Vrije Universiteit Amsterdam

Bachelor Thesis

A Historical Analysis of Energy Consumption in Large-Scale Computer Systems (1990s to 2020s)

Author: Dide Poyraz (2725768)

1st supervisor: Prof. dr. ir. Alexandru Iosup
2nd reader: Dr. Daniele Bonetta

A thesis submitted in fulfillment of the requirements for the VU Bachelor of Science degree in Computer Science

June 2, 2026

Abstract

The increasing demand for computing has led data centers to account for 1-1.5% of global electricity usage, and as demand rises, so does energy consumption. Therefore, there is a need to better understand and optimize the energy consumption of data centers. Existing energy consumption models often rely on peak power usage rather than real-time analysis, leading to unrealistic predictions. This report conducts a historical analysis of the energy consumption of computer systems under realistic conditions from the 1990s to the 2020s using OpenDC, a discrete event data center simulator. By constructing a taxonomy of exemplary computing infrastructures per decade, we develop models to simulate their operation and energy consumption.

Our key contributions include developing a selection procedure to select relevant systems from the TOP500 and Green500 lists, implementing workload scaling methods for diverse configurations, and conducting trace-based simulations to analyze energy consumption trends.

Our findings reveal that Koomey's Law alone is insufficient for evaluating the energy consumption of large-scale computer systems. Comparing different energy efficiency models allows for a more comprehensive evaluation of energy consumption trends. Additionally, Our proposed models capture the nuances of selected workloads and computer systems, providing a more accurate picture of energy consumption and greater flexibility in the selection of such systems and workloads. The inclusion of accelerator-based components in the simulation is particularly significant, as it reflects the substantial improvements in energy efficiency over the past decade. This research provides new insights and a foundation for future studies on sustainable computing practices, offering a better understanding of energy consumption under realistic conditions. The data and the experiments can be found at `https://github.com/didepoyraz/experiments-bsc.git`.

Contents

1 Introduction **1**

1.1 Context . . . 1
1.2 Problem Statement . . . 3
1.3 Research Questions . . . 5
1.4 Research Methodology . . . 7
1.5 Thesis Contributions . . . 8
1.6 Plagiarism Declaration . . . 9
1.7 Societal Impact . . . 9
1.8 Thesis Structure . . . 9

2 Background **10**

2.1 Energy Provisioning in Modern Data Centers . . . 10
2.2 Data Center Simulation . . . 11

3 How to Select Exemplary Computer Systems and Workloads from 1990s to 2020s **15**

3.1 Overview . . . 15
3.2 Selection Processes . . . 15
3.3 Selecting Workloads . . . 18
3.4 Discussion . . . 21

4 How to Model the Operation of Systems Across Time Periods **22**

4.1 Overview . . . 22
4.2 Creating Topologies . . . 22
4.3 Scaling Workloads for Diverse Configurations . . . 25
4.4 Discussion . . . 38

5 Trace-Based Simulation Analysis of the Evolution of Energy Consumption from the 1990s to 2020s **40**

5.1 Overview 40
5.2 Experimental Setup 41
5.3 Evaluation of Energy Consumption Trends from the 1990s to the 2020s . . . 44
5.4 Evaluation of Energy Efficiency from the 1950s to the 2020s 57
5.5 Threats to Validity 57
5.6 Discussion 58

6 Conclusion and Future Work **60**

6.1 Answering Research Questions 60
6.2 Directions for Future Research 61

References **62**

A Reproducibility **69**

A.1 Artifact check-list (meta-information) 69
A.2 Description 69
A.3 Installation 69
A.4 Evaluation and expected results 70
A.5 Methodology 70

1

Introduction

Many daily activities in our society, ranging from business to governance, science to engineering, and shopping to entertainment, rely on advanced computing infrastructure, particularly data centers (1). The magnitude of the electricity computing infrastructures consume and the diversity of applications that make use of them demonstrate why they are seen as the backbone of the digital world. As computing demand continues to grow, optimizing the electrical usage of the ICT industry has become a great concern. The concern includes, in particular, data centers, where storage and processing of data for business and consumer services are hosted. Currently, data centers account for about 1-1.5% of the global electricity usage (2), and around 3% in digitalized societies like the Netherlands (3). Although seemingly a low percentage, it is by no means insignificant – electricity consumption by ICT is very large and continues to grow, for example, the 2024 Electricity report by the IEA states that electricity consumption from data centers could double by 2026, reaching more than 1,000 TWh from 460 TWh in 2022. This projected demand is roughly equivalent to the electricity consumption of Japan (4).

1.1 Context

Following the substantial increase in data center workloads of the past decade, numerous efficiency measures were triggered that resulted in a decrease of 20% in the energy intensity of global data centers since 2010 (5). Simultaneously, demand rose, and the use of data center compute instances increased by up to 550% (5). It is promising to witness the effort of the industry to keep up with the increasing computing demands. Nonetheless, it is not realistic to rely on the efficiency capacity we have achieved until now, as the trends in the increase of compute instances are expected to double again in the coming 3-4 years (5).

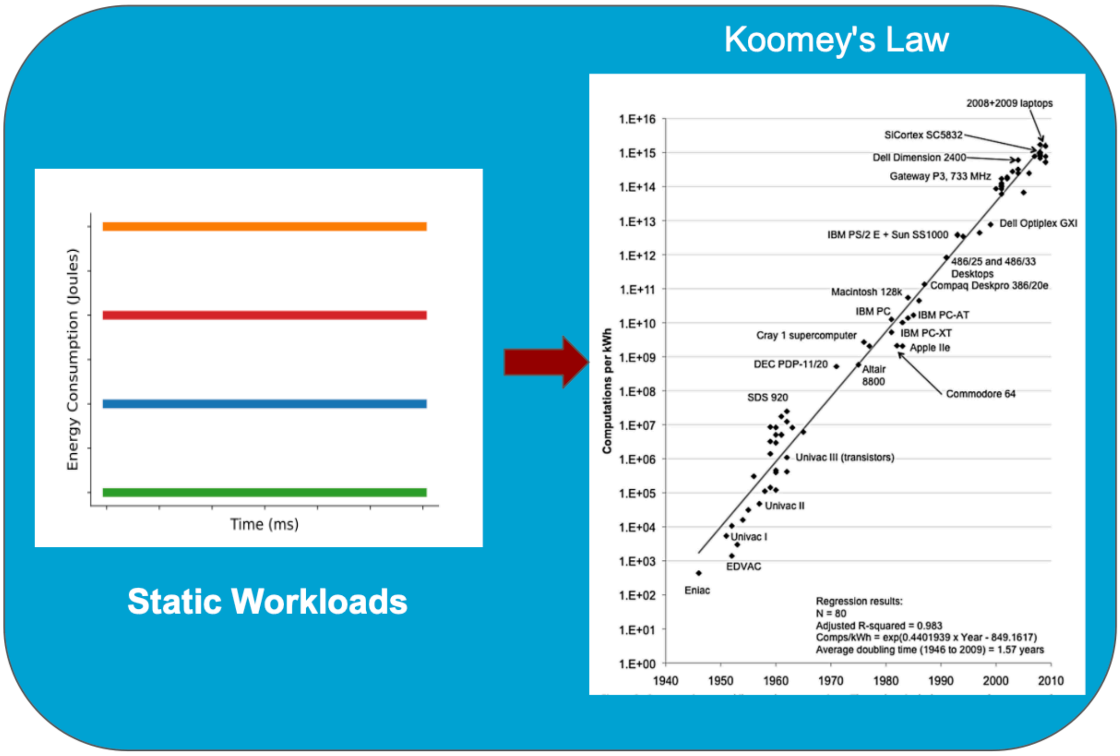

Figure 1.1: Illustration of how Koomey's Law is constructed from static workloads. The left side shows the energy consumption of static workloads over time, and the right side depicts the resulting trend in energy efficiency improvements over the decades.

Instead, we should look for further methods to increase our understanding of the energy consumption in data centers.

One of the impediments in this area is the scarcity of proper modeling and analysis of energy consumption in ICT infrastructure, particularly in large-scale, complex ICT infrastructure such as data centers. Although studies on the matter exist (6) (7), because existing predictions are based on the peak power usage of computers and not a real-time analysis of energy consumption, they fail to be realistic (8). This arises from the limited understanding we have about energy usage under realistic conditions, e.g., with varying computational workloads (2, 5, 9), and more specifically the difficulty of understanding the trade-off between performance and energy when both have dynamic operational patterns.

The models proposed by Koomey et al. (9) form a good foundation for understanding the relationship between the processing power of computers and the electricity required for that performance. In these models, the relationship is calculated by dividing the number of computations at maximum load by the kilo-watt-hour energy consumed by the computing system to complete the computation. Such metrics provide a good first-order approximation of the relationship under static conditions, but when dynamics intervene it can only provide a simplified view. An example of how Koomey's models are formed can be seen in Figure 1.1, where static workloads with constant energy consumption form the ideal, linear curve of Koomey's Law.

In practice, this approximation proves to be simplistic, because the dynamics can be intense – and large changes in workload patterns are highly expected (2, 5, 9). Additionally, large-scale computers are on average loaded fully only at 10-15%, long-term, with high peaks of activity and occasional incidence of significant failures, all of which break the static operational assumptions (9). Even for linear models, estimating power usage accurately is currently challenging, because many of the models use specification-sheet data and not actual, real-world usage data. For example, the specification sheets always include peak energy use, but do not reflect the actual energy consumption for particular operational conditions, such as real workloads managed with specific resource management and scheduling techniques. Overall, the current approach to estimating energy consumption can lead to significantly inaccurate energy-consumption analysis.

1.2 Problem Statement

We posit it would be beneficial for the IT industry, academia, and users of data center services who leverage data centers for their ICT workloads to have an instrument to examine the relationship between energy consumption and compute power for large-scale computer systems demonstrating realistic representations of energy and power usage. An exploration of this approach would make it possible to achieve realistic data modeling as opposed to using peak power consumption as a reference point for energy usage to optimize data center energy management.

Furthermore, a historical trend analysis would make it possible to systematically examine and predict energy consumption through the usage of current and past computer systems data. Analyzing and comparing infrastructures from the past also has the potential of revealing useful but forgotten approaches to apply to our present practices on improving energy efficiency. Our capability to simulate conditions that have disappeared and conditions that currently exist creates the possibility of making a fair comparison on which methodology worked best throughout the history of the last 30 years of computing. This is analogous to building our own "time machine" to take a stroll around the most important

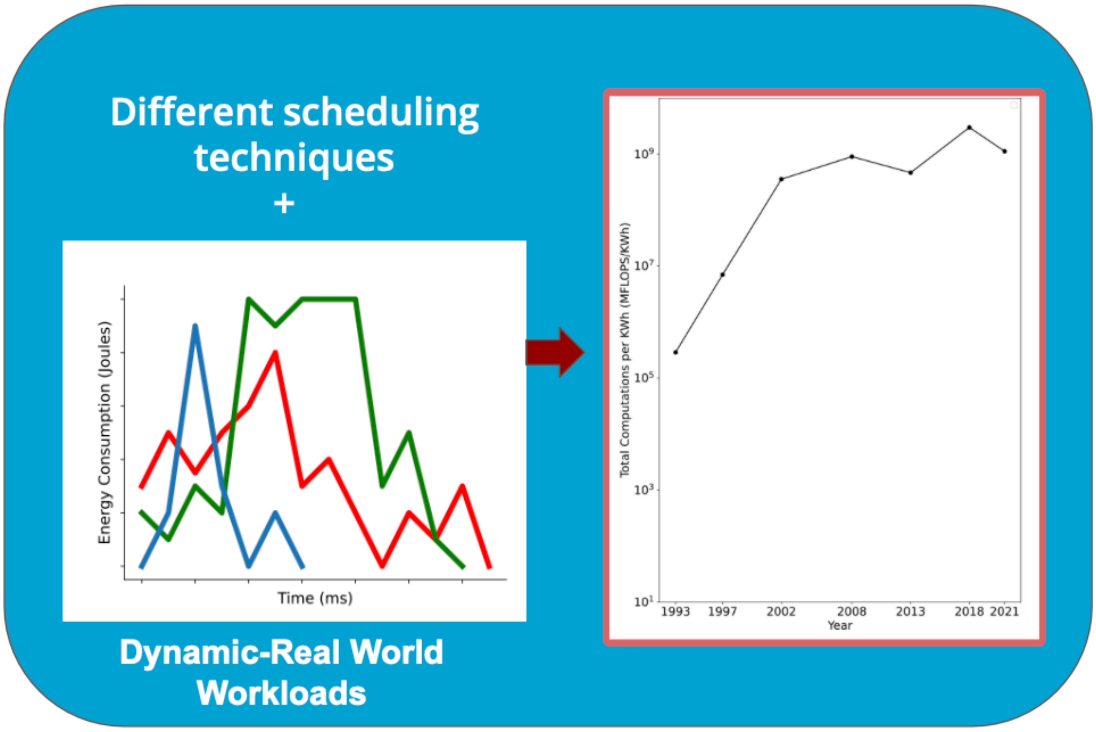

Figure 1.2: The expected impact of dynamic workloads on the energy efficiency of large-scale computer systems. The left side shows the energy consumption of dynamic workloads with different scheduling techniques. The right side depicts the resulting trend in energy efficiency.

large-scale computer systems and draw knowledge from different decades to apply in our present time.

We aim to contribute by creating a historical representation of energy usage in large-scale computer systems to analyze the evolution of the trade-off between performance and energy from the 1990s to the 2020s. We expect the resulting curve to be under Koomey's ideal curve. In figure 1.2, the expected impact of dynamic workloads on the energy efficiency of large-scale computer systems can be seen. We plan to identify relevant infrastructures, according to chosen requirements for each experiment, from different decades and model them using OpenDC which is a state-of-the-art discrete event data-center simulator. Conducting a comparative study of all the decades to conclude our historical analysis is our main objective. The emphasis of this aim is to demonstrate the evolution of the energy consumed through the execution of real-world workloads over time in regard to the extensive time frame chosen. This would be beneficial to evaluate the trends in energy consumption which would create an opportunity to consider improving energy efficiency using realistic modeling and observing the trade-off between performance and energy over 30 years.

Our project aligns with an active field of research, where the complex operation of ICT infrastructures remains a grand challenge, as indicated in the Netherlands manifesto on computer systems and networking research (and tens of key peer-reviewed references within) (10). A distinct scientific challenge in our work is that an accurate and precise methodology to estimate, and possibly to predict, the use of energy incurred by a specific workload on a data center, subject to specific resource management and scheduling, and operational phenomena such as machine failures, does not currently exist.

1.3 Research Questions

RQ1 How to select relevant computer systems and workloads to build a taxonomy of exemplary infrastructures per decade from the 1990s to the 2020s?

Creating a historical representation of the over 30 years of data center operation, from the early 1950s through the early 2020s, requires a good understanding of what makes an infrastructure exemplary, and able to represent the population of computers of its time. We plan to identify various such ICT infrastructures in line with the pace of technological advancement in the field and its reflection in real-world operations, so per decade. To achieve this, we need a set of requirements for what determines a relevant infrastructure. This could include but not be limited to the largest and/or most efficient machines, different architecture types, and relevant workloads executed at the time.

We use the TOP500 and GREEN500 tables, which are released every six months since 1993 and 2013. The TOP500 list covers the period from 1993 to 2023, documenting the most powerful supercomputers, while the GREEN500 list provides insights into the most efficient systems from 2013 to 2023. (11). Although these historical lists are crucial to our study, they are not entirely comprehensive. Information on power consumption metrics, especially under different conditions than peak power, is unavailable for any systems on the TOP500 list or other sources. The absence of such information highlights the need for more detailed and transparent data to fully understand energy consumption trends in computer systems.

RQ2 How to model the operation of data centers from different time periods to analyze the evolution of energy consumption?

After identifying and gathering a historical representation of computer systems in **RQ1**, it is necessary to establish a methodology to model the chosen infrastructures. Modeling infrastructures provide the opportunity to explore hypothetical scenarios without additional constraints on the experiments such as cost and time. We propose using OpenDC, a discrete event simulator to model the operation of data centers from different periods to analyze energy consumption trends.

The scientific challenge here is to (1) create data center models that can capture data centers across many decades. This is challenging because of the lack of detailed information on the system metrics such as the power models of the CPUs and the idle power consumed by the CPUs.

(2) To design discrete-event models for data centers in operation between the 1990s and the 2020s, and implement them in OpenDC as a proof-of-concept. Ensuring our models for workloads and topologies for data centers are compatible with OpenDC is challenging due to the limitations of OpenDC in adapting workloads to diverse system configurations. Adapting these workloads involves maintaining the original task distribution, ensuring consistent computational work through validation methods, and load-balancing to improve system resource utilization.

RQ3 How did energy consumption evolve from the 1950s to 2023 for the execution of realistic workloads in [contemporary] data centers?

Building on our findings from **RQ1** and **RQ2**, we need to analyze our data to understand the evolution of energy consumption over time. A clear understanding of the relationship between performance and energy in computer systems over 30 years is expected to be obtained from this analysis. Currently, OpenDC does not support such a tool to conduct the modeling of a comprehensive comparative analysis of different infrastructures across different generations (period: 1990-2020). Our goal is not only to develop a realistic representation of data center operations across different eras but also to incorporate a new tool to conduct comparative analyses of computer systems of different eras.

The scientific challenge here revolves around interpreting the historical evolution of energy consumption within large-scale computer systems into a coherent and comparative analysis. This task necessitates the synthesis of complex datasets into a comparative framework that can showcase trends and the effectiveness of energy consumption strategies over 30 decades of technological advancement. The challenge of answering **RQ3** is the dynamic nature of the workloads and the efficiency of the infrastructures that are executing them. Traditional models proposed by Koomey et al. (9) have established a fundamental understanding of power consumption and electricity usage under static conditions, however, these models cannot capture the variations of energy consumption due to large changes in workload patterns that are usual in large scale computer systems.

Additionally, the need for a novel approach is further understood by the limitations of historiographical approaches to data centers in capturing the technical details of technological evolution. While historical narratives have been more than adequate at documenting key developments and figures, they often lack the technical depth required to comprehensively analyze the efficiency and energy consumption trends that we need. By leveraging OpenDC, we seek to bridge this gap. Our goal is to develop a realistic representation of data center operations across different eras and to incorporate a new tool to conduct comparative analyses of computer systems of different eras.

In summary, addressing **RQ3** involves a significant inquiry into how advancements in computing technologies and changes in operational paradigms have influenced the energy efficiency of data centers. By systematically analyzing the evolution of energy consumption in the context of real-world workloads, this study aims to identify sustainable and possibly forgotten practices and suggest pathways toward more energy-efficient computing landscapes.

