UNIVERSITY OF MISKOLC

FACULTY OF MECHANICAL ENGINEERING AND INFORMATICS

Enhancing Cloud Simulation Accuracy for Energy-Driven Resource Strategies

PhD dissertation

Author
Hasanein Dakheelallah Habeeb Rjeib

MSc in Computer Engineering

"JÓZSEF HATVANY" DOCTORAL SCHOOL OF INFORMATION SCIENCE, ENGINEERING AND TECHNOLOGY

Head of Doctoral School:
Prof. Dr. Jenő SZIGETI

Supervisor:
Prof. Dr. Gábor Kecskeméti

Miskolc, 2025

Declaration

The author hereby declares that this thesis has not been submitted, either in the same or in a different form, to this or to any other university for obtaining a PhD degree. The author confirms that the submitted work is his own and the appropriate credit has been given where reference has been addressed to the work of others.

Miskolc, 2025

signature of the candidate

Acknowledgements

Writing this dissertation has been a journey filled with guidance, encouragement, and support from many wonderful people. I extend my deepest gratitude to my supervisor, Prof. Dr. Gábor Kecskeméti. His extensive knowledge, attention to detail, and unwavering support have been crucial to my research. Prof. Dr. Gábor's approachability, valuable feedback, and dedication to excellence have motivated me to face challenges with determination and thoroughness.

I hold a special place in my heart for my parents, wife, and lovely children. Their everlasting love, constant support, and prayers have been my strength and motivation throughout this journey. The lessons and values they have instilled in me, along with their faith in my abilities, have been a guiding light, pushing me to strive for excellence.

Hasanein Rjeib

Miskolc, Hungary, 2025

List of abbreviations

VM . . . . . Virtual Machine

PM . . . . . Physical Machine

QoS . . . . . Quality of Service

SLA . . . . . Service Level Agreement

VMC . . . . . Virtual Machine Consolidation

VMM . . . . . Virtual Machine Monitor

VMP . . . . . Virtual Machine Placement

IaaS . . . . . Infrastructure as a Service

PaaS . . . . . Platform as a Service

SaaS . . . . . Software as a Service

OS . . . . . Operating System

CPU . . . . . Central Processing Unit

DVFS . . . . . Dynamic Voltage Frequency Scaling

CSV . . . . . Comma-Separated Values

Contents

List of Figures

List of Tables

1

Introduction

Cloud computing has become a leading model in modern IT infrastructure, offering scalable and on-demand resource provisioning for diverse applications across various domains. Cloud data centers, with their high energy demands, are significant contributor to global energy consumption. As data centers continue to expand, maintaining efficient resource management, energy-aware scheduling, and cost-efficient solutions become more important [2].

Virtualization is central to this issue, abstracting hardware resources into multiple virtual machines (VMs). This leads to better resource sharing and enhances the flexibility of cloud infrastructures. However, improper VM placement and resource allocation can result in reduced utilization and excessive power demands [3]. As cloud providers strive to optimize costs and Quality of Service (QoS) through Service Level Agreements (SLAs), such inefficiencies drive up operational costs and carbon emissions, emphasizing the need for energy-aware resource management strategies [4].

VM consolidation offers a viable solution for cutting energy waste and maximizing cloud resource usage. It optimizes VM placement by continuously reallocating them to a smaller set of physical machines (PMs) based on real-time workload demands [5]. Aggressive VM consolidation can degrade performance and escalate SLA violations caused by bottlenecks and resource contention, underminig some of the energy-saving benefits. Another difficulty is forecasting workloads, as fluctuating demand makes optimal VM placement decisions more challenging [6].

Given the complexity and expense associated with testing new resource management techniques in live cloud environments, researchers rely on cloud simulators to test and verify their algorithms. Simulators offer a controlled, repeatable, and

cost-efficient way to model cloud infrastructures, test scheduling strategies, simulate different workloads, and analyze different energy models without requiring expensive physical infrastructure [7]. Although these simulators are powerful for cloud research, their realism is often limited. For example, modeled energy usage and resource allocation may not fully align with real-world scenarios. Such discrepancies may lead to inaccurate research results, where algorithms that appear effective in simulation fail to deliver expected improvements in real-cloud infrastructure.

For cloud simulators to be valuable, they must accurately replicate the dynamics of real-cloud invironments. This requires detailed modeling of workload behaviors, resource utilization trends, and energy consumption patterns offering realistic power usage predictions that reflect actual hardware performance [8]. However, various cloud simulators adopt distinct assumptions and simplifications, leading to variations in modeling energy consumption and resource usage simulations [9]. Without a standardized evaluation framework to assess realism, researchers could face difficulty in determining which simulator yields the most trustworthy results for their purposes.

The challenges presented above demand a systematic plan and a set of procedures to effectively address them. Therefore, a detailed outline of the research aims is needed to pinpoint the steps to overcome these problems.

1.1 Aims of The Research

1. To devise a framework for evaluating realism of cloud simulators, based on how accurately they model real-world energy consumption and resource utilization. Within this goal, we aim to support researchers to:

   (a) Evaluate the realism of various cloud simulators in terms of their real-world reflection, helping to select or enhance simulators for more accurate representations of real-world environments.

   (b) Outline disparities between simulated outputs and actual measurements, aiding in refining simulations for greater real-world accuracy.

   (c) Enhance the credibility of cloud simulation by aligning results with reliable expectations.

2. To develop a new virtual machine placement algorithm that:

(a) Improves energy efficiency by dynamically consolidating VMs to a reduced set of PMs.

(b) Reduces resource wastage by efficiently allocating CPU and memory to meet real workload demands.

(c) Meets SLA requirements, assuring high performance and quality of service in spite of reduced PMs set.

1.2 Dissertation Guide

This dissertation is organized as follows:

Chapter 2 reviews fundamental concepts in cloud computing and explores key technologies such as virtualization and VM consolidation strategies. Then, it discusses the importance of simulation tools in cloud research, highlighting their features and functionalities. Moreover, it examines energy models and workload characteristics that influence cloud simulation realism. Additionally, this chapter explores previous studies on cloud simulation realism and virtual machine consolidation algorithms to enhance resource management and energy efficiency, outlining the deficiencies that this dissertation aims to address.

Chapter 3 introduces our standardized framework for evaluating the realism in cloud simulation. The chapter defines a unified methodology to compare simulation outputs with real-world data under identical conditions. In addition, it includes a standardized workload profile with five utilization levels, and highlights the data collection process required to generate simulation outputs. Then, it describes the experimental setup, and explores discrepancies in simulation results.

Chapter 4 introduces VMP-ER, a VM placement algorithm to enhance resource utilization while optimizing energy consumption and SLA violations. It presents our algorithm design, experimental evaluation using real-world workload traces, and comparative analysis with existing strategies.

Chapter 5 highlights the significance of our proposed realism scoring framework and VM placement algorithm. It provides a summary of the key contributions in this dissertation and outlines potential future research directions.

2

Background and Literature

2.1 Introduction

Cloud computing has significantly changed how computational resources can be accessed and utilized, offering scalable resources on demand, flexibility, and cost-effectiveness [10]. It has emerged as a fundamental element of today's digital transformation, fostering innovation in fields such as healthcare, finance, and entertainment [11]. As companies continue to transition to cloud-based solutions, the challenges of resource management, performance optimization, and minimizing environmental effects becomes increasingly complicated [12].

The dynamic nature of the cloud has inspired investigations into robust cloud management solutions, realistic modeling environments, and performance optimization strategies [13]. To address these challenges, researchers and professionals leverage the concept of virtualization and the virtual machine consolidation (VMC), trying to fill up VMs on as a few physical hosts as possible to reduce energy consumption in cloud computing, leading to better resource utilization of the cloud datacenter [14].

They often utilize cloud simulation environments, energy-saving algorithms, and workload analytics techniques [15, 16]. Simulators offer a cost-effective and safe platform to evaluate algorithms and strategies, while optimization techniques for energy and resources focus on minimizing operational costs and environmental effects [17].

This chapter gives an overview of cloud computing terminology and its popular models and techniques that are necessary to achieve the aims of our research. In section 2.2, we start with an overview of cloud computing models and

simulation frameworks. Then, we explore cloud simulators techniques to model energy and resources in cloud environments. This section also presents Virtual Machine Consolidation (VMC) strategies and highlights the most widely used techniques and methods to enhance energy efficiency. Finally, a thorough analysis of related studies reveals the strengths and limitations of current research is presented in section 2.3.

2.2 Background

2.2.1 Cloud Computing

Cloud computing represents a framework that delivers flexible and scalable access to computing resources [18]. These resources, including computing power, data storage, and networking, are offered on a pay-per-use basis. This enables organizations to bypass the initial costs and challenges of owning and managing physical infrastructure [19]. According to the National Institute of Standards and Technology (NIST)[i], three widely recognized service models are identified:

1. **Infrastructure as a Service (IaaS):** Provides fundamental computation, storage, and networking services as resources offered on an on-demand basis by a service provider [20]. Users manage their own operating systems and applications, approaching the infrastructure as if it were virtualized hardware [21]. Prominent IaaS providers include Amazon EC2, Google Compute Engine, and Microsoft Azure [22].

2. **Platform as a Service (PaaS):** Features a higher-level development environment or platform that simplifies the management of the underlying infrastructure [23]. Developers can concentrate on deploying applications instead of configuring individual servers [24]. Azure App Service and Google App Engine are good examples.

3. **Software as a Service (SaaS):** Offers standard software applications through a subscription or pay-per-use model, commonly available via a web browser, without the need to install or manage software on local devices [25].

[i] https://www.nist.gov/

As cloud infrastructures continue to expand and grow, effective resource management and accurate simulation modeling are becoming key challenges. Virtual machine placement (VMP) is essential for distributing workloads across physical machines while ensuring energy-efficient cloud operations [26]. In addition, the accuracy of cloud simulators in replicating real-world behaviors is important, as they meant to represent energy consumption and resource utilization of real cloud environments.

2.2.2 Virtual Machine Consolidation (VMC)

Virtualization is a key technology in cloud computing, which enables hardware utilization efficiency and multiple VMs to be hosted in a single PM by leveraging a virtual machine monitor (VMM), or a hypervisor [27–29]. Virtualization can be done on hardware level by running several operating systems (OS) on a physical host, or on OS level allowing applications to be executed within segregated environments while sharing the same kernel [30]. Virtualization can also occur at the container level by using technologies like Docker and Kubernetes [31]. This provides efficient and isolated environments without the overhead of full VMs [32].

Virtualization minimizes reliance on physical hardware and enhances resource efficiency, making it a key enabler of cloud computing [33]. Periodically, incoming requests are evaluated and transformed into virtual machines, thereafter receiving allocation of cloud infrastructure resources. Within the cloud environment, a diverse pool of PM resources with varying capacities exists. An effective VM management is essential to minimize resource wastage and reduce energy inefficiencies, leading to the concept of VM consolidation (VMC) [14, 34].

VMC is a key strategy in cloud computing, enhancing energy efficiency and resource utilization [35]. It focuses on decreasing the number of operational physical machines in a data center by dyanamically reallocating VMs from underutilized PMs to others. Thus, energy consumption and operational expenses can be significantly reduced by putting idle PMs into low power states [36, 37].

The VM consolidation process typically includes three primary steps: (a) Host Detection, finding PMs that are either underloaded or overloaded, (b) VM Selection, choosing VMs to be migrated according to requirements like improving resource usages or energy effect, and (c) VM Placement, finding a suitable target to host VMs, ensuring minimal waste of resources while optimizing

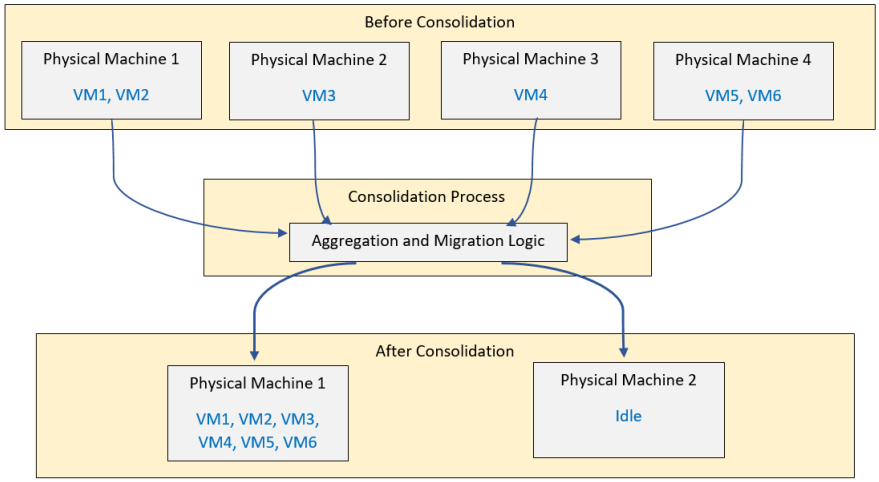

Figure 2.1: Virtual machine consolidation in a cloud data center

energy [38, 39]. Figure 2.1 shows virtual machine consolidation process, where several VMs are consolidated into fewer PMs to increase performance. While in-active PMs are switched off to save energy.

VMC Objectives: VMC strategies are essential for optimizing cloud data centers' performance and efficiency. These strategies focus on achieving energy efficiency, resource utilization, while adhering to Service Level Agreements (SLAs) through different methods, described below:

1. Energy-optimized: This approach focuses on reducing energy consumption by grouping VMs on a fewer servers while shutting down inactive ones. Server power consumption is estimated using energy models, facilitating informed consolidation decisions [36].

2. Resource-optimized: This strategy aims to enhance resource utilization by effectively packing VMs onto available servers. to prevent excessive load on consolidated servers, factors like CPU load, network bandwidth, and memory utilization are taken into account [40].

3. SLA-aware consolidation: This approach ensures that VM consolidation adheres to Service Level Agreement (SLA) requirements. VM performance is observed during and after consolidation to satisfy agreed performance standards [41].

For efficient cloud management, designing a VM consolidation algorithm that effectively balances these objectives is crucial for efficient cloud management.

An optimal strategy should minimize power consumption and enhance resource efficiency, while upholding to SLA-defined performance standards.

Common VMC Approaches: Several approaches and algorithms have been designed to enhance the efficiency of VM consolidation:

1. Heuristic Approaches: Rely on predefined rules to guide the selection and placement of VMs. While computationally efficient, they don't guarantee best possible results [42]. Common approaches include First Fit (FF) which allocates a VM to the first suitable server, Best Fit which assigns a VM to the server with the least remaining capacity, and Next Fit which assigns a VM to the next server in order, enhancing server packing efficiency [43].

2. Metaheuristic Approaches: Offer more complex solutions for VM consolidation. They utilize iterative exploration of the solution space to identify near-optimal VM placement setup [44]. While robust, Metaheuristic demands substantial computational resources and require careful adjustment of parameters. Genetic algorithms (GA), particle swarm optimization (PSO), and ant colony optimization (ACO) are common approaches [45].

3. Workload prediction Approaches: These methods can identify trends in resource usage and estimate future demands, empowering predictive resource consolidation [8]. Although promising, these methods require extensive historical data for training to perform effectively in heterogeneous workload scenarios and they are prone to biases in the data training process. Common approaches are machine learning (ML) and neural networks(NN) [6, 46].

4. Bin-packing Paradigm: Influenced by the principles of the bin-packing problem, Some traditional VMC methods model PMs as bins with constraints like CPU limit and memory capacity, and VMs as items requiring specific resources [34]. Algorithms such as First-Fit Decreasing (FFD) and Best-Fit Decreasing (BFD) are recognized for their simplicity and computational efficiency [42].

   These algorithms serve as a baseline for designing sophisticated or specialized strategies [47]. Power-Aware Best Fit Decreasing(PABFD) extending BFD heuristic by adopting power consumption metrics, and Modified Best-Fit Decreasing(MBFD) prioritizing energy efficiency and SLA adherence, are good examples [48]. By prioritizing VMs with the largest resource demands first, these approaches optimize resource distribution

and minimize the number of active servers. Thus, improving overall energy efficiency [49].

Virtual machine placement (VMP) is a critical factor in optimizing resource utilization and managing energy consumption within the cloud framework [49]. Nonetheless, due to factors such as request diversity, disparities in PM capabilities, resource multidimensionality, and scale, devising an efficient solution is inherently complex [50]. This mapping endeavor necessitates a design that meets the core data center requirements including the reduction of energy consumption and costs while increasing profit [51].

The issue concerning the placement of virtual machines (VMs) and the optimal selection of destinations for migrations can be framed as a multi-objective bin-packing challenge [52]. Here, the objective is to assign items (VMs) to bins (servers) while keeping the number of bins to a minimum. Each VM is characterized by its size, ensuring it fits within the designated container size without exceeding it [35].

2.2.3 Modeling Real-world Behaviors

Since large scale cloud experiments with hundreds or thousands of servers in the cloud can be costly and risky, researchers often utilize simulation tools to replicate the behaviors of actual cloud [53]. Cloud simulators provide means for modeling complex resource management strategies, testing algorithms, and forecasting system performance without the financial implications or risks of real-world deployment [54]. They also facilitate reproducible experiments, which is crucial for analyzing and comparing algorithms within the research community [55].

Cloud simulators vary in features and functionalities. Some address certain features of cloud computing like energy efficiency or workload scheduling. While others are designed to offer great flexibility and personalization, facilitating a wide variety of research subjects [56]. A more comprehensive simulation tool should allow for a more detailed model that better reflects the dynamic nature of data center operations [57]. Additionally, the representation of energy consumption from different states of PMs and VMs is crucial for accurate energy modeling. Missing this can lead to inaccuracies in the simulation results [58].

Furthermore, some cloud simulators provide mechanisms for analyzing and modeling of energy consumption at different levels. (e.g., per server, per

application, per transaction) [59]. In addition, accurate and realistic modeling depends on detailed data about the hardware environment, energy consumption patterns under variable workloads, and environmental factors like temperature control [2], this entails several key aspects:

(A) **Resource Management:** Simulation of how resources are scaled dynamically in response to workload changes, energy implications of scaling operations, and automatic allocation and deallocation of resources in response to changing workloads. Also, a detailed representation of CPU, memory, storage, and network resources, including their energy consumption profiles must exist [60].

(B) **Workload Simulation:** Emulation of actual workloads is essential for cloud simulators in order to take important decisions regarding resource allocation and decision making. This include user interactions and request patterns, and modeling the impact of workload variability and peak usage periods on energy consumption [17].

(C) **Energy Specific Aspects:** Energy consumption realism measures the simulator's ability to accurately estimate energy usage based on workload demands and resource utilization. This is crucial for sustainable cloud management, allowing users to model energy costs and carbon footprints [61].

(D) **Cost modeling:** Accurate representation of costs based on resource usage, including pay-as-you-go and subscription models, in addition to energy consumption. Simulating different scenarios to find cost-effective configurations to reflect real-world billing [62].

One challenge with existing simulators is their varying levels of accuracy. Their internal models may not fully capture real-world power consumption patterns, workload fluctuations, and hardware diversity. To address this, a systematic comparison of simulators, evaluating their internal behaviors under identical configurations and workloads, can ensure that the selected simulator provides a credible basis for evaluating cloud operations.

2.2.4 Cloud Simulators

Various cloud simulators are employed for research and industrial developments. Let's discuss a few of these to illustrate their advantages.

