

The Impact of Process Competition on Energy Consumption: Analysis and Modeling

Eduardo Gomes Campos
Polytechnic School of the University
of São Paulo
São Paulo, SP, Brazil
eduardogc0303@usp.br

Rafaela Sousa de Alencar
Lacerda
Polytechnic School of the University
of São Paulo
São Paulo, SP, Brazil
rafaelalacerda@usp.br

Adnei Willian Donatti
Polytechnic School of the University
of São Paulo
São Paulo, SP, Brazil
adnei.donatti@usp.br

Joberto S. B. Martins
Universidade Salvador (UNIFACS)
Salvador, BA, Brazil
joberto.martins@animaeducacao.com.br

Charles C. Miers
Santa Catarina State University
Joinville, SC, Brazil

Tereza C. M. B. Carvalho
Polytechnic School of the University
of São Paulo
São Paulo, SP, Brazil
terezacarvalho@usp.br

Abstract

With the development of distributed systems, the need to manage the sharing of machines among multiple simultaneous users arises. In the cloud computing context, the instantiation of virtual machines and containers by different users utilizing the same infrastructure leads to a dispute for physical computational resources. In this regard, this paper analyses a process's energy consumption as a function of the competition for computational resources it encounters. Investigating this behavior is fundamental for many applications, such as pricing in cloud computing services, and for task scheduling and load balancing, while increasing energy efficiency. To determine this behavior, experiments were conducted and resulted in a dependency on the number of processor cores of the physical machine hosting the process. As the number of cores increases, the process's energy consumption as a function of the competition it faces transitions from linear to a root function.

CCS Concepts

• **Hardware** → **Energy metering.**

Keywords

Energy Consumption, Resource Competition, Process, Kubernetes

ACM Reference Format:

Eduardo Gomes Campos, Rafaela Sousa de Alencar Lacerda, Adnei Willian Donatti, Joberto S. B. Martins, Charles C. Miers, and Tereza C. M. B. Carvalho. 2026. The Impact of Process Competition on Energy Consumption: Analysis and Modeling. In *ADVANCE: International Workshop on ADVANCEs in ICT Infrastructures and Services, March 25–28, 2026, Florianópolis, Brazil*. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/nmnnnnn.nmnnnnn>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ADVANCE, Brazil

© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
<https://doi.org/10.1145/nmnnnnn.nmnnnnn>

1 Introduction

With the development of distributed systems, handling physical machines shared by multiple users at once is necessary. For instance, in the context of cloud computing, the process of instantiating virtual machines causes a scenario in which computational resources are being competed for by multiple individuals. However, this utilization is usually measured according to the machine specifications and period of use, disregarding how each user uses those resources. Nowadays, energy consumption and sustainability are at high stakes since the world is trying to become more aware of its environmental impact. Since energy consumption has become a relevant cost in cloud computing data centers [30], there is an urge to assess the environmental impact of computer usage, and one way is to measure its energy consumption.

One topic worth mentioning is how the competition for Central Processing Units (CPUs) resources affects the power consumption of a process. In this context, this article proposes an approach to analyze the energy consumption at the process level in these scenarios, which unveils the possibility of modeling process power. In this context, we perform experiments to measure how much energy a process would consume as it faces competition for CPU resources. With that data available, we correlate how high the competition is and how much power the main process consumes. The final objective is to obtain a function $\mathcal{W}(p)$ to represent the power consumed by the process. In this case, p represents the percentage of the CPU used by the competition and, therefore, $p \in [0, 100 - q]$, given that q represents the percentage of the CPU used by the analyzed process.

To establish a reliable $\mathcal{W}(p)$ function, we comprehend diversified variables that might cause an impact on the obtained metrics. Per [14], several factors might impact the power consumption on a computer besides CPU usage. Despite the processor being highlighted as the main energy consumer according to [7], elements such as fans, memory, and disk usage can also have their share of effects on this topic. Moreover, there have been studies that modeled those variables as energy consumption [21]. Thus, we replicate those tests in machines with different hardware configurations to get feasible results.

When it comes to real-world applications, this paper suggests a model that could be used in many scenarios. As it tracks energy consumption at the process level, the function can estimate how much energy will be consumed by a single application running on the computer. For instance, in the context of cloud pricing, it is common to see the hardware selected by the user as the main definer of how much will be charged [1]. However, with our model, the cloud provider could differentiate how each user requires resources from the machines (by tracking its processes). This could lead to different and more accurate prices for each individual. Besides, other interesting applications might be related to large energy forecast models. As an example, during the instantiation of Virtualized Network Functions (VNFs), a theme of interest has been finding ways to allocate them with energy efficiency criteria [34]. With our model, it is possible to estimate how much energy a VNF will consume given the competition for CPU resources on a certain machine. Hence, this could be used to optimize the placement of these functions to reduce total energy consumption.

The rest of the paper is organized as follows: In Section 2, we explain the problem to which our solution will be applied, clarifying on which domains our work collaborates with the state-of-the-art. Section 3 discusses the state-of-the-art models for energy consumption regarding CPU usage and delves into theoretical definitions for our applications. Section 4 we bring up papers that are related to our work on topics such as resource utilization modeling in virtualized contexts, the influence of hardware on energy consumption, etc. These will be presented as theory background to give the necessary presumptions to set up the experiments and to justify the obtained conclusions. Additionally, Section 5, describes the methodology utilized to develop the experiments utilized for the model construction. Section 6 describes both the experiment environment and its main tools. Moreover, Section 8 showcases the results achieved with those tests graphically and discusses the thought process with each experiment’s results. Section 9 concludes the article by summarizing the conclusions obtained from the experiments’ results.

2 Problem Definition and Motivation

In this context, describing problems associated with process resource sharing is imperative. Among those, we highlight the instantiation of VNFs in a network-slicing context [26] [12]. With the development of 5G mobile networks, one of its core components is Network Slicing (NS), which proposes a dynamic provisioning system of networking resources (and functions) to achieve compliance with user needs [31] [5]. In this situation, one relevant theme of research has been the placement of VNFs to achieve high energy efficiency. In this matter, previous work [34] [29] [33] has used models to estimate energy consumption and use this information to apply statistical and algorithmic approaches to provide techniques to optimize the placement of such functions.

Nevertheless, most models do not specify the power usage of a single function but rather analyze the power dissipation of the system as a whole. That is, we cannot analyze the consumption of each VNF and, thus, it becomes harder to estimate the energy consumption of a single slice. Therefore, describing a process’s energy consumption within the context of the entire system is

essential since it would allow measuring the power dissipation of each of its components.

We focus on assessing how a process’s power dissipation behaves when it faces competition for CPU resources. Thus, we describe it as a function $\mathcal{W}(p)$, given that p stands for the competition the process is facing at a certain instant. The behavior of a process’s power dissipation is a key factor for VNF placement heuristics and algorithms generating a network topology that guarantees energy efficiency and balance among slices, for instance. Therefore, this showcases the importance of such a description.