1.4 Research Methodology

To answer **RQ1**, we identify various ICT infrastructures in line with the pace of technological advancement in the field and its reflection in real-world operations, to construct a per decade taxonomy (12) representing systems from the 1990s to the 2020s. We use the TOP500 and GREEN500 tables, which are released every six months since 1993 and 2013, to identify each system in our taxonomy (11).

After establishing a taxonomy of a historical representation of computer systems in **RQ1**, we develop a methodology to model the selected infrastructures. Experiments are conducted using OpenDC (13) to simulate the evolution of energy consumption in large-scale computer systems. Given OpenDC's limitations in adapting workloads to diverse system configurations, we explore methods to scale workloads for compatibility across systems from different decades.

Building on our findings from RQ1 and RQ2, we conduct a trace-based simulation analysis to understand the evolution of energy consumption over time. This analysis aims to provide a clear understanding of the relationship between performance and energy consumption in computer systems over 30 years. The scientific challenge of RQ3 revolves around interpreting the historical evolution of energy consumption within large-scale computer systems into a coherent and comparative analysis. We aim to develop a model showcasing trends and the effectiveness of energy consumption strategies over seven decades of technological advancement.

Our research methodology is guided by the state-of-the-art AtLarge Design Process (14), aligning with its methodological principles of exploring the history and evolution of large-scale computer systems through simulation models. This approach aims to identify sustainable practices and inform future design strategies, contributing to the societal impact emphasized by the AtLarge design framework.

Overall, our approach emphasizes open and reproducible science (15). Our implementations adhere to standard practices and are thoroughly documented, offering both practical guidance and detailed explanations of the underlying theories. Additionally, we open-source our implementation on GitHub to encourage community collaboration and further advancement in the field.

1.5 Thesis Contributions

In answering our research questions, this thesis has produced several contributions.

1. **Conceptual**
   (a) We develop a procedure to select computer systems across different decades to construct a comprehensive taxonomy of technological advancements per decade.
   (b) We develop workload scaling procedures and translate our taxonomy into individual system topologies.
   (c) Our analysis reveals that contrary to Koomey's Law, energy consumption and energy efficiency evolve differently across various classes of computers when considering different workloads. This divergence provides new insights into the complexities of energy consumption in large-scale computing systems.
2. **Technical**
   (a) We create a model to simulate the operation of computer systems from the 1990s to the 2020s to conduct a study on energy consumption trends.
   (b) We implement workload scaling to adapt workloads to diverse system configurations.

(c) We set up and conduct experiments to analyze the energy consumption of computer systems.

1.6 Plagiarism Declaration

I confirm that this thesis work is my own work, and is not copied from any other source (person, Internet, or machine), and has not been submitted elsewhere for assessment.

1.7 Societal Impact

We envision the societal impact of this thesis to have long-term consequences. By developing a process to select systems, a method for scaling workloads, and implementing a toolbox that allows conducting experiments, this study provides a framework that can significantly improve and further encourage the effort for understanding the energy consumption of past and future systems. This methodology not only aids in understanding historical trends but also equips policymakers and industry leaders with valuable insights. Our contributions consisting of selection processes and methods to scale workloads lead to more realistic modeling of energy consumption in data centers.

These contributions emphasize the insufficiency of relying solely on Koomey's Law for understanding energy consumption and advocate for a multi-faceted approach. We expect our contributions to guide future research toward more sustainable and efficient computing practices, thus benefiting society at large.

1.8 Thesis Structure

The structure of the rest of this paper is outlined as follows:

In Chapter 2, we cover the background information relevant to our research. Chapter 3 addresses **RQ1**, detailing the selection processes for exemplary computer systems and workloads from the 1990s to the 2020s. Chapter 4 introduces creating topologies and methodologies for scaling workloads to model system operations, addressing **RQ2**. Chapter 5 answers **RQ3** and presents the evaluation of energy consumption trends. Finally, Chapter 6 summarizes our contributions and proposes future work directions.

2

Background

In this chapter, we provide a comprehensive overview of the core topics relevant to our research. We start with an exploration of energy provisioning strategies in modern data centers in Section 2.1. Section 2.2.1 compares different methods for modeling large-scale computer systems, discussing the benefits and drawbacks of using real data centers, mathematical analysis, and simulation.

Afterward, we detail various simulators and look at conditions to asses simulator suitability to a study on their flexibility, exhaustibility, and scalability. Finally, we present OpenDC as a robust and flexible tool for our study. Our objective is to highlight the existing challenges and methods in accurately understanding, optimizing, and modeling energy consumption in large-scale computing environments.

2.1 Energy Provisioning in Modern Data Centers

Building data center facilities requires determining appropriate power budgets to construct necessary electrical infrastructures to supply power. This power budget is determined using peak power estimates of data center components, such as CPUs. Each data center component comes with a power rating that is meant to represent the maximum power draw of that machine. However, this rating is intentionally conservative and typically not meant to be reached. The primary purpose of this rating is to inform the user of the power infrastructure required to safely supply power to the machine. Equipment manufacturers estimate this peak power by adding up the worst-case power draw of all components of a machine (16). This approach of over-estimating provides a buffer for safe and reliable operation under varying operational loads.

As the demand and use of data centers increase, this buffer can lead to over-provisioning. For example one of the key findings of Fan, Weber, and Barroso states that the gap between the maximum power used by computer systems and their aggregated theoretical peak power usage can be up to 40% (17). This significant gap results in the waste of energy resources. Additionally, it becomes increasingly unlikely for large groups of systems to be at their peak activity levels simultaneously as the size of the group increases.

To utilize the available power budget and minimize the gap, deploying additional compute equipment is an optimal solution (17). However, this is not a trivial task because accurately estimating the power consumption remains challenging. Under-provisioning data centers can lead to a compromise of the availability and stability of the infrastructures. Techniques such as power capping and CPU voltage scaling (Dynamic Voltage and Frequency Scaling, DVFS)(18) help address the issue of utilizing the available power budgets.

Power capping prevents overload situations by setting an upper limit on the power usage of individual servers or the entire data center infrastructure. This method guarantees that the power consumption stays within safe limits during peak activity levels. CPU voltage scaling (DVFS) adjusts the voltage and frequency of a CPU based on the workload demand, managing the energy resources more efficiently and adapting to different performance requirements.

These are a selection of methods to improve energy efficiency, among others that we did not discuss. While these methods significantly improve energy efficiency and resource utilization, a deeper understanding of energy consumption in data centers, particularly how load variations and different system configurations impact it, is crucial. More accurate power budget estimates can prevent over-provisioning and the need for complex solutions, leading to more efficient resource use and better infrastructure utilization. By addressing these issues at the estimation stage, data centers can achieve higher efficiency without the additional effort and expenses associated with implementing corrective measures.

2.2 Data Center Simulation

2.2.1 Comparison of Methods for Modeling Large-Scale Computer Systems

There are several methods to base our experiments to model infrastructures, primarily, the usage of real data centers, mathematical analysis, and simulation.

Exploration via real data centers raises challenges such as a lack of reproducibility of experiments where the cost of retracing the experiment would be too high, reducing the likelihood of finding meaningful premises and funds. Furthermore, operational phenomena (19) affect system performance in complex ways that influence measurements of identical experiments which adds another layer of variability and uncertainty. Additionally, scalability poses a significant issue; if the experiment is orders of magnitude bigger than a typical workload, it would significantly waste energy resources and contradict the aim of the project to understand more about energy efficiency, efficiently.

We can observe from Koomey's models (9) that analysis of energy consumption proves insufficient with mathematical methods of exploration, based on simple mathematical analysis. Although promising to explore the linearity of the trends, it lacks the multi-dimensionality of real-world energy consumption, the ability to match the complexity of dynamic operational decisions taken by data centers, and cannot cope with dynamic computational loads and the varying power consumption they cause.

Simulation-based approaches offer a middle-ground between real-world experimentation and mathematical analysis. Among the benefits, simulation can capture complexity and dynamicity, without the costs incurred by real-world experimentation, and could thus be of real use to this problem (20). Among the drawbacks, simulations depend on complex models that may not capture reality precisely and accurately, and the credibility of the results depends on how well the model captures the essential parts of reality; models also threaten to become as complex as reality and often may become intractable.

All of these methods have benefits and drawbacks to be considered, but a simulator for conducting our experiments appears to be the most compatible choice with our project because we focus on different infrastructures from various periods and workloads, and neither analytical models nor real data centers would be feasible to conduct such large-scale experiments as efficiently and quickly as simulation.

2.2.2 Review of Data Center Simulators

Data centers are complex infrastructures with various components such as (a) user application/workload, (b) Virtual Machine (VM)/host performance, (c) power consumption, (d) resources contention, and scheduling, (e) network communication (21). There are multiple categories for cloud infrastructure simulators, which include but are not limited to general cloud modeling, energy-aware provisioning, and application modeling. It is more realistic to focus on a specific aspect or component when designing a simulator to model data centers because it is unfeasible to model all elements of the system with high accuracy. Furthermore, it is practically impossible to model all components of a system with exact precision as the performance varies and becomes unpredictable due to factors such as different operational phenomena and unpredictable workload patterns. An estimation of the performance with a reasonable relative error is acceptable (21).

Further assessing how suitable a simulator for a study can be decided by looking at the following factors:

- Flexibility is essential for enabling future extensions, enhancing usability, and adapting to the evolving demands of research and development.
- Model exhaustibility, to ensure precise performance estimation, that supports basic features such as job scheduling and duration estimation, as well as supporting advanced features such as power consumption and heat output.
- Scalability can be improved by enabling users to deactivate irrelevant features or adjust accuracy levels, making the simulator more adaptable to various scenarios.

CloudSim (22) is a general cloud modeling simulator focusing on simulating cloud system components including virtual machines, data centers, and resource provisioning policies. Additionally, it includes other single-feature simulators, however, these extend CloudSim in different directions and it is not easy to combine them without extensive effort. One of the single feature simulators that extend CloudSim is GreenCloudSim (23) which focuses on energy-aware cloud computing.

Current data center simulators primarily focus on modeling power consumption related to CPU, network, memory, and cooling infrastructure (21). However, with the increasing popularity of High-Performance Computing (HPC), system architectures are evolving towards heterogeneous nodes where traditional CPUs work alongside accelerators such as GPUs, DPUs, and FPGAs to offload compute-intensive tasks (24, 25, 26). Despite early efforts with simulators like GPGPU-Sim (27) and Barra (28) for NVIDIA GPUs, NetworkCloudSim (29) for network and application models of HPC compute instances, and hybrid architecture simulators such as FusionSim (30) and Multi2Sim (31), these models still lack the flexibility to adapt to new devices and are not advanced in modeling CPU-accelerator cooperation (32). In summary, significant gaps remain in seamlessly integrating accelerator-based simulators within existing frameworks.

The inability of simulators not being able to accurately model all components of data centers becomes particularly problematic when the objective is to evaluate the overall energy consumption of computer systems from the 1990s to the 2020s. While an ideal simulator that captures every component and detail would be invaluable for our study, such a tool does not currently exist. Therefore, it is crucial to carefully consider the trade-offs and decide which components to prioritize when selecting a simulator for the study.

2.2.3 OpenDC

OpenDC is a state-of-the-art discrete event (33), data-center simulator created by the At-Large research group (13). It provides a robust platform for creating various data center configurations tailored to different workloads, making it ideal for examining energy efficiency scenarios. Our experiments using OpenDC model the evolution of energy consumption in large-scale computer systems and implement workload scaling to adapt workloads to diverse configurations.

Among the many classes of simulators, event-driven simulators like OpenDC express reality in steps triggered by significant events, such as new requests sent to the ICT infrastructure, time elapsed, failures occurring, and dynamic operational decisions being made. This event-driven approach ensures a more realistic and dynamic simulation environment.

OpenDC models data center infrastructure as clusters of heterogeneous hosts, with each host representing a node in a data center rack. The resource consumption of applications, CPU usage, is modeled using discretized time slices. Workloads report their resource consumption to the hypervisor at each time slice. Hypervisor consolidates the requests and distributes the resources among the requesting workloads. CPU resources are allocated either through time-sharing (if the workloads are on the same core) or space-sharing (if they are on different cores).

OpenDC is a suitable choice for our study as it supports detailed power consumption metrics for CPUs. Additionally, OpenDC is flexible enough to implement new features, such as workload scaling, which we discuss and implement in Section 4.3. While it currently does not include cooling infrastructure, CPU load remains the primary criterion for estimating energy consumption.

3

How to Select Exemplary Computer Systems and Workloads from 1990s to 2020s

3.1 Overview

In this Chapter, we address **RQ1** by establishing a systematic approach for selecting relevant computer systems and workloads from the 1990s to the 2020s. In 3.2, we outline our methodology for constructing a taxonomy of computer systems. Afterward, in Section 3.3, we discuss the different workloads we select to ensure a comprehensive analysis of different computational demands and operational requirements.

3.2 Selection Processes

Establishing relevant criteria to construct a taxonomy of computer systems must involve a systemic approach to be successful. We begin by looking at the rankings in TOP500 and Green500 (11). Consideration between selecting the finest computers of each decade or the most well-rounded computers to represent a more common range of computer performance for the associated time period proved to be difficult. We conclude that the most viable way to answer **RQ1** would be to study the leading computer systems of each decade to be able to make concise conclusions about the limits of computation the industry achieved thus far and the energy consumption that comes with it.

3.2.1 Dimensions that Characterize the Selection Processes

1. The set of lists being studied.
2. How to select from each list one or several computers?
3. From a list that repeats over time how to consider the subset of each list instance over time?
4. If multiple lists are considered, how to combine the selections?

The dimensions that characterize the selection process represent the different factors and considerations that must be taken into account to systematically and comprehensively select large-scale computer systems. Each dimension addresses a specific component of the selection process to ensure all relevant aspects are covered. The selection criteria can be based on performance, energy efficiency, technological innovations of a specific period, or other relevant factors. Additionally, it is essential to determine how to account for the changes and trends over time. This dimension involves deciding whether to consider every instance of the list, focus on specific periods, or track the evolution of particular systems. By addressing these dimensions, we construct a thorough approach for selecting large-scale computing systems.

3.2.2 Criteria for Comparing Selection Processes

1. Selection of the computers that were energy efficient when they were top ranked but at the same time were considered the most powerful.
2. Clarity in selection, the product of selection is repeatable/reproducible by anyone else without the need for further clarifications
3. Selection where the result would be stable across multi-year selection windows.
4. The gap of intervals of time between samples being consistent.

The dimensions we outline serve as a foundation for developing more specific criteria. Now, we gather criteria to compare the quality and suitability of different selection processes. One of the key indicators of a thorough selection process is stability across multiple years. This means that the selected systems have demonstrated excellence over a significant period. Additionally, it is crucial to ensure that the intervals of time between samples follow a logical approach. This prevents intervals from being too short, which can lead to excessive detail and repetition, or too long, which might cause significant technological advancements to be missed.

3.2.3 Selection Process 1

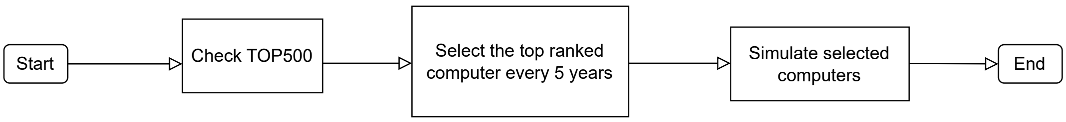

Figure 3.1: Selection Process 1 overview.

For Selection Process 1, we study one list only and choose the top-ranked computer system every 5 years. This process is represented by a flow-chart in Figure 3.1. With this method, we maintain clarity in selection, in that the best computer system is selected from one source and the time interval covers the entire time range of over 30 years in a comprehensive manner. Additionally, we observe that top-ranked computers often remained consistent over extended periods. However, this process has disadvantages as well. The TOP500 list, which ranks powerful computers, may not be the best source for identifying the most energy-efficient systems, unlike the GREEN500 list. Relying solely on the TOP500 may bias the study since it does not provide the specific energy efficiency information required. Although GREEN500 showcases the most efficient computers it is not an indication of the most powerful computers. Additionally, GREEN500 is much shorter and does not cover a wide time range.

3.2.4 Selection Process 2

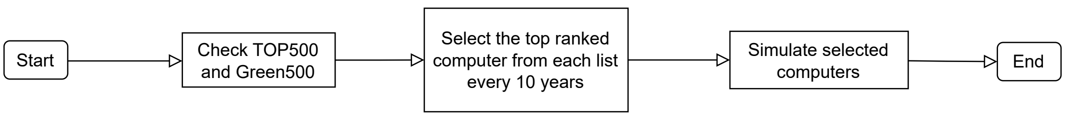

Figure 3.2: Selection Process 2 overview.

For Selection Process 2, we study TOP500 and Green500 simultaneously, choosing the top-ranked computer system every 10 years from each list. This process is represented by a flow-chart in Figure 3.2. This method also has advantages and disadvantages. The significant advantage of Selection Process 2 is choosing the top-ranked computers from both lists to ensure a balanced focus on both computational power and energy efficiency, allowing clarity of selection. However, while this process provides a balanced view of

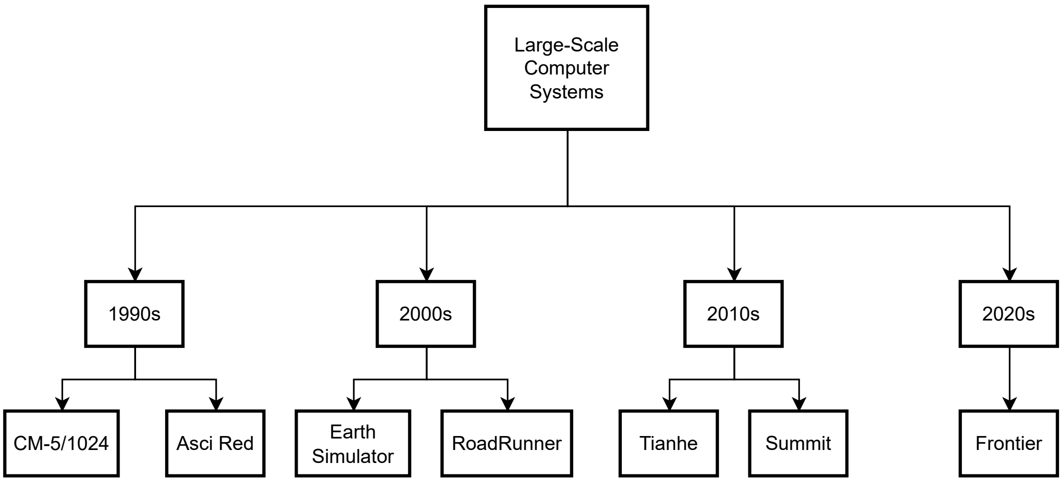

Figure 3.3: Taxonomy of large-scale computer systems per decade from the 1990s to the 2020s.

computational needs, the 10-year gap between the selections may be too long and could potentially miss the important technological advancements that occur within the decade.