CloudSim: Presented by Calheiros et al. [63], CloudSim is one of the leading simulators in cloud computing research. It offers a flexible and adaptable framework for modeling data centers, physical and virtual machines, resource provisioning, and workload scheduling [17]. CloudSim supports various features, such as energy-optimized, resource provisioning methods, and VM migration which are essential for cloud resource optimizations. There are several other simulation tools that are built on CloudSim, either extended to model network behaviors [64], enhance scalability [65], model scientific workflows [66], or to further analyze the performance of cloud-based applications [67, 68].

While CloudSim offers valuable features for energy consumption calculation and state modeling of data center components, it still has some limitations regarding its I/O processing model [69], communication models [70], and the inaccuracy of power calculation [1]. Researchers, inspired by these critiques, have actively contributed to the development of more sophisticated tools that address the limitations identified in CloudSim [9].

DISSECT-CF: "DIScrete event baSed Energy Consumption simulaTor for Clouds and Federations" [71] stands as a powerful simulation framework providing the ability to model energy consumption realistically. The simulator integrates energy consumption models that consider the underlying infrastructure, including servers, I/O bandwidth, and storage components [1]. It provides fine-grained energy modeling, allowing researchers to gain insights into the energy usage patterns of cloud environments.

DISSECT-CF produces highly accurate simulation results in terms of finishing time and energy consumption [72]. The reported error of just around 1% in most cases indicates a high level of precision in capturing the behavior of cloud systems [73]. Nonetheless, DISSECT-CF relies on a general network model, limiting the ability to define specific network devices (e.g., routers, switches) and simulate varied network architectures.

GreenCloud: Offers a balanced relationship between computing power and server energy, employing three different power-saving modes [15]. It enables detailed modeling of the energy consumed by individual data center components, such as servers, links, and switches. Moreover, it provides an in-depth analysis of workload distributions [2].

GreenCloud [74] is a packet-level simulator, built on the top of NS-2 [ii]. Its architecture follows a three-tier data center model: access, aggregation, and core layer. It utilizes two power-saving modes, Dynamic Voltage and Frequency

[ii] `https://www.isi.edu/websites/nsnam/ns/`

Scaling (DVFS) and Dynamic Power Management (DPM), influencing the carbon efficiency of clouds. Long simulation times and high memory requirements are primary limitations in GreenCloud [15].

GroudSim: GroudSim [75] offers a discrete event simulation framework for simulating scientific applications in cloud and grid computing environments. It provides both real and simulated execution of real-world programs, leveraging its integration as a back-end within the ASKALON environment [76].

Groudsim supports both CPU and network resources for cloud and grid systems, task submission capabilities, file transfers, failure handling, and background load management [77]. An abstract GroudEntity class that allows the simulator is offered to adjust and monitor the status of user-defined entities. Compared to other simulators like CloudSim or DISSECT-CF, it experiences performance degradation when applied to large-scale applications, and it only offers basic configuration on the network side, making it difficult to configure a realistic network behavior [78].

SCORE: Simulator for Cloud Optimization of Resources and Energy Consumption [79] is a simulator designed with a specific focus on optimizing the utilization of resources and minimizing energy consumption. While SCORE aims to provide a realistic simulation of data centers by offering simulation of energy-efficient monolithic and parallel scheduling models and for the execution of heterogeneous, realistic, and synthetic workloads, it doesn't provide a detailed networking models [80].

Additionally, SCORE doesn't support VM migration policies which is important for modeling real life scenarios in cloud systems [81]. In real-world data centers, networking and VM migration play a crucial role in overall performance, and detailed simulation of network-related aspects is limited in SCORE [82].

To conclude, cloud simulators exhibit some considerable limitations. Some either use abstract models that do not adequately reflect the complexities of real-world systems, or suffer to model large-scale environments that include thousands of nodes [17]. Moreover, challanges such as the availability of real-world data, simplified energy models, and modeling real network behavior restrict efforts to attain high realism [83].

2.2.5 Realism in Cloud Simulation

Realistic cloud simulation is essential in cloud computing research, reflecting the characteristics of real-world systems. It is useful for assessing resource allocation methods, virtual machine placement approaches, and energy efficiency solutions, without relying on large-scale physical infrastructure [84].

Realistic cloud simulations are important for a variety of reasons: (a) realistic cloud simulations enable accurate evaluation of system performance under varying scenarios, (b) accurate energy modeling is vital for evaluating and implementing approaches to lower the energy footprint of cloud data centers, and (c) they offer valuable insights into resource allocation and scheduling policies, resulting in better cloud resource optimization.

Nevertheless, the realism of cloud simulators is influenced by multiple factors [85]:

- Workload Modeling: using realistic workload traces ensures that the simulation captures real-life VM requests, job durations, and resource usage patterns.
- Energy Consumption Modeling: the precision of simulated energy consumption rely strongly on the selection of energy model.
- Resource Utilization Modeling: realistic simulations require precise representation of CPU, memory, disk, and network usage.
- VM Consolidation Realism: proper modeling of VM placement and migration strategies influences the simulator's accuracy in predicting system behavior.

Despite the significance of realism, there is currently no established standard for direct comparison of cloud simulators. Existing studies either verify specific simulator outputs by comparing them with real-world hardware measurements, or develop sophisticated energy models to represent the dynamic nature of cloud environments [17, 78].

While they remain valuable, these strategies lack a standardized methodology, making it difficult to draw holistic conclusions about realism of simulators. To address the limitations of current evaluation techniques, a realism scoring framework is needed to systematically benchmark cloud simulators, adopting standardized scenarios, configurations, and workloads. By applying uniform

conditions across simulators, it is possible to achieve an unbiased comparison against real-world energy and CPU data.

2.2.6 Workload Modeling

Workloads represent the system's computational tasks and the resources they consume over time [86]. They are very useful in performance benchmarking, testing scalability, and energy optimization [87]. Workload modeling is an essential component of research in cloud computing, impacting resource management, energy optimization, and the overall efficiency of the system [16]. Effective cloud management strategies, such as VM allocation and migration, depends on accurate workload modeling [88].

Cloud workloads can be classified according to different characteristics [89]. For instance, they can be static with consistent resource requirements, or dynamic workloads having fluctuating demands. Furthermore, they can be classified based on their resource consumption characteristics into CPU-bound, I/O-bound, and network-bound [90].

Poor workload modeling can affect the accuracy of simulations, which might mislead the evaluation of optimization strategies [91]. Researchers leverage real-world workload traces (e.g. Google Cluster [92], Microsoft Azure [93], and PlanetLab), offering valuable information about real user behavior and system performance, to make simulations more realistic [94].

2.3 Related Works

The demand for cloud services keeps growing, which means more energy is used and higher emissions of CO_2 are generated [95]. Amazon estimates that up to 42% of a data center's operating expenses are due to the energy it uses. This increased energy usage is a big problem for cloud providers because it makes owning and running data centers more expensive [49]. As we saw in 2.2, server virtualization stands as a pivotal technology within cloud computing systems, permitting the deployment and operation of several virtual machines on a single physical server [96].

In this section, we first explore research works regarding energy efficient VM consolidation strategies. We review recent methodologies, algorithms, and

frameworks proposed in the literature to optimize energy consumption, performance, and resource utilization. Then, we delve deeper into cloud simulation tools, highlighting the existing limitations and shortcomings of available cloud simulators. Our goal is to achieve both enhanced simulation accuracy and realistic emulation of cloud computing environments.

2.3.1 Current VMC Algorithms

Many existing VMC heuristics have focused on the exchange between energy consumption and performance of the system [97–99]. However, reducing energy usage may come at the expense of performance degradation and increased SLA violations due to frequent live migrations. Another challenge is the consequence of incessant VM consolidation on system reliability [40]. This can result in increased system failure probabilities by overburdening certain servers. Therefore, an effective VM consolidation method must consider more than just the ideal energy consumption, such as compliance with resource optimization, SLA, and quality of service requirements [14].

Azizi et al. [49] presented an energy-efficient heuristic algorithm named minPR for optimizing VM placement in cloud data centers. Their approach focuses on minimizing energy consumption while ensuring optimal resource utilization and performance. By introducing the resource wastage factor model, the authors manage VM placement on PMs using reward and penalty mechanisms. The proposed algorithm outperformed other previously discussed heuristics like MBFD and RVMP [100] considering the total number of PMs and total resource waste. However, simulation results showed that the algorithm performance varies depending on the VM specification and the type of workloads.

The proposed approach by Beloglazov et al. [48] splits the VM consolidation problem into hosts overloaded/underload detection, the selection of VMs to be migrated, and VMs allocation. Their algorithm includes sorting the VMs in decreasing order based on CPU utilization and allocating each VM to the host that results in the least increase in power consumption due to the allocation. However, the proposed approach focused on improving energy efficiency without optimizing resource wastage.

The integration of their algorithm and the Planetlab experiment[101] within CloudSim have collectively established a standard benchmark in the field. This integration has catalyzed a trend where researchers, proposing novel algorithms for the Virtual Machine (VM) consolidation problem, frequently adopt their

algorithm as a baseline for comparison, showcasing enhancements through the lens of the Planetlab experiment [102].

A dynamic VM consolidation approach was introduced by Sayadnavard et al. [40] using a Discrete Time Markov Chain (DTMC) model and the e-MOABC algorithm, aiming to balance energy consumption, resource wastage, and system reliability in cloud data centers. The algorithm favors VMs with higher impact on the CPU utilization of the server in order to minimize VM migrations. Devising a resource usage factor technique to utilize PM resources efficiently, Gupta et al. [103] proposed a new VM placement algorithm to minimize the power consumption of the data center by decreasing number of total active PMs. While these approaches showed substantial improvements in energy efficiency, SLA violation need to be considered during the evaluations.

Ghetas [104] proposed the MBO-VM method to reduce energy consumption and minimize resource wastage by maximizing the packaging efficiency. Utilizing a multi-objective Monarch Butterfly Algorithm, the approach considers the CPU and memory dimensions for VM placement optimization. Khan [105] proposed a normalization-based VM consolidation (NVMC) strategy that aims to place VMs while minimizing energy usage and SLA violations by reducing the number of VM live migrations. Despite considering multiple resources of PMs, reliance on CPU-centric calculations and VM sorting poses limitations in heterogeneous environments with varying workloads.

A Load-Balanced Multi-Dimensional Bin-Packing heuristic (LBMBP) to optimize resource allocation in cloud data centers was introduced by Nehra et al. [106]. Nevertheless, the exclusion of resource availability beyond CPU in the power model limits its reliability. Moreover, Mahmoodabadi et al. [107] focused on the bin packing with linear usage cost (BPLUC). They examined the VM placement problem with three dimensions, CPU, RAM, and bandwidth, aiming at reducing the power consumption. They compared their result with PABFD [48], GRVMP [108], and AFED-EF [109]. The approach demonstrates efficiency compared to existing methods.

Following bin-packing heuristics, Sunil et al. [110] proposed energy-efficient VM placement algorithms, EEVMP and MEEVMP, considering server energy efficiency. The researchers aims to optimize energy consumption, QoS, and resource utilization while minimizing SLA violations. However, evaluations of the varied workloads and infrastructures while considering additional resources are necessary to ascertain the adaptability of these algorithms.

An enhanced levy-based particle swarm optimization algorithm with variable sized bin packing (PSOLBP) is proposed by Fatima et al. [111] for

solving the problem combining levy flight and PSO algorithms. An EVMC method for energy-aware virtual machines consolidation was proposed by Zolfaghari et al. [58], integrating machine learning and meta-heuristic techniques to optimize the energy consumption. While considering all resources, including CPU, RAM, storage, and bandwidth, the future load of the servers isn't considered when placing VMs.

Tarafdar et al. [112] proposed an energy-efficient and QoS-aware approach using Markov chain-based prediction and linear weighted sum. The simulation results showed a substantial reduction in the energy consumption, number of VM migrations and SLA violations compared with other VM consolidation approaches. A two-phase energy-aware load-balancing algorithm (EALBPSO) using PSO for VM migration in DVFS-enabled cloud data centers was presented by Masoudi et al. [113]. While demonstrating improvements in power consumption and migration, SLA violation isn't considered during the evaluations.

Many researchers have focused on cloud service selection as well as task assignments and their effect on resource utilization and energy consumption by effectively scheduling user tasks. Nagarajan et al. [114] intorduced a comprehensive survey discussing the advantages and limitations of research investigating the cloud service brokerage concept. They also outlined various open research challenges and provided recommendations. Finally, they introduced an intelligent cloud broker for the effective selection and delivery of cloud services aiming at utilizing cloud resources.

An energy-optimized embedded load-balancing approach that prioritizes the tasks regarding their execution deadline was proposed by Javadpour et al. [116]. It also categorized the physical machines considering their configuration status. The algorithm prioritizes tasks based on execution deadlines and server configuration, utilizing DVFS to reduce energy consumption. A multi-resource alignment algorithm for VM placement and resource management in cloud environment was analyzed by Gabhane et al. [117]. Several algorithms were compared based on CPU and memory utilization as well as the probability of task failure. Multi-resource alignment demonstrated better performance in CPU and memory utilization compared to other algorithms.

Lima et al. [121] introduced a VMP algorithm for optimizing the service allocation process. In their evaluation, they considered two scenarios based on task arrival. The first scenario assumes no prior knowledge of task execution time, and the second scenario assumes all tasks arrive simultaneously. The results indicated a significant improvement in the average task waiting time. A two-phase multi-objective VM placement and consolidation approach employing

Table 2.1: Resources and evaluation metrics used by authors in the literature.

Author	Considered Resources				Metrics Used in Evalution				
	CPU	RAM	B/W	Disk	Energy	Resource Wastage	SLA	#active PMs	#of migrations
Azizi et al. [49]	√	√	-	-	√	√	-	√	√
Sayadnavard et al. [40]	√	√	√	-	√	√	√	-	√
Gao et al. [115]	√	√	-	-	√	√	-	-	-
Panda et al. [100]	√	√	√	-	√	√	√	√	√
Gupta et al. [103]	√	√	-	-	√	√	-	√	-
Ghetas [104]	√	√	-	-	√	√	-	√	-
Khan [105]	√	√	-	√	√	√	√	-	-
Beloglazov et al. [48]	√	-	-	-	√	-	√	√	-
Nehra et al. [106]	√	√	√	√	√	√	-	√	-
Mahmoodabadi [107]	√	√	√	-	√	√	√	-	-
Azizi et al. [108]	√	√	-	-	√	-	-	√	√
Zhou et al. [109]	√	-	√	-	√	√	-	√	√
Sunil et al. [110]	√	-	-	-	√	-	√	√	√
Fatima et al. [111]	√	-	-	-	√	-	-	√	√
Zolfaghari et al. [58]	√	√	√	√	√	-	√	√	√
Tarafdar et al. [112]	√	√	-	-	√	-	√	-	√
Masoudi et al. [113]	√	√	-	-	√	-	-	-	√
Javadpour et al. [116]	√	-	-	-	√	-	√	-	√
Gabhane et al. [117]	√	√	-	-	√	-	-	-	-
Nikzad et al. [118]	√	√	-	-	√	-	√	-	√
Li et al. [119]	√	√	-	-	√	-	√	√	√
Tlili et al. [120]	√	√	√	-	-	√	-	-	√

the DVFS technique was proposed by Nikzad et al. [118]. Using a multi-objective ant colony algorithm, the approach aims to improve energy consumption and SLA violations.

An efficient VM consolidation approach EQ-VMC was introduced by Li et al. [119], which has the goal of optimizing energy efficiency and service quality integrating discrete differential evolution and heuristic algorithms. By considering host overload detection and VM selection, EQ-VMC aims to reduce energy consumption and enhance QoS. Simulation results demonstrated improvements in energy consumption and host overloading risk as well as improved QoS. A Best Fit Decreasing algorithm for VMP formulated as a bin-packing problem was proposed by Tlili et al. [120]. The simulation results demonstrated higher packing efficiency compared to other algorithms. Nevertheless, they did not take SLA violation into account.

While numerous energy-aware algorithms focus primarily on reducing energy consumption within data centers, none have simultaneously considered PMs' heterogeneity and multidimensional resources, minimizing the energy consumption, balancing the resource wastage, and improving SLA altogether. Our proposed algorithm in chapter 4 adopts a more comprehensive approach. In addition to enhancing energy usage, we consider power efficiency, SLA violation, and resource wastage (both CPU and memory) on host servers collectively during VM allocation and placement. Table 2.1 gives a summary of the resources and evaluation metrics used in the literature.

2.3.2 Simulation Accuracy and Realism

Realistic simulation of these data centers is crucial for understanding their performance, optimizing resource utilization, and addressing environmental concerns such as energy consumption [122]. Simulating energy consumption accurately requires a comprehensive energy model that encompasses various aspects, including servers and networking equipment [57]. For instance, modeling the dynamic power consumption of servers based on their utilization levels and accounting for the static power draw is crucial for capturing the nuances of energy usage in a data center.

Moreover, considering the impact of workload variations on energy consumption is necessary for a realistic simulation [61]. A robust energy model should integrate workload-aware power models, reflecting the fluctuating demands on the data center resources. This enables the simulation to mirror real-world

scenarios where energy consumption varies based on the nature and intensity of workloads [123]. To enhance accuracy, researchers often leverage trace-based simulations, where the simulation parameters and workload characteristics are derived from real-world traces. Additionally, continuous refinement of simulation models based on empirical data helps in minimizing the gap between simulation and real world [78].

A number of research works have focused on reproducing works using different simulators aiming to examine the effect of them on operational process. Mann [1] described their experience in porting a VM placement algorithm from one cloud simulator to another, proposing a layer of abstraction for implementing the VM allocation policy using Planetlab workloads. Similarly, Bahwaireth et al. [124] compared several simulation tools by applying different scenarios but failed to establish identical setups among the simulators, which is crucial for accurate and fair comparative analysis. Most existing research lacks a consistent cloud infrastructure across simulators, thereby limiting the validity of the findings.

Bambrick et al. [78] presented a comparative analysis of widely-used simulators, focusing on their supported models, architectures, and high-level features. However, they overlooked the internal behavior of cloud entities across different simulators, which can significantly influence outcomes. Mansouri et al. [17] provided a detailed survey of existing cloud simulators, discussing their features and software architectures but did not explore how varying simulator behaviors impact algorithm implementation. Likewise, Di et al.[92] attempted to reproduce a Google cloud environment using real experimental settings and large-scale production traces [125]. While they successfully demonstrated the simulation system's capability to reproduce real checkpointing and restart events, they did not compare their results against other simulators, thus missing an opportunity to validate their findings across platforms.

Many researchers have sought to enhance the realism of cloud simulations by incorporating energy consumption models, resource utilization metrics, and realistic workload patterns. Alshammari et al.[126] emphasized the necessity for robust validation methods in cloud simulations, as many existing tools struggle to predict energy consumption and resource utilization accurately. While the study provides valuable insights on improvements needed in simulators to better reflect real-world conditions, its reliance on a Raspberry Pi-based testbed may limit the generalizability to more complex data center infrastructures.

Ilager et al. [128] presented a data-driven analysis of private cloud's physical machine-level resource utilization, energy consumption, and thermal behavior over nine months. Their study, based on data from 144 servers, revealed

Table 2.2: Comparison of metrics used for cloud simulation realism in the literature.

