

Optimizing Renewable Energy Utilization in Cloud Data Centers through Dynamic Overbooking: An MDP-based Approach

Tuhin Chakraborty, Carlo Kopp, *Senior Member, IEEE*, and Adel N. Toosi, *Member, IEEE*

Abstract—The shift towards renewable energy sources for powering data centers is increasingly important in the era of cloud computing. However, integrating renewable energy sources into cloud data centers presents a challenge due to their variable and intermittent nature. The unpredictable workload demands in cloud data centers further complicate this problem. In response to this pressing challenge, we propose a novel approach in this paper: adapting the workload to match the renewable energy supply. Our solution involves dynamic overbooking of resources, providing energy flexibility to data center operators. We propose a framework that stochastically models both workload and energy source information, leveraging Markov Decision Processes (MDP) to determine the optimal overbooking degree based on the workload flexibility of data center clients. We validate the proposed algorithm in realistic settings through extensive simulations. Results demonstrate the superiority of our proposed method over existing approaches, achieving better matching with the renewable energy supply by 55.6%, 34.65%, and 40.7% for workload traces from *Nectar Cloud*, *Google*, and *Wikipedia*, respectively.

Index Terms—green computing, sustainable computing, energy efficiency, cloud computing, renewable energy, data centers, VM consolidation, overbooking.

1 INTRODUCTION

THE adoption of sustainability practices to reduce carbon emissions resulting from cloud data center (CDC) operation is of major importance due to the high operational costs and energy demands of CDCs. In addition to techniques for reducing total energy consumption, powering data centers using renewable (green) energy sources, such as solar energy, can significantly reduce carbon emissions. However, the continuity and reliability of renewable energy supply from sources such as solar and wind power is lesser than grid power supplies, due to its stochastic and intermittent nature. In other words, the variable and intermittent nature of both workload demand and renewable energy supply make the goal of powering data centers using renewable energy sources more challenging than established alternatives.

Current data centers are designed with overprovisioning to meet peak demand and mitigate the challenges of serving fluctuating workloads [1], [2]. Recent work proposed resource allocation management for the workload by jointly exploiting VM consolidation and overbooking of resources [3], [4]. Overbooking is the process of allocating more virtual machines (VMs) on a physical server than the server is sized for. This work aims to exploit varying degrees of overbooking to elastically manage power and workloads

when necessary. We exploit energy source-aware resource allocation management so that the energy required to serve the workload favors available green energy where possible.

Due to the complexity of resource allocation in cloud data centers, a large body of solutions designed for energy-efficient resource management algorithms are based on heuristics [5], [6], [7], [8]. Heuristic-based algorithms have become popular in such applications as they can produce acceptable results in a reasonable time frame. However, heuristic methods have their own limitations. They cannot search the entire solution space, and hence their performance depends on the strength of the heuristic within the context of the related problem targeted to be solved. As a result, their performance is not guaranteed. This work aims to deal with the variability of workload demand and renewable energy supply, which are non-deterministic in practice, by modeling them as stochastic processes.

We investigate and model the variability of workload demand and green energy availability as Markov Decision Processes (MDP). Notably, this work considers that the cloud data center is powered by an on-site renewable energy system (e.g., a photovoltaic solar power system) and can use available power from the grid when the supply from renewable resources is inadequate. As this work aims to exploit varying degrees of overbooking to control the power requirement, a challenging question to address is “when and in what capacity should overbooking of resources be applied to better match the power demand of the workload with the available renewable energy supply?” We aim to find a policy to apply optimal overbooking levels. The policy contains actions for the best overbooking levels applied to allocated resources in different time slots.

The key contributions of the paper are as follows:

- T. Chakraborty and C. Kopp are with the Faculty of Information Technology, Monash University, VIC, Australia.
E-mail: tuhin.babai@gmail.com, carlo.kopp@monash.edu
- A. N. Toosi is with the School of Computing and Information Systems, University of Melbourne, Australia. He was formerly with the Faculty of Information Technology, Monash University, VIC, Australia, with which he is still affiliated.
E-mail: adel.toosi@unimelb.edu.au

Manuscript received April dd, 202x; revised Month dd, 202x.