3.2.5 Conclusion on the Selection Process

After considering the advantages and disadvantages of both processes we decide on using Selection Process 1 in Section 3.2.3. Selection Process 1 offers clarity and consistency by choosing the top-ranked computer system every 5 years from the TOP500. While Selection Process 2 is also valuable, it does not encompass a sufficient number of systems. Given the time constraints and scope of our project, we will leave the exploration of Selection Process 2 to future studies.

Figure 3.3 represents the final taxonomy constructed by using the selection process identified. It includes 7 of the most powerful large-scale computer systems from the 1990s to the 2020s.

3.3 Selecting Workloads

Selecting workloads for a comprehensive representation of computer systems across different technological eras is a complex task. The absence of a standard for representing the vast range of computing technologies requires considering various options and developing a method that balances scope and reasonableness.

To address this, we choose two distinct workloads: **Bitbrains** and **SURF Lisa**. These workloads represent different use cases of large-scale computer systems. Bitbrains exemplifies business-critical workloads that specialize in managed hosting and enterprise business computations. In contrast, SURF Lisa represents fixed-work workloads with a higher frequency of tasks, suited for high-throughput HPC computing.

These workloads offer a broad spectrum of operational practices, providing insight into how systems handle varying computational demands. Bitbrains, as a fixed-time workload, involves managing applications that must adhere to strict schedules to prevent significant business losses, emphasizing timely execution and reliability. Meanwhile, SURF Lisa, as a fixed-work workload, focuses on completing a predefined set of computational tasks regardless of time constraints, stressing high throughput and computational capacity. This distinction highlights the different computational demands of fixed-time versus fixed-work workloads, demonstrating the diverse operational requirements of large-scale computer systems.

Instead of generalizing all of computing history into two workloads, we could have selected representative workloads for each decade. Doing so would provide deeper insights into the evolution of computational priorities. By modeling decade-specific practices, we would more accurately reflect how different systems functioned during those periods, offering a clearer understanding of the historical context and technological advancements of each era.

Although selecting representative workloads for each decade is a viable and scientifically valuable approach, due to the scope of our project and time constraints, simplifying the analysis is necessary. This allows us to draw more comprehensive conclusions in a manageable time frame and offers additional advantages for our analysis.

The concept of running identical workloads on both a computer system from the 1950s and a modern system from 2021 is also a scientifically intriguing approach. This method enables a fair and equal comparison of how different systems handle the same tasks. By doing so, we can gain insights into the relative performance and efficiency of these systems across different eras. Specifically, this approach allows us to evaluate what fraction of the workload each system can manage and how efficiently they perform these tasks, highlighting the advancements in computational capabilities and resource management over time.

3.3.1 Bitbrains

Bitbrains is a service provider and the Bitbrains workload, an operational trace representative of business-critical operations, includes applications that require continuous availabil-

Properties	Bitbrains	SURF Lisa
Workload Type	Fixed-Time	Fixed-Work
Average Job Duration	28 days	3.1 hours
Number of Jobs	50	6,295

Table 3.1: Comparison of workload characteristics for Bitbrains and SURF Lisa.

ity at precise scheduled times to prevent significant business losses. Business-critical workloads often comprise applications in the solvency domain, such as Monte Carlo simulation-based financial modeling, as well as email, database, and management services(34).

Bitbrains exemplifies a fixed-time workload (35), where the success of its operations relies heavily on adhering to predefined schedules for offering real-time services. Strict scheduling is a core characteristic that defines the functionality of Bitbrains' services.

3.3.2 SURF Lisa

The SURF Lisa9 cluster is an HPC data center in the Netherlands. Its workload consists of 6,295 jobs executed over seven days, with job durations ranging from less than an hour to several days. We select only tasks that run on a single node, and this node only executes a single task. These are nodes without GPUs and use the Intel XEON Silver 4110 CPU [1]. All tasks run on 16 cores.

SURF Lisa exemplifies a fixed-work workload (35), characterized by a set amount of computational tasks that must be completed irrespective of the time required. The higher frequency of tasks in SURF Lisa highlights its focus on processing numerous jobs.

3.3.3 Comparison of Workloads

Table 3.1 presents a high-level overview of the Workload Type, Average Job Duration, and the Number of Jobs for each workload. We see that SURF Lisa has a higher frequency of jobs with smaller job durations compared to Bitbrains. Table 3.2 presents a statistical description of the CPU characteristics of each workload.

[1] `https://ark.intel.com/content/www/us/en/ark/products/123547/intel-xeon-silver-4110-processor-11m-cache-2`
`html`

Workload Name	Properties	Mean	Min	Q1	Median	Q3	Max	Standard Dev
Bitbrains	CPU Usage (MHz)	2,449.3	0.0	20.8	129.9	217.91	22,035	6,011.1
	CPU Cores	3.9	1	1	2	4	32	5
SURF Lisa	CPU Usage (MHz)	21,863.9	0.0	3,024	27,552	33,600	33,600	13,433.4
	CPU Cores	16	16	16	16	16	16	0

Table 3.2: Comparison of CPU characteristics for Bitbrains and SURF Lisa.

Key insights from Table 3.2 include:

1. SURF Lisa has a higher CPU usage compared to Bitbrains, by an order of magnitude.
2. Bitbrains shows variability in the number of CPU cores per task. In contrast, SURF Lisa has a constant number of 16 cores for each task.

3.4 Discussion

In this Chapter, we established Selection Processes to be able to choose relevant large-scale computer systems in Section 3.2 and explained our selection of workloads in Section 3.3. These Sections are essential to address **RQ1** on the selection of computer systems and workloads.

First, we explored various selection processes to establish a taxonomy of large-scale computer systems. After evaluating the advantages and disadvantages of different approaches, we opted for a process that studies top-ranked systems every five years from the TOP500 list. Afterward, we discussed the selection of workloads that focused on two types: Fixed-Time and Fixed-Work. Bitbrains is a business-critical trace and it exemplifies a Fixed-Time workload. Surf Lisa exemplifies Fixed-Work HPC tasks. These two workload types highlight the different operational requirements and computational demands of large-scale computer systems.

4

How to Model the Operation of Systems Across Time Periods

4.1 Overview

In this section, as a step toward answering **RQ2**, we translate the taxonomy we constructed in Section 3.2 into individual system topologies in Section 4.2 for simulation in OpenDC. We then explore how workload traces can be scaled for use on different system topologies while preserving the original characteristics of the jobs in Section 4.3. The goal of this chapter is to ensure that our systems and workloads can be accurately and effectively represented in OpenDC to be able to analyze energy efficiency between the 1990s and the 2020s in Chapter 5.

4.2 Creating Topologies

Creating topologies requires more information than a taxonomy, therefore we delve deeper into the architectures of the systems that we choose. We only take into account the compute nodes of all architectures. The information for the CPU core speed and the number of cores are straightforward for all architectures. However, inferring the configurations of the hosts and clusters poses challenges for some architectures. In these cases, we calculate the node and host count based on the total number of CPUs or CPU cores available in the entire system. This highlights the need for more comprehensive publications of technical specifications for computer systems. While such transparency might be seen as disadvantageous for competitors, it is invaluable for scientists to study these systems to be able to contribute to the existing body of knowledge and improve industry standards.

System Name	Installation Year	Host Count	Core Count	Core Speed (MHz)	Peak Power (W)
Frontier (36, 37)	2021	9,472	64	2,000	240
Summit (38, 39)	2018	4,608	44	3,070	190
Tianhe (40, 41)	2013	17,792	24	2,200	115
RoadRunner (42, 43)	2008	11,340	9	3,200	90
Earth Simulator (44)	2002	640	8	1,000	-
Asci Red (45, 46)	1997	4,536	2	200	-
CM-5/1024 (47, 48)	1993	1,024	1	32	-

Table 4.1: System topology information for all chosen systems.

System	**CPU**
Frontier	AMD EPYC 7713 Trento
Summit	IBM POWER9
Tianhe	Intel Xeon E5-2692v2
Roadrunner	PowerXCell
Earth Simulator	NEC
Asci Red	Pentium Pro
CM-5/1024	SuperSPARC

Table 4.2: CPUs of all chosen systems.

Table 4.1 presents the system topology information for our selected systems. This includes the installation year, total host count, core count per host, and core speed. Additionally, the table provides peak power ratings based on the Thermal Design Power (TDP) from the CPU specifications. TDP is only available for four systems Frontier, Summit, Tianhe, and RoadRunner; for the remaining systems, this information is not available. OpenDC requires the combination of the peak power and idle power rating, however idle power ratings are not available for any system. Therefore, we estimate the idle power rating to be 60% of the peak power rating for those systems where peak power data was available.

Table 4.2 showcases the CPUs of each system from our taxonomy. Most of these CPUs are general-purpose processors, with the exceptions being the NEC, SuperSPARC, and PowerXCell. The NEC processor is a vector processor, an early design precursor to modern GPUs (44). The SuperSPARC is a RISC microprocessor, known for its scalable architecture and efficiency (48). Lastly, the PowerXCell (43) is a hybrid processor, combining conventional CPU capabilities with high-performance processing elements similar to GPUs.

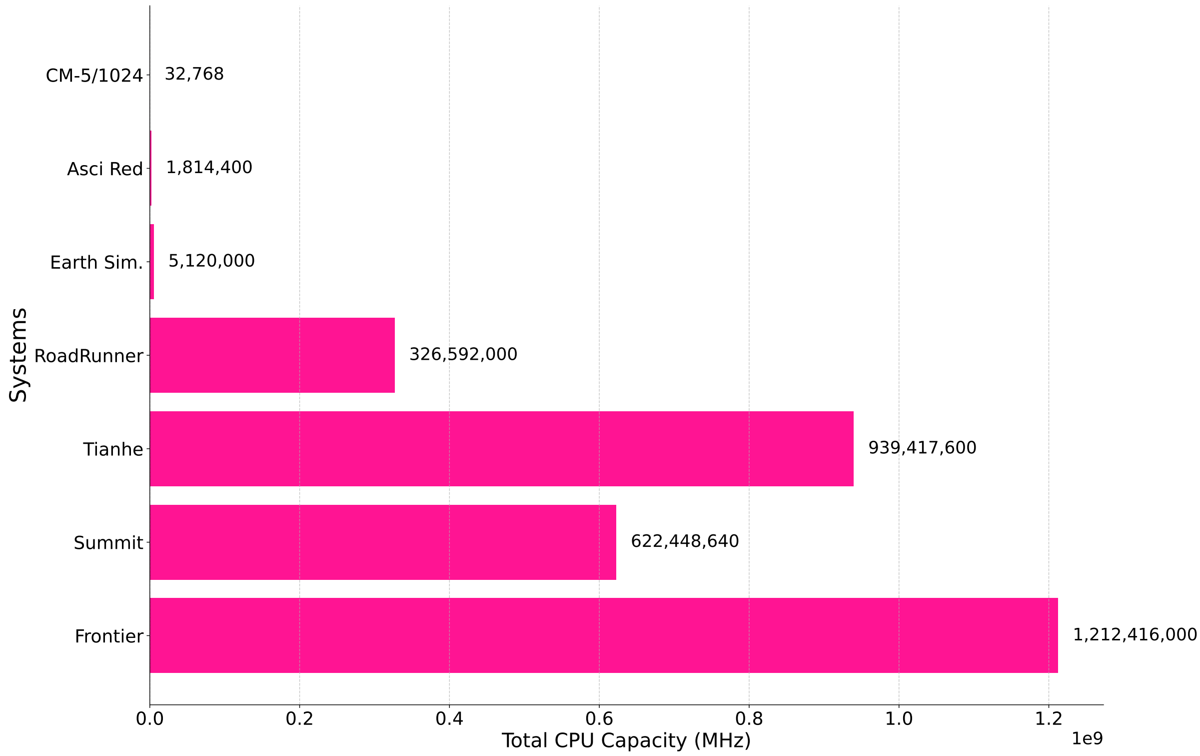

Figure 4.1: Total CPU capacity of each system.

Currently, OpenDC only accounts for CPU core count and core speed, without considering different CPU architectures or types. Consequently, the impact of variations in CPU architectures or types on energy consumption is not modeled in our simulations.

Figures 4.1 and 4.2 showcase the total CPU capacity of each system and the per-host CPU capacity of each system, respectively. We see that the per-host capacities and total system CPU capacities differ in progression. For instance, while Summit has a lower total CPU capacity than Frontier, it has a greater per-host CPU capacity. This can lead to different performance and energy efficiency characteristics. Systems with higher per-host CPU capacities may perform better for tasks requiring more computational power per node. In contrast, systems with a higher total CPU capacity can handle a larger volume of tasks. Acknowledging these differences is important for selecting appropriate workloads based on the available system resources.

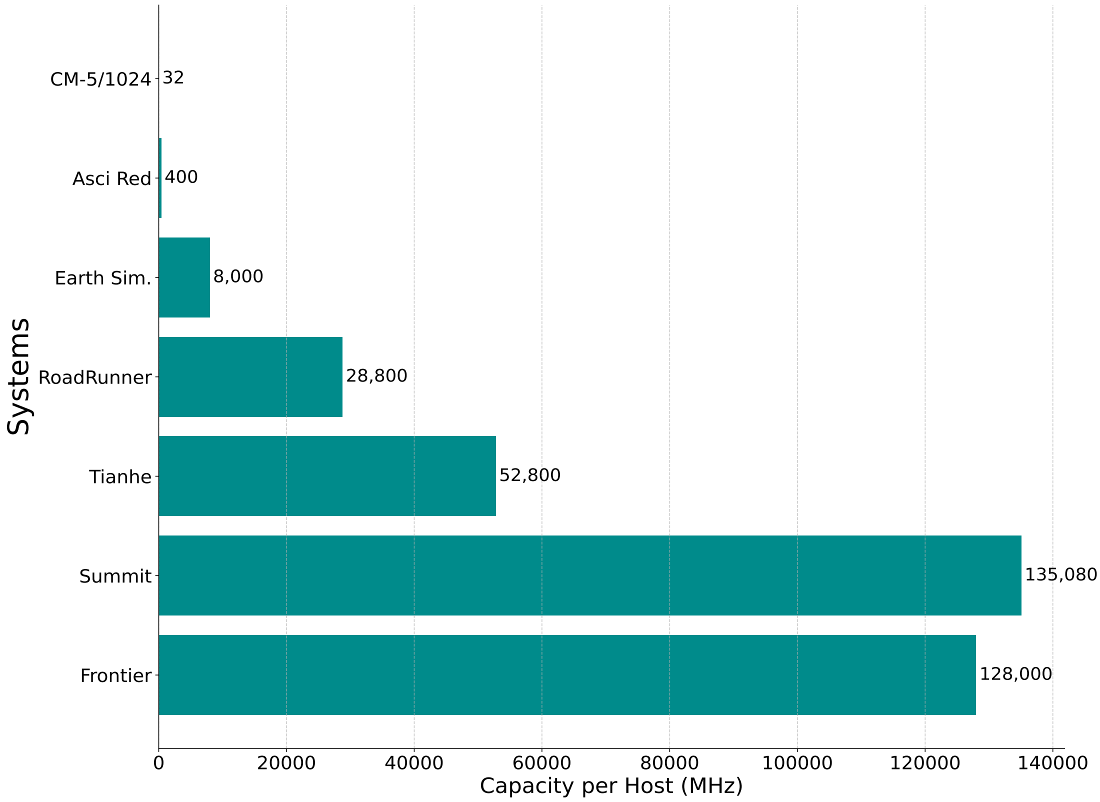

Figure 4.2: CPU capacity per host of each system.

4.3 Scaling Workloads for Diverse Configurations

The workloads we use are traces designed to imitate the resource consumption patterns of real-world applications without performing real computations. However, using trace data limits the ability to select different configurations because tasks are dependent on the original computational resources under which the data was designed for (49). As a result, the trace data is only applicable to similar machine configurations, restricting its flexibility in diverse simulations.

Scaling workloads refers to the process of adapting the computational demands of a workload to align with the configurations of different computer systems. This process considers components such as runtime, CPU core counts, CPU capacity, and the number of hosts to ensure compatibility and optimal performance across various system topologies (35, 49).

This is a complex task because it contrasts with the typical practices usually employed. For instance, in serverless computing where Function-as-a-Service (FaaS) is utilized, clients have workloads consisting of specific tasks with defined compute requirements. They request resources from cloud service providers, which then allot the necessary computational resources(50). However, in a simulation setting, our approach deviates from this conventional method. Instead of requesting specific computational resources for our workloads, we aim to run the same workloads across various systems within OpenDC. Therefore, we must take an unconventional approach: rather than obtaining additional resources to meet the workload demands, we need to adapt the workloads to fit within the existing resource constraints of each simulated system. By doing so, we achieve accurate performance metrics and resource utilization across diverse environments.

Currently, OpenDC does not support the scaling of workloads according to different configurations. It can simulate workloads that match the configuration of the system being analyzed. However, if there is a significant mismatch between the workload's resource requirements and the system's available resources, such as when the system is much larger than the workload or vice versa, OpenDC is not able to adjust the workload to fit the duration and CPU requirements appropriately. This limitation affects the accuracy and relevance of simulations involving diverse configurations.

Our research on analyzing the energy consumption of various computer systems from different decades, ranging from systems with 200 MHz core speeds to those with 1000 MHz, leads to the requirement to consider different resource configurations. Given the importance of accurately modeling these configurations, we must ensure that the workloads we use are consistent and simulate these systems with the highest accuracy. Therefore, we explore various methods to scale workloads to fit systems from different decades.

4.3.1 Requirements

To ensure that our method for workload scaling is well-designed to be adapted for various system configurations and workload types, we begin by gathering the requirements. Before finalizing our process for scaling workloads, we explore several methods, and each iterative approach allows us to refine our strategy. This process helps us to identify our priorities and define the scope of our approach.