Study	Comparison within one simulator		Comparison across multiple simulators			
	real energy measurements?	real CPU measurements?	Identical setup?	Identical VM allocation?	Identical Power model?	Real data used?
Mann [1]	-	√	√	√	-	√
Bahwaireth et al. [124]	-	√	-	√	√	-
Makaratzis et al. [127]	-	-	√	√	√	-
Alshammari et al.[126]	√	√	√	-	-	√
Bambrik et al. [78]	-	-	√	-	√	-
Mansouri et al. [17]	-	-	√	√	√	-
Ilager et al. [128]	√	√	-	-	√	√
Tian et al. [129]	-	-	√	√	√	-
Proano et al. [130]	-	√	-	-	√	√
Sakellar et al. [131]	-	-	-	-	√	-
Umar et al. [132]	-	-	√	√	-	-
Estrada et al. [134]	√	√	√	-	-	√
Ismail et al. [7]	-	-	-	√	√	-
Di et al. [92]	-	√	√	-	-	√
Shahid et al. [133]	-	-	√	√	√	-
Mane et al. [135]	√	-	√	-	-	√

non-linear relationships between utilization and energy/thermal metrics. While the absence of VM-level data restricts deeper analysis of application-specific impacts, the work provides a valuable foundation for further research and optimization strategies.

Several cloud simulation surveys have been done, but only few tried to address energy driven aspects. Some categorize simulation tools based on attributes, availability, and features provided [78, 131, 132]. Others focused on system architecture aspects and modeling support of simulators while choosing suitable tools based on their specific requirements [7, 9, 129]. Makaratzis et al.[127] conducted an in-depth study of cloud simulation frameworks, highlighting the significance of accurate energy models for evaluating potential energy usage. They compared multiple simulation tools to examine their strategies for energy prediction.

The study by Makaratzis et al. [127] underlined that linear interpolation model is considered as highly precise because of their proficiency in approximating intermediate utilization points. However, the absence of real-world data for all levels of utilization, hindering the ability to reach firm conclusions concerning the accuracy of these models.

While existing approaches classify simulators based on their features, architectures, or specific applications, a significant gap remains in evaluating their realism, especially in terms of energy efficiency and resource allocation. These approaches often overlook how accurately these simulators replicate real-world behavior, which limits their reliability for research requiring precise cloud environment modeling. Table 2.2 presents a comparison of existing research based on the realism metrics they incorporate.

2.4 Summary

This chapter provided an in-depth exploration of the fundamental ideas and studies relevant to cloud computing, virtualization, virtual machine consolidation, and cloud simulation. An introduction to cloud computing initiated the discussion, highlighting its essential features and service layers. The discussion then shifted to virtualization, a cornerstone of cloud computing, emphasizing its ability to abstract resources, handle workload, and optimize cloud operations. Next, the chapter delved into an examination of VM consolidation approaches used to optimize resource utilization and energy consumption in cloud data centers.

The exploration continued with an overview of cloud simulation, analyzing various simulators such as CloudSim, DISSECT-CF, GreenCloud, and GroudSim. These tools play a vital role for cost-effective and scalable evaluation of cloud resource management strategies. The discussion focused on the strengths and weaknesses of these simulators, particularly in terms of their support for energy consumption analysis and workload representations, as these factors directly impact the reliability of simulation-based research. Studies focusing on the comparative analysis of cloud simulators and their accuracy in mimicking real-world behaviors were analyzed.

Finally, the chapter emphasized the importance of a unified evaluation framework to assess the realism of cloud simulators. This discussion lays the groundwork for Chapter 3, where a novel realism scoring framework for cloud simulators is introduced, based on their ability to accurately model energy consumption and resource utilization. Chapter 4 further expands on this discussion by introducing an energy-efficient virtual machine placement algorithm that optimize energy efficiency and resource utilization while maintaining SLA adherence.

3

Realism in Cloud Simulation: Definitions and Scoring Framework

3.1 Introduction

Cloud simulators play an important role in evaluating cloud systems performance. They minimize costs and risks associated with experimenting in live environments [17, 55]. Nevertheless, the usefulness of cloud simulators depends on their ability to replicate real-world behaviors accurately [78]. This replication is crucial for tasks such as virtual machine migration, resource scheduling [38], and energy consumption minimization, in cloud data centers [136].

Realistic simulation in the context of a cloud environment is expected to be a comprehensive and accurate emulation of real-world systems, applications, and workloads within a cloud infrastructure [84]. This simulation must replicate the behavior, performance, constraints, and dynamics of the actual environments to provide credible insights and outcomes. This includes:

- reliable predictions about system behavior and performance under various conditions.
- opportunities for optimizing resource usage to reduce energy consumption.
- and sustainable cloud practices by simulating and recommending energy-efficient configurations and operations [137].

Achieving realism in cloud simulation is a complex task because of the diversity of cloud workloads, variance of resource demands, and complex energy behaviors [138]. However, realism's standardized assessment allows for improvement of simulators, which is beneficial for both research and industry applications [17]. In this chapter, we explore the concept of realism in cloud simulation, with particular focus on energy consumption accuracy and resource utilization. We propose a systematic framework for defining, measuring, and scoring realism across different simulators.

In addition, we evaluate the framework's applicability and highlight its potential to improve the quality and consistency of cloud simulation research through a detailed case study and experiments. We define measurable metrics for energy consumption, resource utilization, and performance evaluation, exploring CloudSim and DISSECT-CF as two widely used cloud simulators. Through rigorous experimentation and comparison with real-world data, we demonstrate how our realism score can bring valuable insights into the strengths and limitations of simulators, guiding researchers and practitioners in selecting the most suitable tool for their needs.

This chapter continues with the following: Section 3.2 defines realism in cloud simulators, setting up a unified metric system to benchmarck simulation tools. Section 3.3 presents the workload profile to represent realistic workloads, encompases different utilization levels to reflect real-world cloud data center tasks. Section 3.4 describes the process of collecting data from real and simulated environment. Section 3.5 presents the metrics used to calculate realism score for cloud simulators. In Section 3.6, we demonstrate the required prerequisites for having identical setup among simulators, ensuring fair comparison. Section 3.7 covers experiments and simulations to examine realism. Section 3.8 concludes the chapter, and highlight the outcomes.

3.2 Designing Unified Metric System

Realistic simulation environments provide opportunities for users to make a decisive assesments about performance, power efficiency, and budget control. These assesments can be readily translated into real-world operations. Cloud simulators diverge in the way they present realism, some offer resource scheduling mechanisms, others target power consumption measurements, and some provide distinct services like memory utilization and network communications.

Each acts in different way considering the internal configuration and design aims.

Nevertheless, the majority of cloud simulators target resource utilization and energy consumption measurements as these metrics are important in cloud data centers. To determine the degree to which a cloud simulator accurately reflects real-world behavior, a unified realism score mechanism is essential as it determines how close a simulator can model real-life environments. This should provide a standardized score so that cloud simulators can be compared based on accuracy they provide. Such comparisons would foster usability and effectiveness of the simulator in research and applications.

Additionally, setting up a unified metric helps in benchmarking simulation tools. This is helpful when identifying the effectiveness of imitating real-world experiments, providing guidance to users prioritizing their simulation tool selection, and also for developers to improve their tools by identifying the strenghes and weaknesses and setting a room for improvements. The first step towards unified realism score is to define what realism means in cloud computing simulation.

Realism Definition: Realism is *"the proportion to which the environment and behavior of simulators conform to real-world cloud system operations"*. It should encompass aspects such as workload characteristics, resource utilization, task scheduling, energy consumption, and network dynamics. Realism ensures that findings derived from simulations are transferable to real-world systems expanding their practical value.

Our proposed realism score framework aims to:

1. Offer a standardized approach to evaluate and compare the realism of different cloud simulators.

2. Assess the alignment between simulated and real-world data using comparable metrics.

3. Inspire simulator developers to aim at realistic modeling by iterative adjustments.

Finding one cloud simulator that can model all aspects and behaviors of real clouds is challenging task. Our primary focus is on resource utilization and energy consumption metrics in cloud simulation, as these factors play a crucial role in the virtual machine consolidation decision-making process, a critical part of energy optimization. Therefore, we can split realism in cloud simulation into two main key components:

- **Resource Utilization:** The degree to which the simulated computational resources align with real-world under a predefined setting with varying workload characteristics.

- **Energy Consumption:** The level of accuracy that a simulator demonstrates in terms of power usage patterns in cloud environments compared to actual power draw, considering various states such as idle, stress, and dynamic workloads.

3.3 Workload Profiles

Workload profiles are comprehensive representations of the resource utilization and operational patterns of applications in cloud platforms across time. It sheds light on how system resources are utilized as time progresses, guiding decisions with respect to resource distribution, performance analysis, and system architecture. They are essential for reproducing and examining results across different simulation tools [139]. By capturing resource usage patterns and user demands across time, workload profiles serve as benchmarks for assessing the effectiveness and realism of simulation models.

Workload profiles feature several primary components and properties. For instance, resource utilization trends indicate the range of system resources (e.g., CPU, memory, and bandwidth) during run time. Execution patterns represent another important property, which include the time of workload execution along with the count of tasks executed. These patterns strongly affect performance of systems and are often used to capture a variety of operational features and scenarios. Additionally, behavioral patterns illustrate the variations in workloads depending on load changes. Combined, these attributes give a holistic view of workload dynamics, enabling clear representation of real-world cloud environments [16, 128].

In the absence of realistic workload profiles, simulation results can be misleading or overly optimistic. Thus, leading to incorrect assessments in cloud management performance analyses. Based on discussion above, a five-level utilization framework is introduced as a standardized workload profiling. This framework captures primary functional states in cloud data centers, transitioning from minimal to peak resource utilization. Each level demonstrates unique characteristics and real world examples observed in cloud environments:

- **Idle Uitilization:** Minimal resource utilization (0-2%), often matched to off-peak hours where the system is in a low-power state or standby mode. this level of utilization could last for three hours a day in average [16]. The main reason behind including this level is to model baseline energy usage, which is crucial for power efficiency analysis.
- **Low Utilization:** Periods of low demands in which workloads are light (up to 30% utilization). This level is the most common in data centers, typically reflecting applications requiring minimal resources, few test scripts, or small web services. It highlights system behavior for lightweight operations and it may continue for up to 15 hours per day [88, 128].
- **Medium Utilization:** Moderate periods (30-60%) to reflect regular and steady operations. This level of utilization reflects scenarios like running a moderate data analytics or business operations and it might be observed for 3 hours everyday [89].
- **High Utilization:** Periods of high demand where CPU is heavily used (60-90% utilization). These could represent peak times during the day when more users are active. High utilization explores periods of heavy transactional workloads and it may take around 2 hours per day [128].
- **Maximum Unitilization:** In real data centers, some PMs might experience extreme load conditions or a full-utilization (90-100%) state, such as running computationally intensive simulations or modeling tasks. This level of utilization may endure an average of one hour per day [16].

The main reason behind introducing five levels of utilization is to provides a balanced representation of diverse operational states while ensuring simplicity and reproducibility. A lower number of levels fail to represent critical transitions, such as the gradual increase from idle to low utilization. On the other hand, increasing the number of levels can add unnecessary complexity and limit the practical applicability of the model, as smaller variations may have negligible effects on the energy efficiency and system performance.

To record the highest attainable samples for dynamic and varying utilization per level, the average utilization of each phase is expected to be approximate to *(min + max) / 2*, where *min and max* represent the minimum and maximum percentages of a given utilization level. For instance, when measuring the medium utilization level (between 30% and 60%) over one hour, the average utilization for that hour should be approximately 45%.

To illustrate, this approach ensures variance in utilization levels between the minimum and maximum (30% and 60%) and promotes unbiased behavior in

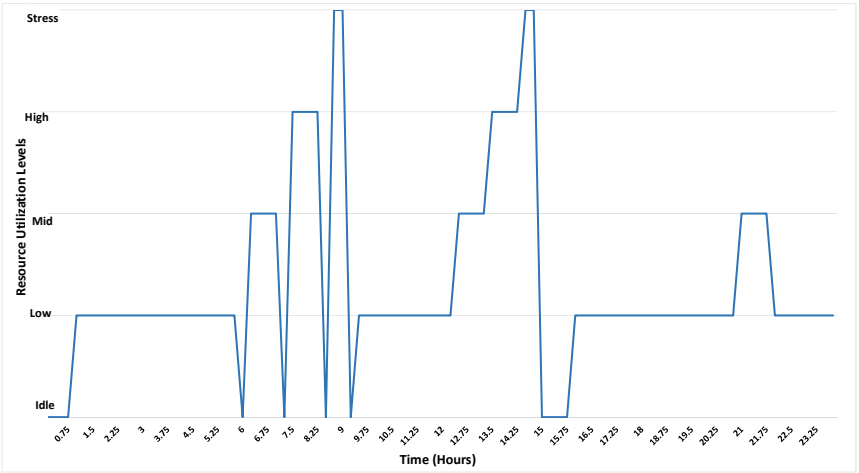

Figure 3.1: Utilization cycle test for a period of 24 hours

simulators, thereby achieving a balanced number of increases and decreases in utilization during that period. It also reflect dynamic nature of cloud workload in which resources allocation are changing dynamically based on the kind of running applications.

Real-World Context: Consider the comparison with vehicle emissions testing, such as the Worldwide Harmonized Light Vehicle Test Procedure (WLTP). Just as WLTP incorporates multiple measures to analyze vehicle performance under a range of conditions, this framework integrates statistical and correlation analysis to offer a thorough evaluation of simulator realism.

The five-level framework integrates the advantages of real workloads (offering authentic trends and recording subtle variations and irregularities) and synthetic workloads (convenient to generate and replicate, offering ability to control specific features) by enabling a systematic yet versatile model to ensure a blend of realism and reliable reproduction. To capture the dynamic nature of cloud workloads, several common patterns, that servers in a normal cloud system might experience during a day, are introduced in Figure 3.1.

The figure also illustrates direct and gradual transitions among various utilization levels to reflect different types of applications where resource demands differs with the amount of resources they utilize. The first part of the figure shows direct transitions from idle state to all other states while confirming that

reference energy usage is modeled accurately. This highlights the increase in resource utilization when launching various types of applications.It also helps in identifying each utilization level seperately while giving a better understanding on simulator behaviors for each given phase. For instance, capturing transitions from idle to low is critical since idle power consumption is non-zero, and power models in simulation tools must precisely reflect this baseline utilization level.

Additionally, transitions from idle to high levels are common in fault-tolerant systems, idle nodes might be kept on standby mode, stepping in for a failed active node and taking over its full operational load. Later, the figure illustrates a gradual increase from idle to stress phases, passing through all intermediate levels of utilization. This progression reflects the rise in user requests during working hours, offering insights into system scalability, and can also represent the calibration process of cloud resources.

The final part of the figure represents periods with minimal user interactions and low demands, featuring occasional spikes that could correspond to backup operations or other maintenance tasks performed during non operational periods. The transitions among different levels of utilizations were chosen to benchmark simulation tools and also to facilitate reproducibility.

3.4 Data Collection

In real-life cloud computing environments, dynamic energy consumption is calculated based on actual power measurements from hardware components, typically using power meters or built-in sensors.

The process of collecting data from real-world environments is crucial, as this data are needed to evaluate the accuracy of simulation tools. By comparing the results from both real and simulated environments, it is possible to determine the level of accuracy for simulators in terms of measurement precision. There are numerous tools commonly used to record data from servers in cloud data centers, including metrics such as CPU, memory, network usage, and their respective energy consumption. Each component exhibits a different power-to-utilization ratio, depending on its internal design and the type of applications it executes.

3.4.1 Real-world Data Collection

Gathering real-world data is a key step in ensuring authentic realism score for cloud simulators. Accurate measurements on resource utilization and energy consumption establish a reference in which simulated outcomes can be evaluated. There is a wide range of tools for monitoring resource utilization and energy consumption, each providing multiple functionalities and levels of accuracy. For instance, external power meters can measure the actual power drawn by individual servers or entire racks, Intelligent Power Distribution Units (PDUs)[i] can measure and report power consumption for the connected equipment.

Many modern servers have built-in sensors that monitor power consumption and report metrics like voltage, current, and temperature. In addition, software tools like Intel's RAPL (Running Average Power Limit) and AMD's PowerNow provide APIs to read power consumption metrics directly from the CPU.[ii] Amazon has AWS CloudWatch to monitor services for its resources, providing insights into several metrics, such as CPU utilization and memory usage. There is also a variety of tools for recording data, such as Azure Monitor, Nagios, and Zabbix, which provide CPU utilization and other performance metrics. [iii]

Users are expected to adhere to specific measurement guidelines when gathering real data. Initially, workload traces should be obtained from cloud environments running a range of applications with varying resource demands, aligning with the utilization levels specified in section 3.3. CPU utilization must be measured at fine intervals to accurately capture dynamic workload fluctuations. Moreover, hardware and software configurations should be recorded to maintain consistency and to avoid potential biases due to infrastructure differences. Data collected on CPU utilization is used as input to simulators, and resulting simulation measurements should align with corresponding real measurements with specified time intervals.

In summary, despite the complexity and cost associated with hardware tools, they provide more accurate measurements and real-time data recording compared to software tools, while software monitoring tools offer cost-effective measurements with acceptable precision. In fields like computer science and cloud computing research, it's common to extract specific parts of large datasets to focus on particular utilization levels or to study specific behavioral patterns.

[i] https://www.hpe.com/psnow/doc/c04123329
[ii] https://powerapi.org/reference/formulas/rapl/
[iii] https://www.zabbix.com/integrations/azure

This is usually done by specifying the required utilization levels and applying data acquisition technique for extraction while ensuring data continuity by selecting consecutive intervals.

3.4.2 Simulated Data Collection

Simulation tools involve different models for resource utilization and power consumption to represent practical scenarios. This enables users to gain valuable insights in handling their infrastructure and pinpoint the elements or the applications that mostly affect system behavior. Additionally, such tools guide cloud providers and researchers on strategies to save energy and enhance efficiency, thereby fostering the adoption of energy-efficient solutions that optimize total system performance.

To maintain uniformity and comparability across simulators, it is important to adopt a standardized procedure for collecting results. This includes using identical power model and utilization levels (or workload behaviors) across simulations. By doing so, we can eliminate variability caused by differing setups and focus on evaluating the core behavior of each simulator. Several power models are used to capture the characteristics of hardware environments. For instance, resource utilization can be effectively modeled linearly with power consumption [140]. The simplicity of this model makes it a suitable choice for benchmarking and allows for straightforward integration across simulators.

The first step is to configure simulators to replicate the real-world hardware setup as closely as possible. This may include parameters such as CPU, memory, network bandwidth, disk, operating system, and room temperature. It is important to note that not all simulators can simulate all of these parameters. In such cases, the metrics used should be clearly defined to ensure transparency and enable fair comparisons among simulators.

Data gathered from the real world infrastructure, following our utilization cycle from figure 3.1, is input into the simulation framework to compare the accuracy of resource allocation between the real and simulated environments. The data collected from real-world sources will be directly compared to their simulated equivalents. Ultimately, the realism score will be based on the error rate between real and simulated outputs.

By this stage, the expectation is to have a comprehensive table of data from both real-world and simulated measurements, aligned based on timestamps,

Table 3.1: A sample of real data gathering for simulation

Time interval	U_{real_i}	P_{real_i}	U_{simu_i}	P_{simu_i}
t_1	U_{real_1}	P_{real_1}	U_{simu_1}	P_{simu_1}
t_2	U_{real_2}	P_{real_2}	U_{simu_2}	P_{simu_2}
t_3	U_{real_3}	P_{real_3}	U_{simu_3}	P_{simu_3}
...	...	...	...	...
...	...	...	...	...
t_n	U_{real_n}	P_{real_n}	U_{simu_n}	P_{simu_n}

and including both CPU and power consumption measurements. This step ensures comparability across the datasets. Table 3.1 shows a data sample for score calculation, where $\{t_1,t_2,...,t_n\}$ define time intervals during which data collection took place. U_{real_i} and U_{simu_i} represent real-world and simulated utilization measurements, while P_{real_i} and P_{simu_i} reflect the power readings from both real-world and simulation environments. Now that the data has been collected, we can proceed to calculate our proposed realism score.