In this sense, we proceed by formulating a sample VNF placement problem described as: let \mathcal{M} be a set composed of different machines that would be used to create Network Slices. Then, \mathcal{S} is a set of the created slices and \mathcal{F} is a set of VNFs to be placed on these machines. Thus, each element from \mathcal{F} belongs to a certain element from \mathcal{S} , since, in this problem, a VNF composes a fraction of a slice running in a certain machine.

To exemplify such a problem, we describe it with Figure 1. In this situation, we have 2 network slices (S1 and S2) built with 3 different machines (M1, M2, M3). Each machine hosts a few network functions (F1, F2, ..., F6, F7), and each of them either belongs to S1 or S2, as previously suggested with the problem definition.

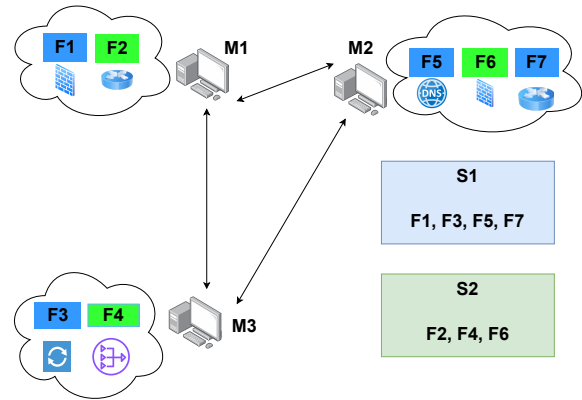


Figure 1: Diagram representing a topology for the NS

$$\mathcal{M} = [m_1, m_2, \dots, m_a] \quad (1)$$

$$\mathcal{S} = [s_1, s_2, \dots, s_b] \quad (2)$$

$$\mathcal{F} = [f_1, f_2, \dots, f_c] \quad (3)$$

Then, we define the power dissipated by a single VNF as $W(f_j)$, $j \in [1, c]$, the power dissipated by each slice as $P(s_i)$, $i \in [1, b]$, and the power dissipated by whole slice group as $P'(\mathcal{S})$:

$$P(s_i) = \sum_{j=1}^c q_{i,j} \cdot W(f_j) \quad (4)$$

$$P'(\mathcal{S}) = \sum_{i=1}^b P(s_i) = \sum_{i=1}^b \sum_{j=1}^c q_{i,j} \cdot W(f_j) \quad (5)$$

$$q_{i,j} = \begin{cases} 1 & \text{if } f_j \in s_i \\ 0 & \text{else} \end{cases} \quad (6)$$

As we wish to minimize the power consumption (4) of all the slices in \mathcal{S} , there is an urge to develop a description for $W(f_j)$. Since VNFs could be initially placed in any machine, we suggest the following structure for its power consumption $W(f_j)$, given that p_{mk} represents the competition for resources on a certain machine m_k :

$$W(f_j) = \text{minimum}[\mathcal{W}(p_{m1}), \mathcal{W}(p_{m2}), \dots, \mathcal{W}(p_{ma})] \quad (7)$$

This means that one can choose which machine m_k to allocate a VNF based on how the competition for resources in each m_k affects the VNF's power consumption. With this construction, the description we found for $\mathcal{W}(p)$ is useful in optimization problems for the placement of VNFs, considering the competition for processing resources. Moreover, it would allow a per-slice energy analysis even if s_i shares the same physical machine with other NS. Thus, displaying the potential of our research. With this motivation, this paper aims to answer the following research questions:

- Q1: How to describe Energy Consumption mathematically?
- Q2: How do CPU usage and energy consumption correlate?
- Q3: How to estimate the energy consumption of a process?
- Q4: How does competition for resources affect the energy consumption of a process?

By the end of our research, we expect to prove or refute the following hypothesis: the energy consumption of a constant process is constant, regardless of the competition for resources it faces.

3 Background

Energy is the basic resource needed for performing human activities in the current century, especially in the field of computing. However, its definition is easily misunderstood for that of *Power*, and it is important to differentiate those terms. Energy (E) represents the total work done by a system over a specified period of time (T), as shown in (8), whereas power (P) refers to the rate at which the system performs that work [9]. Therefore, power can be defined as the infinitesimal amount of energy consumed or produced in a very small time interval, as shown in (9). In computer systems, it is common to use power instead of energy when analysing the consumption profile. Since power is an instantaneous quantity, it offers more precise insights into how energy consumption varies dynamically over time.

$$E = \int_0^T P(t) dt \quad (8)$$

$$P(t) = \frac{dE(t)}{dt} \quad (9)$$

The energy consumption of a computer derives from several of its components, such as CPU, memory, storage, fans, and the network interface card. However, as the CPU consumes the most energy compared to other components, it is common in power modeling to take the energy consumed by the processor as representative of the entire machine's consumption [9] [15] [16]. Work [7] also highlights that the energy consumed by memory and network devices is insignificant compared to that of the CPU.

The relationship between power consumption and CPU usage is a common subject of study in the computing field [9] [13]. In this context, several studies have been conducted over the years to determine what this relationship is. Even though some studies propose that different functions (such as polynomial and non-linear [25] [22]) can characterize the relationship between energy consumption and CPU usage, it is generally accepted that this behavior could be linearly represented. The research in [7] has significantly influenced power modeling for data centers and proposes this linear correlation and also the model (10) to represent it mathematically. In (10), P_u is the power consumption of the server as a function of the CPU usage, u . P_{idle} and P_{max} are constants, the average power consumption when the server is idle and at its maximum capacity, respectively.

$$P_u = (P_{max} - P_{idle})u + P_{idle} \quad (10)$$

Studies such as [16] have shown that this linear model can accurately describe a server's power consumption. Therefore, as a way of simplifying hands-on measurement, this article will consider the CPU as the only component responsible for a machine's energy consumption. We will also consider the relationship between energy consumption and CPU usage as linear, as it is classically considered by literature [9] [13].

However, this state-of-the-art relationship is only representative of the energy consumption of a computer, and cannot be used at the process level. The survey [9] extensively reviews power models and finds that only [27] proposes a model for the energy consumption of a process. Moreover, [27]'s model doesn't consider the CPU usage, and does not provide a relationship between that and the energy consumption of a process. Therefore, literature does not provide a model for the relationship between a process's energy consumption and the machine's CPU usage. In this context, this paper aims to empirically find such a model that works at the process level.

Virtualization technologies allow multiple activities to be run independently on the same physical machine, facilitating isolated environments that are essential for accurate performance and energy consumption analysis [17]. VMs, a hypervisor-based type of virtualization, provide robust security and isolation at the hardware level [36]. However, the overhead associated with VMs can significantly impact the energy consumption profile of individual processes, thereby complicating precise measurement and analysis [4].