As a crucial part of validating our workload scaling process, we employ the squashed area method (49). The squashed area method serves as a measure of the amount of processing power for a workload by calculating the resource consumption of each task in the workload. Resource consumption is calculated by the Equation 4.1, where R is the resource consumption, U is the CPU usage, and T is the task duration.

This method helps us identify any differences in computational work before and after scaling. By comparing the squashed area of the original workload trace with that of the scaled workload trace, we can confirm whether the total computational work remains constant or changes within acceptable limits, even as the resources and duration are adjusted. This validation ensures the integrity and accuracy of our scaling process across different system topologies.

$$R = U \times T \tag{4.1}$$

It is important to note that any scaling of a workload to match different topologies inevitably alters the original distribution of the workload. Balancing the trade-off between maintaining the modeling accuracy of different systems and preserving the integrity of the workload is a critical consideration when defining the methodology of scaling workloads.

Below is an overview of the most important requirements that we find the workload scaling must adhere to.

R1 Ensure workload compatibility across all system topologies

The workload should be executable across all system topologies that we select in Section 3.2. The scaling factor should be flexible enough to adjust the workload demands whether the system has limited resources or more than enough capacity. By doing so, the workload can be used to evaluate systems from all decades for a comprehensive analysis of energy consumption.

R2 Preserve workload integrity during scaling

Scaling the workload should preserve its fundamental characteristics and behavior. Maintaining this consistency is crucial for ensuring the validity of our results. Altering basic characteristics, such as workload distribution, computational intensity, and duration, without a reasonable approach introduces variability that can invalidate our findings. This would make it difficult to attribute differences in energy efficiency solely to the systems being compared.

R3 Load-balance to improve system resource-utilization

The workload should be adapted to effectively leverage the available system resources. This ensures that the system shows its best performance without being under or over-utilized.

Proper utilization leads to more accurate performance measurements and improves our ability to observe the true capabilities and limitations of a system.

4.3.2 Refining the Workload Scaling Process

To develop an effective and accurate method for workload scaling, we experiment with an iterative process of evaluation. Each phase is built upon the previous one, identifying the weaknesses and improving the overall strategy. Below we present each phase and the improvements we make.

Phase 1: Initial Proportional Scaling of Resources and Duration

In the first phase, we scale the resources and duration of tasks in proportion to the system being modeled. For each task, we determine a scaling factor based on the CPU capacity of the workload task relative to the system being modeled. We then adjust each task's CPU requirements and duration according to this factor. However, this results in tasks taking as much as 500 years to complete, which is highly unrealistic for several reasons. It is impractical to assume that a task could run continuously for five centuries (Except if we are simulating the supercomputer Deep Thought from The Hitchhiker's Guide to the Galaxy) because the energy requirements over an extended period would evolve and change significantly.

Additionally, in the context of service-based workloads such as Bitbrains, scaling the duration alters the original characteristics of the workload significantly, resulting in not adhering to **R2**. No client can wait for a service to finish for such an extensive period of time. Therefore, we conclude that it is necessary to limit the duration of the scaled tasks according to the nature of the specific workload. Not having a lower limit on the duration also poses a problem, and it is essential to establish realistic boundaries for task durations to preserve the integrity of the workloads.

This approach assumes perfect scaling of CPU usage, meaning that CPU usage is directly proportional to the CPU specifications of the system. However, in reality, performance does not increase linearly with resources due to factors such as overheads, bottlenecks, and resource contention. These factors also contribute to the energy efficiency of the CPUs being different. Due to limited information being available on CPU power consumption metrics and the scope of our project, we proceed with the assumption of perfect scaling for CPUs.

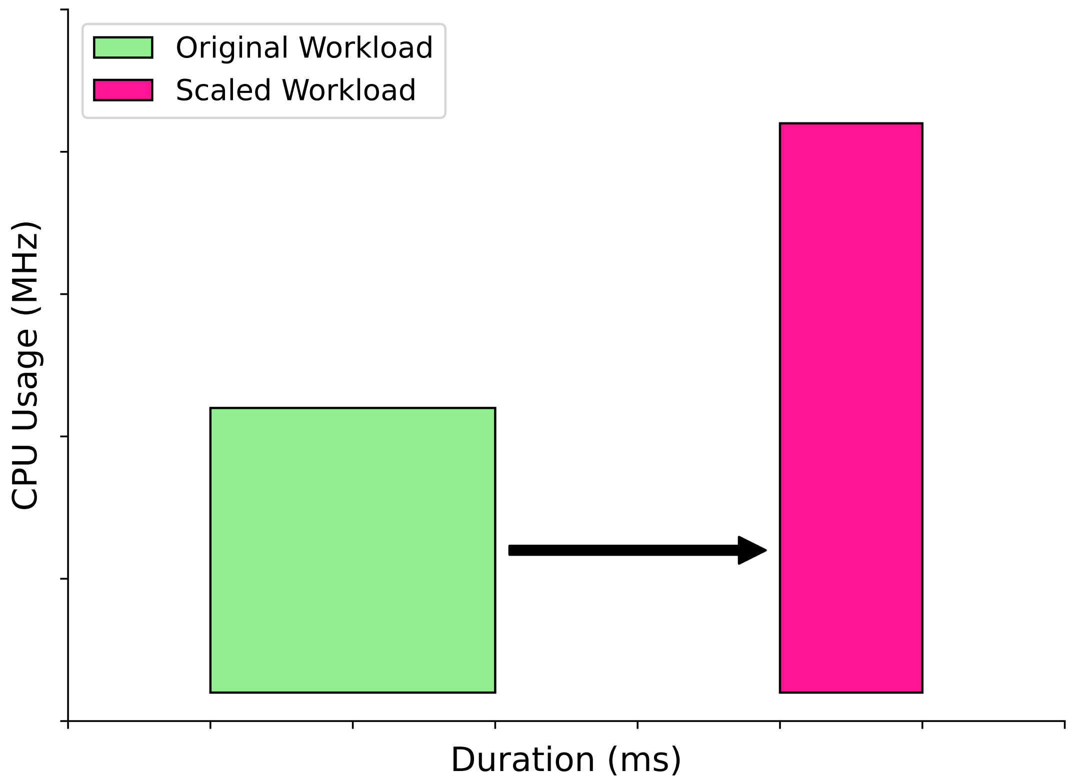

Figure 4.3: Demonstration of the scaling up process for fixed-work workloads.

Figure 4.3 shows how a fixed-work workload scales up and Figure 4.4 shows how it scales down. The illustration of the scaling process is a simple example to show the perfect scaling and how the shape of the workload changes, however, it does not show the cases where task duration becomes an unrealistic value such as 500 years. For example, if the original task in the workload uses 2 MHz of CPU and takes 2 milliseconds to execute, and we have a system with a CPU capacity of 1 MHz, the scaling factor is calculated as $1/2$. This means the new CPU usage is 1 MHz, and the duration is 4 milliseconds. Conversely, if the system has 4 MHz capacity, the same task can be executed in just 1 millisecond instead of 2.

Phase 2: Incorporating Constraints on Task Duration

In the second phase, we introduce constraints on task duration, allowing us to execute workloads on all systems and successfully meet requirement **R1**. These constraints are tailored to specific workloads being scaled:

Fixed-Work Trace Duration Constraints:

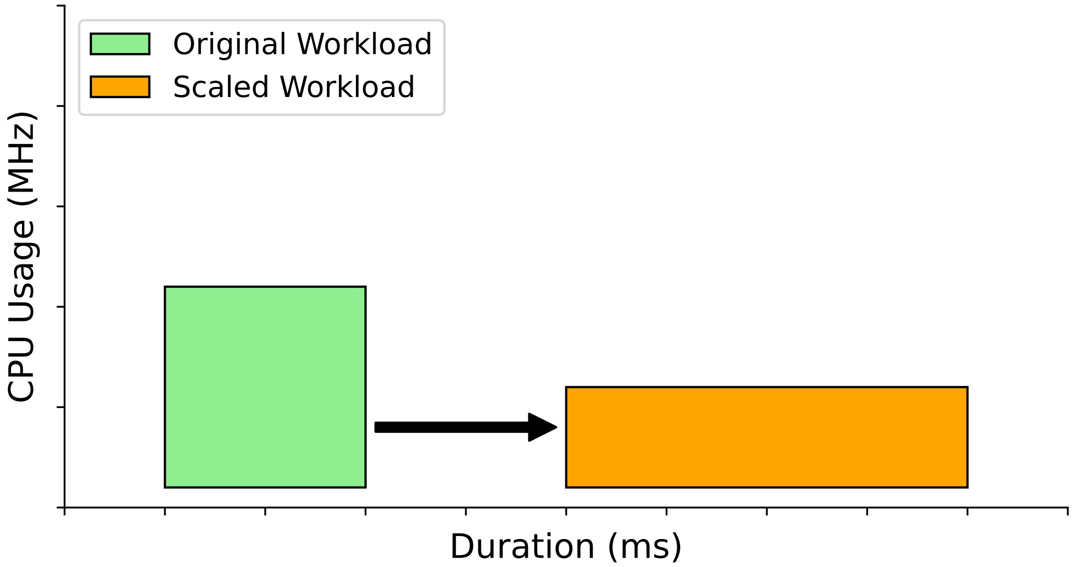

Figure 4.4: Demonstration of the scaling down process for fixed-work workloads.

- Upper Bound: The maximum duration of any task is limited to the total duration of the trace. If scaling causes a task's duration to exceed this limit, it will be capped at the trace's total duration.

- Lower Bound: The minimum duration of any task is limited to 1 second as per the bounded slowdown (51).

Fixed-Time Trace Duration Constraints:

Due to the service-based nature of the fixed-time trace (Bitbrains) we use, explained in 3.3, where workloads are designed to run at specific, fixed times to meet service level agreements (SLAs), it is not appropriate to alter the runtime of the tasks. Therefore, we scale the CPU requirements of the workload according to the specific system being used, while keeping the task duration unchanged. This approach ensures that the original characteristics of the workload are preserved while adapting the computational load to different systems.

Figure 4.5 illustrates how a fixed-time workload scales down. The computational work done is decreased by the process because smaller CPU resources result in less work being completed in the same duration. Figure 4.6 shows the scaling up process where the scaled computational work is greater than the original because larger CPU resources allow more work to be done in the same duration.

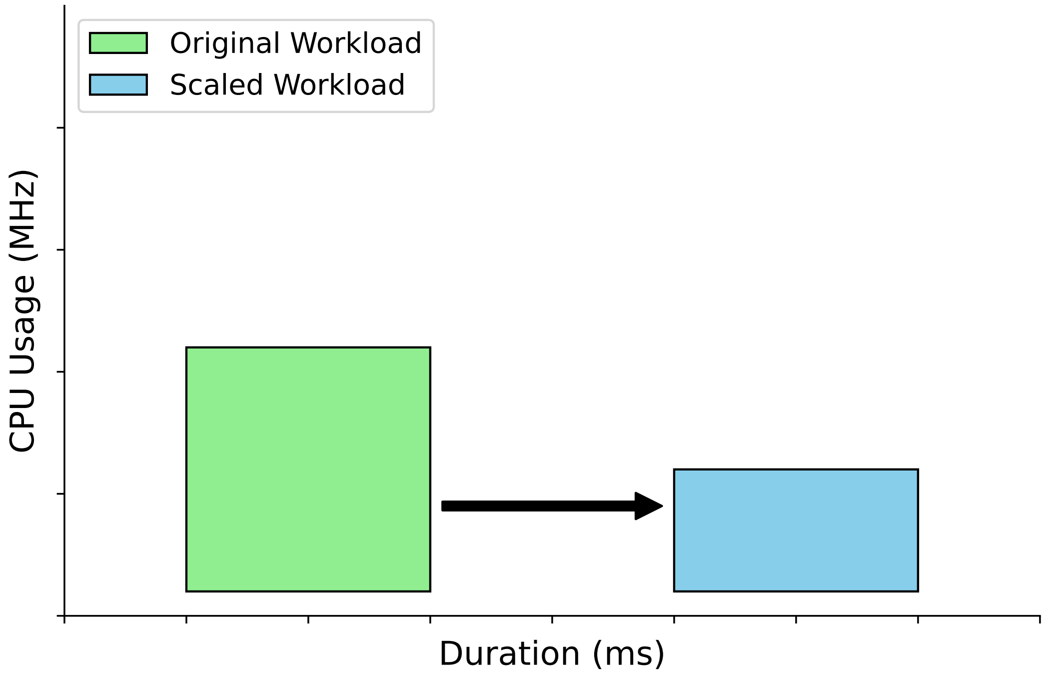

Figure 4.5: Demonstration of the scaling down process for fixed-time workloads.

Phase 3: Enhancing Resource Utilization through Task Duplication

The first two processes focus on maintaining workload integrity and ensuring compatibility across different system topologies. However, they do not sufficiently optimize system resource utilization for smaller systems that do not have enough resources to complete the workload. In Scaling Process 3, we aim to address all three of our requirements: **R1**, **R2**, and **R3**. Building on the foundations of the first two processes, we introduce task duplication. This involves replicating incomplete tasks and distributing them to available nodes to improve resource utilization. This allows for smaller systems to be able to complete a bigger fraction or all of the workload allowing for a better comparison of performance between systems. Figure 4.7 showcases how task duplication is performed for the Fixed-Time Bitbrains workload. The incomplete tasks are duplicated to ensure the amount of computational work done is the same. Figure 4.8 shows how task duplication looks like for the Fixed-Work Surf Lisa Workload where the incomplete tasks due to limited duration are duplicated. For instance, the fixed-work task duplication in Figure 4.8 is limited to 3 milliseconds of task duration instead of being scaled down to 4 milliseconds. The amount of computational work done is 4 (measured in MHz * ms), and the incomplete portion of the task is duplicated to keep the computational work of the scaled task the same as the

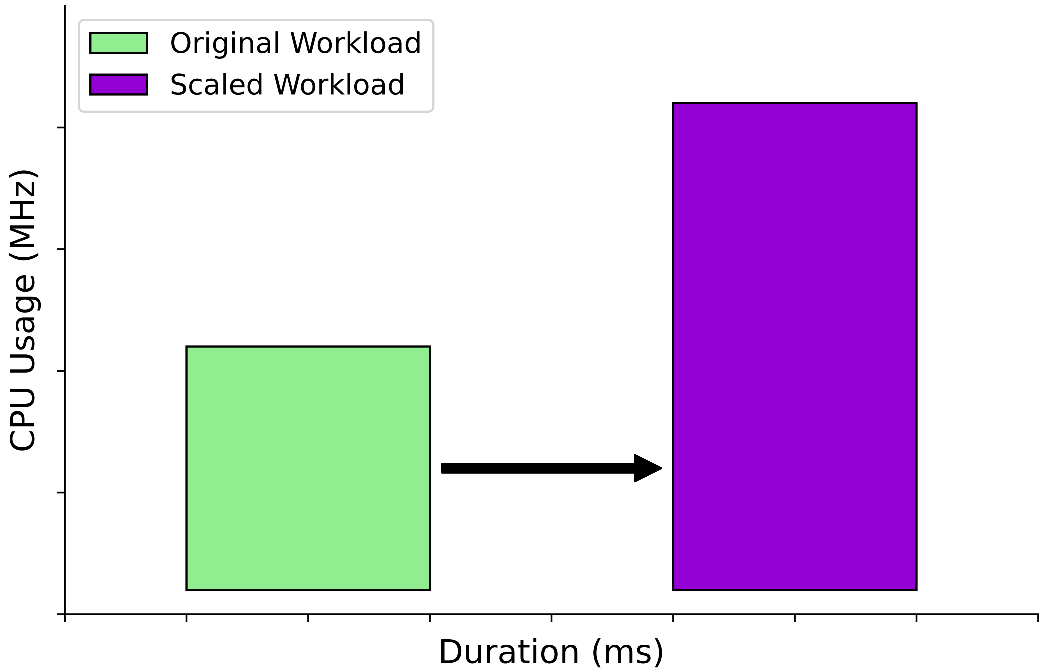

Figure 4.6: Demonstration of the scaling up process for fixed-time workloads.

original task. Then, with the limited duration the computational work done becomes 3 (MHz * ms), and the incomplete portion, which is 1, is duplicated as another task.

4.3.3 Scaling Method Employed for Bitbrains

Our approach keeps the fundamental characteristics of the Bitbrains intact by keeping the duration of tasks unchanged and the workload consistent. However, keeping the duration unchanged inevitably results in some tasks being cut off before completion. We record these interrupted tasks to gain insight into what fraction of the workload can be accomplished within the specified time frame, helping us understand the capability of systems from different decades for service-based workloads.

We transform the workload by applying a scaling factor to each task, derived from the trace and the specific system to be modeled. A scaling factor is a numerical value used to adjust the target variable relative to a baseline. In our context, the baseline is defined by the computational demand and CPU requirements as provided by the workload trace. This approach ensures that the workload is appropriately scaled to reflect the characteristics and capabilities of the system being modeled.

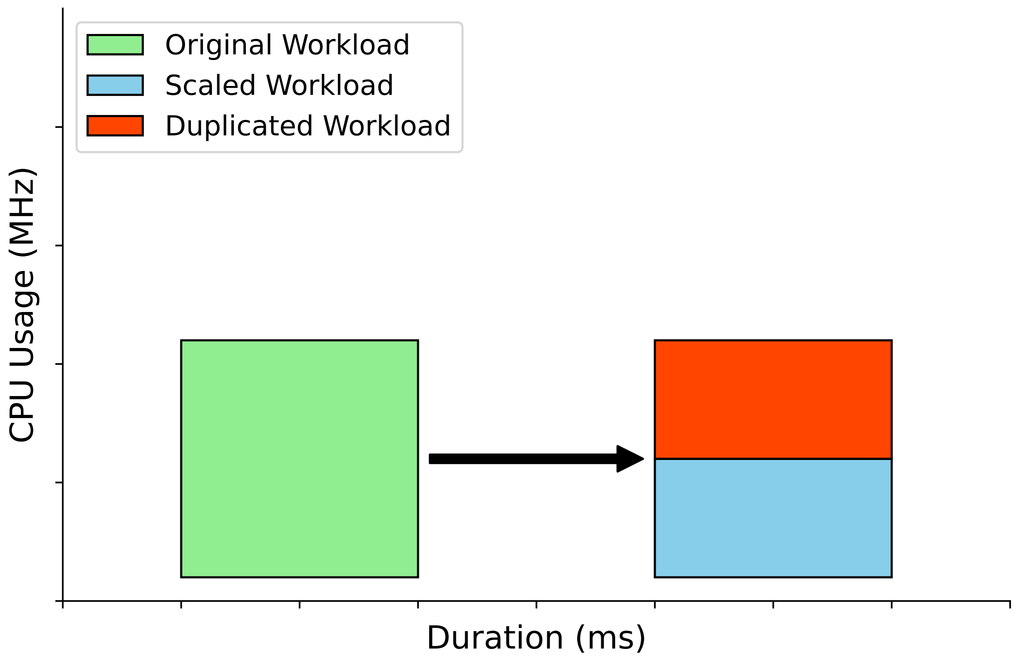

Figure 4.7: Task Duplication for fixed-time workloads.