3.5 The Realism Score Calculation

Our realism score offers a structured method for assessing the precision of cloud simulators in reproducing real-world data patterns. The score indicates the divergence between simulated and real-world measurements adopting multiple statistical measures, promoting a rigorous and consistent evaluation approach. This section describes the framework and supporting equations leading to the final realism score, targeting both the errors magnitude and the alignment of trends within real and simulated data sets.

Additionally, the framework fosters improvements in cloud simulators, particularly in energy modeling and resource usage precision. The primary goal is to evaluate simulation accuracy through various aspects, accounting for error magnitude, energy consumption alignment for workload profiles, and unified realism score incorporating all these factors. One of the steps envoloved in error calculation is evaluating the absolute difference between two measurements for each data point. This is useful as absolute error (AE) gives an indication of the discrepancy in each measurement.

Let R_{Measure} and S_{Measure} be the real and simulated measurements for each data point. The absolute error can be calculated as follow:

$$\text{AE} = |R_{\text{Measure}} - S_{\text{Measure}}|$$

This step allows both overestimates and underestimates to have an effect on the error metric without cancelling each others. To calculate the mean absolute differences across all data points, Mean Absolute Error (MAE) provides a measure of the overall deviation of simulated measurements from the real-world data where lower MAE indicates better alignment between real and simulated data:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |R_{\text{Measure}_i} - S_{\text{Measure}_i}|$$

where n represent the number of data points used for comparison. To make error values comparable across diverse workloads with varying scales of utilization, Mean Absolute Percentage Error (MAPE) is used to provide a direct measure of how much the simulated measurements deviate from the real-world measurements in percentage terms, making it a widely used metric for simulation accuracy.

$$MAPE = \frac{1}{n}\sum_{i=0}^{n} \frac{|R_{\text{Measure}_i} - S_{\text{Measure}_i}|}{R_{\text{Measure}_i}} \times 100\% \tag{3.1}$$

Since the lower value of MAPE indicate better precision, and we want higher realism score to demonstrate better simulation, we subtract from 100. The realism score (R_{Score}) can be calculated as:

$$R_{\text{Score}} = W_p \times (100 - MAPE_p) + W_u \times (100 - MAPE_u) \tag{3.2}$$

where W_p and W_u are the weights for power and utilization and they can be adjusted according to the focus of the evaluation, $MAPE_p$ and $MAPE_u$ are the mean absolute percentage error for power and utilization respectively.

In our framework, we set W_p to 0.8 and W_u to 0.2 as power consumption precision holds greater importance in our evaluation. In addition, the choice of our weights is mainly attributed to varying scale and variability of CPU utilization to power consumption. CPU utilization is expressed as a percentage from 0% to

100%, while power consumption is measured in watts and depends on hardware specifications. It is worth mentioning that R_{Score} in equation 3.2 is centered on CPU utilization measurements with their related energy consumption. To consider others resources such as memory and bandwidth, additional weights for each H/W component might be encorporated.

Illustrative example: Assume $MAPE_p$ is 30%, and $MAPE_u$ is 10%. R_{Score} can be calculated as follows:

$$R_{\text{Score}} = 0.8 \times (100 - 30) + 0.2 \times (100 - 10) = 74.$$

A Realism Score of 74 show that the simulator achieves relatively good performance (the closer to 100 the better).

3.6 Establishing Comparable Simulation Setup

Before applying the realism scoring framework, a preliminary investigation has to be conducted to set up equivalent configurations between, the two to be compared, cloud simulation tools. In this section, we use CloudSim and Dissect-CF as examples. This is necessary to establish a relevant comparison of their realism, as simulators differ significantly in their structural designs and core functionalities. Directly comparing results obtained with default settings would be misleading because of the inherent differences in modeling practices of simulators.

The realism score depends on the precise reproduction of PMs and VMs on both simulators with the same specifications (e.g., CPU cores, processing speed, and power model). In our case, we choose to implement a PM with same specification as the *HP ProLiant DL560 Gen9* with Intel Xeon E5-4650 v4, 64 GB RAM, and 1 GB bandwidth. For the workload data, we implement new mechanism in DISSECT-CF simulator so that it can load the same Planetlab [141] workload structure in CloudSim. For a good quality realism score, the compared simulators must have the same amount of energy consumed for their respective PMs.

3.6.1 Trace Loading Mechanism

In cloud simulation, workload loaders are important for the interpretation and execution of input workloads within the simulated environment. These loaders handle tasks allocation, resource usage emulation, and trace parsing. To ensure consistency in experiment setups, identical workload loaders has to be implemented. This also helps in eliminating discrepancies due to fluctuations in job scheduling, resource provisioning, and submission time, ensuring that any observed differences in simulation outcomes are caused by the internal models of simulators.

One of the advantages of CloudSim that attracts many researchers is that it has a builtin workload traces(Planetlab workload). It contains information from 10 days about CPU usage from around 1000 VMs, these information can be found in *examples/workload/planetlab* folder in CloudSim. The CPU load data are stored as simple text files in which each file contains 288 values reflecting the CPU utilization of one VM for a day. Thus, each value in a file represents a CPU utilization taken every 5 minutes.

Beloglazov et al. [48] have made some arrangements so they could evaluate their algorithm with realistic data for testing. They have implemented *UtilizationModelPlanetLabInMemory* class for the cloudlet utilization model which reads the utilization values from a file. For their experimentation setup, they have made the *PlanetLabRunner* class with some helper classes (*PlanetLabHelper* and *PlanetLabConstants*) to provide parameters for simulation. These parameters include the name of the folder corresponding to a specific date of the Planetlab data in which the folder consists of many files contain the CPU values for a VM. Finally, they have created *helper* class to set up PMs and VMs based on the data in the constants class.

In order to use the Planetlab data in the DISSECT-CF simulator, we introduced a new trace loading mechanism [139]. This mechanism aims to generate jobs based on Planetlab workload trace, enabling these jobs to utilize the VMs throughout simulation process. We created two new classes, *PlanetLabFolderReader* class in which it is responsible for choosing the experiment date, then *PlanetLabFileReader* is created to open the files inside the folder and create a job for each line in the files. This class implements the *CreateJobFromLine* method in the DISSECT-CF simulator. Table 3.2 shows a brief description of trace loading mechanism. Jobs created in DISSECT-CF have:

Table 3.2: Trace loading mechanism in DISSECT-CF

```
1:  Open Planetlab directory.
2:  Choose specific day for loading data.
3:  For each file inside the folder:
4:      While the file has line:
5:          Construct a job from line.
6:          Setup the submission time and the executable value.
7:  Add the job to List of jobs.
```

- Start time: we configure each job with different starting time with a 300 seconds interval between any two succissive jobs so that each job can run on the same VM once the previous job finishes.

- Job type: the *PlanetLabFileReader* class will insure that all the 288 jobs to be created have the same executable value so that they could all run on a specific VM later (DISSECT-CF have *VMSetPerKind* map in which it bond a VM type to certain type of jobs)

3.6.2 Physical Machine Setup

Even when executing identical workloads, resource utilization and power consumption can be affected due to factors like CPU design, energy efficiency, and other hardware specification of the PM. For example, a server featuring advanced power-saving mechanism may exhibit lower energy than an outdated server. This would distort the interpretation of realism when compared to simulated results. Hence, ensuring identical hardware configurations, power measurement strategies, and workload execution settings enhance realism assessments.

Many differences have been observed during the implementation of PMs on both simulators. In order to implement a PM in CloudSim (called *Host*) while getting results regarding energy and utilization, we used CloudSim power package. We first created a data center (*PowerDataCenter class*) object in which we could add PM (*PowerHostUtilizationHistory*) to it. For host creation, we needed to specify the ID, RAM, network bandwidth, storage, number of CPUs, and power model. We also built a new power model to reflect the energy consumption of the server according to the CPU utilization percentage.

This is done by extending (*PowerModelSpecPower*) class and implementing its (*getPowerData*) function.

For DISSECT-CF, creating PM (*PhysicalMachine*) was more complicated than CloudSim because DISSECT-CF tries to imitate real life cloud infrastructure in more detail. We needed to create a repository object representing the disk, connect it to network, and defining a power model for the power characteristics descriptions. Also, DISSECT-CF defines three consumption models (CPU, memory, and network) inside the power model of a PM, so we modify the consumption model to have the CPU amount of energy separately.

In addition to *HP ProLiant DL560 Gen9*, we incorporated two additional power models for *HP ProLiant ML110 G4* and *HP ProLiant ML110 G5* server types within DISSECT-CF. These models are designed to reflect identical energy consumption compared to CloudSim. Furthermore, we implemented power transition generators to capture energy consumption by considering multiple states of both PMs and VMs.

Concerning the initial allocation of VMs to PMs, a VM allocation policy was implemented to mirror the behavior of CloudSim. This policy takes into account more realistic measurements, including memory and bandwidth utilization of the PMs, in addition to the current CPU utilization. For the sake of a straightforward comparison between the two simulators, we focused solely on the initial VM allocations without involving any VM migrations. The primary objective was to assess differences in energy consumption rather than delve into the decision-making processes related to VM migration by the VM scheduler.

It is worth mentioning that DISSECT-CF doesn't allow querying energy directly from the PM. Unlike CloudSim, which allows direct queries for current power consumption of hosts, each PM may be associated with an energy meter object that consistently monitors its usage. DISSECT-CF aligns more closely with real data centers, where energy consumption is typically collected using monitoring tools and sensors.

3.6.3 Virtual Machine Setup

The configurations of virtual machines must be identical across simulators to ensure fair comparisons in the realism scoring framework. Resource allocation, performance, and energy consumption, are directly influenced by VM

characteristics. Thus, discrepancies in VM configurations can introduce biases that are affecting realism evaluations. Parameters like virtual CPU (vCPU), memory size and type, bandwidth limit, scheduling policies, and VM placement strategies, should be uniform.

VMs in CloudSim have the following specifications: ID, Million Instructions Per Second (MIPS), image size, bandwidth, number of cores, and task scheduler. CloudSim has separate classes for VM and the task to be run. To launch a task in the VM, one can create two separate objects for the VM (*Vm*) and for the task (*Cloudlet*) and then it is the responsibility for *Broker* to submit task to VM. VMs in DISSECT-CF can be created by invoking *requestVM* method in the *PhysicalMachine* object. *requestVM* requires a *VirtualAppliance* object representing the functional virtual machine images in the system, as well as a resource constraints representing the amount of resources the VM uses compared to the hosting PM.

To launch a task in DISSECT-CF, we invoked the *newComputeTask* method and specified the task length, processing limit, and *ConsumptionEventHandler*. Once task is assigned to VM, it's not possible for the VM to change its utilization. To have a VM with varying levels, we made sure that the task finishes in specific period of time so we can launch another task with different utilization.

In order to have the same behavior for the VM in DISSECT-CF, we have created a VM and control the total number of instructions to be executed considering the processing capability (for instance, a task with 5 million instructions would take 5 seconds to finish if it runs on a VM with 1 million instruction/ second power) so that VM can run for specific period of time with identical utilization percentage as if it was running in CloudSim simulator.

3.7 Experiments

To examine the realism of CloudSim and DISSECT-CF simulators, an experimental system is designed for enabling clear evaluation under consistent settings. During the simulation, we turned off any simulator specific features (e.g., VM migration, energy saving mechanism) except if they are applied in both simulators. This is useful to ensure that differences observed are stemmed from core simulator logic and not from different configurations.

3.7.1 Data Acquisition and Simulation Setup

A fixed input structure was developed to make sure both simulators obtain the same dataset. A 24-hour time span specifying time intervals, CPU utilization values, and real measurements of energy consumption was specified. these measurements were collected using raw data from several sources, all precisely matched to the server hardware described earlier. a five-minute average measurement intervals were taken(i.e., 288 data points per simulator) to provide a side-by-side comparison of the simulators.

The data is extracted from multiple reputable sources while ensuring alignments with Intel Xeon E5-2650 processor. Idle, low, and some of medium utilization data with their respective energy readings were extracted from the dataset in [142], trimmed to timeframes where CPU usage stayed within idle (0-2%) and low (3-30%) thresholds. Mid(30-60%), high(60-90%), and stress(90-100%) utilization data were obtained from published real-world measurements running several kind of applications (e.g., Agisoft Metashape[iv], Blender[v], and MATLAB) on an Intel Xeon processors.

We implemented a structured data extraction process to align the real-world dataset samples with our workload profile. First, we applied a running average across the entire dataset to obtain a representative baseline of utilization levels. Next, we identified intervals in the dataset where utilization levels match those in our workload profiles, to reflect real-world behavior. Following the selection of appropriate windows, we performed a normal average calculations for the extracted samples to ensure that the final 288 data points, to be loaded into simulation, preserve the original utilization trends for each segment in the real dataset.

After selecting the required samples, we combined them to produce a composite workload, preserving the natural variations and trends of real-world workloads. Both simulators were instructed to generate CPU utilization and power consumption values to a csv file every five minutes while ensuring identical timestamps between real and simulated measurements for fare comparison.

[iv] https://www.agisoft.com/
[v] https://www.blender.org/

3.7.2 Realism Score Calculation

Once the real and simulated datasets were aligned, we measured the absolute error between each real data point and its matching simulation output by using Excel formulas. We then calculated the MAPE to determine the overall deviation for both CPU utilization and energy consumption values using equation 3.1. We continued with our realism score calculation, including both CPU and energy measurements, as discussed in section 3.5.

Our final realism score for CloudSim and DISSECT-CF simulators is presented in table 3.3. From the table, we can see that DISSECT-CF better matches actual energy data. It yields 19.89% error for $MAPE_p$ compared to CloudSim which has 21.56%. Possible interpretation that the default mechanism of DISSECT-CF's internal power modeling analyze real consumption patterns a bit more precisely than CloudSim's one, more likely in responding to transitions betweeen idle and all other active states, power measurement for the first five minutes, or even in handling spikes of energy measurements.

While CloudSim shows slightly more accurate representation of CPU utilization percentage error($MAPE_u$) of 0.05% compared to DISSECT-CF (1.34%), both reflect delicate variations in CPU and energy precisions as their overall realism scores adhere closely. Nevertheless, the error margin indicates that these tools, although practical for academic studies, might need adjustments for a more accurate energy or resource scheduling at scale. Figure 3.2 shows a complete energy consumption aquired from running the same workload on PM in both simulators for 24-hours period, compared to the real power readings.

Table 3.3: Realism score for DISSECT-CF and CloudSim simulators

Simulator	$MAPE_p$	$MAPE_u$	R_{Score}
$DISSECT-CF$	19.89	1.34	83.82
$CloudSim$	21.56	0.05	82.74

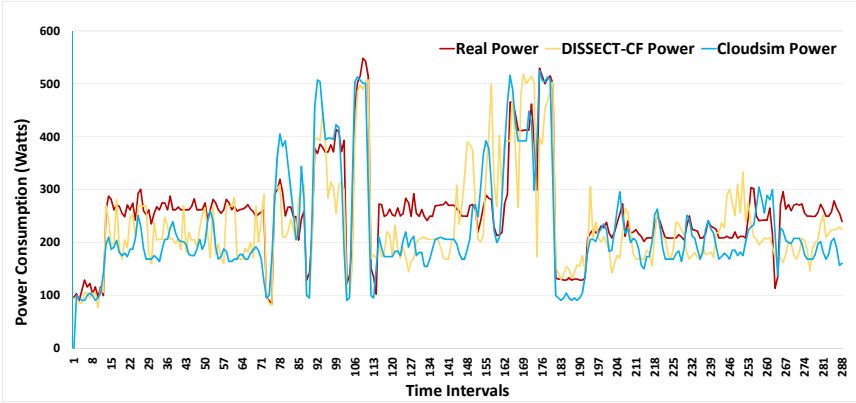

Figure 3.2: Real vs. simulated power consumption

3.7.3 Insights from Complete Planetlab Experiment

To further explore simulation realism and to reinforce insights derived from our evaluation, additional study using the complete real-world workload trace of Planetlab experiment is conducted. Although the main analysis centered on a single PM for comparability, our goal is to investigate the behavior of both simulation tools in a broader and more authentic workload circumstances.

By applying the full Planetlab workload in DISSECT-CF, we aim to study broader patterns and to detect any simulator-specific actions, such as task scheduling, resource overloading, and energy consumption patterns, that further boost precision and scalability. In addition, we provide our insights on some ciritical behaviors found while implementing experiment.

Table 3.4 shows the parameters used for evaluating Planetlab experiment. VMs are initially created with requested resources adhering to the nominal values specified in the workload. In CloudSim, these values govern the initial allocation of VMs to PMs. However, following the initial re-optimization phase, CPU values undergo modification in alignment with the Planetlab trace, resulting in considerably reduced CPU loads. Additionally, memory utilization values for VMs are set to zero during this phase. This implies that, from the first re-optimization onward, VMs occupy significantly fewer CPU and memory resources compared to their initial placement.

Table 3.4: Parameters of Planetlab experiment

PM Types	#CPUs for each PM	CPU (MIPS) for PMs	RAM (MB) for PMs	Power Model for PMs
2	2, 2	1860, 2660	4000, 4000	HP ProLiant G4,G5

VM Types	#CPUs for each VM	CPU (MIPS) for VMs	RAM (MB) for VMs
4	1,1,1,1	2500, 2000, 1000, 500	870, 1740, 1740, 613

Figure 3.3 shows the power consumption characteristics of *HP Proliant G5 and G4* servers for different CPU load intervals. For the Planetlab experiment in CloudSim, The curve of power efficiency for G5 servers demonstrates a more favorable trend from 0% to 30%. Compared to the range between 30% and 60%, this may result in suboptimal decisions by the algorithm.

To illustrate, in a scenario where the algorithm must choose between the two servers, one with CPU load of 20% and the other with 60% load, to host a VM with CPU load corresponding to 10% of PMs' capacity, the algorithm might favor the PM with a CPU load of 20% due to its superior power efficiency. However, this contradicts the VM consolidation concept, which advocates for consolidating as much load as possible onto well-utilized PMs while attempting to free up lightly loaded ones.

Figure 3.4 depicts a comparison of energy consumption between CloudSim and DISSECT-CF utilizing an identical infrastructure setup, we run the simulation using Planetlab workload for 10 days. The increased energy consumption in DISSECT-CF is attributed to the consideration of additional factors that impact energy measurements in real-life scenarios. This includes the potential positive power draw of switched-off PMs, as well as the energy consumption associated with switching PMs on and off, all of which CloudSim neglects.

In addition, the initial assessment of power demand in CloudSim occurs at the start, when the VMs have zero CPU load, resulting in the assumption that each PM's power consumption is also zero at this point. Consequently, the energy consumption of the data center during the first 5 minutes is inaccurately recorded as 0. In contrast, DISSECT-CF accurately estimates energy consumption from the beginning, leading to higher recorded energy values over the first five minutes. It is noteworthy that to ensure a fair comparison between the two simulators, we configured the VM allocation so that both have an identical VM-to-PM allocation map.