- We propose a green-energy-aware architecture to manage the resources of a CDC.
- We model the green-energy-aware resource allocation management problem as a discrete finite state Markov Decision System problem.
- We propose a solution methodology by exploiting the dynamic adjustment of varying overbooking levels according to the availability of renewable energy.
- We propose the novel concept of *Green SLA*, which allows for an agreement between CDC clients and CDC providers to negotiate the degree to which overbooking can be applied to CDC clients' workloads in favor of renewable energy availability.
- We conduct simulation-based experiments using real-world data derived from various workload traces including a community cloud, *Wikipedia* and *Google* traces, and renewable energy availability data from practical solar systems.

The remainder of the paper is organized as follows: The background of renewable energy aware sustainable CDCs is explained in Section 2. This is followed by the proposed system with the details of problem formulations in Section 3. Our proposed algorithm is explained in Section 4. Section 5 presents the details of the performance evaluations including the results. Finally, the conclusions and future works are discussed in Section 6.

2 RELATED WORK AND MOTIVATION

This section presents an overview of the related work and the motivation underpinning this work.

2.1 Related Work

Several works have explored the area of resource management for cloud data centers in an energy-efficient manner. Most of these works addressed a particular component of cloud infrastructure and proposed energy-efficient management techniques for the targeted components. Some works, such as [17], [18], [19], considered power consumption by the CDC network fabric and proposed energy-efficient ways to manage it. CARPO [17] exploited the correlation between traffic flows to consolidate the less correlated traffic into the same network link to improve efficiency. However, Heller et al. [18] observed that the dominant cause of power consumption in CDC network fabrics are switching components when powered on, and that as little as 1-2 Watts per port are saved by powering off unused ports. This work did not consider the edge switches of the empty servers as part of their energy-saving mechanism. Maestro et al. [20] investigated a decentralized architecture for cloud resource management based on blockchain technology, while disregarding constraints related to energy sources. Cziva et al. [19] minimized network-wide communication cost by exploiting temporal network traffic information while not considering the servers' power consumption.

Several studies considered servers and network components together as part of their mechanisms for improving energy efficiency, specifically [1], [4], [21], [22]. Jin et al. [21] considered the host and network jointly to decide upon VM placements in targets to optimize power consumption.

They proposed a mechanism to find a suitable host for the VM placement by utilizing a depth-first search. Fang et al. [22] investigated the placements of VMs in the hosts and network routing together. They proposed to form groups of VMs that share higher mutual traffic and then assign these to the same rack. This way, they aimed to minimize traffic between racks to permit unutilized switches to be powered off to save energy. Zheng et al. [1] observed that the workloads in different servers and the data center network (DCN) traffic flows do not peak simultaneously. Based on this observation, they consolidate servers and traffic to save more energy, considering correlations between workloads. Thus their proposed power optimization strategy, *PowerNetS*, jointly minimizes the power consumption of servers and DCN [1]. VM placements and consolidation strategies are also used in [4] and [23] to reduce the energy requirement of a data center. Consolidating the VMs in a minimal subset of physical hosts can effectively reduce the requirement by pushing the unused hosts into idle mode. Son et al. [4] also turned the network switches off to save more energy when not utilized. None of these works considered powering CDCs with a renewable energy source. None of their proposed solutions also considered energy supply source awareness as part of their resource management framework.

Workload distribution across various geographical locations has been proposed to optimize operating costs and balance workloads. Research conducted by Ammari et al. [24] aimed to address applications with a bounded delay when deployed in Distributed Green Data Centers (DGDCs). Their study focused on the cost-effective scheduling of various heterogeneous applications while ensuring the delay-bound constraints of different tasks were met. Previously, Yuan et al. [25] proposed a Geography-Aware Task Scheduling (GATS) method that considers spatial disparities in DGDCs. This approach maximizes the total profit to the DGDC provider by intelligently scheduling tasks in all applications. Additionally, Yuan et al. [26] introduced a multi-objective optimization technique for DGDCs that aims to maximize the profit of DGDC providers and minimize the average task loss for all applications. The method jointly determines the division of tasks among multiple Internet service providers (ISPs) given the task service rates of each Green Data Center (GDC). However, these research works necessitated the consideration of delay-bound constraints to effectively schedule workloads. This requirement may unfortunately not always be feasible for all types of services.