For the tasks that are incomplete after scaling, we employ task duplication which is explained in Section 4.3.4.

Firstly, we adjust the CPU resources of the task by changing the CPU capacity of the task and the total CPU Core count available per node to match the system's CPU resources. In the workload trace, the total CPU capacity is represented, whereas in the system topology, the CPU core speed is provided. Therefore we need to calculate the total CPU capacity of the system. To obtain the total CPU capacity of the system **(C)** of the system, we multiply the CPU core count **(N)** by the CPU core speed **(S)** which is defined in Equation 4.2.

- **CPU Core Count (N):** The total number of CPU cores available in a single node of the system.
- **CPU Core Speed (S):** The CPU core speed per node, measured in MHz.
- **CPU Capacity (C):** The CPU capacity needed to run the server in the workload trace, measured in MHz.

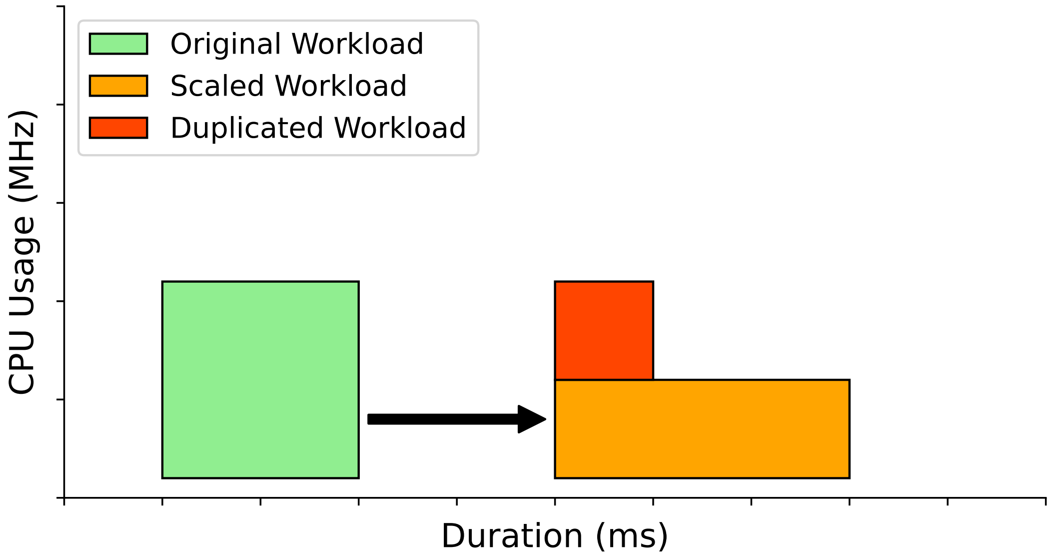

Figure 4.8: Task Duplication for fixed-work workloads.

$$C = S \times N \tag{4.2}$$

The scaling factor (α) adjusts the workload based on the difference between the CPU capacity of the system node and the CPU capacity required by the task in the workload trace. It is calculated by dividing the system CPU capacity per node (C_{system}) by the trace CPU capacity per node (C_{trace}) (see Equation 4.3).

$$\alpha = \frac{C_{\text{system}}}{C_{\text{trace}}} \tag{4.3}$$

If the system has more CPU capacity than the tasks require, the scaling factor (α) will be greater than 1. This indicates that the system has excess resources for the workload, necessitating an increase in the workload's computational demands to match the system's higher CPU capacity.

Conversely, **if the system has less CPU capacity than the tasks require**, the scaling factor (α) will be less than 1, indicating that the system cannot handle the original computational demand. In this case, the CPU requirements of the workload will need to be scaled down accordingly to match the system's lower CPU capacity.

We scale the CPU usage of the task (U_{scaled}) by multiplying the task's CPU usage (U_{task}) by the scaling factor (α), see Equation 4.4. We then scale the CPU core count (N_{scaled}) according to the scaled CPU usage (U_{scaled}) and system CPU core speed (S_{system}) and round the resulting value to the nearest integer in Equation 4.5.

$$U_{\text{scaled}} = U_{\text{task}} \times \alpha \tag{4.4}$$

$$N_{\text{scaled}} = \frac{U_{\text{scaled}}}{S_{\text{system}}} \tag{4.5}$$

By using Equations 4.4 and 4.5, we ensure that regardless of the CPU capacity of the system that is being modeled the percentage of CPU usage relative to CPU capacity will be kept constant, but the resulting CPU usage will be lower or higher depending on the system's capacity. Although keeping the percentage of CPU usage constant leads to low utilization in smaller systems with already low resources, it maintains the workload consistency (**R2**) and how a workload behaves relative to available CPU resources.

If we had taken a different approach and maximized the percentage of CPU usage across all nodes instead of keeping it constant, smaller systems would become over-utilized. This would lead to performance degradation and potential overloading of those systems. Additionally, it would disrupt the proportional computational demands of the workload, resulting in a less accurate basis for comparing different systems.

We expect the resulting squashed area to be larger compared to the original task for systems with a scaling factor greater than 1 and smaller for systems with a scaling factor less than 1. This stems from our constraint on keeping the duration of the trace unchanged. Therefore, the total computational work calculated by the squashed area will be higher for larger systems because they can complete more work within the same time frame. In contrast, it will be lower for smaller systems with limited capacity due to completing less work within the same duration.

4.3.4 Scaling Method Employed for Surf Lisa

The fixed-work Surf Lisa scaling trace follows a similar approach to the Bitbrains trace in terms of scaling the CPU usage and CPU count by the scaling factor. However, instead of keeping the duration unchanged, we also scale the duration. If the scaled duration exceeds the original task duration, we limit it to the end time of the trace, referred to as the maximum duration of the whole trace. If the scaled duration is shorter than 1 second we limit it to 1 second, referred to as the bounded slowdown (51). Additionally, to better utilize the smaller systems we implement duplication of tasks that are left incomplete due to the limited duration. After scaling and duplicating the tasks, we validate our approach by comparing the squashed area of the original workload trace to that of the scaled workload trace. If the squashed areas are identical, it confirms that the computational workload has remained constant despite changes in computational resources and duration.

Scaling the Duration

First, we scale the original duration of the task defined in Equation 4.6, where the original task duration (T_{orig}) is divided by the scaling factor (α). This adjustment ensures that for smaller systems, where the scaling factor is less than 1, the duration is appropriately increased and for larger systems, where the scaling factor is greater than 1, the duration is shortened to match the increased capacity, covering both scenarios.

$$T_{\text{scaled}} = \frac{T_{\text{orig}}}{\alpha} \tag{4.6}$$

After scaling the duration we check for the edge cases. We handle two scenarios: when the duration is shorter than 1 second and when the duration exceeds the end time of the entire trace. In scenarios where the duration is shorter than 1 second, we limit it by the lower bound of 1 second which corresponds to the bounded slowdown (51) and in scenarios where the duration exceeds the end time of the entire trace, we limit it by an upper bound.

Limiting the Duration by the Lower Bound

In scenarios where the scaled duration is shorter than 1 second, we impose a lower bound of 1 second. Limiting the lower bound is crucial for maintaining practical and manageable task durations. For the lower bound, we need to adjust the CPU usage to reflect the same amount of work in 1 second as initially intended in the scaled duration. We calculate the new CPU usage by multiplying the scaled CPU usage (U_{scaled}) with the scaled task duration (T_{scaled}). Finally, given that we have enforced a minimum duration of 1 second, we divide the product by 1. While dividing by 1 mathematically does not change the value, it conceptually normalizes the CPU usage for the enforced lower bound (see Equation 4.7).

$$U_{\text{min}} = \frac{U_{\text{scaled}} \times T_{\text{scaled}}}{1} \tag{4.7}$$

Limiting the Duration by the Upper Bound

In scenarios where the scaled duration exceeds the end time of the entire trace, we impose an upper bound corresponding to the trace's end time. This ensures that the task durations do not surpass the trace duration. The constrained task remains incomplete with this limitation. This results in the modeled system being unable to execute the full workload, however, this is not due to a lack of resources but rather our effort to maintain a consistent scaling of tasks to preserve the workload characteristics.

We do not adjust the CPU usage in this case because the aim is to execute the maximum CPU work within the limited upper bound. The maximum duration available T_{max}, relative to the trace end time, is calculated using the Formula 4.8, where T_{end} is the trace end time, and T_{start} is the task start time.

$$T_{\text{max}} = T_{\text{end}} - T_{\text{start}} \tag{4.8}$$

Task Duplication

To handle incomplete tasks, we duplicate them and set all of their durations to T_{max}. The CPU usage for these duplicated tasks corresponds to the leftover work from the original constrained tasks. This approach ensures that the full computational workload is executed by the system while maintaining consistent scaling. In the cases where the leftover work exceeds the system CPU capacity per node, we split the duplicated task into another set of sub-tasks that has the same limited duration but the maximum system CPU capacity per node.

To better understand this process, let's take a look at the specifics. We first calculate the job completion ratio to find out the fraction of the task that was completed. We divide the resource consumption of the scaled task by the resource consumption of the original task to find the job completion ratio, as shown in Equation 4.9, where ρ is the job completion ratio, R_{scaled} is the resource consumption of the scaled task, and R_{orig} is the resource consumption of the original task (defined in Equation 4.1).

$$\rho = \frac{R_{\text{scaled}}}{R_{\text{orig}}} \tag{4.9}$$

To calculate the remaining CPU usage after the task is limited by the maximum duration, we use Equation (4.10). We first calculate the leftover CPU work by multiplying R_{orig} and uncompleted fraction of the work (1 - ρ), then find the remaining CPU usage by dividing it by the maximum duration available (T_{max}). We divide it by the maximum duration because for all the duplicated tasks the duration is T_{max}, as mentioned before.

$$U_{\text{r}} = \frac{R_{\text{orig}} \times (1 - \rho)}{T_{\text{max}}} \tag{4.10}$$

Finally, we find the number of duplicate rows (D) needed by dividing the remaining CPU usage (U_{r}) by the system CPU capacity per node (C_{system}) in Equation 4.11. This ensures that the remaining CPU work does not exceed the CPU capacity of the system per node and is instead dispersed into sub-tasks.

$$D = \frac{U_{\text{r}}}{C_{\text{system}}} \tag{4.11}$$

Task duplication for incomplete tasks ensures that the system's computational resources are effectively utilized. Unlike maximizing CPU usage across all tasks, which would not preserve the workload integrity (**R2**), this method focuses solely on duplicating incomplete tasks. Although this approach might lead to over-utilization in smaller systems, it provides a basis for comparison that reflects realistic workload conditions. The advantage of task duplication is that it ensures load-balancing to improve system resource utilization (**R3**).

For smaller systems, this can result in potential overloading, which is expected when running the same workload across systems from different decades. Naturally, older systems have fewer resources and are prone to overloading under modern workloads. Nevertheless, we can still make an effort to balance the number of duplicate sub-tasks generated. Limiting the number of duplicate tasks can prevent over-utilization but might result in a lower job completion ratio, indicating a trade-off between system performance and workload completion. Different scaling configurations such as limiting or not limiting the duplicate tasks generated can be done to observe various outcomes and comprehensively evaluate system performance in future studies.

4.4 Discussion

This chapter enabled us to introduce and explain creating topologies and scaling workloads to accurately model systems in OpenDC for **RQ2**. In Section 4.2, we presented the system topology information necessary to model systems in OpenDC. Then, we summarized the CPU information of each system to observe the differences in CPU technologies across decades. Finally, we looked into the per host and total CPU capacity per system to compare the capacities of the different systems we model.

In Section 4.3 we explored methods for scaling workloads to align with different system configurations, focusing on requirements such as ensuring compatibility, preserving workload integrity, and load-balancing. Our iterative process led to a final scaling method that keeps task durations unchanged for Fixed-Time workloads and scales both CPU usage and duration for Fixed-Work workloads. Task duplication was used to handle incomplete tasks to improve system resource utilization by load-balancing. By validating our approach with the squashed area method, we confirmed the accuracy of our scaling process.

5

Trace-Based Simulation Analysis of the Evolution of Energy Consumption from the 1990s to 2020s

5.1 Overview

In this chapter, we conduct a trace-based simulation analysis of the evolution of energy consumption in large-scale computer systems from the 1990s to the 2020s. This section is crucial for answering **RQ3** to provide insights into how energy efficiency has progressed over the decades. We detail our experimental setup in Section 5.2, including the system topologies, workloads, and the power consumption model employed in OpenDC. Afterward, we evaluate energy consumption trends in Section 5.3 by simulating both Fixed-Work and Fixed-Time workloads on our chosen systems, we analyze trends in energy efficiency and compare our findings with Koomey's Law and TOP500 metrics. Finally, in Section 5.4 we combine the results of our Honours Report and Section 5.3 to get a broader picture of how energy efficiency progressed across seven decades.

Main Findings

MF1 Koomey's law alone proves insufficient to evaluate the energy efficiency of large-scale computer systems.

MF2 Our proposed models, Fixed-Time and Fixed-Work using OpenDC, capture the nuances and complexities of various workloads and computer systems, but their accuracy depends on how well the parameters are set and the exhaustiveness of the simulator.

MF3 Our method to scale workloads and compare the performance of systems provides flexibility in the selection and simulation of different workloads and systems.

MF4 Fixed-Time and Fixed-Work models indicate a deceleration in energy efficiency starting from the 2000s.

MF5 Comparing different energy efficiency models, allows more comprehensive evaluation of trends.

MF6 Variations in workload characteristics in simulation highlight different energy efficiency nuances present in large-scale computer systems.

MF7 Including accelerator-based components, such as GPUs, provides a more accurate reflection of the energy efficiency improvements in the last decade.

MF8 Regardless of how energy efficient a system is, if the workload executed does not appropriately match the system's capacity, energy waste will be significant. Efficient utilization of resources is crucial to minimize energy consumption and maximizing computational efficiency.

5.2 Experimental Setup

This section details our experimental setup; the inputs, configurations, physical machine, and methodology.

We present the summary of our experimental setup in Table 5.1.

System Topology Setup

The fixed-work and fixed-time traces are provided along with an environment description of the data center's topology. This topology includes clusters of machines, each specified by the number of cores, core speed (GHz), number of hosts, memory (GB) available to each host, and core count per host.

Component	Specification
Operating System	MacOS Sonoma
Processor	Apple M1
Number of Cores	8 cores
Memory	16 GB RAM
Python Implementation	CPython
Python Version	3.11.5

Table 5.1: Experimental setup.

For each host, the total memory is fixed at 512 GB. This decision stems from OpenDC's current limitations in detailed memory simulation. OpenDC only conducts a binary check to confirm whether the memory capacity is adequate for the simulated tasks. It does not account for memory-related variations in energy consumption or performance in its output. By setting a constant memory value across all systems, we simplify the simulation process to allow for consistent comparisons and reduce the complexity introduced by variable memory configurations. This approach prioritizes ensuring sufficient memory for task execution while considering the limitations in the current simulation model.

OpenDC uses a linear power model for all system topologies. This approach is taken because no conclusive information is available on the specific power models of CPUs. The linear power model simplifies the simulation process by assuming a direct proportionality between resource usage and power consumption. In practice, the linear models may oversimplify the power consumption patterns because power consumption does not have a proportional relationship with factors such as the intensity and type of workload used, the number of cache hits/misses, and the frequency of the CPU (52). However, due to the lack of detailed information on how the specific CPUs in our system topologies are affected by these parameters, we are forced to use a simplified assumption for the power consumption model.

TOP500 uses the base clock speed of CPUs in the LINPACK benchmark to ensure consistency across different systems and have a reliable metric for performance comparison (11). We adopt this approach in the system topology of OpenDC, using the base clock speed of CPUs in the CPU Core Speed metric. By doing so, we align with the established practices of TOP500, which provides a consistent and reproducible framework for performance evaluation that is comparable with industry standards.

Due to the varied scale of the computer systems we model, our traces cover both high-demand scenarios on smaller systems and low-system utilization scenarios on larger, modern systems. To ensure an equal baseline for comparison of energy efficiency, we filter out the idle hosts of the systems before analyzing the results. Idle hosts are identified by the hosts that have 0 guests running in the output metrics of the simulation. Focusing on the active resource usage allows us to generate accurate energy efficiency results with OpenDC that align more closely with the energy efficiency data provided by TOP500 (53).

5.2.1 Scope and Limitations of Simulating Power Consumption

Power use in computer systems is driven by more than just the power model and CPU frequency. Computer systems include losses in power supplies and electricity used by disk drives, GPUs, network hardware, cooling infrastructure, and other components (52). The energy efficiency associated with these components does not necessarily improve at rates comparable to the trends identified in our project.

We do not consider these additional components in our power consumption model, similar to the approach taken by Koomey, which focuses primarily on the energy use of CPUs without accounting for the entire power consumption of the data center infrastructure (9). Additionally, OpenDC cannot currently simulate the power consumption of all the diverse components indicated. As discussed in Section 2.2.2 it is currently unfeasible to model all components of a data center with high accuracy, therefore it is more realistic to focus on specific components. While this simplification means we do not capture the full scope of power usage, it aligns with Koomey's models and ensures our results are reproducible. In summary, this allows us to maintain a manageable scope and the clarity of our findings to provide insights that can be integrated into the broader research conducted by our peers.