Containers, though not as secure, are lightweight and can be rapidly instantiated [4] [36]. They also provide sufficient isolation for most application scenarios and have a negligible impact on the host machine's energy consumption [23]. This makes containers particularly suitable for performance isolation and energy consumption research. Furthermore, Kubernetes is an open source system for automating deployment, scaling, and management of containerized applications [20]. The dynamic resource allocation it implements in containerized environments enhances the efficiency and flexibility of resource usage [10]. This further mitigates the overhead concerns typically associated with virtualization.

By using containerization, we can more accurately assess the energy consumption of individual processes, ensuring that the overhead introduced by the virtualization layer remains minimal and

Table 1: Summary of comparison between papers

Research questions	[35]	[14]	[37]	[32]	This work
Q1	✗	✓	✗	✓	✓
Q2	✓	✓	✓	✓	✓
Q3	✗	✗	✗	✗	✓
Q4	✗	✗	✗	✗	✓

does not distort the measurements. Thus, we chose containerization to investigate the energy consumption of a process, benefiting from the low overhead, rapid deployment, and effective resource management capabilities inherent to this technology.

The phenomenon of two or more simultaneous processes competing for computational resources is called “resource competition” [23]. When multiple containers operate on the same physical machine, they compete for processing resources. Thus, there is resource competition associated with processing resource sharing, and it is imperative to learn how that affects the energy consumption of each container. However, the container’s energy consumption can be broken down to the process level. Therefore, learning the relationship between a process’s energy consumption and the resource competition it encounters enables the development of generalized solutions. These solutions would not only be applicable to containerized environments but could also extend to other process-based applications.

4 Related Work

When looking for work related to ours, our search method consisted of searching the IEEEExplore platform with the following advanced search command, with keywords related to our research: (“All Metadata”:energy consumption) AND (“All Metadata”:monitoring) AND (“All Metadata”:virtualization) AND (“All Metadata”:cpu usage) OR (“All Metadata”:energy efficiency)). Filtering the articles from the last 9 years, from 2015 to 2024, we found that [35] and [37] were the resulting papers that related best to our work. Besides, we found two other articles that relate to our work, [14] and [32], when reviewing literature and reading through the survey [9]. This survey is relevant in the Energy Consumption Modeling field, since it thoroughly reviews models for energy consumption in data centers. [9] has 410 citations and is cited by 733 papers.

Work [37] uses test scenarios to assess how the energy consumption of Virtual Machines (VMs) behaves when sharing virtual network infrastructure and CPU cores. Throughout the article, it evaluates how the distribution of network traffic between VMs results in different values of power being dissipated by the machines altogether. Moreover, it also delves into how the competition for processing resources at CPU cores generates different energy consumption for the VM cluster. Nevertheless, unlike our contribution, this paper does not analyze the power consumption at the process level, only from the point of view of the VM.

Besides, [14] and [32] provide a similar analysis of how power consumption correlates to CPU utilization. This correlation is assessed by executing experiments in which they measure the total power consumed by the PC as it faces a gradual increase in CPU utilization over time. Afterward, [14] shows how linear and non-linear empirical functions can describe the behavior of the power

dissipated on similar machines. On the other hand, [32] uses a polynomial function to approximate $W(p)$. Still, differentiating from our work, the first one does not consider different hardware configurations, and neither of them assesses process-level energy consumption.

Furthermore, [35] introduces a study into the power consumption behavior of Docker containers. This article investigates how the containers’ instantiations create energy overhead and how their power dissipation behaves when facing gradually increasing loads over time in different applications. For example, they run containerized versions of Nginx servers and submit them for a test that increases the number of requests throughout the experiment, establishing a scenario of competition for processing resources related to our work. Even though this paper assesses energy consumption in containers and includes CPU-sharing elements, it does not propose any process-level power representation or model.

Finally, Table 1 presents a summary of the contributions of each paper mentioned alongside our production. Although the CPU use and energy consumption relation is a well-discussed topic in the literature, it remains to be introduced an analysis of process-level energy consumption regarding the total use of the CPU. Hence, this paper proposes a study that complements previous work done on this subject.

5 Proposal

To solve the problem of describing the impact of competition on the energy consumption of a reference process (a VNF, for instance), we propose an empirical method. In this sense, our method consists of running various experiments and observing how the power dissipation of a specific process behaves by collecting data on certain metrics. In the end, with extensive experimentation and data collection, we can describe the behavior of power dissipation. Accordingly, our method relies on three pillars:

- **Baseline Process:** We establish a baseline process - the baseline process has a constant workload and serves as a reference for the competition.
- **Observed Metrics:**
 - resource usage - we observe the resource usage (more specifically, CPU) of the machine and of the baseline process. This allows us to measure the intensity of competition through its resource usage; and
 - power consumption per process - we gather data on the power consumption of the baseline process.
- **Experimentation Scenarios:** we define scenarios in which we change variables, such as hardware (experiments are reproduced in different machines with different hardware); active cores (within the same machine); and competition

growth (e.g., none, gradual increase). This allows us to investigate how each of these variables affects the observed metrics.

6 Experiments environment

Machine name	CPU	Total RAM available	Disk Space available	Number of Threads
Controller	Intel(R) Core(TM) i7-4770	7.67 GB	1 TB	8
Worker 1	Intel(R) Core(TM) i5-3330	15.5 GB	1 TB	4
Worker 2	Intel(R) Core(TM) i7-2600	15.5 GB	500 GB	8
Worker 3	Intel(R) Core(TM) i7-2600	7.63 GB	500 GB	8
Worker 4	Intel(R) Core(TM) i5-8500	3.65 GB	1 TB	6
Worker 5	Intel(R) Core(TM) i7-2600	7.63 GB	500 GB	8

Table 2: Machines configurations

To correlate CPU competition usage and process energy consumption, it is necessary to execute tests to attempt to fit the obtained data into a mathematical model that represents this correlation. To achieve this, we created an environment in which we could carry out these experiments. In this context, we assembled 6 machines (hardware configurations in the table 2) and used them to create a Kubernetes Cluster (Client version 1.28.1 and Server version 1.28.6).

Creating a cluster allows automating tests on multiple machines with centralized telemetry capabilities. In this sense, we deploy pods running Docker images with specific test routines so that they can be run as pods on each computer. When it comes to this cluster architecture, the machines are labeled as Controller or Worker. The Controller is responsible for managing the cluster. That is, it makes sure the requirements defined by the user (such as scheduling pods and guaranteeing connection between nodes) are met. On the other hand, the Worker nodes are responsible for hosting the pods running the test routines. To execute the tests and fetch the data obtained from them, we created a four-agent scheme: Scaphandre pods to extract energy information from the Worker nodes, a Prometheus agent so that this data could be exported, a Grafana agent to plot graphs with the exported data, and a Python client to interact with the Controller API.