Although we cannot definitively determine which simulator performs better

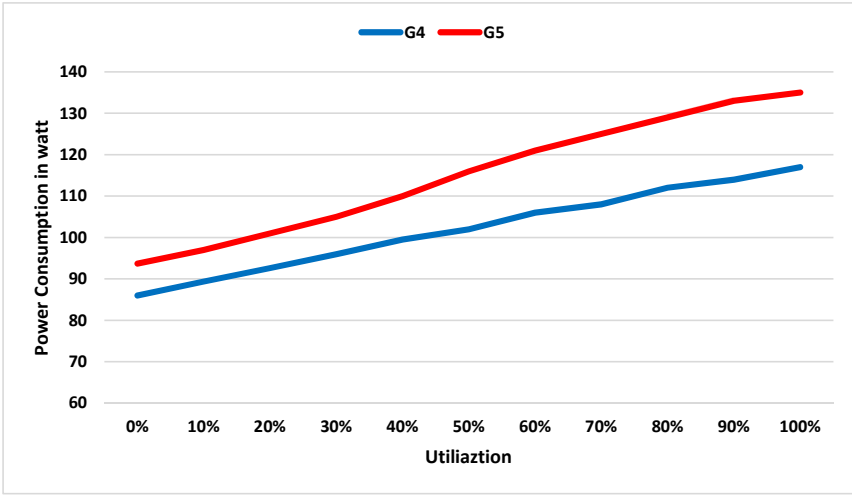

Figure 3.3: Power usage in watt of different servers at different utilization levels [1]

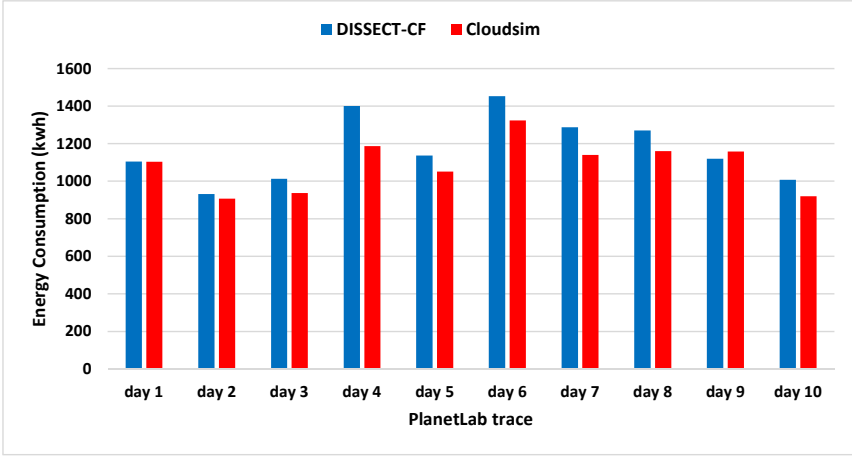

Figure 3.4: Energy consumption using Planetlab workload

in terms of realism due to the lack of real energy measurements for the entire Planetlab experiment, observations from the single PM analysis in 3.7.2 provide additional evidence suggesting that DISSECT-CF produces more realistic energy measurements. Finally, DISSECT-CF exhibited significantly faster simulation times compared to CloudSim. This is helpful for conducting very large scale simulation experiments.

3.8 Summary

The prevalence and complexity of cloud computing systems emphasize the value of simulation tools for designing, tuning, and evaluating system performance. Considering the limitations in forecasting energy usage in large-scale systems, it is vital for simulators to mitigate this complexity. In this chapter, we defined realism in cloud simulation, with a particular focus on power measurements and the reflection of resource utilization between real and simulated environments.

Next, we introduced a realism scoring framework for assessing cloud simulation tools, emphasizing the value of matching simulated results to actual data behaviors. We presented a five-level workload profile for evaluating simulation tools, and encorporated a clear data collection procedure, to guide researchers and developers in obtaining comparable data for score calculations. The profile incorporates various characteristics of real-life workloads, while also allowing the framework to be tailored to diverse workloads and evaluation needs.

A comparison between CloudSim and DISSECT-CF simulators was conducted to evaluate their accuracy in replicating real-world data by assessing their realism scores for CPU utilization and power measurement accuracy. Our analysis showed that, while both emphasized their capability to model real-world scenarios, DISSECT-CF offered a little better precision than CloudSim in simulating power behavior, while CloudSim showed slightly improved accuracy in resource utilization percentages. Furthermore, we discussed our endeavor to port the Planetlab experiment and its evaluation setup from CloudSim to DISSECT-CF aiming at achieving a more realistic simulation.

Having established that simulator selection can offer a dependable starting point to develop more effective solutions, our next focus will be on designing a VM placement algorithm focused on maximizing energy efficiency and resource utilization, while meeting SLA requirements. In cloud data centers, where energy efficiency, resource allocation, and SLA requirements are vital, a reliable simulation environment encourages both researchers and practitioners

to develop methods that optimize resource utilization and energy efficiency within practical limitations.

4

An Energy Efficient and Resource Optimized Virtual Machine Placement Algorithm

4.1 Introduction

In large-scale cloud environments, with numerous high-performance computing devices, energy inefficiency becomes a significant concern due to the underutilization of resources [136]. Virtualization addresses this issue by allowing multiple instances of VMs to be hosted on a single physical server, thereby reducing operational costs, power consumption, and resource wastage [41, 143]. To further improve resource utilization and energy efficiency, VMs are migrated from underutilized servers allowing those servers to be transitioned to a low-power state [144].

As discussed in chapter 2, VM consolidation allows for flexible redistribution of resources across multiple hosts. Nevertheless, maintaining a balance between performance and energy consumption minimization is still a challenge. A variety of existing solutions prioritize consolidation strategies aiming at enhancing power efficiency while ignoring resource utilization and SLA, leading to increased latency and possible financial penalties for cloud service providers [145–147]. Other solutions overlook possible power inefficiencies and potential resource wastage trying to achieve better SLA [148–150].

This research gap urges for more holistic approaches that tackle these challenges by improving resource utilization and energy efficiency while adhering to SLA

requirements.. In this chapter, we present VMP-ER, a novel VM placement algorithm that optimizes (1) energy consumption, (2) resource wastage, and (3) SLA violation within cloud data centers, in the context of a multi-objective bin-packing problem. Unlike other placement algorithms that are focused on the data center's overall power and resources, we consider resource usage in addition to power efficiency factors at the host level. The main idea is to balance resources accross PMs, leading to overall improvements in the data center resources.

In chapter 3, we highlighted that the selection of cloud simulators entails finding a balance among realism, practical implications for accessibility and simplicity, and comparability with current research. We employ CloudSim in the upcoming step to facilitate direct comparisons with existing algorithms that have already been implemented in CloudSim.

This chapter continues with the following: Section 4.2 presents our methodology of implementing VMP-ER algorithm, introducing several models such as power efficiency, resource wastage, and SLA. Section 4.3 demonstrates scenarios and experiments designed to evaluate the performance of our algorithm compared to others. Section 4.4 summarizes the results achieved by our algorithm.

4.2 Methods

Virtual Machine Placement (VMP) aims to minimize the number of physical servers required to host VMs while generating migration maps for live migration. However, excessive migration of VMs during consolidation may degrade application performance and response times. Additionally, efforts to minimize energy consumption through VM consolidation may inadvertently violate the SLA between service providers and customers.

Figure 4.1 gives an example of improved VM placement aiming at reducing the power consumption and resource wastage. In this figure, 9 VMs are placed on 4 different PMs utilizing resources to some extent. It can be seen that the PMs' resources are not utilized properly. One way to improve resource utilization is by migrating VMs from server 1 to server 2 and from server 4 to server 3 so that server 2 and server 3 become more utilized and well balanced in term of resource usages, while servers 1 and 4 are switched off to save energy. All servers in the figure are assumed to be Lenovo ThinkSystem SD535 V3, where power consumption at different utilization levels is stated in Table 4.1.

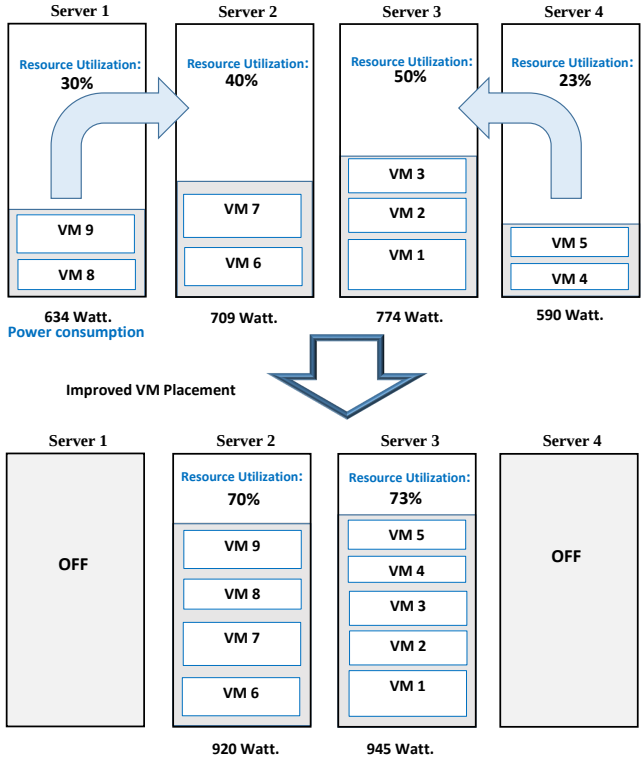

Figure 4.1: Example of improved VM placement

The resource wastage factor represents the amount of allocated resources that are left idle, which negatively impacts resource effectiveness in cloud infrastructure. Poor VM placement, underutilized physical machines, and unbalanced resource allocation can escalate resource wastage. Incorporating the resource wastage factor into the energy-efficiency calculation aids in alleviating potential VM migrations caused by resource shortages on hosting servers. Additionally, it facilitates the efficient utilization of a PM's resources and fosters a balance among remaining resources across multiple dimensions.

When two VMs operate on the same server, the server's CPU and memory utilization are approximated by adding up the individual CPU and memory utilizations of the VMs. For instance, if one VM requests 30% CPU and 40% memory, and another VM requests 20% CPU and 20% memory, the combined utilization of the server hosting both VMs would be estimated at 50% CPU and 60% memory, respectively, reflecting the summation of the utilization vectors. To prevent the server's CPU and memory usage from reaching full

Table 4.1: Power usage in watts of different servers at different utilization levels

Utilization (%)	ThinkSystem SR850 V3	ThinkSystem SD535 V3
0	375	222
10	580	449
20	670	552
30	758	634
40	848	709
50	947	773
60	1058	842
70	1164	920
80	1436	999
90	1879	1199
100	1916	1321

capacity (100%), it becomes necessary to set an upper limit on the resource utilization per server, which is denoted by a threshold value. This precautionary measure is rooted in the understanding that operating at full capacity can lead to significant performance deterioration and also because VM live migration consumes some CPU processing resources on the migrating node.

We model the power consumption, SLA and resource wastage of a data center in the following subsections. Following the earlier works of [49, 115], we exploit the resource wastage factor to devise the resource usage at the PM level. Table 4.2 includes the list of notations used in this work for better readability.

4.2.1 Power Consumption Modeling

The power consumption of PMs is described as a scale of CPU utilization with linear relation. So, a server's power consumption is a function of its CPU utilization, as shown below:

$$\mathbb{P}(h) = \begin{cases} \mathbb{P}^{idle}(h) + (\mathbb{P}^{max}(h) - \mathbb{P}^{idle}(h)) \times \mathbb{U}^{cpu}(h), & if\ \mathbb{U}^{cpu}(h) > 0, \\ 0, & \text{otherwise}, \end{cases} \tag{4.1}$$

where $\mathbb{P}^{idle}(h)$ and $\mathbb{P}^{max}(h)$ are the power usage of host h when it is within an idle and 100% utilization state. $\mathbb{U}^{cpu}(h)$ denotes the normalized CPU

Table 4.2: List of notations used in this chapter

Notation	Description
$\mathbb{H}$	The set of servers or hosts
$\mathbb{A}(\mathbb{H})$	The set of active hosts, where $\mathbb{A} \subset \mathbb{H}$
$\mathbb{V}$	The set of VMs
$\mathbb{Z}$	The set of (VM,PM) pairs
$\mathbb{M}()$	Processing power in MIPS
h^{ch}	The chosen server for hosting
$\mathbb{R}^q(v)$	Requested resources for VM v
$\mathbb{R}^f(h)$	Available/free resources of host h
$\mathbb{P}^e(h)$	Power efficiency of host h
$\mathbb{E}^e(h)$	Energy efficiency for host h
$\mathbb{P}(h)$	Current power consumption of host h
$\mathbb{P}'(h,\, v)$	Expected power consumption of host h after hosting VM v
$\mathbb{S}(h)$	The current SLA violation of a host h
$\mathbb{S}'(h,\, v)$	The expected SLA violation of host h after hosting VM v
$\mathbb{R}^w(h)$	Resource wastage factor of host h
$\mathcal{U}^{cpu}(h)$	Normalized CPU usage of host h
$\mathcal{U}^{ram}(h)$	Normalized RAM usage of host h
$\mathcal{F}^{cpu}(h)$	Normalized free/remaining CPU of host h
$\mathcal{F}^{ram}(h)$	Normalized free/remaining RAM of host h
$\mathbb{P}^{max}(h)$	Maximum power for a host h when it is 100% utilized
$\mathbb{P}^{idle}(h)$	Power of host h when it is idle
$\mathbb{P}^{dif}(h, v)$	The increase in power consumption of host h after hosting VM v
$\mathbb{S}^{dif}(h, v)$	The difference between $\mathbb{S}'(h,\, v)$ and $\mathbb{S}(h)$
$\mathbb{F}(h, v)$	Denotes $\mathbb{P}^{dif}(h, v)$ multiplied by $\mathbb{S}^{dif}(h, v)$
$V_h(t)$	SLA violation of host h at time t
$A^T(h)$	Total time in which host h is active

utilization. The overall power draw for the data center will be the summation of all the PMs in the data center.

4.2.2 Power Efficiency Modeling

A high-performance server consuming excessive power may be outperformed by a mid-range server delivering better computational power per watt. To improve efficiency evaluation, we introduced a power efficiency metric to determine the amount of computing power that servers provide per unit of their peak power

consumption. Our algorithm leverages this metric to assign VMs to servers that offer the highest performance while minimizing power wastage.

If we consider a physical machine h with a CPU capacity of $mips(h)$ (expressed in Million Instructions Per Second, or MIPS) and a maximum power consumption of $\mathbb{P}^{max}(h)$ (measured in watts) reflecting the overall power draw from all the PM dimensions (CPU, RAM, network, and disk storage), we can define the power efficiency of h as follows:

$$\mathbb{P}^{e}(h) = \frac{mips(h)}{\mathbb{P}^{max}(h)}. \tag{4.2}$$

$\mathbb{P}^{e}(h)$ is designed to assess PM efficiency based on its peak power consumption. This approach ensures that VMs are assigned to PMs that deliver the highest processing power per watt of energy consumed. For instance, the machine in our running example from table 4.1, Lenovo ThinkSystem SD535 V3, boasts a computational power of 288,000 MIPS and a maximum power consumption of 1321 watts, yielding a power efficiency score of 218.02. Conversely, Lenovo ThinkSystem SR850 V3 offers a computational power of 456,000 MIPS and a maximum power consumption of 1916 watts, resulting in a power efficiency score of 237.99. Consequently, Lenovo SR850 V3 demonstrates superior power efficiency compared to ThinkSystem SD535.

4.2.3 Resource Wastage Modeling

When VMs are deployed on servers, some allocated resources may remain unused, leading to resource wastage. The available remaining resources on individual servers can significantly differ across various VM placement strategies. To effectively harness multidimensional resources, the subsequent equation is employed to quantify the potential expense incurred from the unused resources of a single server:

$$\mathbb{R}^{w}(h) = \frac{|\mathcal{F}^{cpu}(h) - \mathcal{F}^{ram}(h)| + \varepsilon}{\mathcal{U}^{cpu}(h) + \mathcal{U}^{ram}(h)} \tag{4.3}$$

where $\mathcal{U}^{cpu}(h)$ and $\mathcal{U}^{ram}(h)$ are the normalized used CPU and RAM, while $\mathcal{F}^{cpu}(h)$ and $\mathcal{F}^{ram}(h)$ indicate the normalized remaining CPU and RAM. The idea of having ε is that resource wastage cannot be entirely eliminated even when hosts are fully utilized, as there is always some degree of resource waste.

Additionally, it helps to avoid zero values when both CPU and RAM have identical normalized utilization rates. ε can be set to any small positive number like 0.0001 for instance.

The equation identifies cases where resource utilization can be improved across various dimensions and ensures a balanced distribution of remaining resources on each server. This is achieved by prioritizing PMs that give the lowest average resource waste in all considered dimensions when hosting certain VMs. We exploit the $\mathbb{R}^w(h)$ factor to determine the normalized resource usage for each PM where the one with more balanaced resources across all dimensions will have a higher priority to host VMs.

4.2.4 SLA Modeling

Ensuring adherence to SLAs is vital in cloud computing, as it affects system reliability, user satisfaction, and potential financial penalties for service providers. SLA violation indicates situations in which a host cannot fulfill a specific VM's request for a certain amount of resources at a given time. We expressed this with the following equation:

$$\mathbb{S}(h) = \frac{\sum_{t=0}^{T} V_h(t)}{A^T(h)} \tag{4.4}$$

where T denotes the total time required to execute the input workload, and $V_h(t)$ represents the binary value for host h, which is 0 if no violation occurs at that time and 1 otherwise. The value for $A^T(h)$ represents the overall time in which host h was active. A violation occurs when the host experiences 100% CPU utilization.

4.2.5 SLA Conscious Energy Efficiency Model

Improper VM placement and resource management can result in high power consumption, underutilized resources, and frequent SLA violations, degrading overall system performance. To tackle this, our approach determines the feasibility of deploying a VM on a server by considering its influence on power efficiency and potential SLA violation rates. Let $\mathbb{E}^e(h)$ represent our target function that encompasses the energy consumption, resource wastage, and SLA

violations of host h within a data center. As a first step toward our algorithm, we define $\mathbb{P}^{dif}(h, v)$. This denotes a potential power draw improvement of a host if it is also hosting VM v:

$$\mathbb{P}^{dif}(h, v) = \mathbb{P}'(h, v) - \mathbb{P}(h). \tag{4.5}$$

Similarly, we define $\mathbb{S}^{dif}(h, v)$ as the potential SLA violation change of a host when it starts hosting a new VM v:

$$\mathbb{S}^{dif}(h, v) = \mathbb{S}'(h, v) - \mathbb{S}(h). \tag{4.6}$$

We next define the overall improvement of the host as follows:

$$\mathbb{F}(h, v) = \begin{cases} \mathbb{P}^{dif}(h, v), & if\ \mathbb{S}^{dif}(h, v) = 0, \\ \mathbb{S}^{dif}(h, v), & if\ \mathbb{P}^{dif}(h, v) = 0, \\ \mathbb{P}^{dif}(h, v) \times \mathbb{S}^{dif}(h, v), & if\ both\ terms \neq 0, \\ 1, & otherwise. \end{cases} \tag{4.7}$$

We are using only non-zero improvement components, so we can use this in the denominator of our overall target function:

$$\mathbb{E}^{e}(h) = \frac{\mathbb{P}^{e}(h)(1 - \mathbb{R}^{w}(h))}{\mathbb{F}(h, v)}. \tag{4.8}$$

4.2.6 The VMP-ER Algorithm

Now, we are ready to discuss Algorithm 1, which builds on the previously discussed equations. In general, our algorithm, just like any typical VMP, takes a list of PMs and a list of VMs as input, and then it produces a set of VM–PM pairs as output, in which each VM is hosted by certain hosts as guided by our target function.