Additionally, Li et al. [27] conducted an investigation into the optimization of workflow execution costs on Infrastructure-as-a-Service (IaaS) clouds while adhering to service-level agreements with users. Zhou et al. [28] provided a thorough review of deep reinforcement learning-based approaches in cloud scheduling, encompassing recent research in this domain, with a specific focus on resource scheduling in Cloud Computing. Meanwhile, Zhang et al. [29] explored the management of cloud resources using deep reinforcement learning to automate and efficiently negotiate appropriate configurations in a complex cloud environment. However, these research endeavors did not specifically address the energy-related and green computing aspects of cloud data centers.

TABLE 1: Literature comparison

Research Work	Consume Brown Energy	Consume Green Energy	Apply Migration	Apply Varied Dynamic Overbooking	Consider On-site Renewable	Green-aware Dynamic Resource Management	Consider Stochasticity
Le et al. [9]	✓	✓	✗	✗	✗	✗	✗
Celesti et al. [10]	✓	✓	✗	✗	✗	✗	✗
Li et al. [11]	✓	✓	✗	✗	✓	✗	✓
Toosi et al. [12]	✓	✓	✓	✗	✓	✗	✗
Farahnakian et al. [13]	✓	✗	✓	✗	✗	✗	✓
Son et al. [4]	✓	✗	✓	✓	✗	✗	✗
Goiri et al. [14]	✓	✓	✗	✗	✓	✗	✓
Hasan et al. [15]	✓	✓	✗	✗	✓	✗	✗
Khosravi et al. [16]	✓	✓	✓	✗	✓	✗	✓
Chakraborty et al. [5]	✓	✓	✓	✗	✓	✓	✗
OUR WORK	✓	✓	✓	✓	✓	✓	✓

Several research studies have concentrated on the direct incorporation of renewable energy to provide power to data centers. Goiri et al. [14] proposed a delay in serving workloads to align them with available renewable energy supply, where workloads could be deferred. Hasan et al. [15] proposed to compromise non-core and independent features, which could be isolated to align their execution with green energy availability. Khosravi et al. [16] proposed migrating VMs between geographically diverse sites to exploit local availability of renewable energy. Similarly, Liu et al. [30] proposed geographical load balancing with “follow the renewables” routing to manage the workloads. Lin et al. [31] extended that concept by combining brown and green energy to achieve a net-zero brown energy system. Toosi and Buyya [32] presented a fuzzy logic-based load-balancing algorithm without a priori knowledge to solve the same problem. These reactive solutions are based on routing the incoming workload to “follow the renewables” when receiving job requests. However, none of these research works considered VM management inside the CDCs that exploits overbooking techniques to maximise renewable energy utilization.

Chakraborty et al. [5] aimed to address this critical issue by proposing a heuristic-based approach that utilized a fixed and pre-defined amount of overbooking. However, their approach failed to consider the power consumption overhead of VM management and was limited to reactive responses to renewable energy generation. Additionally, it did not take into account the probabilistic nature of both workload and renewable energy generation. In contrast, this work deals with the variability of workload demand and renewable energy supply, which are non-deterministic in practice, and then uses MDP-based dynamic overbooking to match this variability. It also explores VM management with varying levels of overbooking to identify appropriate matches between energy consumption and renewable energy availability, while also considering the energy overhead required to manage VMs and apply overbooking.