Then, a tool with process-level power telemetry is needed to gather the goal data. Even though there have been previous software that provided these capabilities [11] [8], Scaphandre [28] stands out due to its natural compatibility with Kubernetes and Prometheus, making it more suitable for our distributed measurements. By using the Powercap RAPL sensors integrated into Intel CPUs (which have

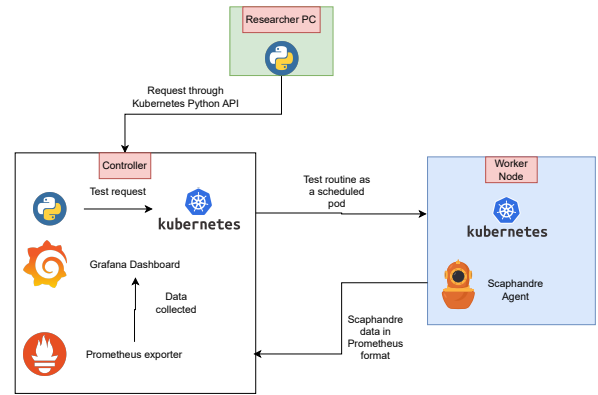


Figure 2: Diagram representing the test environment and its main agents

been previously validated and used to assess their power consumption [24]), it can track the power dissipated by the host as a whole and estimate energy consumption by processes. Thus, by selecting a process on the machine (by tracking its PID, for instance) the agent can estimate how much energy the chosen element consumes, enabling the aforementioned process-level analysis.

Moreover, the Prometheus and Grafana agents act together to obtain this power information and expose it graphically on time-series dashboards (over the test duration). Finally, the Python client consists of a tool to interact with the controller through the Kubernetes API. Through this interaction, we could run tests remotely by authenticating and deploying the test pods.

7 Baseline

In order to validate our method, we ran two initial experiments. In both, we analyzed the power consumption of a baseline process, but in different scenarios. Through the command-line tool for Kubernetes, kubectl [19], we were able to use the Stress tool [3] to generate 2 constant loads, that is, the baseline process and its competition. These two tests were run on Worker 1. Scaphandre scraped power consumption data, which was stored in Prometheus, and Grafana was used to export it to a .csv file.

The first scenario was of no competition for resources, that is, with only the baseline process consuming resources (in this case, CPU). We found that its power consumption (shown in Figure 3) varies from 8.7 to 9.8 Watts, which is a 11.9% variation of the average value, proving that the consumption of the baseline process is in fact constant. Then, the second scenario consisted of running another process alongside the baseline, and identical to it. After some time, the second process was deactivated, enabling comparison between the first scenario (no competition) and the second (with competition). In Figure 4, the sample on the left side, with an average value of 12.5 Watts, corresponds to Scenario 2, and the sample on the right, with an average of 9.75 Watts, corresponds to Scenario 1.

This proves that our method enables us to describe the impact competition has on the energy consumption of a reference process,

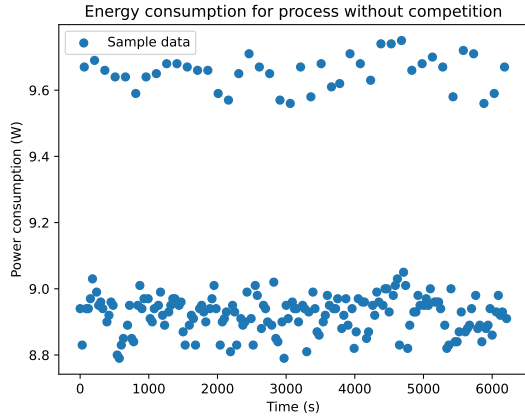


Figure 3: Power consumption of a constant process

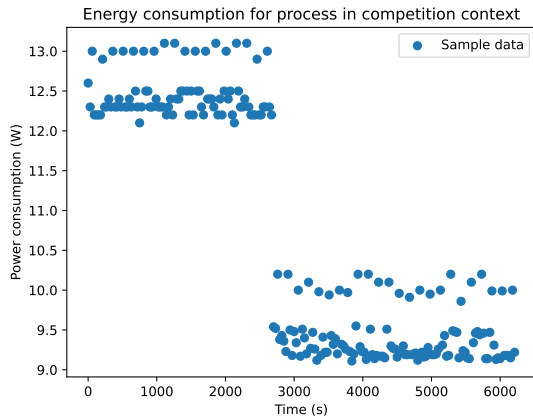


Figure 4: Power consumption with and without competition

through empirical means. Furthermore, this raises the possibility that competition increases the energy consumption of a baseline process, contradicting our initial hypothesis that the energy consumption of a constant process is constant. With this motivation, we ran more experiments in order to study the behavior of a process’s energy consumption as it faces resource competition.

8 Experiments and Results

Figure 5 displays sequentially the tests routine. As Section 6 described, we initially code the test routine and, with the communication with the Kubernetes Python API, the Controller is able to fetch the necessary information to set up the test. In this situation, the API, through kubectl commands, is able to create the necessary jobs and services to run the test. Afterwards, the Kubernetes agents on the Controller and Worker nodes are able to set up the pods with the test scripts. Besides, they also communicate in order to get the energy data collected by Scaphandre in the Worker running the routine. These sets are sent to the Controller, which stores them with the Prometheus volume. After the test is finished, the data

stored in Prometheus is then compiled in a .csv file by the API, which is sent to the Researcher PC. The information is filtered and processed with statistical analysis, giving the final results, which are discussed in this section.

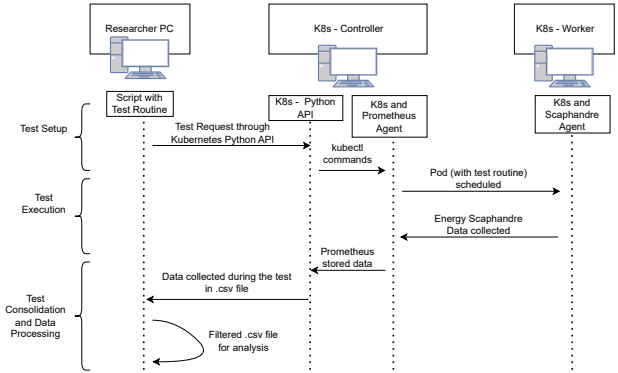


Figure 5: Sequential Diagram describing the test routine

To assess the competition effects on CPU energy consumption, we used both the CpuLimit [2] and Stress [3] tools to generate processing loads and test routines. While the Stress tool generated the load itself (such as running a C script in a loop), CpuLimit was responsible for controlling the load to achieve the expected percentage of usage required for the investigation.

Regarding the test routine, their main structure consisted of running a main process with a constant CPU load and gradually increasing the competition by instantiating smaller processes over time. During the experiment, the main process is monitored, and its power consumption is exposed in Grafana dashboards. Then, the plotted data was analyzed statistically to evaluate whether the data would fit in a specific model for $\mathcal{W}(p)$.

8.1 Resource Competition Experiment - Gradual Increase

Initially, we proposed the previously detailed routine: the main constant process facing escalating competition for CPU resources while getting its energy measured. In our investigation, we propose that the competition starts at 0% and increases 5% at a time every 6 minutes (to gather enough data targeting reducing the effects of outliers), until the total processor usage reaches approximately 100%. These cycles were repeated 8 times to reduce the effect of possible outliers and the obtained data was saved in a CSV file. The obtained data was saved and published at Zenodo for public access[6]. Then, by using a Python program that receives a math model from the user (such as linear, quadratic, cubic, etc.), we fit the data into the model and evaluated its correlation by using a t-test. To get started, we executed this experiment on all Worker machines.

