnes, F. Halbah, A. Rocha, and J. Stubbs, "Lessons learned from the chameleon testbed," in *2020 USENIX Annual Technical Conference (USENIX ATC 20)*, pp. 219–233, USENIX Association, 2020.

[12] F. Xia, L. T. Yang, L. Wang, and A. Vinel, "Internet of things," *International journal of communication systems*, vol. 25, no. 9, p. 1101, 2012.

[13] Cisco Systems, "Annual internet report (2018–2023)," white paper, Cisco Systems, 2020.

[14] R. Buyya and S. N. Srirama, *Fog and edge computing: principles and paradigms.* John Wiley & Sons, 2019.

[15] M. Iorga, L. Feldman, R. Barton, M. Martin, N. Goren, and C. Mahmoudi, "Fog computing conceptual model," tech. rep., Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, 2018.

[16] Atos SE, "Spec-sheet bullsequana edge," tech. rep., Atos SE, 2019.

[17] Atos SE, "Quality control in production line," tech. rep., Atos SE, 2020.

[18] S. Svorobej, P. Takako Endo, M. Bendechache, C. Filelis-Papadopoulos, K. M. Giannoutakis, G. A. Gravvanis, D. Tzovaras, J. Byrne, and T. Lynn, "Simulating fog and edge computing scenarios: An overview and research challenges," *Future Internet*, vol. 11, no. 3, 2019.

[19] SwitchDoc Labs, *SunControl - Advanced Solar Controller / Charger / Sun Tracker / Data Gathering Grove/Header.* SwitchDoc Labs LLC, 2018.