Xu et al. [33] conducted a study on the optimization of renewable energy utilization in environmentally sustainable cloud data centers, with a primary emphasis on admission control and battery charge and discharge scheduling. In contrast, our research uses the grid as the backup energy source and focuses on dynamically adjusting the overbooking level based on the availability of renewable energy, an aspect that has not been thoroughly explored in the current literature. Nevertheless, both approaches share common ground in employing a finite-horizon MDP approach. MDP-based solutions and reinforcement learning approaches con-

tinue to be extensively researched in the field of process industries [34] [35].

2.2 Motivation

CDC clients are service providers who subscribe to resources from CDC operators in order to run and manage end-user applications. To become more environmentally friendly, we assume that CDC clients agree to accept some levels of intermittent resource overbooking from the CDC operators in exchange for incentives such as carbon tax benefits or discounts, when exposed to overbooking, as a part of a service level agreement (SLA). We have named this part of the agreement as *Green SLA*. *Green SLA* is a suitable option for services that are commonly underutilized or applications that can handle occasional overbooking, such as batch jobs, background processing, and optional tasks. Overbooking, however, may result in a degradation of Quality of Service (QoS), e.g., delayed response time for end users of the services provided by cloud clients (service providers). To ensure compatibility with the incentives received, CDC clients should manage their applications accordingly.

In this work, we propose that CDC operators take advantage of *Green SLA* opportunities for overbooking in order to reduce brown energy consumption. In practice, CDC operators may use overbooking and impose CPU over-commitment without even informing clients [36]. For example, VMware considers up to 200% CPU over-booking, according to its best practice recommendations, as a performance sizing guideline. Moreover, it is common that CDC providers offer services at different QoS levels and prices. For example, Amazon Web Services (AWS) offer interruptible Spot Instances [37] in their spare capacity for less than the On-Demand price. This work proposes an optimized policy for applying overbooking in order to match the supply of renewables with data center power consumption, motivated by a proposed *Green SLA* which incentivizes both parties to use renewables.

3 SYSTEM MODELING AND PROBLEM STATEMENT

This section presents an overview of our system model with the key issues addressed in this paper in a typical CDC scenario, followed by the power consumption models of the CDC components. Then, we present a discrete-time finite-state MDP to handle and model this issue and to find the optimal overbooking policy as a solution.

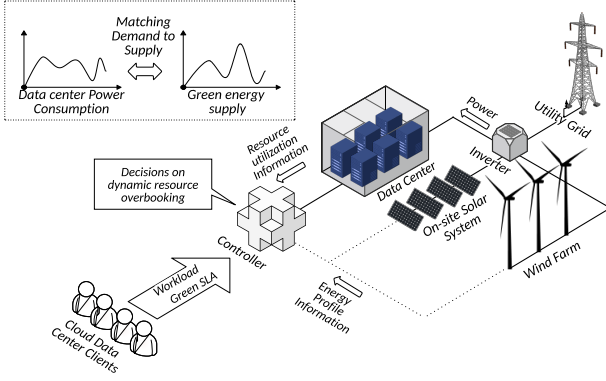


Fig. 1: Schematic view of the system.

3.1 System Overview

A schematic diagram of the proposed system with a typical CDC scenario is shown in Fig. 1.

Input component: The CDC clients subscribe to resources from the data center to manage and run applications for the end-users. In order to provide services to the end-users, CDC clients transmit their workloads to be processed, while complying with the agreed *Green SLA* standards. These workloads are subsequently allocated to virtual machines (VMs) residing within the data center infrastructure. The *Controller* component of the system receives workload requests from the CDC clients, accompanied by the resource utilization and energy profile information obtained from the power system. The *Controller* module proceeds to provide dynamic resource overbooking using these information. A detailed explanation of the *Controller* component is provided later in this section and in Section 3.3.

Data center components: The system considers a data center that consists of multiple physical machines serving the user requests by assigning VMs with the required virtual CPUs (vCPUs), Memory, and Disk. Different components and activities that consume power such as physical hosts, switches, and cooling systems, are considered. Note that, since the method is designed to manage workload and their associated resources, we only focus on the computational load-variant energy components. Thus, we disregard the other components that do not directly vary with the computational load, such as power control systems and lighting, as they remain load-invariant. Power consumption models for all components are reported in Section 3.2.