After plotting the data, we proposed that the main process behavior could be represented by either a linear or an n-root function. For Workers 1 and 4 (Figure 6 and Figure 8, respectively), the result was better described by a linear profile. On the other hand, Workers 2 (Figure 7), 3 (Figure 8), and 5 (Figure 10) were more accurately

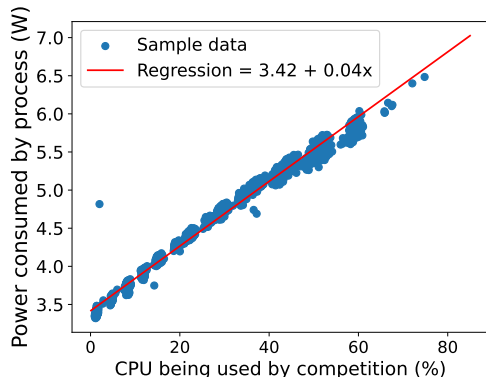


Figure 6: Worker 1 results

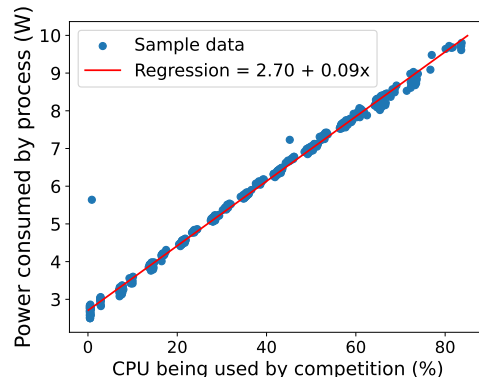


Figure 9: Worker 4 results

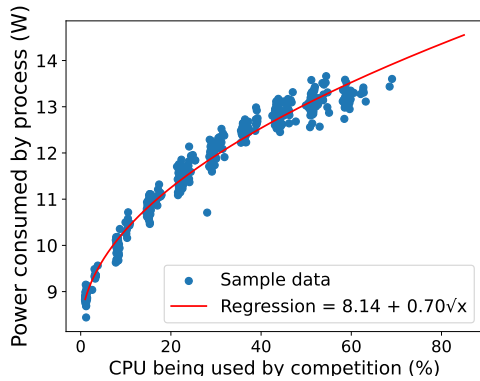


Figure 7: Worker 2 results

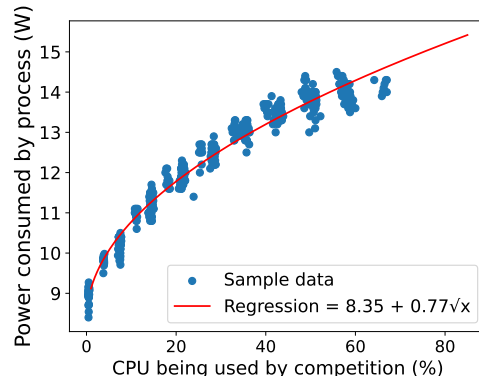


Figure 10: Worker 5 results

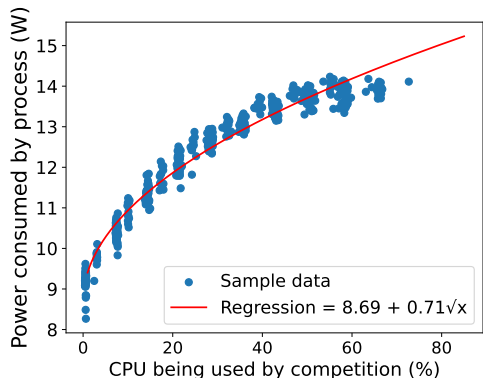


Figure 8: Worker 3 results

pictured by an n-root function. Besides, it is known from Table 2 that Workers 2, 3, and 5 have the same number of virtual cores on their CPUs (8), whilst Workers 1 and 4 have a lower number (4 and 6, respectively). As there have been previous studies that proposed a correlation between the number of active threads and energy usage [18], we investigate the relationship between the number of virtual cores and the power consumption profile.

8.2 Resource Competition Experiment - Gradual Increase with capped CPU

Then, we suggested executing the same tests on 8-thread worker machines to investigate the effect of vCPUs. However, before running them, we would deactivate some of the cores to simulate a 4 or 6-thread computer and see if the function profile becomes linear. The resulting dataset from these experiments is also publicly available in Zenodo [6].

When reducing the number of threads from 8 to 4 for the machines Worker 2 (Figure 11), 3 (Figure 12), and 5 (figure 13), we got a linear profile for $\mathcal{W}(p)$, just like the results for Worker 1 (the machine with 4 cores originally). On the other hand, the results for capping the resources to 6 threads were mixed.

We executed this version of the test on Worker 2 and 3, and then, after gathering the data, we found out, as displayed in Figures 15 for Worker 2 and 16 for 3, that the best model for fitting this information was the n-root (differentiating from the original 6-core machine - Worker 4 - that provided a linear profile). However, we also did linear fits, and the t-statistics were not significantly different from those with the n-root model (even though they were slightly worse). This is graphically showcased by Figure 14 for Worker 2 and by Figure 17 for Worker 3. In this scenario, we propose that there is a

correlation between the number of virtual cores and the profile for $\mathcal{W}(p)$.

8.3 Analysis & Discussion

With the obtained data, it is possible to draw conclusions related to the $\mathcal{W}(p)$ profile and the machine’s CPU. First of all, for all machines with an n-root behavior, it is worth noting that not only was the function’s type to represent the power consumption the same, but also the parameters were very similar (that is, a and b on $f(x) = a + b\sqrt{x}$). On Workers 2,3 and 5, the variation in a did not surpass 7%, while for b , the maximum difference reached the 10% threshold. With that, as these machines had significant differences in available RAM and disk space, it seems that the CPU type overrules any of these hardware variations since their function $\mathcal{W}(p)$ had approximately the same parameters. This fact confirms the affirmation made in previous studies [7] [9] that the CPU can be considered the most relevant source of energy consumption on a machine. Moreover, regarding the thread’s discussion, we propose a correlation between the function profile and the number of virtual cores available. In this case, we suggest that for lower numbers of cores, the $\mathcal{W}(p)$ behavior is linear (such as the results for Worker 1 and capped results for Workers 2,3 and 5 suggest). As the number of these components increases, they gradually progress to an n-root behavior as they reach 6 cores (Worker 4 provided a linear pattern, whilst capped versions of Workers 2 and 3 had a mixed shape). Then, as it reaches higher quantities of cores (up to 8, on our experimentation), the profile shifts completely to an n-root function (as Workers 2,3 and 5 results propose). Following this line, it is possible to assume that most modern hardware would follow the latter version of $\mathcal{W}(p)$.