One crucial component that has been omitted in the modeling but deserves discussion is the role of GPUs (Graphics Processing Units) in large-scale computer systems, as detailed in Section 2.2.2. With the rise of high-performance computing (HPC) platforms, accelerator-based components, such as GPUs, have been widely adopted over the last decade due to their ability to handle parallel processing tasks efficiently. Their contribution to the performance and efficiency of large-scale systems is significant and should not be ignored when evaluating computational performance and energy efficiency. For instance, Summit (2018) features NVIDIA Volta V100 GPUs in each node(39), and each GPU offers the performance of up to 32 CPUs. However, the energy consumption of GPUs is also higher with their high performance. While Summit's CPUs have a power consumption (TDP) of 190W, its GPUs consume 300W. It is important to note that the energy efficiency of CPUs versus GPUs depends on the specific application and evaluation methods used (54). Most modern large-scale systems now incorporate GPUs into their compute

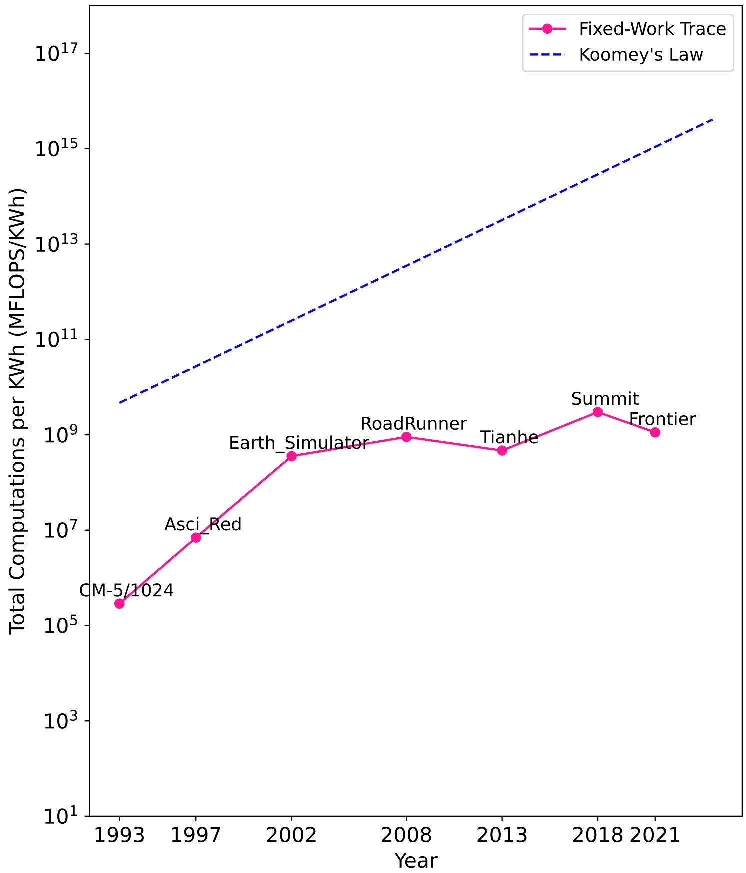

Figure 5.1: Evolution of energy consumption for the fixed-work trace.

nodes, with 9 out of the top 10 systems on the June 2024 TOP500 list utilizing GPUs, emphasizing their widespread adoption and importance (55).

5.3 Evaluation of Energy Consumption Trends from the 1990s to the 2020s

Fixed-Work SURF Lisa Trace

Figure 5.1 shows the number of computations per kilo-watt-hour (kWh) of electricity consumed for various system topologies from 1993 to 2021 for the Fixed-Work SURF Lisa trace. To better visualize the trends, the plot uses a logarithmic scale. Significant improvements in energy efficiency are apparent from the progression of the plot. However, these improvements do not align with the pace that Koomey has suggested. Particularly, starting from the 2000s the rate of increase of efficiency appears to slow down.

The first 20 years, from 1993 to 2002, show a steep increase in energy efficiency. Asci Red demonstrates a significant improvement after CM-5/1024 and this trend continues with the Earth Simulator in 2002, marking almost two orders of magnitude increase in the energy efficiency.

From 2003 onwards, the increase in energy efficiency starts to slow down. RoadRunner (2008), Tianhe (2013), Summit (2018), and Frontier (2021) show incremental increases. The overall pace of improvement decelerates compared to the early 1990s.

The resulting plot only accounts for the different scheduling techniques and dynamic workloads as we used the Surf Lisa trace. However, it does not account for failure models or interference in the execution. The absence of these types of non-linearities is crucial to consider for a more comprehensive evaluation of energy efficiency trends in computational systems. However, even without the inclusion of the additional non-linearities we still see the resulting trends diverging from Koomey's Law.

Koomey's Law posits that the energy efficiency of computations doubles approximately every 1.5 years (9). However, the trends depicted in our findings suggest that while there have been significant gains in energy efficiency the improvements have not sustained the exponential growth rate predicted by Koomey's Law, especially after the 2000s. According to our plot for SURF Lisa, the energy efficiency of computations doubles approximately every 2.04 years. This divergence can be attributed to a variety of factors. Notably, our study takes into account dynamic workloads and various scheduling techniques, unlike Koomey's model, which assumes static conditions and peak power usage. The introduction of dynamic workloads introduces variability and more realistic computational loads, leading the efficiency trends to fall below Koomey's idealized curve. This approach provides a more accurate representation of energy efficiency of the fluctuating energy usage nature of real-world tasks.

An interesting point for discussion is that Summit (2018) has a higher energy efficiency than Frontier (2021). This difference arises from variations in their CPU capacities per host and total system CPU capacity, as shown in Figures 4.1 and 4.2. While Frontier has almost twice the total system CPU capacity, Summit has a higher CPU capacity per host. This means that Frontier can handle more tasks overall due to its greater total capacity, however, Summit can perform more computations per individual host. Both of the traces we are running do not fully utilize all hosts of either system and are not overly demanding. In traces where the computational demand is not high Summit appears to be more efficient. The lower power consumption of Summit makes it more suitable for smaller workloads. Conversely, for larger and more demanding computations requiring a higher number of hosts being active, Frontier's larger total CPU capacity would compensate for its higher energy consumption making it more effective for bigger workloads. This highlights an important principle in large-scale computing: no matter how energy efficient a system is if the workload does not appropriately match the system's capacity, energy waste will be significant. Efficient utilization of resources is crucial to minimizing energy consumption and maximizing computational efficiency. To correct these biases in the future, the next step should involve using more computationally demanding workloads to assess the true performance and efficiency of large-scale computer systems.

As data-centric processing tasks and customized computing components become increasingly prevalent in modern data centers, the limitation of OpenDC in simulating accelerator-based components such as GPUs, DPUs, FPGAs, and TPUs should be a significant consideration for future studies (24). Addressing this limitation is crucial for accurately representing energy efficiency trends. Figure 5.1 reflects the industry's shift from a CPU-centric architecture to more performance-driven and customized computing methods.

For example, the absence of GPUs in our simulations has had a notable impact on the systems we modeled. From the 7 systems we selected, 3 include GPUs in their compute nodes, corresponding to the last 3 systems in the plot: Titan (2013), Summit (2018), and Frontier (2021). This supports the assumption that the efficiency gains of the last decade seem smaller due to the exclusion of GPUs. Especially, the nearly identical energy efficiency of Frontier (2021) and RoadRunner (2008) raises concerns. According to TOP500 metrics, Frontier's energy efficiency is 120 times greater than Roadrunner's based on Rpeak and power consumption metrics, while our plot shows only a 1.24-fold increase. This exclusion likely skews our results which leads to an under-representation of the true efficiency improvements in the last decade. If this assumption is true we can expect large-scale computer systems' energy efficiency to increase more steeply due to the wide adoption of GPUs for scientific computing, machine learning, high-performance computing, and cryptocurrency mining (4).

We plan to incorporate the modeling of accelerator-based components, such as GPUs, in future studies. This addition will allow for a more accurate picture of energy efficiency trends that reflect the impact of the change from a homogeneous architecture to a more heterogeneous architecture where CPUs and accelerators are combined. Moreover, while modeling the current state of the art is possible with adequate information on energy metrics, system architectures, and evaluation methods, the rapid pace of change in these industry practices makes it difficult to capture these trends and predict the future progres-

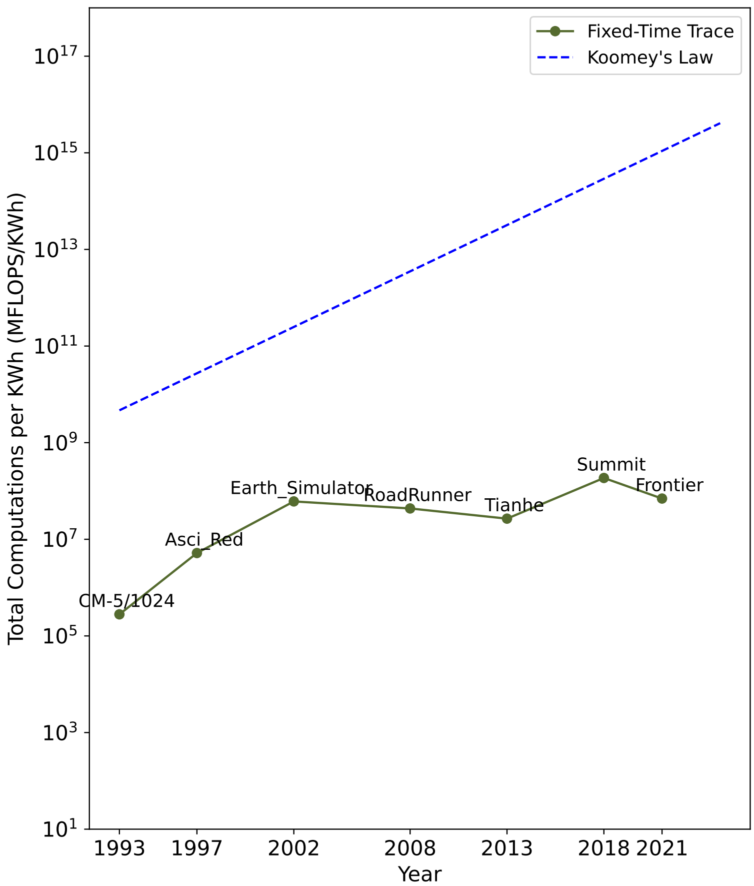

Figure 5.2: Evolution of energy consumption for the fixed-time trace.

sion of energy efficiency. By including these new components in the data center model, our simulations will better align with the evolving landscape of data center operations.

Fixed-Time Bitbrains Trace

Figure 5.2 illustrates the number of computations per kilowatt-hour (kWh) of electricity consumed for our chosen system topologies from 1993 to 2021 for the Fixed-Time Bitbrains Trace. The overall trend in efficiency progression is similar to that observed in the SURF trace with a steep increase in the efficiency pre-2000s and a deceleration post-2000s. However, the rate of increase in energy efficiency is slower with a doubling time of 3.21 years and the efficiency values for systems post-2000s are lower. This discrepancy can be attributed to the differing characteristics of the two traces. The Bitbrains trace comprises larger, longer-duration tasks that necessitate more hosts being active simultaneously, resulting in higher energy consumption relative to the total computations performed. In contrast,

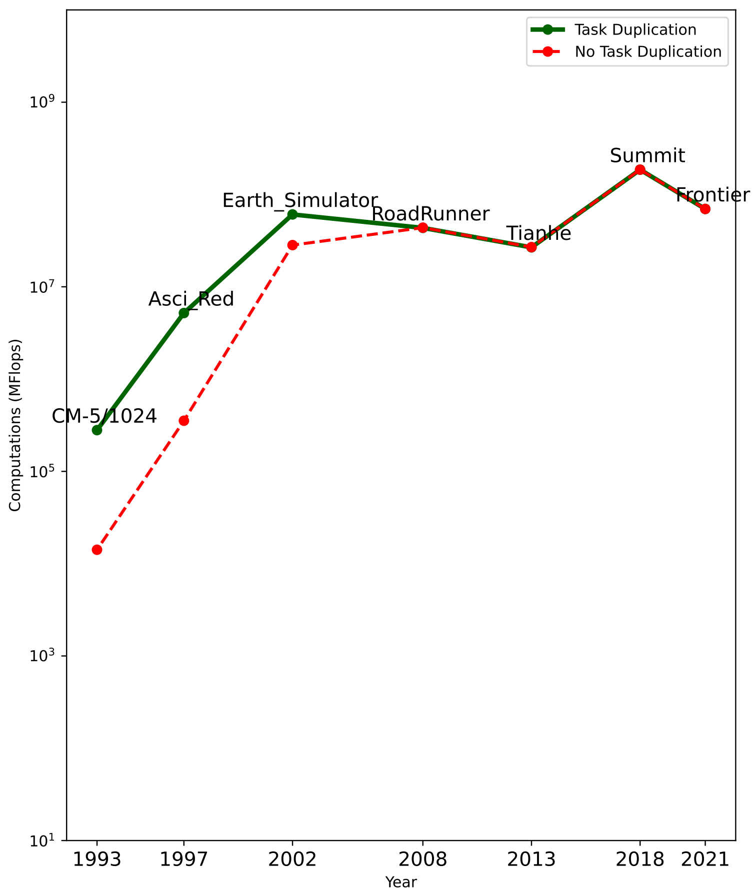

Figure 5.3: Impact of task duplication on energy efficiency for Bitbrains trace.

the SURF trace features smaller, high-frequency tasks with shorter durations, allowing for fewer hosts to be active simultaneously. For instance in Table 5.4 the host utilization is presented for each system for Bitbrains, Frontier (2021) requires 44 active hosts to handle the Bitbrains workload while requiring only 5 active hosts for the SURF workload in Table 5.3 where the host utilization is shown for SURF Lisa workload. This means, that despite SURF handling six times more computational work than Bitbrains, the nature and frequency of its tasks enable more efficient energy usage. This results in higher energy efficiency for SURF, whereas the continuous availability and substantial simultaneous host activity in Bitbrains lead to greater energy consumption per computation, thus lowering its overall energy efficiency.

5.3.1 Impact of Task Duplication on the Fixed-Time Trace

The systems pre-2000s show closer efficiency values and progression to SURF trace than post-2000s systems. This is due to how we employ task duplication. In Figure 5.3 the impact of task duplication on energy efficiency of pre-2000s systems can be observed. Task duplication is implemented only for systems that do not have sufficient resources to run the full workload. This strategy ensures better system utilization and makes the workload fully executable across all hosts. For the Bitbrains workload, this approach enhances efficiency for pre-2000s systems to use their hosts in a more balanced and efficient manner, thereby increasing overall efficiency. Task duplication allows these systems to achieve higher efficiency values for pre-2000s systems. For ASCI Red (1997) and CM-5/1024 (1993), where most hosts are unable to complete their tasks due to resource constraints, task duplication is applied more aggressively. This results in efficiency metrics that are more consistent with those of the SURF workload. On the other hand, the Earth Simulator, which has fewer tasks that are unable to complete, requires less task duplication. As a result, its efficiency is lower compared to the SURF workload.

5.3.2 Evaluation of the TOP500 Energy Efficiency Model

The Green500 list calculates energy efficiency using the same metric we use (MFLOPS per kWh), but in different units, GFLOPS per Watt. Additionally, instead of summing the power consumption of all hosts as our traces do, the total system power is estimated by measuring the power consumption of a single host and then multiplying it by the number of active hosts (N) (53). Deriving power consumption from one unit is necessary because Green500 benchmarks are conducted on actual computer systems, not simulations or mathematical models, and measuring the power consumption of every host would be costly and complicated, therefore, this method provides a practical and straightforward estimate of the total power consumption for the entire system.

Figure 5.4 shows the total computations per kWh (MFLOPS/kWh) of our chosen system topologies with metrics taken from the TOP500 list. Most of the systems modeled are not available on the Green500 list, therefore the metrics are taken from the TOP500 list, and the efficiency is calculated using the same method as Green500 (53) where the Rmax value of a system that represents the maximum performance achieved when the LINPACK benchmark is executed is divided by the power consumption value. For CM-5/1024 (1993) and ASCI Red (1997), power consumption information is not available on TOP500, therefore they are omitted from this plot.

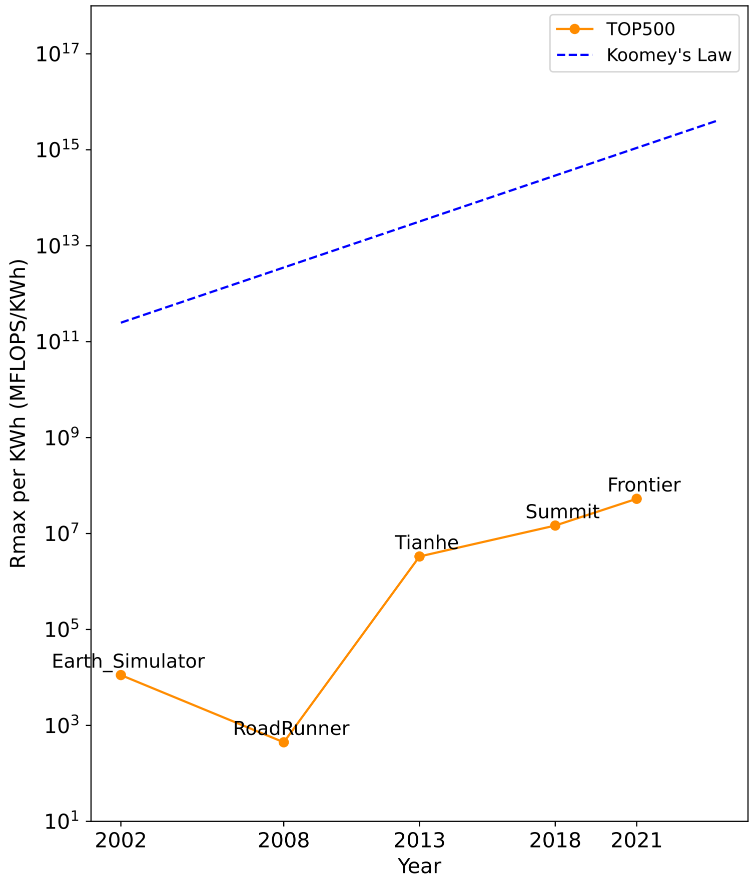

Figure 5.4: Evolution of energy consumption for the TOP500 list.

Comparing our results with the TOP500 metrics is a valuable method to validate our findings. Since TOP500 metrics include all components of a data center infrastructure, such as GPUs and cooling systems, we can assess whether the inclusion of these components supports our assumptions and results about energy efficiency trends. Consequently, we can draw supported conclusions based on alternative methods of evaluation. Utilizing three different evaluation methods—simulation, mathematical models, and real benchmarks—provides a comprehensive evaluation of energy efficiency trends.

A high-level overview of the plot reveals that the TOP500 metrics support the rate of increase in energy efficiency that Koomey suggests. The efficiency doubles every 1.25 years instead of every 1.5 years as Koomey suggests. This is not surprising; unlike the traces we use, the computational workload of the LINPACK benchmark is well-balanced across all hosts (53). Consequently, the progression of energy efficiency trends appears steeper. This indicates that when workloads are optimally distributed and systems are fully utilized, the advancements in energy efficiency are more pronounced.