Power system: Several energy sources are also considered to power the data center, including onsite solar panels to generate green energy and coal-based brown energy from the grid. The controller tracks and manages power consumption by exploiting dynamic overbooking levels per the proposed algorithm in Section 4.

This work encourages on-site renewable without local storage such as lithium batteries in CDCs, similar to greener services presented in [38]. Recent works such as [14], [39], [40] specifically avoided batteries because of 1) the high procurement and maintenance cost of batteries, 2) the shorter lifespan relative to the other components, and 3) the fact that they are not fully recyclable due to technical constraints and regulatory gaps that present challenges in disposal. In

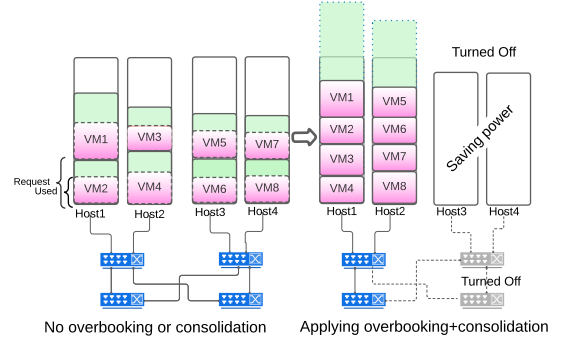


Fig. 2: Example of how consolidation with overbooking can reduce power consumption [5].

addition, the cost of expanding energy storage installations to fully support a CDC operation without a grid connection can become prohibitively expensive, especially as the size of the data center grows. Thus, in this work, we assume procuring power from grid as a fallback source when renewable power is insufficient. We consider inverters to combine multiple power sources. Inverters can supply power from renewable sources, send excess power back to the grid, and switch to grid power when renewable energy is insufficient.

Controller component: The objective of this study is to synchronize the energy consumption of a data center with the supply of renewable energy. The *Controller* module evaluates the present resource allocation, workload, and availability of green energy to monitor the system's performance. It employs a discrete-time finite-state Markov Decision Process (MDP) based approach, dependent on the workload and solar power history, to generate a policy output (elucidated in Sec. 4) that outlines actions for optimal overbooking levels. These strategic outputs are employed to dynamically allocate resources with varying overbooking levels at each time interval (e.g., hourly throughout a day) to control and better align the energy requirement of the data center with the supply of green energy. The MDP model's different components and our strategy's working procedures are expounded upon in detail in Section 4.

Our proposed method only applies a varying degree of overbooking when the supply of renewable energy is inadequate. As described above, we assume that there is a *Green SLA* between CDC operators and clients, which enables occasional overbooking when renewables are scarce. This agreement aims to motivate both parties to adapt greener services knowingly and with mutual benefits, as explained above in Section 2.2. In the next section, we explain the procedure for reducing power consumption followed by an in-depth discussion of consumption models.

Fig. 2 illustrates how to reduce power consumption by exploiting consolidation and overbooking through VM migration activity. A group of VMs hosted by the corresponding physical hosts (servers) is shown in the left part of Fig. 2. The green area represents the maximum level of overbooking that can be applied to the VM's requested resources (green + pink) according to the *Green SLA*. The implementation of overbooking along with consolidation can lead to a lower resource usage, and thus, can save energy, as shown in the right part of Fig. 2. Note that idle

TABLE 2: Some basic notations for problem formulation

Symbols	Description
h_i	The i^{th} host in the data center
H	$\forall i, h_i \in H$, set of all hosts in the data center,
$ H $	Total number of hosts in the CDC
s_i	i^{th} switch in the data center
S	The set of all network switches in the CDC, $\forall i, s_i \in S$
$ S $	The total number of network switches in the data center
$P(h_i)$	Power consumption of host i
P_{id}	Idle power consumption of host
P_{pk}	Peak power consumption of host
u_i	CPU utilization percentage of host i
α_i	The number of VMs placed in host i
$P(s_i)$	Power consumption of switch i

(base) power consumption is a constant factor consumed by hosts and switches regardless of how much load they are serving [4]. This work aims to save this energy as much as possible and whenever a host is idle, we turn it off.