A conclusion taken from this is that the power consumption of a process behaves differently on each machine. Then, when deciding where to instantiate a process (regarding attempting to use less energy), the fact that some machines have a linear or n-root profile brings intricacies when choosing the PC in which the process will dissipate less power. To understand them, we shall propose an explanation of the possible $\mathcal{W}(p)$ derivatives. When taking the linear version of the function ($\mathcal{W}_{lin}(p) = a + bp$), the derivative

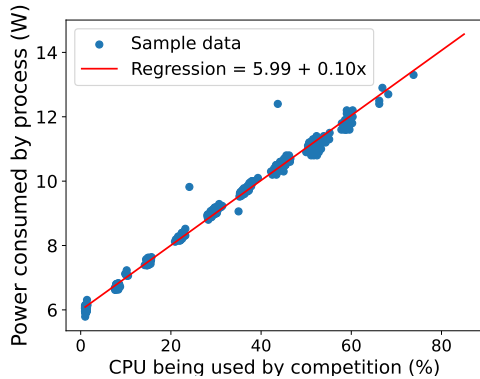


Figure 11: Worker 2 results with 4 cores

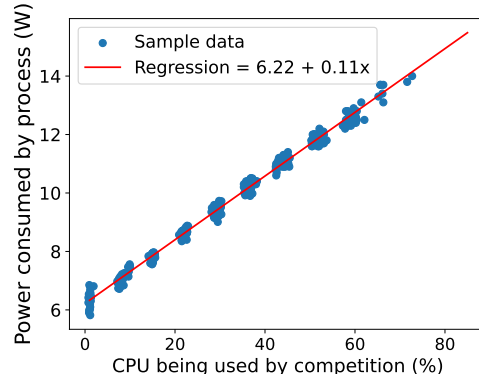


Figure 12: Worker 3 results with 4 cores

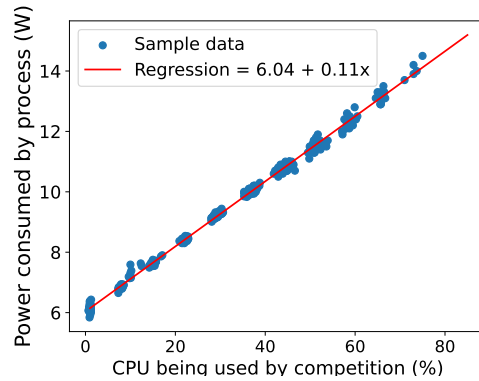


Figure 13: Worker 5 results with 4 cores

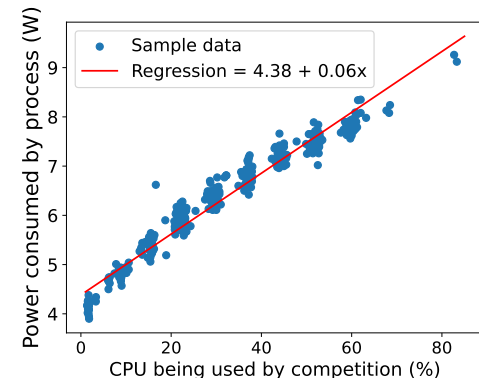


Figure 14: Worker 2 results with 6 cores and linear fit

returns:

$$\dot{\mathcal{W}}_{lin} = \frac{d\mathcal{W}_{lin}(p)}{dp} = b \tag{11}$$

Given that b is constant, the function increases at a constant rate. That is, independent of how much CPU is being used by competing processes, the process energy consumption will increase with the same

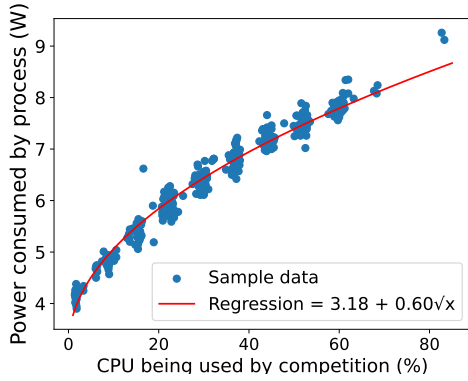


Figure 15: Worker 2 results with 6 cores and n-root fit

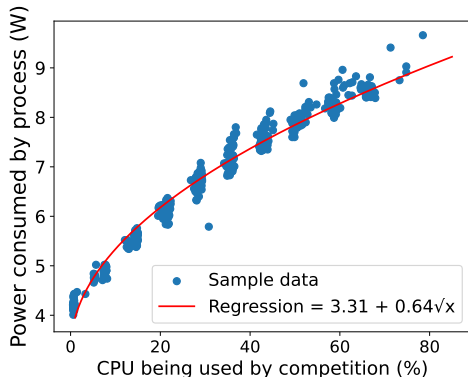


Figure 16: Worker 3 results with 6 cores and n-root fit

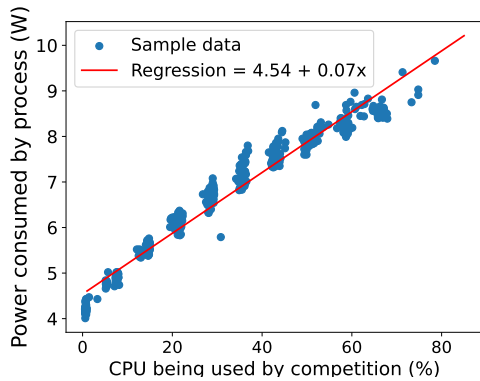


Figure 17: Worker 3 results with 6 cores and linear fit

intensity. On the other hand, with the n-root model ($\mathcal{W}_{rt}(p) = c + d \sqrt[p]{p}$), the derivative is:

$$\dot{\mathcal{W}}_{rt} = \frac{d\mathcal{W}_{rt}(p)}{dp} = \frac{d}{np^{1-\frac{1}{n}}} \quad (12)$$

In this case, the derivative depends on p . Moreover, it depends in a way that for lower values for p , the derivative is greater, whilst, for higher values, the derivative becomes tinier. To exemplify the relevance of these different behaviors, we propose a hypothetical scenario. We assume that, for a certain process, $\mathcal{W}_{lin}(p_i) < \mathcal{W}_{rt}(p_i)$ for a certain $p_i \in [0, 100 - q]$ and let $D(p) = \mathcal{W}_{lin}(p) - \mathcal{W}_{rt}(p)$. Therefore, this shows $D(p_i) < 0$, and that means it is more energetically friendly to deploy the process on the machine with the linear profile. However, the derivative of $D(p)$ shows that (assume $k = 1 - \frac{1}{n}$):

$$\dot{D} = \frac{dD(p)}{dp} = b - \frac{d}{np^k} \quad (13)$$

$$\forall p > \sqrt[k]{\frac{d}{nb}}, \dot{D} > 0 \quad (14)$$

$$\therefore \exists p_j > p_i \mid D(p_j) = 0 \quad (15)$$

Thus, conclusion 15 provides scenario ($p > p_j$) in which $D(p) > 0$. Therefore, if $p_j < 100 - q$, there will be an interval for p (that is, $]p_j, 100 - q[$) for the competition in which it will be better (energy-wise) to use the machine with the n-root behavior rather than the linear one for the same p .