As expected, the inclusion of GPUs in the compute nodes reveals that the more recent systems Tianhe (2013), Summit (2018), and Frontier demonstrate higher efficiency than older systems.

Going into more system-specific details highlights the seemingly low energy efficiency of RoadRunner (2008) compared to Earth Simulator (2002). This trend is also observed in the Fixed-Time Bitbrains workload in Figure 5.2. Although RoadRunner topped the TOP500 in 2013, it is not among the most energy-efficient systems. The TOP500 list focuses on performance rather than energy efficiency, unlike the Green500. Consequently, a system with excellent performance but high energy consumption, like RoadRunner, may achieve top rankings but face real-world limitations due to substantial operational costs. Therefore, while RoadRunner achieved the number one spot, its practicality is limited by its significant energy demands. This emphasizes that even top-ranked systems may not be useful or practical if they are not energy efficient, as the high operational costs make them unsuitable for sustained high-performance computational tasks.

Additionally, Frontier appears to be more efficient than Summit which validates our assumption that with a more computationally demanding trace, like the LINPACK benchmark, Frontier, and large-scale computer systems in general, will show their true performance and efficiency capabilities.

5.3.3 Comparative Analysis of all Energy Efficiency Models

Figure 5.5 shows the total computations per kWh (MFLOPS/kWh) for all the energy efficiency models we have considered so far: Koomey's Law, Fixed-Work SURF Lisa trace, Fixed-Time Bitbrains trace, and the TOP500 for all our chosen system topologies.

The adoption of GPUs has significantly improved energy efficiency, as evidenced by the increase in computations per kWh in TOP500 compared to Fixed-Time and Fixed-Work traces. Our traces show that without the inclusion of GPUs, we would see a noticeable decline in both energy efficiency and performance.

Another distinction is that the efficiency values of the TOP500 are lower than those for the Bitbrains and SURF Lisa workloads. This discrepancy can be due to several factors. Firstly, the lack of precise power specifications for each system means the power consumption output from OpenDC may be affecting the results. Secondly, although a balanced workload, the LINPACK benchmark focuses on performance maximization rather than energy optimization. The tasks in LINPACK are designed to push the system to its limits, often resulting in higher energy usage and consequently lower energy efficiency.

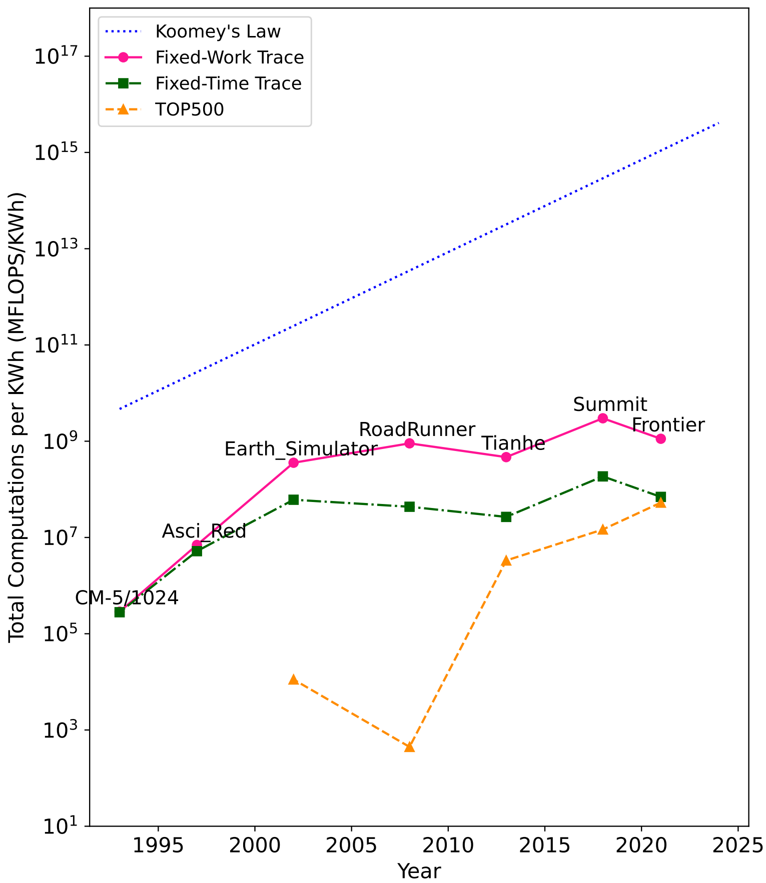

Figure 5.5: Evolution of energy efficiency of energy efficiency models.

When analyzing system behavior over time it is more meaningful to consider the general trends in efficiency rather than exact values. This allows us to identify overarching patterns and long-term improvements or declines in energy efficiency. Neither the TOP500 data nor our system modeling in OpenDC are perfectly accurate or realistic; they simply represent the best available methods for their respective purposes.

System	Year	Fixed-Work (%)	Fixed-Time (%)	Koomey's Law (%)	Top500 (%)
CM-5/1024	1993	-	-	-	-
Asci Red	1997	2329.46	1756.51	759.79	-
Earth Simulator	2002	5013.76	1070.37	3430.95	-
RoadRunner	2008	152.84	-28.51	3438.89	-96.03
Tianhe	2013	-48.25	-38.48	3445.9	746826.21
Summit	2018	541.41	594.88	3452.8	340.87
Frontier	2021	-62.49	-62.49	696.66	260.94

Table 5.2: Fluctuations in energy efficiency metrics over time. To read this table, compare for each system the values in its row with the previous system. Values in the rows represent the change over the system in the previous row. (System CM-5 does not have a prior in our study, which is indicated by the signs in its row).

Table 5.2 summarizes the fluctuations in energy efficiency over time in percentage for all of the energy efficiency models we used thus far: Fixed-Work, Fixed-Time, Koomey's Law, and TOP500. This table allows us to observe the stability and sensitivity of the four energy efficiency models we analyzed until this point. We calculate the fluctuations using Equation 5.1.

$$\Delta\% = \left(\frac{E_{\text{current}} - E_{\text{previous}}}{E_{\text{previous}}}\right) \times 100 \tag{5.1}$$

Where:

- E_{current} represents the energy efficiency of the current system.
- E_{previous} represents the energy efficiency of the previous system.

Large fluctuations in the efficiency metrics indicate that a metric is highly sensitive to the type of workload used and the system configurations. This leads to the results not being stable across different systems and decades.

Among the evaluation methods, Koomey's law stands out as the most stable, with changes in energy efficiency consistently positive and increasing over the years. This stability might suggest that Koomey's law is a reliable method for evaluating and analyzing energy efficiency, however, it does not reflect the real-world scenario where energy efficiency does not increase in a perfect linear fashion. As discussed in 1.1, real-life energy efficiency trends are influenced by various factors that cause non-linear progress and fluctuations. Therefore, the large fluctuations observed in the other methods are more valuable as they provide a more realistic glimpse into the complexities of energy efficiency over time.

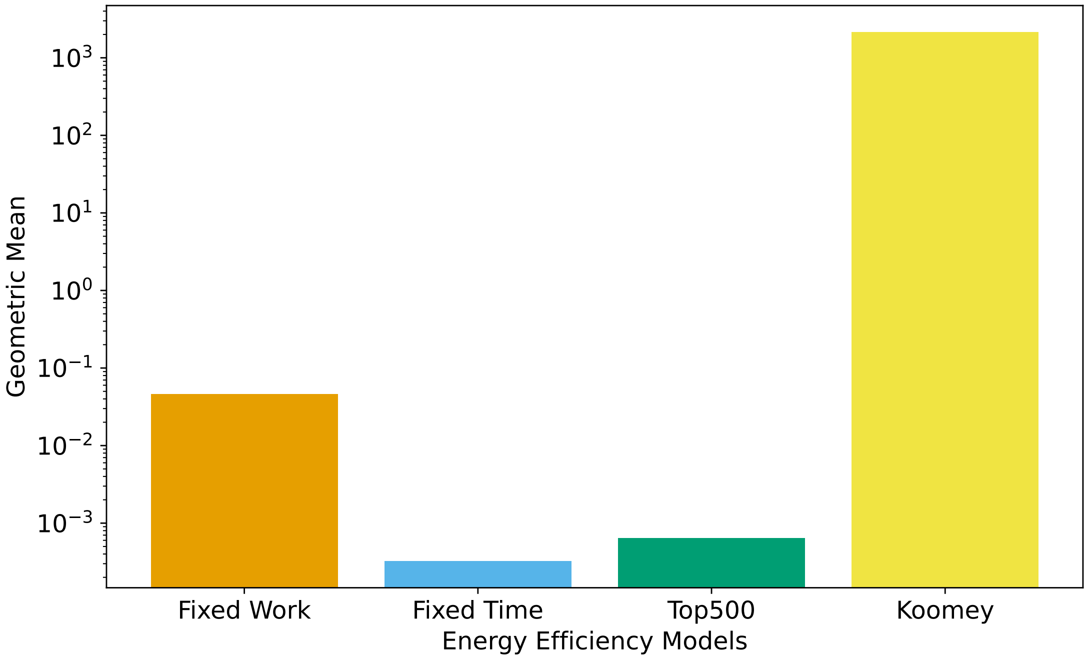

Figure 5.6: Geometric mean of the fluctuations per energy efficiency model over time.

It is important to acknowledge the trade-off between sensitivity and stability. The sensitivity observed in the other methods such as Frontier (2018) showing a 62.5% decrease or Tianhe showing a 48.2% decrease (both were discussed in the evaluation of Fixed-Work Trace), can offer deeper insights into specific variations and trends. However, this sensitivity can also highlight parameter-specific variations. Differentiating between the parameter-specific variations and the general energy efficiency trends is crucial when using the Fixed-Work and Fixed-Time methods to ensure that the insights gained are accurate and reflective of true performance.

Figure 5.6 showcases a bar chart of the geometric means of the percentage changes across four different energy efficiency metrics based on Table 5.2. The y-axis uses a logarithmic scale to accommodate the wide range of values. We calculate the percentage fluctuation of energy efficiency for each system compared to its predecessor using the following formula:

Fixed-Time and Top500 models show relatively small geometric mean values, which are around 10^{-3}. Fixed-Work model shows a higher geometric mean, in the range of 10^{-2} and 10^{-1}. Koomey's model shows the largest geometric mean, larger than 10^3. This is due to Koomey's model having large positive increases because of its linear curve, whereas the

System Name	Host Count	Active Host Count	Host Utilization (%)
Frontier	9,472	3	0.03
Summit	4,608	3	0.07
Tianhe	17,792	3	0.02
RoadRunner	11,340	3	0.03
Earth Simulator	640	5	0.78
Asci Red	4,536	44	0.97
CM-5/1024	1,024	595	58.1

Table 5.3: System utilization of Surf Lisa

TOP500, Fixed-Work, and Fixed-Time models do not follow a linear progression. This bar chart illustrates the unrealistic improvements in energy efficiency Koomey expected and how real-life benchmarks and simulations diverge from this prediction.

System Utilization

Tables 5.3 and 5.4 present a summary of the host information for all systems under the Fixed-Work and Fixed-Time traces. Only the active host count and host utilization columns differ for both of the tables. Table 5.3 shows lower overall host utilization, while Table 5.4 indicates higher host utilization. This discrepancy arises from the nature of the traces, as introduced in Section 3.3 and discussed in more detail in Section 5.3.

High system utilization percentages usually indicate a more efficient use of resources, as more of the available capacity is being actively used. For example, using a large system such as Frontier only to have four active hosts would be a significant waste because other systems can operate at a lower cost. Particularly for larger systems, it is inefficient to have a substantial portion of the system inactive.

In our study, we subjected the systems to the same workload to assess efficiency differences and capabilities across different decades, resulting in varying host utilization percentages for each system. Future studies could benefit from subjecting the systems to a consistent host utilization percentage, such as 60% or 70%, reflecting the average utilization of typical systems. This approach offers an alternative to using identical workloads and has the potential to reveal additional insights into energy efficiency trends by standardizing utilization conditions.

System Name	Host Count	Active Host Count	Host Utilization (%)
Frontier	9,472	44	0.46
Summit	4,608	44	0.95
Tianhe	17,792	45	0.25
RoadRunner	11,340	45	0.40
Earth Simulator	640	69	10.78
Asci Red	4,536	315	6.94
CM-5/1024	1,024	1024	100

Table 5.4: System utilization of Bitbrains

Relative impact of the Observed Energy Efficiency Models with each other

Assessing the relative importance of our traces, TOP500, and Koomey's law is essential to understanding which approaches are best suited for different cases.

Koomey's law provides a simple, clear number presenting the doubling of energy efficiency over a specific period, drawing connections between improvements in peak performance over time. Its simplicity is useful for a high-level trend analysis making way for more nuanced approaches. However, its applicability is limited due to not accounting for the complexities and non-linearities present in real-world systems.

The TOP500 list, from which we extract statistical patterns, is based on the LINPACK benchmark. While this provides valuable insights into the performance of the world's top supercomputers, it proves to be limited by its focus on a single benchmark and specific classes of computer systems.

In contrast, our proposed models are not limited to a single benchmark, a specific class of computer systems, or peak power ratings provided by manufacturers. By using OpenDC to simulate systems of our choice we offer a flexibility that the other two methods cannot match. This enables us to capture the nuances and complexities of various workloads and computer systems, even with just two classes of workloads and a limited taxonomy of computer systems. However, this flexibility comes with its constraints. While simulations are beneficial for addressing non-linearities and specific parameters, their accuracy depends on how well the parameters are set and the exhaustiveness of the simulator, as discussed in Section 2.2.2 and in Section 5.3.3.

5.4 Evaluation of Energy Efficiency from the 1950s to the 2020s

5.4.1 Fixed-Time Bitbrains

Figure 5.7 represents the evolution of energy consumption from the assembly of the full time frame of the 1950s to the 2020s, merging the results of our honours report (56) with our bachelor thesis.

The y-axis represents the total computations per kilowatt-hour (MFLOPS/KWh) on a logarithmic scale, and the x-axis represents the years from 1954 to 2021. The solid brown line with markers indicates the actual measured energy efficiency for various computer systems over the decades and the blue dashed line represents Koomey's Law.

The trends depicted in our findings suggest that the energy efficiency of computations doubles approximately every 3.83 years. This doubling time aligns with the observed patterns from the 1990s to the 2020s in Section 5.3. However, our comprehensive analysis over seven decades of computing provides a more extensive and detailed understanding of the long-term evolution in energy efficiency.

Initially, IBM 701 demonstrates an energy efficiency of approximately 10^3. Afterward, there is a period of stagnation starting from CDC 6600 until Cray-2. This suggests that advancements in computational power during this period did not significantly improve energy efficiency. This is due to energy efficiency not being a primary concern during those decades, showcasing the focus on increasing computational power instead.

Overall, this Figure presents a comprehensive picture of computing starting from the 1950s until today.

5.5 Threats to Validity

OpenDC Limitations. One key limitation arises from the use of simulations with OpenDC, despite the advantages of simulation analysis, data center simulators cannot yet perfectly replicate all aspects of real-world data center components as explained in Chapter 2. OpenDC is not able to simulate all components of a data center, such as cooling infrastructure and accelerator-based components like GPUs. This means, our results may not fully capture the energy consumption patterns of large-scale computer systems.

Workload Limitations. Additionally, the workloads utilized in this study may not adequately represent the true performance and efficiency capabilities of the systems, as they may not be computationally demanding or diverse enough. This limitation restricts the ability to fully capture the systems' potential under varied and intensive real-world scenarios.

Limitations of the Absence of Detailed CPU metrics. Another limitation is the reliance on the available data for system specifications due to the absence of detailed power consumption metrics of CPUs in available system specifications. Therefore, we assume a simplified power model that assumes a direct proportionality between resource usage and power consumption. This oversimplification may result in inaccuracies in estimating the true energy efficiency of the systems, as it does not account for the complexities and nuances of real-world power consumption.

5.6 Discussion

Our analysis revealed that the energy efficiency of large-scale computer systems has experienced varied progression over seven decades, diverging from Koomey's Law. While significant improvements were observed from the 1990s onwards, our findings indicate a deceleration in efficiency gains post-2000s. Main findings can be found in Figure 5.1 and detailed analysis of the results can be read in Sections 5.3 and 5.4.

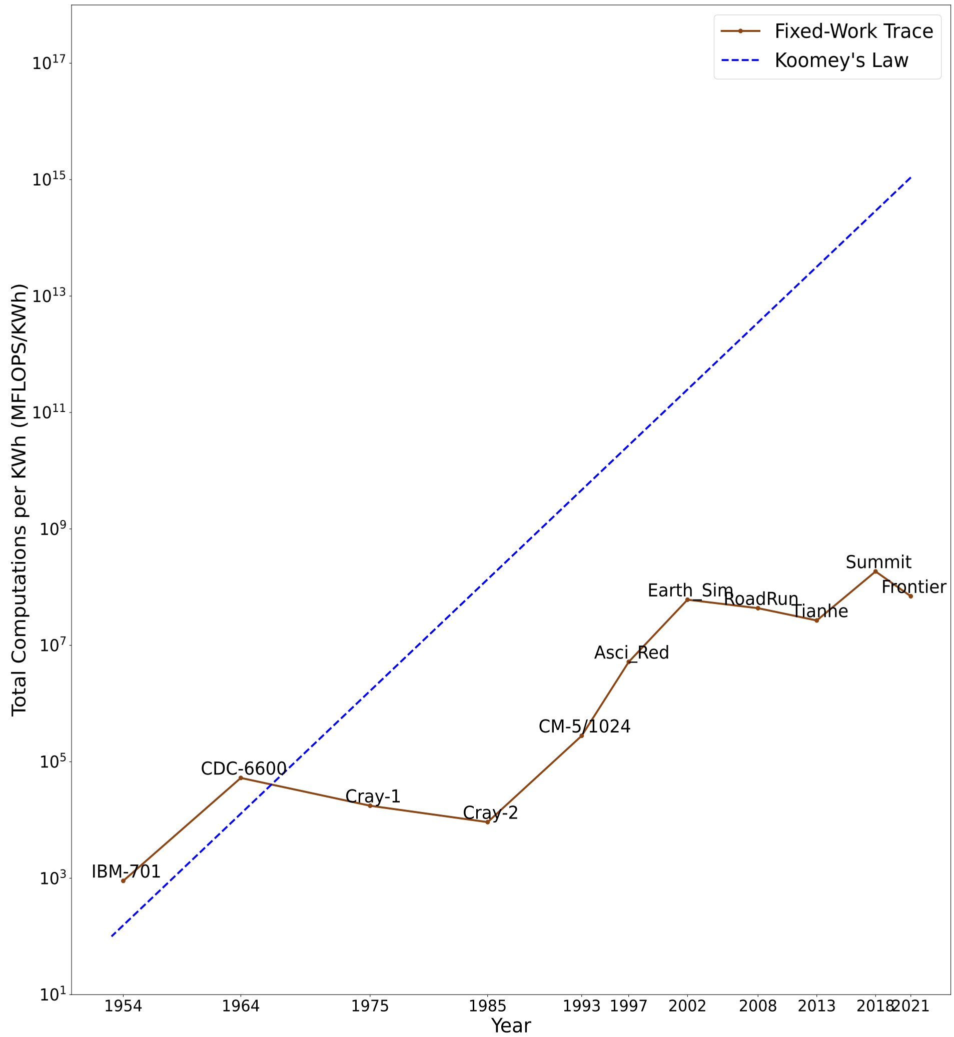

Figure 5.7: Evolution of energy efficiency in large-scale computer systems from the 1950s to the 2020s.