First, the VMs are sorted in decreasing order according to CPU demands. For each VM in the VM list, the algorithm checks the set of active servers that have enough resources to accommodate the VM request and calculates the energy efficiency for each one based on Equation (4.8). To calculate $\mathbb{E}^{e}$ for

Algorithm 1: VMP-ER Placement Algorithm

Require : $\mathbb{H}$ and $\mathbb{V}$
Ensure : $\mathbb{Z}$: where $\mathbb{Z}$ is $\{(v_1, h_1), (v_2, h_2), (v_3, h_3), ...\}$
1 **forall** $v_i \in \mathbb{V}$ *such that* $\mathbb{M}(v_i) \geq \mathbb{M}(v_{i+1})$ **do**
2 **if** $\mathbb{R}^q(v_i) \leq \mathbb{R}^f(h_j)$ **then**
3 $h^{ch} = \arg\max_{h_j \in \mathbb{A}(\mathbb{H})} \mathbb{E}^e(h_j)$
4 **end**
5 **else**
6 $h^{ch} = \arg\max_{h_k \notin \mathbb{A}(\mathbb{H})} \mathbb{E}^e(h_k)$
7 **end**
8 $\mathbb{Z} \cup (v_i , h^{ch})$
9 **end**
10 **return** $\mathbb{Z}$

each host, the algorithm determines the variation of power consumption before and after hosting VM v using Equation (4.5). Thus, the host with the lowest increase in the power consumption will have a better chance at hosting.

The $\mathbb{P}^e$ factor helps in choosing the server with the highest power efficiency. This means that the host with a higher $\mathbb{P}^e$ is preferred, since it can host more VMs with a lower increase in power consumption compared to other PMs. This leads to minimizing the overall power consumption of the DC by utilizing as few hosts as possible. The potential SLA violation impact before and after VM placement is calculated using equation 4.6.

The resource usage factor of the nominated servers is calculated by considering two dimensions, CPU and RAM, based on Equation (4.3). This factor is important for selecting an appropriate PM in terms of resource utilization across multiple dimensions. Additionally, it aids in balancing the resource wastage of PMs by giving priority to those with higher resource utilization.

The energy-efficiency metric for each server will be obtained based on the power effiency, resource usage, and potential occurrence of SLA, which all contribute to the selection of PMs. The most suitable one will be chosen to host the VM while pairing between VM–PM is updated (line 3). If there is no active server to accommodate the VM, the algorithm selects an appropriate host from the set of inactive servers that has the most energy-efficienct metric (line 6). The chosen host will be associated to the VM (line 8). Finally, when all the VMs are allocated to their hosts, the algorithm returns the final set of VM–PM pairings (line 10).

Let there be n number of VMs in set $\mathbb{V}$ and m number of PMs in set $\mathbb{H}$. The algorithm takes $\mathcal{O}(n \log n)$ to sort VMs in descending order based on resource demands. For each VM, the algorithm checks all active PMs to find a suitable host and then checks all inactive ones in case there is no active host available. All calculations for $\mathbb{E}^e(h)$ take a constant amount of time ($\mathcal{O}(1)$) and can be ignored. Thus, the overall time to sort VMs and to check all hosts takes $\mathcal{O}(n \log n + nm)$. As the number of PMs is smaller than the number of VMs in the cloud system, we can express the complexity for the algorithm as $\mathcal{O}(n \log n + n^2)$. As n increases, the overall complexity can be simplified to $\mathcal{O}(n^2)$.

The fundamental concept of including the $\mathbb{R}^w(h)$ factor in the choice of PM is to enhance the utilization of a physical machine's resources across various dimensions while ensuring a balanced distribution of remaining resources. It also helps with mitigating unnecessary potential migrations due to resource contention, thus increasing the energy optimization for the whole data center.

Since $\mathbb{R}^w(h)$ represents the normalized resource wastage of the host, $(1-\mathbb{R}^w(h))$ reflects the normalized resource usage, which we include in our calculation of the energy-efficiency factor ($\mathbb{E}^e(h)$). This means that the host that has an acceptable amount of resource usage will be prioritized for hosting the VM and thus lead to more utilized hosts while minimizing the number of powered-on servers overall. This indicates a reduction in the total power consumption of the data center caused by decreasing the number of active hosts.

To reduce the data center's power consumption, overloaded and underloaded machines are not considered for hosting. This exclusion occurs when a physical machine's remaining capacity surpasses a specific threshold, either exceeding an upper threshold or falling below a lower one. Consequently, VMs on underloaded PMs will be migrated to other hosts, enabling the shutdown of those machines to save energy.

4.3 Experiments

A modern data center is characterized by its heterogeneity, comprising various generations of PMs with differing configurations, particularly in processor speed and capacity. These servers are typically integrated into the data center incrementally, often replacing older legacy machines within the infrastructure [151]. Thus, we considered a heterogenous data center that contains two types of servers from different generations, as shown in table 4.3.

Table 4.3: Configuration of PMs

PM Type	CPU (MIPS)	RAM (GB)	Bandwidth (Gbps)
SD535 V3	288,000	192	100
SR850 V3	456,000	512	100

Table 4.4: Configuration of VMs

VM Type	CPU (MIPS)	RAM (GB)	Bandwidth (Gbps)
m5.large	3100	8	8
m5.xlarge	6200	16	8
m5.2xlarge	12,400	32	10
m5.4xlarge	24,800	64	12
m5.8xlarge	49,600	128	20

Five types of VMs are used, as shown in table 4.4. Also, we set all VMs to have a single CPU core, and we assumed task independency in which there are no communications required among the VMs during the task execution. To assess the performance of VMP-ER, we utilized various metrics including the number of active PMs, total power draw, and resource waste.

4.3.1 Representative Example

Let us first consider a scenario in which two types of Lenovo servers, SD535 and SR850, are available within a data center for hosting VMs. For simplicity, let us assume that the following resources are currently available in these servers: one active SD535 server has 2604 MIPS and 2800 MB available, while one active SR850 server has 3724 MIPS and 1600 MB available. Now, let v_1, v_2, and v_3 represent VMs awaiting placement within the data center. The resource requirements for each VM are as follows: v_1(1800 MIPS, 1100 MB), v_2(1600 MIPS, 1000 MB), and v_3(1100 MIPS, 520 MB).

Through the analysis of various algorithms, it has been observed that some algorithms prefer SR850 servers, while others favor SD535 servers. For instance, LBMBP [106] and EEVMP [110] algorithms would opt to place v_1 on a SR850 server due to its higher power efficiency factor compared to SD535. Subse-

quently, v_2 would be placed on a SD535 server, since the available resources on the SR850 server are insufficient. For v_3, EEVMP would activate a new physical machine to accommodate it, as neither SD535 nor SR850 has adequate resources.

In contrast, our algorithm takes resource usage into account while identifying the most energy-efficient PM among the active PMs. This involves calculating the $(1 - \mathbb{R}^w(h))$ term for both PMs using Equation (4.3), resulting in values of 0.942 for SD535 and 0.861 for SR850. Subsequently, the power efficiency factor $(\mathbb{P}^e(h))$ is computed using Equation (4.2), yielding values of 218.02 for SD535 and 237.99 for SR850.

The difference in power consumption before and after placing v_1 is then determined, resulting in a difference of 27.2 watts for SD535 and 28.8 watts for SR850. Assuming no SLA violations occur, the E-efficiency factor $(\mathbb{E}^e(h))$ for both servers is calculated to be 7.55 for SD535 and 7.14 for SR850, leading to the selection of SD535 to host v_1. For v_2 and v_3, since only the SR850 server possesses sufficient CPU and RAM resources, both VMs are hosted on the SR850 server without the need to activate another physical machine.

Consequently, our proposed algorithm utilizes fewer PMs, leading to increased energy savings for the data center. Additionally, resource wastage is minimized as PM resources are utilized properly. While the LBMBP algorithm moves VMs to different PMs during the migration phase, it tends to result in higher energy consumption compared to the proposed algorithm. This is primarily because more energy is consumed during migrations. In data centers with diverse PMs and VM instances operating at scale, our algorithm is expected to deliver better performance.

In addition, our proposed algorithm enhances the quality of service of running PMs and minimizes the threat of PM overload by considering both the current and future resource usage status. This approach ensures effective destination PM selection. We consider both the load and dependability of PMs. Thus, the election of safer and balanced PMs for hosting results in properly utilized PMs and hence reduces the likelihood of potential migrations. This reduction significantly improves the performance degradation due to the migration metric. By enhancing the servers' resource usages, the algorithm ensures that fewer PMs will be able to host more VMs. This improvement becomes viable when the number of VMs increase.

For large instances of VMs like Amazon EC2 M5.4xlarge (24,800 MIPS, 64,000 MB) and M5.8xlarge (49,600 MIPS, 128,000 MB), the LBMBP algorithm initially places them on SR850 servers but later migrates them to SD535 servers

to minimize resource wastage. This approach performs better with certain types of workloads where VMs are not fully utilized most of the time (workloads do not request 100% of their demands). However, the likelihood of SLA violations will increase when these VMs operate at full utilization.

Considering the energy consumed by VM migration, our proposed algorithm obtains better results than the LBMBP by achieving better resource balancing and energy optimization on average. This is accomplished by avoiding frequent VM migrations and incorporating the SLA metric into the PM selection process, thereby improving QoS.

4.3.2 Evaluation

In our experiment, we used real data taken from the Planetlab workload trace [101]. Cloudsim was used to perform the experiments. Figure 4.2 indicates the number of PMs required to host varying numbers of VMs. VMP-ER outperforms EEVMP and LBMBP in terms of the total number of active PMs due to its packing efficiency aiming at allocating more VMs to a single PM while taking load balancing into account. In contrast, EEVMP and LBMBP tend to place most VMs on SR850 servers first, as their decisions are based mostly on power efficiency. As the number of VM requests increase, both algorithms start to activate more PMs to accommodate these requests, as previously discussed in this section.

EEVMP requires a higher number of PMs, as shown in the figure. This is primarily because EEVMP focuses on the lowest increase in power consumption when hosting particular VMs without considering resource wastage, often leading to the selection of an idle PM for initial VM placement. It is noteworthy that LBMBP involves two phases in VM placement: an initial placement phase and a replacement phase, where VMs are migrated to different PMs. In our comparison, we only consider the initial placement phase in the evaluation.

Figure 4.3 compares the energy consumption for different numbers of VMs, ranging from 10 to 200, considering different types of VMs, as shown in table 4.4. From the figure, we observe similar performance for a small number of VMs. However, our algorithm begins to outperform EEVMP and LBMBP as the number of VMs increase. Since VMP-ER requires fewer active PMs to accommodate VM requests, it results in lower energy consumption. Additionally, balancing the resource placement of VMs leads to more efficient utilization and

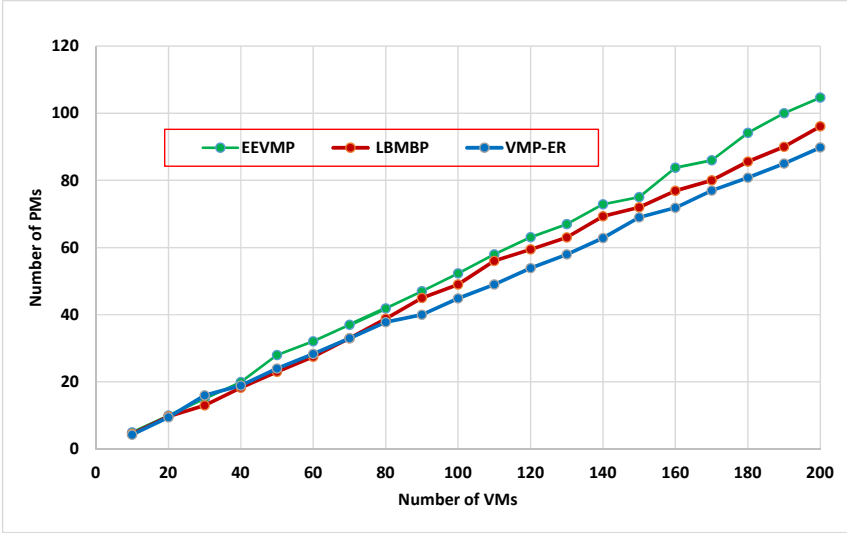

Figure 4.2: The number of PMs required to host a given number of VMs

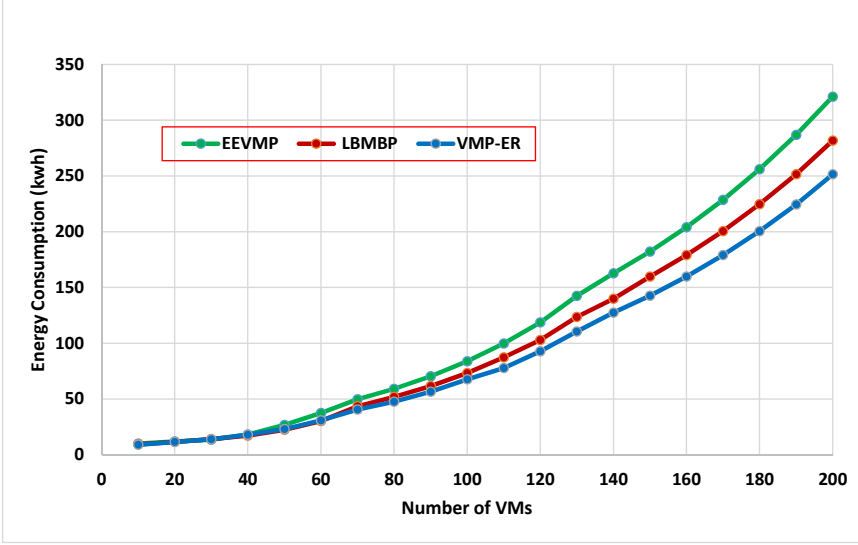

Figure 4.3: Energy consumption for a given number of VMs

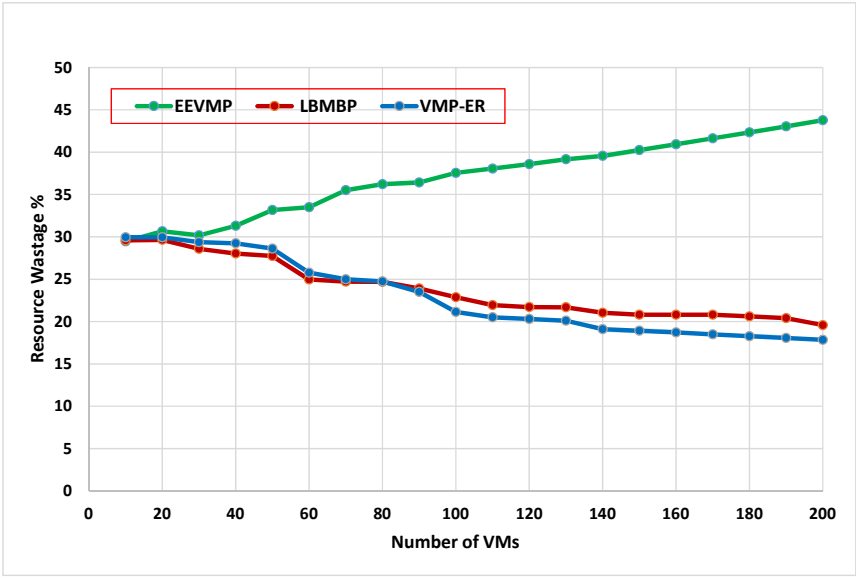

Figure 4.4: Average resource wastage for a given number of VMs

avoids unnecessary potential migrations, positively impacting the overall data center energy consumption.

In contrast, LBMBP tends to migrate VMs in its second phase to achieve better placement, which consequently results in additional energy consumption due to migration. EEVMP shows higher energy consumption as the number of VMs increase because the algorithm primarily focuses on CPU resources when allocating VMs, neglecting other resources like RAM. This results in an increased number of migrations for workloads where VMs request a higher amount of RAM, thereby affecting energy consumption. Figure 4.4 shows the percentage of average resource wastage (CPU and RAM) when hosting different numbers of VMs. A reduction in the number of active hosts indicates an improvement in the average utilization of resources. For a small number of VMs, LBMBP performs better than both EEVMP and VMP-ER. Comparing the performance of our algorithm to LBMBP, both algorithms utilize resources effectively with minimal resource waste.

However, as the number of VMs increase, our algorithm starts to yield better results. This is because VMP-ER maintains higher resource utilization by incorporating the resource usage factor at the host level, leading to more efficient host selection and overall energy saving in the data center. Finally, VMP-

ER achieves a slightly lower average SLA violation rate of 10.14% compared to EEVMP and LBMBP, which have violation rates of 11.49% and 13.33%, respectively.

4.4 Summary

Given the diverse characteristics of PMs and VMs, along with the multidimensional resources and large-scale infrastructure of the cloud environment, virtual machine placement has emerged as a significant area of research. This chapter introduces an efficient algorithm aimed at minimizing power draw and resource waste.

The proposed algorithm achieves power consumption reduction by decreasing the number of running PMs. It focuses on merging VMs onto the least possible number of PMs, minimizing the total energy consumption of the data center. Moreover, the algorithm adopts an energy-awareness strategy by prioritizing more efficient PMs in placement decisions. Decreasing the number of running PMs and targeting energy-efficient hardware deliver dramatic decreases in energy consumption.

Complementing power optimization, VMP-ER achieved notable improvements in resource efficiency and SLA compliance compared to the other tested algorithms. By effectively balancing resource optimization, energy efficiency, and SLA adherence, VMP-ER serves as a reliable and efficient strategy for optimizing cloud resource management.

5

Conclusion

5.1 Summary

This dissertation focused on improving resource utilization and energy efficiency in cloud data centers, while also ensuring compliance with service level agreements. To accomplish this, the study analyzed virtualization technologies, VM consolidation strategies, and cloud simulation techniques to devise a more efficient and realistic strategy for cloud resource management. This dissertation contributes to the field by addressing key issues regarding the realism of cloud simulations, VM placement optimization, and energy-aware scheduling.

The research started with an introduction to the problem space, followed by a comprehensive background on cloud computing, detailing its architecture, key enabling technologies, and approaches to resource management. Particular attention was given to virtualization and VM consolidation, both of which are essential for optimizing cloud operations. Moreover, the chapter presented cloud simulation as a key tool for evaluating resource management strategies, underlining the significance of realism in the simulations. A review of current cloud simulators and comparative studies identified the challanges in accurately modeling energy consumption and resource utilization in virtualized environments.

Chapter 3 built on this foundation by introducing a novel realism scoring framework designed to evaluate and improve accuracy of cloud simulations. Using Mean Absolute Percentage Error, the framework analyzed differences between simulated CPU utilization and energy consumption with real-world measurements. The research employed controlled experiments with identical setups across simulators, highlighting considerable discrepancies in how they

model energy consumption and resource allocation. These findings underlined the importance of improving internal modeling in cloud simulators to produce research outcomes that are both reliable and applicable.

Chapter 4 furthered the research by proposing a VM placement algorithm aimed at optimizing resource utilization and energy consumption in cloud data centers. The algorithm, incorporating dynamic resource allocation techniques and SLA constraints, demonstrated better performance compared to existing VM placement strategies. Extensive simulations validated the algorithm's effectiveness, leveraging insights from the realism scoring framework introduced in chapter 3. Results also demonstrated that the proposed approach significantly increased energy efficiency without sacrificing application performance or adherence to SLA requirements.