This conclusion could be useful for virtualization scheduling scenarios. As previously detailed, when deciding the network topology for the instantiation of NS, the agent responsible for selecting the machine could use the $\mathcal{W}(p)$ models to assess which machine would receive the user's process in question. Alongside performance metrics, the agent would be able to balance performance and energy consumption (with the model) to achieve the necessary Quality of Service without hurting the environment as much.

Besides, such results could be used in more generalized virtualization scenarios, such as pricing the utilization of cloud resources. Instead of considering only the type of hardware that is being used, cloud companies could also assess how much energy the user's processes are consuming. In this case, enterprises could more accurately evaluate how many resources are being consumed by the user, which would lead to a value that would represent the usage better than only considering hardware.

9 Considerations & Future work

In the resource competition context, we evaluated the state of the art. We learned that the relationship between energy consumption and CPU usage at the process level was not yet consolidated. Thus, we applied an empirical method that, given its results and analysis, successfully establishes a function $\mathcal{W}(p)$ to represent the energy consumed by a process.

The conducted experiments showed that the profile of a process's energy consumption as a function of the CPU used by the competing processes is dependent on the number of CPU virtual cores. When there are 4 threads, that relationship is linear. For 6 cores, a transition between linear and n-root is seen. Then, when 8 threads are available, the behavior seen is of an n-root. Our experiments suggest that, for a higher number of threads, the n-root behavior remains.

In this study, we observe that a lower number of cores means a linear relationship between energy consumption and competition CPU usage, meaning the rate of energy consumption growth matches the rate of CPU usage growth. However, a higher number of cores leads to an n-root behavior for that relationship. This means that at low levels of CPU usage, energy consumption grows at a higher rate, whereas at high levels of CPU usage, energy consumption grows at a lower rate. This leads to the conclusion that, when the competition is at low CPU usage, it is less energy-consuming to run a process on a machine with fewer cores. On the other hand, when the competition is at high CPU usage, it consumes less energy to run a process on a machine with more cores. These results significantly impact resource allocation and pricing in cloud services, as competition should be a factor considered when allocating new resources to minimize energy consumption and when charging users.

In future work, we aim to extend our analysis to machines with a higher number of CPU threads to further validate the n-root behavior observed. Additionally, extending our research to include VMs would be highly beneficial. By measuring and modeling the energy consumption profile of a process as a function of the competition it faces within VMs, we could uncover new insights that enhance our understanding of resource management or pricing in virtualized environments.

Acknowledgments

The authors acknowledge FAPESP for supporting the thematic project SF12 - Slicing Future Internet Infrastructures (2018/23097-3 - MCTIC/CGI), as well as the Scientific Initiation proposals 2023/13381-4 and 2023/13383-7. We also thank the ANIMA Institute and CNPq (140303/2021-9) for research scholarships. Finally, we are grateful to UDESC and the LabP2D laboratory for their partnership.