3.2 Energy Model

This section presents power consumption models representing the different components of the system and management activity of a CDC.

3.2.1 Compute Energy Model

The power consumption of a server is typically modeled by its CPU power and the widely-used model proposed by Pelley et al. [41], which is a linear function that considers the base idle power consumption and the consumption due to CPU utilization, with the power consumed increasing linearly as CPU utilization increases. This model has also been adopted in other recent works, such as [4], [5], [42], [43]. Therefore, the power consumption of any host h_i is modeled below:

$$P(h_i) = \begin{cases} P_{id} + (P_{pk} - P_{id}) \cdot u_i & \text{if } \alpha_i > 0 \\ 0 & \text{if } \alpha_i = 0 \end{cases} \quad (1)$$

where P_{id} and P_{pk} are the power consumption of a host in its idle and peak states, which are constant factors, u_i is the dynamic power consumption of host, which is linearly related to the CPU utilization percentage, and α_i is the number of VMs placed in host i .

The *compute* subsystem of a CDC comprises a set of hosts. Its power consumption during time slots 0 to T is determined by the Equation 2:

$$P_C = \sum_{t=0}^T \left(\delta_t \cdot \sum_{i=1}^{|H|} P(h_i) \right) \quad (2)$$

where $|H|$ is the total number of hosts and δ_t is the duration of the respective time slot.

3.2.2 Network Energy Model

The network power is mainly modeled by its consumption of active switching components [18] as adopted by recent works, such as [5], [43]. Thus, we assume that the power consumed by the network is predominantly determined by the number of active (On) switches. The power consumption of any switch s_i , $P(s_i)$, is modeled as:

$$P(s_i) = \begin{cases} P_{sw} & \text{if } s_i \text{ is ON} \\ 0 & \text{if } s_i \text{ is OFF} \end{cases} \quad (3)$$

where P_{sw} is the power consumption of the switch.

We compute the network subsystem energy consumption using equation 3, over time period 0 to T by the given Equation 4:

$$P_S = \sum_{t=0}^T \left(\delta_t \cdot \sum_{i=1}^{|S|} P(s_i) \right) \quad (4)$$

Note that power consumption of the network subsystem in a CDC is typically low, comprising approximately 5% of the total power consumption, primarily due to the energy usage of network switches [18], [44].

3.2.3 CRAC Model

The thermal management of a data center is typically handled by a Computer Room Air Conditioning (CRAC) unit, which we assumed to be the only cooling system facility available in this study. More sophisticated cooling systems can be added to the model if needed. To estimate the power consumption by the CRAC subsystem, we adopt a highly-cited model from the [45]. This work is also adopted by the recently published works [5], [46]. According to this model, the Coefficient of Performance (*CoP*), which is a function of the cold air supply temperature T_s , determines the efficiency of the CRAC subsystem. *CoP* represents the ratio of the total power of the compute subsystem to the power consumption required to extract the dissipated heat by the CRAC subsystem.

$$P_{AC} = P_C / CoP(T_s) \quad (5)$$

where P_C represents compute power and P_{AC} represents cooling system power. Either diminishing the compute load or increasing the temperature of supplied cold air can reduce the power consumption of the cooling system. The following regression model, adopted from [45], estimates the *CoP*.

$$CoP(T_s) = aT_s^2 + bT_s + c. \quad (6)$$

where $a = 0.0068$, $b = 0.0008$, and $c = 0.458$. Equations 5 and 6 indicate that an increasing value of T_s causes a reduction in cooling power when the compute system power remains the same. Now, a function of compute power consumption P_C and supplied cold air temperature T_s can measure the total energy consumption for the cooling system and it is represented below (by using (5), (2)):

$$P_{AC}(P_C, T_s) = \frac{\sum_{t=0}^T \left(\delta_t \cdot \sum_{i=1}^{|H|} P(h_i) \right)}{CoP(T_s)} \quad (7)$$