6

Conclusion and Future Work

In this chapter, we conclude our work in this thesis. We set out to develop an accurate model of energy consumption in large-scale computer systems to gain a comprehensive understanding of energy efficiency across different decades in computing. To achieve this, we detail a set of three research questions in Chapter 1. We now answer these research questions:

6.1 Answering Research Questions

RQ1 How to select relevant computer systems and workloads to build a taxonomy of exemplary infrastructures per decade from the 1990s to the 2020s?

To answer this question, we construct a taxonomy of exemplary computing infrastructures. We define two selection procedures on how to choose systems by decade using the TOP500 and Green500 lists. The results of this study are presented in Chapter 3. Afterward, we use this taxonomy to model the chosen systems using OpenDC.

RQ2 How to model the operation of data centers from different time periods to analyze the evolution of energy consumption?

To address **RQ2**, we create individual system topologies for each selected system in Section 4.2. We look at the technical specifications of each system to gain information on the CPU speed, Host Count, and Cluster Count. Afterward, we compare the CPU and the CPU capacities of each system.

We develop a workload scaling technique in Section 4.3 to adapt fixed-time and fixed-work trace types for diverse system configurations. We explore this method to ensure all chosen systems are able to execute our traces while maintaining workload integrity and improving system resource utilization in systems with insufficient resources. The details of our scaling method can be read in Section 4.3.2.

RQ3 How did energy consumption evolve from the 1950s to 2023 for the execution of realistic workloads in [contemporary] data centers?

For the final research question, **RQ3**, we evaluated the energy consumption of our chosen systems to gain insights into the trends of energy consumption over time. Our trace-based simulation analysis from the 1950s to the 2020s highlighted significant improvements in energy efficiency. However, it also revealed a deceleration in the rate of efficiency gains post-2000s. Our findings suggest an observed doubling time of energy efficiency of computations approximately 3.83 years, while Koomey's Law posits that the energy efficiency of computations doubles approximately every 1.5 years. Detailed results of our findings can be read in Chapter 5 in Sections 5.3 and 5.4.

6.2 Directions for Future Research

This work initiates an exploration into the energy efficiency of large-scale computer systems over several decades and highlights key trends. The potential for future research in this area is vast. Based on our analysis, we identify the most promising directions for future research:

1. Utilizing a more extensive taxonomy and various types of workloads that differ in computational demand, workload type, and distribution is valuable for future studies to conduct sensitivity analyses between different methods of workload types and energy efficiencies.

2. Leveraging the developed tool for comparing systems and scaling workloads to adapt to different systems, presents numerous opportunities to explore various facets of data centers comprehensively.

3. Incorporating the modeling of accelerator-based components, such as GPUs, to provide a more accurate representation of energy efficiency trends.

References

[1] Cisco Systems. **Cisco Global Cloud Index: Forecast and Methodology, 2016–2021**. Cisco Systems, 2016. 1

[2] U.S. Energy Information Administration. **Electricity consumption**. U.S. Energy Information Administration, n.d. Accessed on 2024-04-01. 1, 2, 3

[3] Statistics Netherlands. **Electricity supplied to data centers, 2017-2019**, 2021. Accessed: 2024-07-04. 1

[4] IEA. **Electricity 2024, Paris**. IEA, 2024. Accessed on 2024-07-04. 1, 46

[5] Eric Masanet, Arman Shehabi, Nuoa Lei, Sarah Smith, and Jonathan Koomey. **Recalibrating global data center energy-use estimates**. *Science*, **367**(6481):984–986, 2020. 1, 2, 3

[6] Lotfi Belkhir and Ahmed Elmeligi. **Assessing ICT global emissions footprint: Trends to 2040 recommendations**. *Journal of Cleaner Production*, **177**:448–463, 2018. 2

[7] Anders Andrae and Tomas Edler. **On Global Electricity Usage of Communication Technology: Trends to 2030**. *Challenges*, **6**:117–157, 04 2015. 2

[8] Xiaobo Fan, Wolf-Dietrich Weber, and Luiz Andre Barroso. **Power provisioning for a warehouse-sized computer**. In *Proceedings of the 34th Annual International Symposium on Computer Architecture*, ISCA '07, page 13–23, New York, NY, USA, 2007. Association for Computing Machinery. 2

[9] Jonathan Koomey, Stephen Berard, Marla Sanchez, and Henry Wong. **Implications of Historical Trends in the Electrical Efficiency of Computing**. *IEEE Annals of the History of Computing*, **33**(3):46–54, 2011. 2, 3, 6, 12, 43, 45

[10] Alexandru Iosup, Fernando Kuipers, Ana Lucia Varbanescu, Paola Grosso, Animesh Trivedi, Jan Rellermeyer, Lin Wang, Alexandru Uta, and Francesco Regazzoni. **Future Computer Systems and Networking Research in the Netherlands: A Manifesto**, 2022. 5

[11] Top500.org. **TOP500**. `https://www.top500.org`. Online; accessed 2024-04-01. 5, 7, 15, 42

[12] Robert C. Nickerson, Jan Muntermann, and Upkar Varshney. **Taxonomy Development in Information Systems: A Literature Survey and Problem Statement**. In *AMCIS 2010 Proceedings*, number 125, 2010. 7

[13] Alexandru Iosup, Georgios Andreadis, Vincent van Beek, Matthijs Bijman, Erwin van Eyk, Mihai Neacsu, Leon Overweel, Sacheendra Talluri, Laurens Versluis, and Maaike Visser. **The OpenDC Vision: Towards Collaborative Datacenter Simulation and Exploration for Everybody**. In *16th International Symposium on Parallel and Distributed Computing, ISPDC 2017, Innsbruck, Austria, July 4-6, 2017*, pages 1–10, 2017. 7, 14

[14] Alexandru Iosup, Laurens Versluis, Animesh Trivedi, Erwin van Eyk, Lucian Toader, Vincent van Beek, Giulia Frascaria, Ahmed Musaafir, and Sacheendra Talluri. **The AtLarge Vision on the Design of Distributed Systems and Ecosystems**. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 1765–1776, 2019. 8

[15] Alessandro Vittorio Papadopoulos, Laurens Versluis, André Bauer, Nikolas Herbst, Jóakim von Kistowski, Ahmed Ali-Eldin, Cristina L. Abad, José Nelson Amaral, Petr Tůma, and Alexandru Iosup. **Methodological Principles for Reproducible Performance Evaluation in Cloud Computing**. *IEEE Transactions on Software Engineering*, **47**(8):1528–1543, 2021. 8

[16] J. Mitchell-Jackson, J.G. Koomey, B. Nordman, and M. Blazek. **Data center power requirements: measurements from Silicon Valley**. *Energy and Resources Group, University of California, Berkeley, Lawrence Berkeley National Laboratory, AT&T*, 2001. 10

[17] Xiaobo Fan, Wolf-Dietrich Weber, and Luiz Andre Barroso. **Power provisioning for a warehouse-sized computer**. *SIGARCH Comput. Archit. News*, **35**(2):13–23, jun 2007. 11

[18] Mark Weiser, Brent Welch, Alan Demers, and Scott Shenker. **Scheduling for Reduced CPU Energy**. In *First Symposium on Operating Systems Design and Implementation (OSDI 94)*, Monterey, CA, November 1994. USENIX Association. 11

[19] Alexandru Uta, Alexandru Custura, Dmitry Duplyakin, Ivo Jimenez, Jan Rellermeyer, Carlos Maltzahn, Robert Ricci, and Alexandru Iosup. **Is Big Data Performance Reproducible in Modern Cloud Networks?** In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*, pages 513–527, Santa Clara, CA, February 2020. USENIX Association. 12

[20] Jerry Banks. **Introduction to simulation**. In *Proceedings of the 31st Conference on Winter Simulation: Simulation—a Bridge to the Future - Volume 1*, WSC '99, page 7–13, New York, NY, USA, 1999. Association for Computing Machinery. 12

[21] Ilyas Bambrik. **A Survey on Cloud Computing Simulation and Modeling**. *SN Computer Science*, **1**(5):249, 2020. 12, 13

[22] Rodrigo Calheiros, R. Ranjan, Anton Beloglazov, Cesar De Rose, and Rajkumar Buyya. **CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms**. *Software Practice and Experience*, **41**:23–50, 01 2011. 13

[23] Dzmitry Kliazovich, Pascal Bouvry, Yury Audzevich, and Samee Khan. **GreenCloud: A Packet-Level Simulator of Energy-Aware Cloud Computing Data Centers**. **62**, pages 1–5, 12 2010. 13

[24] **Data Center 2030**, 2023. Accessed: 2024-07-07. 13, 46

[25] Gaurav Chawla. **Accelerators Will Play a Key Role in 5G and Edge Computing**. *Dell Technologies Blog*, March 16 2020. 13

[26] Unai Lopez-Novoa, Alexander Mendiburu, and Jose Miguel-Alonso. **A Survey of Performance Modeling and Simulation Techniques for Accelerator-Based Computing**. *IEEE Transactions on Parallel and Distributed Systems*, **26**(1):272–281, 2015. 13

[27] GPGPU-Sim Development Team. **GPGPU-Sim Manual**, n.d. Accessed: 2024-07-07. 13

[28] Sylvain Collange, Marc Daumas, David Defour, and David Parello. **Barra: a parallel functional simulator for GPGPU**. pages 351–360, 08 2010. 13

[29] Saurabh Kumar Garg and Rajkumar Buyya. **NetworkCloudSim: Modelling Parallel Applications in Cloud Simulations**. In *2011 Fourth IEEE International Conference on Utility and Cloud Computing*, pages 105–113, 2011. 13

[30] Vitaly Zakharenko, Tor Aamodt, and Andreas Moshovos. **Characterizing the Performance Benefits of Fused CPU/GPU Systems Using FusionSim**. pages 685–688, 01 2013. 13

[31] Rafael Ubal, Byunghyun Jang, Perhaad Mistry, Dana Schaa, and David Kaeli. **Multi2Sim: a simulation framework for CPU-GPU computing**. In *Proceedings of the 21st International Conference on Parallel Architectures and Compilation Techniques*, PACT '12, page 335–344, New York, NY, USA, 2012. Association for Computing Machinery. 13

[32] Unai Lopez-Novoa, Alexander Mendiburu, and Jose Miguel-Alonso. **A Survey of Performance Modeling and Simulation Techniques for Accelerator-Based Computing**. *IEEE Transactions on Parallel and Distributed Systems*, **26**(1):272–281, 2015. 13

[33] Jerry Banks, David Nicol, John Carson, and Barry Nelson. *Discrete-Event System Simulation* . Pearson Deutschland, 2013. 14

[34] Siqi Shen, Vincent Van Beek, and Alexandru Iosup. **Statistical Characterization of Business-Critical Workloads Hosted in Cloud Datacenters**. In *2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pages 465–474, 2015. 20

[35] Lieven Eeckhout. **FOCAL: A First-Order Carbon Model to Assess Processor Sustainability**. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, ASPLOS '24, page 401–415, New York, NY, USA, 2024. Association for Computing Machinery. 20, 25

[36] Oak Ridge Leadership Computing Facility. *Frontier User Guide*. Oak Ridge National Laboratory, 2024. `https://docs.olcf.ornl.gov/systems/frontier_user_guide.html`. 23

[37] AMD. **AMD EPYC 7713**. `https://www.amd.com/en/products/processors/server/epyc/7003-series/amd-epyc-7713.html`, 2024. Accessed: 2024-07-09. 23

[38] Oak Ridge Leadership Computing Facility. *Summit User Guide*. Oak Ridge National Laboratory, 2024. `https://docs.olcf.ornl.gov/systems/summit_user_guide.html`. 23

[39] NVIDIA. **Accelerated Computing and the Democratization of Supercomputing**. Technical report, NVIDIA Corporation, 2018. Technical Overview. 23, 43

[40] Jack Dongarra, Michael Heroux, and Piotr Luszczek. **High-Performance Conjugate-Gradient Benchmark: A New Metric for Ranking High-Performance Computing Systems**. Technical Report icl-utk-970-2017, Innovative Computing Laboratory, University of Tennessee, 2017. Accessed: 2024-07-09. 23

[41] TechPowerUp. **Intel Xeon E5-2692 v2**. `https://www.techpowerup.com/cpu-specs/xeon-e5-2692-v2.c1669`, 2024. Accessed: 2024-07-09. 23

[42] Marc Snir, David A. Bader, David J. Kerbyson, Arndt Bode, and Larry K. Johnsson. **A Blueprint for Introducing Disruptive Technology into HPC Systems**. *Journal of Computing Research*, **8**(4):44–54, 2008. Accessed: 2024-07-09. 23

[43] Michael Gschwind, H. Peter Hofstee, Brian Flachs, Martin Hopkins, Yukio Watanabe, and Takeshi Yamazaki. **Synergistic Processing in Cell's Multicore Architecture**. *IEEE Micro*, **26**(2):10–24, 2006. Accessed: 2024-07-09. 23

[44] Nico Lokker. **NEC Earth Simulator System**. `http://www.lokker.net/THOCP/hardware/nec_ess.htm`, 2024. Accessed: 2024-07-09. 23

[45] Timothy G. Mattson and Greg Henry. **An Overview of the Intel TFLOPS Supercomputer**. Technical report, Intel Corp., 1998. `http://www.lokker.net/THOCP/hardware/nec_ess.htm`. 23

[46] Intel. **Intel Pentium Pro Processor 200 MHz, 256K Cache, 66 MHz FSB**. `https://www.intel.com/content/www/us/en/products/sku/49952/intel-pentium-pro-processor-200-mhz-256k-cache-66-mhz-fsb/specifications.html`, 2024. Accessed: 2024-07-09. 23

[47] TOP500. **Supercomputer: Frontier**. `https://top500.org/system/166997/`, 2024. Accessed: 2024-07-09. 23

[48] G. Blanck and S. Krueger. **The SuperSPARC microprocessor**. In *Digest of Papers COMPCON Spring 1992*, pages 136–141, 1992. 23

[49] Carsten Ernemann, Baiyi Song, and Ramin Yahyapour. **Scaling of Workload Traces**. In Dror Feitelson, Larry Rudolph, and Uwe Schwiegelshohn, editors, *Job Scheduling Strategies for Parallel Processing*, pages 166–182, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. 25, 26

[50] Johann Schleier-Smith, Vikram Sreekanti, Anurag Khandelwal, Joao Carreira, Neeraja J. Yadwadkar, Raluca Ada Popa, Joseph E. Gonzalez, Ion Stoica, and David A. Patterson. **What serverless computing is and should become: the next phase of cloud computing**. *Commun. ACM*, **64**(5):76–84, apr 2021. 26

[51] Alexandru Iosup, Dick H. J. Epema, Carsten Franke, Alexander Papaspyrou, Lars Schley, Baiyi Song, and Ramin Yahyapour. **On Grid Performance Evaluation Using Synthetic Workloads**. In Eitan Frachtenberg and Uwe Schwiegelshohn, editors, *Job Scheduling Strategies for Parallel Processing, 12th International Workshop, JSSPP 2006, Saint-Malo, France, June 26, 2006, Revised Selected Papers*, **4376** of *Lecture Notes in Computer Science*, pages 232–255. Springer, 2006. 30, 36

[52] Miyuru Dayarathna, Yonggang Wen, and Rui Fan. **Data Center Energy Consumption Modeling: A Survey**. *IEEE Communications Surveys Tutorials*, **18**(1):732–794, 2016. 42, 43

[53] R. Ge, X. Feng, H. Pyla, K. Cameron, and W. Feng. **Power Measurement Tutorial for the Green500 List**. Technical report, June 27 2007. 43, 49, 50

[54] Sparsh Mittal and Jeffrey S. Vetter. **A Survey of Methods for Analyzing and Improving GPU Energy Efficiency**. *ACM Comput. Surv.*, **47**(2), aug 2014. 43

[55] Top500.org. **TOP500**. `https://www.top500.org/lists/top500/2024/06/`. Online; accessed 2024-07-01. 44

[56] Dide Poyraz. **A Historical Analysis of Energy Consumption in Large-Scale Computer Systems (1950s to 1990s)**, July 22 2024. Honours Research Report, Vrije Universiteit Amsterdam, Bachelor of Science degree in Computer Science. 57

Appendix A

Reproducibility

A.1 Artifact check-list (meta-information)

- **Program: OpenDc** (`https://github.com/atlarge-research/opendc`)
- **Compilation: CPython 3.11.5, Kotlin 1.9.22**
- **Run-time environment: Local machine**
- **Hardware: Apple M1, 16 GB of RAM, 494 GB of NVME storage**
- **Experiments: Available at** `https://github.com/didepoyraz/experiments-bsc.git`
- **Publicly available: Source code publicly available**
- **Code licenses: OpenDC is MIT Licensed and Copyrighted (c) by AtLarge Research**

A.2 Description

A.2.1 How to access

- Experiments: `https://github.com/didepoyraz/experiments-bsc.git`
- OpenDC: `https://github.com/atlarge-research/opendc`

A.3 Installation

Clone the repository containing the experiments:

```
> git clone https://github.com/didepoyraz/experiments-bsc.git
```

A.4 Evaluation and expected results

If run successfully, the modeling results and energy consumption CSV files will be populated with the outputs according to which the trace is executed. Furthermore, the "plot_results.ipynb" notebook will contain the relevant plots of each experiment.

A.5 Methodology

Submission, reviewing, and badging methodology:

- `https://www.acm.org/publications/policies/artifact-review-badging`
- `http://cTuning.org/ae/submission-20201122.html`
- `http://cTuning.org/ae/reviewing-20201122.html`