References

- [1] [n. d.]. Amazon pricing for VMs. <https://aws.amazon.com/pt/ec2/pricing/on-demand/>. Accessed: 2025-12-16.
- [2] [n. d.]. CpuLimit repository. <https://github.com/opsengine/cpulimit>. Accessed: 2024-07-11.
- [3] [n. d.]. Stress tool description. <https://man.archlinux.org/man/stress.1>. Accessed: 2024-07-11.
- [4] Vaibhav Aggarwal and B. Thangaraju. 2020. Performance Analysis of Virtualisation Technologies in NFV and Edge Deployments. In *2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, 1–5. doi:10.1109/CONECCT50063.2020.9198367
- [5] Patrick Kwadwo Agyapong, Mikio Iwamura, Dirk Staehle, Wolfgang Kiess, and Anass Benjebbour. 2014. Design considerations for a 5G network architecture. *IEEE Communications Magazine* 52, 11 (2014), 65–75. doi:10.1109/MCOM.2014.6957145
- [6] Campos. 2024. *Test results for Competition for CPU resources experiments*. doi:10.5281/zenodo.13221177
- [7] Jeffrey Chase, Darrell Anderson, Prachi Thakar, Amin Vahdat, and Ronald Doyle. 2001. Managing Energy and Server Resources in Hosting Centres. *Operating Systems Review - SIGOPS* 35, 103–116. doi:10.1145/502034.502045
- [8] Hui Chen, Youhuizi Li, and Weisong Shi. 2012. Fine-grained power management using process-level profiling. *Sustainable Computing: Informatics and Systems* 2, 1 (2012), 33–42. doi:10.1016/j.suscom.2012.01.002
- [9] Miyuru Dayarathna, Yonggang Wen, and Rui Fan. 2016. Data Center Energy Consumption Modeling: A Survey. *IEEE Communications Surveys & Tutorials* 18, 1 (2016), 732–794. doi:10.1109/COMST.2015.2481183
- [10] Zhijun Ding, Song Wang, and Changjun Jiang. 2023. Kubernetes-Oriented Microservice Placement With Dynamic Resource Allocation. *IEEE Transactions on Cloud Computing* 11, 2 (2023), 1777–1793. doi:10.1109/TCC.2022.3161900
- [11] Thanh Do, Suhilb Rawshdeh, and Weisong Shi. 2009. pTop: A Process-level Power Profiling Tool.
- [12] Adnei Willian Donatti, Marcia Cristina Machado, Marvin Alexander Lopez Martinez, Sabino Rogério S. Antunes, Eli Carlos Figueiredo Souza, Sand Correa, Tiago Ferreto, José Augusto Suruagy, Joberto S. B. Martins, and Tereza Cristina Carvalho. 2025. Energy Efficiency in Network Slicing: Survey and Taxonomy. *IEEE Access* (2025), 1–20. doi:10.1109/ACCESS.2025.3590365
- [13] Meryeme El Yadari, Ali Yahyaouy, Stéphane Le Masson, Khalid El Fazazy, and Hamid Gualous. 2022. Study of the correlation between server resources utilization and energy consumption. In *2022 10th International Conference on Systems and Control (ICSC)*, 125–130. doi:10.1109/ICSC57768.2022.9993950
- [14] Xiaobo Fan, Wolf-Dietrich Weber, and Luiz Andre Barroso. 2007. Power provisioning for a warehouse-sized computer. *SIGARCH Comput. Archit. News* 35, 2 (jun 2007), 13–23. doi:10.1145/1273440.1250665
- [15] Xiaobo Fan, Wolf-Dietrich Weber, and Luiz Andre Barroso. 2007. Power provisioning for a warehouse-sized computer. In *Proceedings of the 34th Annual International Symposium on Computer Architecture (San Diego, California, USA) (ISCA '07)*. Association for Computing Machinery, New York, NY, USA, 13–23. doi:10.1145/1250662.1250665
- [16] Vishal Gupta, Ripal Nathuji, and Karsten Schwan. 2011. An analysis of power reduction in datacenters using heterogeneous chip multiprocessors. *SIGMETRICS Perform. Eval. Rev.* 39, 3 (dec 2011), 87–91. doi:10.1145/2160803.2160867
- [17] Nancy Jain and Sakshi Choudhary. 2016. Overview of virtualization in cloud computing. In *2016 Symposium on Colossal Data Analysis and Networking (CDAN)*, 1–4. doi:10.1109/CDAN.2016.7570950
- [18] Hiroki Kataoka, Dilawaer Duolikun, Tomoya Enokido, and Makoto Takizawa. 2015. Power Consumption and Computation Models of a Server with a Multi-core CPU and Experiments. In *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*, 217–222. doi:10.1109/WAINA.2015.127
- [19] Kubernetes 2024. Command line tool (kubectl). <https://kubernetes.io/docs/reference/kubectl/>. [Online; accessed 29-October-2024].
- [20] Kubernetes 2024. Production-grade container orchestration. <https://kubernetes.io/>. [Online; accessed 23-July-2024].
- [21] Adam Lewis, Soumik Ghosh, and Nian Feng Tzeng. 2008. Run-time Energy Consumption Estimation Based on Workload in Server Systems. *HotPower*.
- [22] Hui li, Giuliano Casale, and Tariq Ellahi. 2010. SLA-driven planning and optimization of enterprise applications. *WOSP/SIPEW'10 - Proceedings of the 1st Joint WOSP/SIPEW International Conference on Performance Engineering*, 117–128. doi:10.1145/1712605.1712625
- [23] Youhuizi Li, Jiancheng Zhang, Congfeng Jiang, Jian Wan, and Zujie Ren. 2019. PINE: Optimizing Performance Isolation in Container Environments. *IEEE Access* 7 (2019), 30410–30422. doi:10.1109/ACCESS.2019.2900451
- [24] Unai Lopez-Novoa. 2019. Exploring Performance and Energy Consumption Differences between Recent Intel Processors. In *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI)*, 263–267. doi:10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCL.2019.00088
- [25] Xiaoyan Ma, Changgeng Zhang, Shurong Li, and Xinsheng Yang. 2021. A Data-Driven Based Energy Consumption Modeling for Heterogeneous Servers in Data Centers. In *2021 IEEE 5th Conference on Energy Internet and Energy System Integration (EI2)*, 3109–3114. doi:10.1109/EI252483.2021.9713131
- [26] Joberto S. B. Martins, Tereza C. Carvalho, Rodrigo Moreira, Cristiano Bonato Both, Adnei Donatti, João H. Corrêa, José A. Suruagy, Sand L. Corrêa, Antonio J. G. Abelem, Moisés R. N. Ribeiro, José-marcos S. Nogueira, Luiz C. S. Magalhães, Juliano Wickboldt, Tiago C. Ferreto, Ricardo Mello, Rafael Pasquini, Marcos Schwarz, Leobino N. Sampaio, Daniel F. Macedo, José F. De Rezende, Kleber V. Cardoso, and Flávio De Oliveira Silva. 2023. Enhancing Network Slicing Architectures With Machine Learning, Security, Sustainability and Experimental Networks Integration. *IEEE Access* 11 (2023), 69144–69163. doi:10.1109/ACCESS.2023.3292788
- [27] Alexander Nowak, Tobias Binz, Frank Leymann, and Nicolas Urbach. 2013. Determining Power Consumption of Business Processes and Their Activities to Enable Green Business Process Reengineering. In *2013 17th IEEE International Enterprise Distributed Object Computing Conference*, 259–266. doi:10.1109/EDOC.2013.36
- [28] Benoit Petit. 2023. *Scaphandre*. <https://github.com/hubblo-org/scaphandre>.
- [29] Chuan Pham, Nguyen H. Tran, Shaolei Ren, Walid Saad, and Choong Seon Hong. 2020. Traffic-Aware and Energy-Efficient vNF Placement for Service Chaining: Joint Sampling and Matching Approach. *IEEE Transactions on Services Computing* 13, 1 (2020), 172–185. doi:10.1109/TSC.2017.2671867
- [30] Meikel Poess and Raghunath Othayoth Nambiar. 2008. Energy cost, the key challenge of today's data centers: a power consumption analysis of TPC-C results. *Proc. VLDB Endow.* 1, 2 (Aug. 2008), 1229–1240. doi:10.14778/1454159.1454162
- [31] Peter Rost, Ignacio Berberana, Andreas Maeder, Henning Paul, Vinay Suryaprakash, Matthew Valenti, Dirk Wübben, Armin Dekorsy, and Gerhard Fettwies. 2015. Benefits and challenges of virtualization in 5G radio access networks. *IEEE Communications Magazine* 53, 12 (2015), 75–82. doi:10.1109/MCOM.2015.7355588
- [32] Cheng-Jen Tang and Miao-Ru Dai. 2011. Dynamic computing resource adjustment for enhancing energy efficiency of cloud service data centers. In *2011 IEEE/SICE*

- International Symposium on System Integration (SII)*. 1159–1164. doi:10.1109/SII.2011.6147613
- [33] Amir Varasteh, Basavaraj Madiwalar, Amaury Van Bemten, Wolfgang Kellerer, and Carmen Mas-Machuca. 2021. Holu: Power-Aware and Delay-Constrained VNF Placement and Chaining. *IEEE Transactions on Network and Service Management* 18, 2 (2021), 1524–1539. doi:10.1109/TNSM.2021.3055693
- [34] Zihao Wang, Lei Zhuang, Feijie Zhou, and Ruimin Wang. 2023. Energy Efficient VNF Placement Algorithm Using Reinforcement Learning in NFV-Enabled Network. In *2023 IEEE International Conference on Control, Electronics and Computer Technology (ICCECT)*. 625–629. doi:10.1109/ICCECT57938.2023.10141342
- [35] Mehul Warade, Kevin Lee, Chathurika Ranaweera, and Jean-Guy Schneider. 2023. Monitoring the Energy Consumption of Docker Containers. In *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*. 1703–1710. doi:10.1109/COMPSAC57700.2023.00263
- [36] Junzo Watada, Arunava Roy, Raturaj Kadikar, Hoang Pham, and Bing Xu. 2019. Emerging Trends, Techniques and Open Issues of Containerization: A Review. *IEEE Access* 7 (2019), 152443–152472. doi:10.1109/ACCESS.2019.2945930
- [37] Chi Xu, Ziyang Zhao, Haiyang Wang, Ryan Shea, and Jiangchuan Liu. 2017. Energy Efficiency of Cloud Virtual Machines: From Traffic Pattern and CPU Affinity Perspectives. *IEEE Systems Journal* 11, 2 (2017), 835–845. doi:10.1109/JSYST.2015.2429731