Thus, the total energy consumption by the CRAC system, compute, and network of CDC over the time slots 0 to T can be formulated as:

$$P_T = P_{AC}(P_C, T_s) + P_C + P_S \quad (8)$$

Which is:

$$P_T = \sum_{t=0}^T \left(\delta_t \cdot \sum_{i=1}^{|S|} P(s_i) \right) + \left(\frac{\alpha T_s^2 + \beta T_s + \gamma + 1}{\alpha T_s^2 + \beta T_s + \gamma} \right) \cdot \sum_{t=0}^T \left(\delta_t \cdot \sum_{i=1}^{|H|} P(h_i) \right) \quad (9)$$

3.2.4 Energy Model for Virtual Machines (VM) Migration

This work considers live VM migration technology, which has proven to be a powerful management technique for data center operators to optimize the placement of VMs in a non-disruptive manner [47]. Virtualization technology giants, such as VMware and Xen, have successfully adopted live VM migration in a memory-to-memory strategy using a pre-copying mechanism [48], [49], [50]. This method migrates the physical memory image to the new destination host while the VM in the source host is running. The energy overhead during live migration activity can be measured as the sum of two components.

1) The *first component* calculates the overhead for the migration effort that depends on the size of the memory-to-memory transfer due to migration activity through the interconnect fabric. We adopt [50] to calculate this additional amount of energy consumed (denoted by E_m), which is obtained as the increased amount of consumption arising from live VM migration. They model this energy as the linear regression of the network traffic for the memory size of the VM that is being migrated (V_m).

$$E_m = aV_m + b \quad (10)$$

where $a = 0.512$, $b = 20.165$, and the R^2 is as high as 0.993.

2) The *second component* of the energy overhead during the live migration process calculates the energy overhead of keeping both the source and destination hosts turned on during the migration. Note that we trigger VM migration when evacuating a host to be turned off to reduce power consumption. So, before turning off the source host, the energy consumed by the evacuating (source) host j is modeled as:

$$E_h = \tau \times P(h_j) \quad (11)$$

where τ is the time required for the migration of VMs of host j . So, the energy overhead for the migration at any time-slot t is modeled as:

$$P(M_t) = E_m + \sum_{\text{Evacuating Hosts}} E_h \quad (12)$$

The total energy overhead for the migration activity P_M during time slots 0 to T is determined by the (13):

$$P_M = \sum_{t=0}^T \delta_t \cdot P(M_t) \quad (13)$$

3.3 Problem Formulation

To better match the data center's power consumption with the supply of green energy, we aim to find an optimized policy to apply varying degrees of overbooking while serving the workload. As shown in the schematic overview in Fig. 1, the *Controller* manages the resource allocation to serve the workload depending on the availability of green energy. So, the Controller keeps track of the current allocations, workloads, and green energy supplies. We consider a discrete-time finite-state MDP-based algorithm to find the optimal policy within a fixed time horizon (e.g., a day). The output of the policy contains actions for the selection of optimal overbooking levels applied to allocating resources in each time slot. We discuss different components of the MDP model below.

3.3.1 States

To implement the discrete-time finite-state MDP we must divide the time horizon into a set of identical discrete time slots. In this work, we consider single-day optimization and we thus divided each day into 24 one hour time slots. But the model can be simply generalized to any other time horizon and respective number of time slots. We focus on the hourly time cycles of a day since the weather data availability is often on an hourly basis, and services are billed per hour by popular providers such as AWS.

The system state $S(t)$ at time slot t includes 1) the hour of the day, 2) workload demand, 3) available renewable energy, and 4) the level of overbooking. The state space of the *hour of the day* at time t is given as $H(t) \leq H_M \in [1, 24] \cap \mathbb{N}$, which is the set of natural numbers (\mathbb{N}) from 1 to 24, both inclusive.