Overall, this dissertation contributes to the field of cloud computing and simulation research by filling two essential gaps: the need for standardized approaches to validate cloud simulators accuracy and the need for more effective VM scheduling strategies to enhance energy efficiency. The findings offer valuable insights for researchers and practitioners seeking to optimize cloud resource management while ensuring precise simulation outcomes.

5.2 Contributions to Science

This dissertation contributes to the field of cloud computing research with the following two theses:

Thesis 1: *I proposed a scoring framework to evaluate the realism of cloud simulators in terms of their accuracy in energy consumption and resource utilization. This framework establishes a systematic strategy to determine the fidelity of simulation results in contrast to real-world cloud behavior. A unified evaluation equation is defined to mesaure simulators realism, exploiting the overall deviation of simulated results to real world measurements. A standardized workload configuration, encompassing 5 utilization levels and incorporating various transitions to reflect the dynamic behavior of cloud environments, is introduced to enable fair assessment across different simulators. The study showed that simulators differ in their ability to replicate realistic cloud behavior, offering a basis for researchers to analyze and improve cloud simulation accuracy, and paving the way for more realistic and effective simulations for cloud environments.* [P2, P5]

Thesis 2: *I introduced VMP-ER (virtual machine placement for energy and resource optimization), an algorithm to enhance energy efficiency and optimize resource utilization in cloud data centers. The algorithm was designed to reduce energy usage and prevent resource waste while upholding SLA commitments. The algorithm uses PM level calculations when placing VMS, ensuring a balance between different resources across PMs. Experimental results showed that VMP-ER demonstrated better performance, compared to other strategies, in terms of energy efficiency and resource utilization by minimizing the number of active physical machines.* [P1, P3, P4]

5.2.1 Author's Publications During Research

(P1) Rjeib Hasanein, Kecskemeti Gabor. "VMP-ER: An Efficient Virtual Machine Placement Algorithm for Energy and Resources Optimization in Cloud Data Center". Algorithms. 2024 Jul 5;17(7):295. (Scopus Index Q2).

(P2) Rjeib Hasanein, Kecskemeti Gabor. "An investigation on implementing a scenario on different cloud simulators". MULTIDISZCIPLINÁRIS TUDOMÁNYOK: A MISKOLCI EGYETEM KÖZLEMÉNYE. 2022; 12(3): 256-63.

(P3) Rjeib Hasanein, Kecskemeti Gabor. "Energy-Aware VM Consolidation in Cloud Datacenter". In: Iványi, Péter (eds.) Abstract book for the 17th MIKLÓS IVÁNYI INTERNATIONAL PHD & DLA SYMPOSIUM : ARCHITECTURAL, ENGINEERING AND INFORMATION SCIENCES. Pécs, Hungary : Pollack Press (2021) 227 p. p. 107. ISBN: 9789634298113.

(P4) Rjeib Hasanein, Kecskemeti Gabor. "Energy-Aware VM migration in Fog computing: A literature Review". In: Molnár, Dániel; Molnár, Dóra (eds.) XXIV. Tavaszi Szél Konferencia 2021: Absztrakt kötet Bp, Hungary : Association of Hungarian PHD and DLA Students (2021) 667 p. pp. 404-404. Paper: 532 , 1 p. ISBN: 9786155586996. Scientific.

(P5) Rjeib Hasanein, Kecskemeti Gabor. "Realism in Cloud Simulation: A Scoring Framework". Simulation. (Scopus Index Q2). (Under Review).

5.2.2 Other Publications

(P5) Nsaif M, Kovásznai G, Rjeib Hasanein, Malik A, de Fréin R. "Evaluating RNN Models for Multi-Step Traffic Matrix Prediction". In 2024 IEEE 3rd Conference on Information Technology and Data Science (CITDS) 2024 Aug 26 (pp. 1-6). IEEE.

(P6) Ali NS, Alhilali AH, Rjeib Hasanein, Alsharqi H, Al-Sadawi B. "Automated attendance management systems: systematic literature review". International Journal of Technology Enhanced Learning. 2022;14(1):37-65. (Scopus Index Q3).

5.3 Future Research Directions

We have identified several future research that could merit further exploration. First, our current realism scoring framework introduced in chapter 3 focuses on assessing CPU utilization and power consumption accuracy. We aim to extend the framework by incorporating additional hardware resources (e.g. memory utilization, disk I/O, and network usage), thermal data, and cooling strategies. A more extensive collection of resource metrics enables researchers to enhance the precision of cloud simulations and to improve alignment with real-world cloud systems.

Second, we plan to leverage machine learning and artificial intelligence techniques to improve efficiency in VM deployment and resource utilization by dynamically learn and adjust placement strategies using historical workload and adapting to current system conditions. Finally, we aspire to explore mutli-level consolidation strategies (containers within VMs across multiple hosts), and investigate whether it provides additional energy savings or it reduces the effectiveness of consolidation due to the overhead of managing and migrating containers.

Bibliography

[1] Zoltán Ádám Mann. Cloud simulators in the implementation and evaluation of virtual machine placement algorithms. *Software: Practice and Experience*, 48(7):1368–1389, 2018.

[2] Avita Katal, Susheela Dahiya, and Tanupriya Choudhury. Energy efficiency in cloud computing data centers: a survey on software technologies. *Cluster Computing*, 26(3):1845–1875, 2023.

[3] Mohammad Masdari and Mehran Zangakani. Green cloud computing using proactive virtual machine placement: challenges and issues. *Journal of Grid Computing*, 18(4):727–759, 2020.

[4] Piotr Nawrocki and Mateusz Smendowski. A survey of cloud resource consumption optimization methods. *Journal of Grid Computing*, 23(1):5, 2025.

[5] Nisha Chaurasia, Mohit Kumar, Rashmi Chaudhry, and Om Prakash Verma. Comprehensive survey on energy-aware server consolidation techniques in cloud computing. *The Journal of Supercomputing*, 77:11682–11737, 2021.

[6] Tahseen Khan, Wenhong Tian, Guangyao Zhou, Shashikant Ilager, Mingming Gong, and Rajkumar Buyya. Machine learning (ml)-centric resource management in cloud computing: A review and future directions. *Journal of Network and Computer Applications*, 204:103405, 2022.

[7] Azlan Ismail. Energy-driven cloud simulation: existing surveys, simulation supports, impacts and challenges. *Cluster Computing*, 23(4):3039–3055, 2020.

[8] Deepika Saxena and Ashutosh Kumar Singh. Workload forecasting and resource management models based on machine learning for cloud computing environments. *arXiv preprint arXiv:2106.15112*, 2021.

[9] Yousef Sanjalawe et al. Cloud computing simulators: A review. In *2023 24th International Arab Conference on Information Technology (ACIT)*, pages 1–14. IEEE, 2023.

[10] Sukhpal Singh Gill, Huaming Wu, Panos Patros, Carlo Ottaviani, Priyansh Arora, Victor Casamayor Pujol, David Haunschild, Ajith Kumar Parlikad, Oktay Cetinkaya, Hanan Lutfiyya, et al. Modern computing: Vision and challenges. *Telematics and Informatics Reports*, page 100116, 2024.

[11] Alexander Benlian, William J Kettinger, Ali Sunyaev, Till J Winkler, and Guest Editors. The transformative value of cloud computing: a decoupling, platformization, and recombination theoretical framework. *Journal of management information systems*, 35(3):719–739, 2018.

[12] Prateek Kumar Soni and Harshita Dhurwe. Challenges and open issues in cloud computing services. In *Advanced Computing Techniques for Optimization in Cloud*, pages 19–37. Chapman and Hall/CRC, 2024.

[13] Mohammed Joda Usman, Abdul Samad Ismail, Gaddafi Abdul-Salaam, Hassan Chizari, Omprakash Kaiwartya, Abdulsalam Yau Gital, Muhammed Abdullahi, Ahmed Aliyu, and Salihu Idi Dishing. Energy-efficient nature-inspired techniques in cloud computing datacenters. *Telecommunication Systems*, 71:275–302, 2019.

[14] Junzhong Zou, Kai Wang, Keke Zhang, and Murizah Kassim. Perspective of virtual machine consolidation in cloud computing: a systematic survey. *Telecommunication Systems*, pages 1–29, 2024.

[15] Salil Bharany, Sandeep Sharma, Osamah Ibrahim Khalaf, Ghaida Muttashar Abdulsahib, Abeer S Al Humaimeedy, Theyazn HH Aldhyani, Mashael Maashi, and Hasan Alkahtani. A systematic survey on energy-efficient techniques in sustainable cloud computing. *Sustainability*, 14(10):6256, 2022.

[16] Mohammad Masdari and Afsane Khoshnevis. A survey and classification of the workload forecasting methods in cloud computing. *Cluster Computing*, 23(4):2399–2424, 2020.

[17] Najme Mansouri, R Ghafari, and B Mohammad Hasani Zade. Cloud computing simulators: A comprehensive review. *Simulation Modelling Practice and Theory*, 104:102144, 2020.

[18] Ali Sunyaev and Ali Sunyaev. Cloud computing. *Internet computing: Principles of distributed systems and emerging internet-based technologies*, pages 195–236, 2020.

[19] Tharam Dillon, Chen Wu, and Elizabeth Chang. Cloud computing: issues and challenges. In *2010 24th IEEE international conference on advanced information networking and applications*, pages 27–33. Ieee, 2010.

[20] Sulav Malla and Ken Christensen. Hpc in the cloud: Performance comparison of function as a service (faas) vs infrastructure as a service (iaas). *Internet Technology Letters*, 3(1):e137, 2020.

[21] Mario Santana. Infrastructure as a service (iaas). In *Cloud Computing Security*, pages 65–70. CRC Press, 2020.

[22] Prateek Mathur. Cloud computing infrastructure, platforms, and software for scientific research. *High Performance Computing in Biomimetics: Modeling, Architecture and Applications*, pages 89–127, 2024.

[23] Keke Gai and Annette Steenkamp. A feasibility study of platform-as-a-service using cloud computing for a global service organization. *Journal of Information Systems Applied Research*, 7(3):28, 2014.

[24] Deepali Bajaj, Urmil Bharti, Anita Goel, and SC Gupta. Paas providers and their offerings. *International Journal of Scientific & Technology Research*, 9(2):4009–4015, 2020.

[25] WeiTek Tsai, XiaoYing Bai, and Yu Huang. Software-as-a-service (saas): perspectives and challenges. *Science China Information Sciences*, 57:1–15, 2014.

[26] Nancy Jain and Sakshi Choudhary. Overview of virtualization in cloud computing. In *2016 Symposium on Colossal Data Analysis and Networking (CDAN)*, pages 1–4. IEEE, 2016.

[27] Amarildo Rista, Jaumin Ajdari, and Xhemal Zenuni. Cloud computing virtualization: a comprehensive survey. In *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*, pages 462–472. IEEE, 2020.

[28] Aditya Bhardwaj and C Rama Krishna. Virtualization in cloud computing: Moving from hypervisor to containerization—a survey. *Arabian Journal for Science and Engineering*, 46(9):8585–8601, 2021.

[29] Ankita Desai, Rachana Oza, Pratik Sharma, and Bhautik Patel. Hypervisor: A survey on concepts and taxonomy. *International Journal of Innovative Technology and Exploring Engineering*, 2(3):222–225, 2013.

[30] K Gouda, Anurag Patro, Dines Dwivedi, and Nagaraj Bhat. Virtualization approaches in cloud computing. *International Journal of Computer Trends and Technology (IJCTT)*, 12(4):161–166, 2014.

[31] Chao-Chun Chen, Min-Hsiung Hung, Kuan-Chou Lai, and Yu-Chuan Lin. Docker and kubernetes. *Industry 4.1: Intelligent Manufacturing with Zero Defects*, pages 169–213, 2021.

[32] Aaqib Rashid and Amit Chaturvedi. Virtualization and its role in cloud computing environment. *International Journal of Computer Sciences and Engineering*, 7(4):1131–1136, 2019.

[33] Yuping Xing and Yongzhao Zhan. Virtualization and cloud computing. In *Future Wireless Networks and Information Systems: Volume 1*, pages 305–312. Springer, 2012.

[34] Alexandre HT Dias, Luiz HA Correia, and Neumar Malheiros. A systematic literature review on virtual machine consolidation. *ACM Computing Surveys (CSUR)*, 54(8):1–38, 2021.

[35] Leila Helali and Mohamed Nazih Omri. A survey of data center consolidation in cloud computing systems. *Computer Science Review*, 39:100366, 2021.

[36] Jaspreet Singh and Navpreet Kaur Walia. A comprehensive review of cloud computing virtual machine consolidation. *IEEE Access*, 2023.

[37] B Prabha, K Ramesh, and PN Renjith. A review on dynamic virtual machine consolidation approaches for energy-efficient cloud data centers. *Data Intelligence and Cognitive Informatics: Proceedings of ICDICI 2020*, pages 761–780, 2021.

[38] Mirsaeid Hosseini Shirvani, Amir Masoud Rahmani, and Amir Sahafi. A survey study on virtual machine migration and server consolidation techniques in dvfs-enabled cloud datacenter: taxonomy and challenges. *Journal of King Saud University-Computer and Information Sciences*, 32(3):267–286, 2020.

[39] Rahmat Zolfaghari and Amir Masoud Rahmani. Virtual machine consolidation in cloud computing systems: Challenges and future trends. *Wireless Personal Communications*, 115(3):2289–2326, 2020.

[40] Monireh H Sayadnavard, Abolfazl Toroghi Haghighat, and Amir Masoud Rahmani. A multi-objective approach for energy-efficient and reliable dynamic vm consolidation in cloud data centers. *Engineering science and technology, an International Journal*, 26:100995, 2022.

[41] Harmeet Kaur and Abhineet Anand. Review and analysis of secure energy efficient resource optimization approaches for virtual machine migration in cloud computing. *Measurement: Sensors*, 24:100504, 2022.

[42] Nagma Khattar, Jagpreet Sidhu, and Jaiteg Singh. Toward energy-efficient cloud computing: a survey of dynamic power management and heuristics-based optimization techniques. *The Journal of Supercomputing*, 75:4750–4810, 2019.

[43] Fikru Feleke Moges and Surafel Lemma Abebe. Energy-aware vm placement algorithms for the openstack neat consolidation framework. *Journal of Cloud Computing*, 8(1):2, 2019.

[44] Hamid Talebian, Abdullah Gani, Mehdi Sookhak, Ahmed Abdelaziz Abdelatif, Abdullah Yousafzai, Athanasios V Vasilakos, and Fei Richard Yu. Optimizing virtual machine placement in iaas data centers: taxonomy, review and open issues. *Cluster Computing*, 23:837–878, 2020.

[45] Mala Kalra and Sarbjeet Singh. A review of metaheuristic scheduling techniques in cloud computing. *Egyptian informatics journal*, 16(3):275–295, 2015.

[46] Suraj Singh Panwar, Man Mohan Singh Rauthan, and Varun Barthwal. A systematic review on effective energy utilization management strategies in cloud data centers. *Journal of Cloud Computing*, 11(1):95, 2022.

[47] Spyros Angelopoulos, Shahin Kamali, and Kimia Shadkami. Online bin packing with predictions. *Journal of Artificial Intelligence Research*, 78:1111–1141, 2023.

[48] Anton Beloglazov and Rajkumar Buyya. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. *Concurrency and Computation: Practice and Experience*, 24(13):1397–1420, 2012.

[49] Sadoon Azizi, Maz'har Zandsalimi, and Dawei Li. An energy-efficient algorithm for virtual machine placement optimization in cloud data centers. *Cluster Computing*, 23(4):3421–3434, 2020.

[50] Vaneet Garg and Balkrishan Jindal. Resource optimization using predictive virtual machine consolidation approach in cloud environment. *Intelligent Decision Technologies*, 17(2):471–484, 2023.

[51] Seyyed Meysam Rozehkhani, Farnaz Mahan, and Witold Pedrycz. Efficient cloud data center: An adaptive framework for dynamic virtual machine consolidation. *Journal of Network and Computer Applications*, 226:103885, 2024.

[52] Manoel C Silva Filho, Claudio C Monteiro, Pedro RM Inácio, and Mário M Freire. Approaches for optimizing virtual machine placement and migration in cloud environments: A survey. *Journal of Parallel and Distributed Computing*, 111:222–250, 2018.

[53] Weiwei Lin, Siyao Xu, Ligang He, and Jin Li. Multi-resource scheduling and power simulation for cloud computing. *Information Sciences*, 397:168–186, 2017.

[54] Christopher Carothers, Alois Ferscha, Richard Fujimoto, David Jefferson, Margaret Loper, Madhav Marathe, Pieter Mosterman, Simon JE Taylor, and Hamid Vakilzadian. Computational challenges in modeling and simulation. *Research Challenges in Modeling and Simulation for Engineering Complex Systems*, pages 45–74, 2017.

[55] Mohamed Abu Sharkh, Ali Kanso, Abdallah Shami, and Peter Öhlén. Building a cloud on earth: A study of cloud computing data center simulators. *Computer Networks*, 108:78–96, 2016.

[56] Pericherla S Suryateja. A comparative analysis of cloud simulators. *International Journal of Modern Education & Computer Science*, 8(4), 2016.

[57] Chaoqiang Jin, Xuelian Bai, Chao Yang, Wangxin Mao, and Xin Xu. A review of power consumption models of servers in data centers. *applied energy*, 265:114806, 2020.

[58] Rahmat Zolfaghari, Amir Sahafi, Amir Masoud Rahmani, and Reza Rezaei. An energy-aware virtual machines consolidation method for cloud computing: Simulation and verification. *Software: Practice and Experience*, 52(1):194–235, 2022.

[59] Bilal Ahmad, Zaib Maroof, Sally McClean, Darryl Charles, and Gerard Parr. Economic impact of energy saving techniques in cloud server. *Cluster Computing*, 23(2):611–621, 2020.

[60] Congfeng Jiang, Tiantian Fan, Honghao Gao, Weisong Shi, Liangkai Liu, Christophe Cérin, and Jian Wan. Energy aware edge computing: A survey. *Computer Communications*, 151:556–580, 2020.

[61] Alessio Balsini, Luigi Pannocchi, and Tommaso Cucinotta. Modeling and simulation of power consumption and execution times for real-time tasks on embedded heterogeneous architectures. *ACM SIGBED Review*, 16(3):51–56, 2019.

[62] David Ockwell, Joanes Atela, Kennedy Mbeva, Victoria Chengo, Rob Byrne, Rachael Durrant, Victoria Kasprowicz, and Adrian Ely. Can pay-as-you-go, digitally enabled business models support sustainability transformations in developing countries? outstanding questions and a theoretical basis for future research. *Sustainability*, 11(7):2105, 2019.

[63] Rodrigo N Calheiros, Rajiv Ranjan, Anton Beloglazov, César AF 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(1):23–50, 2011.

[64] 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. IEEE, 2011.

[65] Bhathiya Wickremasinghe, Rodrigo N Calheiros, and Rajkumar Buyya. Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications. In *2010 24th IEEE international conference on advanced information networking and applications*, pages 446–452. IEEE, 2010.

[66] Weiwei Chen and Ewa Deelman. Workflowsim: A toolkit for simulating scientific workflows in distributed environments. In *2012 IEEE 8th international conference on E-science*, pages 1–8. IEEE, 2012.

[67] Rodrigo N Calheiros, Marco AS Netto, César AF De Rose, and Rajkumar Buyya. Emusim: an integrated emulation and simulation environment for modeling, evaluation, and validation of performance of cloud computing applications. *Software: Practice and Experience*, 43(5):595–612, 2013.

[68] Florian Fittkau, Sören Frey, and Wilhelm Hasselbring. Cdosim: Simulating cloud deployment options for software migration support. In *2012 IEEE 6th International Workshop on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA)*, pages 37–46. IEEE, 2012.

[69] Hamza Ouarnoughi, Jalil Boukhobza, Frank Singhoff, Stéphane Rubini, and Erwann Kassis. Considering i/o processing in cloudsim for performance and energy evaluation. In *High Performance Computing: ISC High Performance 2016 International Workshops, ExaComm, E-MuCoCoS, HPC-IODC, IXPUG, IWOPH, P^ 3MA, VHPC, WOPSSS, Frankfurt, Germany, June 19–23, 2016, Revised Selected Papers 31*, pages 591–603. Springer, 2016.

[70] Alberto Núñez, Jose L Vázquez-Poletti, Agustin C Caminero, Gabriel G Castañé, Jesus Carretero, and Ignacio M Llorente. icancloud: A flexible and scalable cloud infrastructure simulator. *Journal of Grid Computing*, 10:185–209, 2012.

[71] Gabor Kecskemeti. Dissect-cf: a simulator to foster energy-aware scheduling in infrastructure clouds. *Simulation Modelling Practice and Theory*, 58:188–218, 2015.

[72] Vincenzo Emanuele Martone, Michele Mastroianni, and Francesco Palmieri. Evaluation of cloud simulators for energy-related applications. In *2023 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, pages 142–147. IEEE, 2023.

[73] René Ponto, Gabor Kecskemeti, and Zoltán Á Mann. Comparison of workload consolidation algorithms for cloud data centers. *Concurrency and Computation: Practice and Experience*, 33(9):e6138, 2021.

[74] Dzmitry Kliazovich, Pascal Bouvry, and Samee Ullah Khan. Greencloud: a packet-level simulator of energy-aware cloud computing data centers. *The Journal of Supercomputing*, 62:1263–1283, 2012.

[75] Simon Ostermann, Kassian Plankensteiner, Radu Prodan, and Thomas Fahringer. Groudsim: An event-based simulation framework for computational grids and clouds. In *Euro-Par 2010 Parallel Processing Workshops: HeteroPar, HPCC, HiBB, CoreGrid, UCHPC, HPCF, PROPER, CCPI, VHPC, Ischia, Italy, August 31–September 3, 2010, Revised Selected Papers 16*, pages 305–313. Springer, 2011.

[76] Simon Ostermann, Kassian Plankensteiner, Daniel Bodner, Georg Kraler, and Radu Prodan. Integration of an event-based simulation framework into a scientific workflow execution environment for grids and clouds. In *Towards a Service-Based Internet: 4th European Conference, ServiceWave 2011, Poznan, Poland, October 26-28, 2011. Proceedings 4*, pages 1–13. Springer, 2011.

[77] Simon Ostermann, Gabor Kecskemeti, and Radu Prodan. Multi-layered simulations at the heart of workflow enactment on clouds. *Concurrency and Computation: Practice and Experience*, 28(11):3180–3201, 2016.

[78] Ilyas Bambrik. A survey on cloud computing simulation and modeling. *SN Computer Science*, 1(5):249, 2020.

[79] Damián Fernández-Cerero, Alejandro Fernández-Montes, Agnieszka Jakóbik, Joanna Kołodziej, and Miguel Toro. Score: Simulator for cloud optimization of resources and energy consumption. *Simulation Modelling Practice and Theory*, 82:160–173, 2018.

[80] Saiqin Long, Yuan Li, Jinna Huang, Zhetao Li, and Yanchun Li. A review of energy efficiency evaluation technologies in cloud data centers. *Energy and Buildings*, 260:111848, 2022.

[81] Mohit Kumar, Subhash Chander Sharma, Anubhav Goel, and Santar Pal Singh. A comprehensive survey for scheduling techniques in cloud computing. *Journal of Network and Computer Applications*, 143:1–33, 2019.

[82] Oleg Sukhoroslov and Andrey Vetrov. Towards fast and flexible simulation of cloud resource management. In *2022 International Conference on Modern Network Technologies (MoNeTec)*, pages 1–8. IEEE, 2022.

[83] Gabriel Dulac-Arnold, Nir Levine, Daniel J Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning*, 110(9):2419–2468, 2021.

[84] David Perez Abreu, Karima Velasquez, Marilia Curado, and Edmundo Monteiro. A comparative analysis of simulators for the cloud to fog continuum. *Simulation Modelling Practice and Theory*, 101:102029, 2020.

[85] Divya Kapil, Varsha Mittal, and Atika Gupta. Cloud computing and simulation paradigms: A technical exploration and analysis. In *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–6. IEEE, 2024.

[86] Jiechao Gao, Haoyu Wang, and Haiying Shen. Machine learning based workload prediction in cloud computing. In *2020 29th international conference on computer communications and networks (ICCCN)*, pages 1–9. IEEE, 2020.

[87] Binbin Feng and Zhijun Ding. Application-oriented cloud workload prediction: A survey and new perspectives. *Tsinghua Science and Technology*, 30(1):34–54, 2024.

[88] Eduard Zharikov, Sergii Telenyk, and Petro Bidyuk. Adaptive workload forecasting in cloud data centers. *Journal of Grid Computing*, 18(1):149–168, 2020.

[89] Orawat Yodnual and Roungsan Chaisricharoen. Optimized classification for organizational workload. In *2021 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering*, pages 313–317. IEEE, 2021.

[90] Avita Katal, Susheela Dahiya, and Tanupriya Choudhury. Workload characterization and classification: A step towards better resource utilization in a cloud data center. *Pertanika Journal of Science & Technology*, 31(5), 2023.

[91] Korrapati Sindhu, Karthick Seshadri, and Chidambaran Kollengode. Workload characterization and synthesis for cloud using generative stochastic processes. *The Journal of Supercomputing*, 78(17):18825–18855, 2022.

[92] Sheng Di and Franck Cappello. Gloudsim: Google trace based cloud simulator with virtual machines. *Software: Practice and Experience*, 45(11):1571–1590, 2015.

[93] Bulbul Gupta, Pooja Mittal, and Tabish Mufti. A review on amazon web service (aws), microsoft azure & google cloud platform (gcp) services. In *Proceedings of the 2nd International Conference on ICT for Digital, Smart, and Sustainable Development, ICIDSSD 2020, 27-28 February 2020, Jamia Hamdard, New Delhi, India*, 2021.

[94] Shobhana Kashyap and Avtar Singh. Prediction-based scheduling techniques for cloud data center's workload: a systematic review. *Cluster Computing*, 26(5):3209–3235, 2023.

[95] Lotfi Belkhir and Ahmed Elmeligi. Assessing ict global emissions footprint: Trends to 2040 & recommendations. *Journal of cleaner production*, 177:448–463, 2018.

[96] Oluwasanmi Richard Arogundade and Kiran Palla. Virtualization revolution: Transforming cloud computing with scalability and agility, 2023.

[97] Bhagyalakshmi Magotra, Deepti Malhotra, and Amit Kr Dogra. Adaptive computational solutions to energy efficiency in cloud computing environment using vm consolidation. *Archives of computational methods in engineering*, 30(3):1789–1818, 2023.

[98] Abdelaziz Said Abohamama and Eslam Hamouda. A hybrid energy–aware virtual machine placement algorithm for cloud environments. *Expert Systems with Applications*, 150:113306, 2020.

[99] Zhihua Li, Kaiqing Lin, Shunhang Cheng, Lei Yu, and Junhao Qian. Energy-efficient and load-aware vm placement in cloud data centers. *Journal of Grid Computing*, 20(4):39, 2022.

[100] Sanjaya K Panda and Prasanta K Jana. An efficient request-based virtual machine placement algorithm for cloud computing. In *Distributed Computing and Internet Technology: 13th International Conference, ICDCIT 2017, Bhubaneswar, India, January 13-16, 2017, Proceedings 13*, pages 129–143. Springer, 2017.

[101] Larry Peterson, Andy Bavier, Marc E Fiuczynski, and Steve Muir. Experiences building planetlab. In *Proceedings of the 7th symposium on Operating systems design and implementation*, pages 351–366, 2006.

[102] Suraj Singh Panwar, MMS Rauthan, Varun Barthwal, Sachin Gaur, and Nidhi Mehra. Study of energy-efficient virtual machine migration with assurance of service-level agreements. In *International Conference on Cryptology & Network Security with Machine Learning*, pages 761–785. Springer, 2023.

[103] Madnesh K Gupta and Tarachand Amgoth. Resource-aware virtual machine placement algorithm for iaas cloud. *The Journal of Supercomputing*, 74:122–140, 2018.

[104] Mohamed Ghetas. A multi-objective monarch butterfly algorithm for virtual machine placement in cloud computing. *Neural Computing and Applications*, 33(17):11011–11025, 2021.

[105] Minhaj Ahmad Khan. An efficient energy-aware approach for dynamic vm consolidation on cloud platforms. *Cluster Computing*, 24(4):3293–3310, 2021.

[106] P Nehra and Nishtha Kesswani. Efficient resource allocation and management by using load balanced multi-dimensional bin packing heuristic in cloud data centers. *The Journal of Supercomputing*, 79(2):1398–1425, 2023.

[107] Zahra Mahmoodabadi and Mostafa Nouri-Baygi. An approximation algorithm for virtual machine placement in cloud data centers. *The Journal of Supercomputing*, 80(1):915–941, 2024.

[108] Sadoon Azizi, Mohammad Shojafar, Jemal Abawajy, and Rajkumar Buyya. Grvmp: A greedy randomized algorithm for virtual machine placement in cloud data centers. *IEEE Systems Journal*, 15(2):2571–2582, 2021.

[109] Zhou Zhou, Mohammad Shojafar, Mamoun Alazab, Jemal Abawajy, and Fangmin Li. Afed-ef: An energy-efficient vm allocation algorithm for iot applications in a cloud data center. *IEEE Transactions on Green Communications and Networking*, 5(2):658–669, 2021.

[110] Shilpa Sunil and Sanjeev Patel. Energy-efficient virtual machine placement algorithm based on power usage. *Computing*, 105(7):1597–1621, 2023.

[111] Aisha Fatima, Nadeem Javaid, Tanzeela Sultana, Mohammed Y Aalsalem, and Shaista Shabbir. An efficient virtual machine placement via bin packing in cloud data centers. In *Advanced Information Networking and Applications: Proceedings of the 33rd International Conference on Advanced Information Networking and Applications (AINA-2019) 33*, pages 977–987. Springer, 2020.

[112] Anurina Tarafdar, Mukta Debnath, Sunirmal Khatua, and Rajib K Das. Energy and quality of service-aware virtual machine consolidation in a cloud data center. *The Journal of Supercomputing*, 76:9095–9126, 2020.

[113] Javad Masoudi, Behnam Barzegar, and Homayun Motameni. Energy-aware virtual machine allocation in dvfs-enabled cloud data centers. *IEEE Access*, 10:3617–3630, 2021.

[114] Rajganesh Nagarajan and Ramkumar Thirunavukarasu. A review on intelligent cloud broker for effective service provisioning in cloud. In *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 519–524. IEEE, 2018.

[115] Yongqiang Gao, Haibing Guan, Zhengwei Qi, Yang Hou, and Liang Liu. A multi-objective ant colony system algorithm for virtual machine placement in cloud computing. *Journal of computer and system sciences*, 79(8):1230–1242, 2013.

[116] Amir Javadpour, Arun Kumar Sangaiah, Pedro Pinto, Forough Ja'fari, Weizhe Zhang, Ali Majed Hossein Abadi, and HamidReza Ahmadi. An energy-optimized embedded load balancing using dvfs computing in cloud data centers. *Computer Communications*, 197:255–266, 2023.

[117] Jyotsna P Gabhane, Sunil Pathak, and Nita Thakare. An improved multi-objective eagle algorithm for virtual machine placement in cloud environment. *Microsystem Technologies*, pages 1–13, 2023.

[118] Badieh Nikzad, Behnam Barzegar, and Homayun Motameni. Sla-aware and energy-efficient virtual machine placement and consolidation in heterogeneous dvfs enabled cloud datacenter. *IEEE Access*, 10:81787–81804, 2022.

[119] Zhihua Li, Xinrong Yu, Lei Yu, Shujie Guo, and Victor Chang. Energy-efficient and quality-aware vm consolidation method. *Future Generation Computer Systems*, 102:789–809, 2020.

[120] Takwa Tlili and Saoussen Krichen. Best fit decreasing algorithm for virtual machine placement modeled as a bin packing problem. In *2023 9th International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 1261–1266, 2023.

[121] David HS Lima, Andre LL Aquino, and Marilia Curado. A virtual machine placement algorithm for resource allocation in cloud-based environments. In *Workshop de Gerência e Operação de Redes e Serviços (WGRS)*, pages 113–124. SBC, 2023.

[122] Kazi Main Uddin Ahmed, Math HJ Bollen, and Manuel Alvarez. A review of data centers energy consumption and reliability modeling. *IEEE access*, 9:152536–152563, 2021.

[123] Jiechao Liang, Weiwei Lin, Yangguang Xu, Yubin Liu, Ruichao Mo, and Xiaoxuan Luo. Energy-aware parameter tuning for mixed workloads in cloud server. *Cluster Computing*, 27(4):4805–4821, 2024.

[124] Khadijah Bahwaireth, Lo'ai Tawalbeh, Elhadj Benkhelifa, Yaser Jararweh, and Mohammad A Tawalbeh. Experimental comparison of simulation tools for efficient cloud and mobile cloud computing applications. *EURASIP Journal on Information Security*, 2016:1–14, 2016.

[125] Paul H Hargrove and Jason C Duell. Berkeley lab checkpoint/restart (blcr) for linux clusters. In *Journal of Physics: Conference Series*, volume 46, page 494. IOP Publishing, 2006.

[126] Dhahi Alshammari, Jeremy Singer, and Timothy Storer. Performance evaluation of cloud computing simulation tools. In *2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, pages 522–526. IEEE, 2018.

[127] Antonios T Makaratzis, Konstantinos M Giannoutakis, and Dimitrios Tzovaras. Energy modeling in cloud simulation frameworks. *Future Generation Computer Systems*, 79:715–725, 2018.

[128] Shashikant Ilager, Adel N Toosi, Mayank Raj Jha, Ivona Brandic, and Rajkumar Buyya. A data-driven analysis of a cloud data center: statistical characterization of workload, energy and temperature. In *Proceedings of the IEEE/ACM 16th International Conference on Utility and Cloud Computing*, pages 1–10, 2023.

[129] Wenhong Tian, Minxian Xu, Aiguo Chen, Guozhong Li, Xinyang Wang, and Yu Chen. Open-source simulators for cloud computing: Comparative study and challenging issues. *Simulation Modelling Practice and Theory*, 58:239–254, 2015.

[130] Julio Proaño, Carmen Carrión, and Blanca Caminero. Empirical modeling and simulation of an heterogeneous cloud computing environment. *Parallel Computing*, 83:118–134, 2019.

[131] Georgia Sakellari and George Loukas. A survey of mathematical models, simulation approaches and testbeds used for research in cloud computing. *Simulation Modelling Practice and Theory*, 39:92–103, 2013.

[132] Talha Umar, Mohammad Nadeem, and Mohammad Sajid. Simulation tools for cloud computing: A comparative study. In *Advances in Data-driven Computing and Intelligent Systems: Selected Papers from ADCIS 2022, Volume 2*, pages 239–251. Springer, 2023.

[133] Muhammad Asim Shahid, Muhammad Mansoor Alam, and Mazliham Mohd Su'ud. A systematic parameter analysis of cloud simulation tools in cloud computing environments. *Applied Sciences*, 13(15):8785, 2023.

[134] Rebeca Estrada, víctor Asanza, Danny Torres, Adrian Bazurto, and Irving Valeriano. Learning-based energy consumption prediction. *Procedia Computer Science*, 203:272–279, 2022.

[135] Sarika Mane, Makarand Kulkarni, and Sudha Gupta. Energy prediction for efficient resource management in iot-enabled data centres. In *Technologies for Energy, Agriculture, and Healthcare*, pages 158–166. CRC Press, 2025.

[136] Ayaz Ali Khan and Muhammad Zakarya. Energy, performance and cost efficient cloud datacentres: A survey. *Computer Science Review*, 40:100390, 2021.

[137] Muhammad Asim Shahid. A systematic survey of simulation tools for cloud and mobile cloud computing paradigm. *Journal of Independent Studies and Research Computing*, 20(1), 2022.

[138] Pratik Thantharate. Scale-it: distributed and realistic simulation frameworks for testing cloud-based software. In *2023 10th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, pages 300–306. IEEE, 2023.

[139] Hasanein Rjeib and Gábor Kecskeméti. An investigation on implementing a scenario on different cloud simulators. *MULTIDISZCIPLINÁRIS TUDOMÁNYOK: A MISKOLCI EGYETEM KÖZLEMÉNYE*, 12(3):256–263, 2022.

[140] Miyuru Dayarathna, Yonggang Wen, and Rui Fan. Data center energy consumption modeling: A survey. *IEEE Communications surveys & tutorials*, 18(1):732–794, 2015.

[141] KyoungSoo Park and Vivek S Pai. Comon: a mostly-scalable monitoring system for planetlab. *ACM SIGOPS Operating Systems Review*, 40(1):65–74, 2006.

[142] Rebeca Estrada, Danny Torres, Adrian Bazurto, Irving Valeriano, et al. Learning-based energy consumption prediction. *Procedia Computer Science*, 203:272–279, 2022.

[143] Amir Varasteh and Maziar Goudarzi. Server consolidation techniques in virtualized data centers: A survey. *IEEE Systems Journal*, 11(2):772–783, 2015.

[144] Rahul Yadav, Weizhe Zhang, Keqin Li, Chuanyi Liu, and Asif Ali Laghari. Managing overloaded hosts for energy-efficiency in cloud data centers. *Cluster Computing*, pages 1–15, 2021.

[145] Sanjaya K Panda and Prasanta K Jana. An energy-efficient task scheduling algorithm for heterogeneous cloud computing systems. *Cluster Computing*, 22(2):509–527, 2019.

[146] Altaf Hussain, Muhammad Aleem, Abid Khan, Muhammad Azhar Iqbal, and Muhammad Arshad Islam. Ralba: a computation-aware load balancing scheduler for cloud computing. *Cluster Computing*, 21:1667–1680, 2018.

[147] S Kanagasubaraja, M Hema, K Valarmathi, Naveen Kumar, Bala Pradeep M Kumar, and N Balaji. Energy optimization algorithm to reduce power consumption in cloud data center. In *2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, pages 1–8. IEEE, 2022.

[148] S Anithakumari and K Chandrasekaran. Monitoring and management of service level agreements in cloud computing. In *2015 International Conference on Cloud and Autonomic Computing*, pages 204–207. IEEE, 2015.

[149] Behrooz Nobakht, Stijn De Gouw, and Frank S de Boer. Formal verification of service level agreements through distributed monitoring. In *European Conference on Service-Oriented and Cloud Computing*, pages 125–140. Springer, 2015.

[150] Faiza Qazi, Daehan Kwak, Fiaz Gul Khan, Farman Ali, and Sami Ullah Khan. Service level agreement in cloud computing: Taxonomy, prospects, and challenges. *Internet of Things*, page 101126, 2024.

[151] Wei-Hua Bai, Jian-Qing Xi, Jia-Xian Zhu, and Shao-Wei Huang. Performance analysis of heterogeneous data centers in cloud computing using a complex queuing model. *Mathematical Problems in Engineering*, 2015(1):980945, 2015.