The state space of *workload demand* at time t is given as $W(t) \leq W_M \in \mathbb{Z}^+$, where \mathbb{Z}^+ is the set of non-negative integers and W_M is the maximum workload the system can serve. The workload is measured by the number of VM instances needed to be served during time period t . The *available renewable energy* at time t is represented by a discrete random value $G(t) \leq G_M \in \mathbb{Z}^+$. $G(t)$ represents the amount of green power generated by the onsite renewable power system and G_M is the maximum amount of renewable power that can be generated by the system. Similarly, $O(t) \leq O_M \in \mathbb{Z}^+$ is a discrete value denoting the *overbooking level* at time t , where O_M is the maximum overbooking level possible in the system. Therefore, the state of the system at time t , $S(t)$, is denoted by:

$$S(t) \triangleq [H(t), W(t), G(t), O(t)] \in \mathbb{S} \quad (14)$$

where \mathbb{S} stands for all possible states.

3.3.2 Actions

At the beginning of each time slot, the system determines the control action to increase or decrease the amount of overbooking. Here, the action of adjusting the overbooking basically changes the active resource allocation schemes and results in changes in energy consumption to serve the workload leading to a better match with the available green energy. The amount of overbooking leads to a different scheme of resource allocation including consolidation and VM migrations. A following state $S(t+1)$ depends only on the current state $S(t)$ and the decision maker's action. Therefore, it satisfies the Markov property of the model that the state transitions must be independent of actions and states encountered before the current decision step. Our model decides to perform an action in each state denoted by A , where depending on the overbooking level a at the current state $S(t)$, the model decides to increase or decrease the amount of overbooking by $|a - b|$ to reach overbooking state b at the state $S(t+1)$. The actions change the states from one to another and achieve different rewards. The objective of the system is to calculate the optimal policy in form of a mapping from states to actions such that the reward is maximized. In this case, we need to select the overbooking level b for the next state $S(t+1)$ such that it can return the maximum reward. The reward function is discussed in the next subsection.

3.3.3 Reward Function

At each time slot, the process is at state $S(t)$, and we choose a possible action A . The process randomly moves to the next state $S(t+1)$ at the next time slot, and gives the corresponding reward $R(S(t), A)$. Our model is designed in such a way that a higher reward is produced when power consumption is closer to available green power. Updating the overbooking level has an energy overhead, as it needs to be achieved by changing the active resource allocation. Thus, the energy requirements has two parts, one is to serve the workload, and another is the overhead to update resource allocation (e.g., the energy overhead of VM migration for host evacuations and consolidations). The sum of these two energy requirement components needs to be as close as possible to the green energy supply to achieve optimal results. Let, G be the green energy supply and E represents the sum of the following two energy requirement components: 1) the energy required to serve the workload denoted as $E(W(t))$ (derived by Equation 9) and 2) the energy overhead of updating the overbooking level from a to b , denoted as $E_O(|a - b|)$ (derived by Equation 13), that is, $E = E(W(t)) + E_O(|a - b|)$. We define $X = G - E$, and $Y = \frac{X}{E_{MAX}}$, where E_{MAX} is the highest electrical energy required, i.e., at the full capacity of all the hosts in the CDC. Moreover, we have introduced a controlling factor, represented by ρ ($0 \leq \rho \leq 1$), which provides additional flexibility by enabling a weighted preference to be set in situations where the sum of energy consumption components (E) exceeds the green energy (G), or vice versa. When ρ is set to a higher value, it emphasizes a more conservative approach to using brown energy, prioritizing environmental friendliness. In contrast, when ρ is set to a lower value, it focuses on lowering *Green SLA* violations and adopting a more conservative overbooking strategy. Thus, we define the reward function as:

$$R = \begin{cases} \rho \times (1 - Y) & \text{if } G \geq E \\ (1 - \rho) \times (1 + Y) & \text{if } G < E \end{cases} \quad (15)$$

where G represents the green energy supply and E denotes the sum of two energy components. The function is designed to provide a higher reward when the energy consumption E is closer to the available green energy G , with a flexible reward for situations where E exceeds or falls short of G .

3.3.4 Transition probabilities

The probability of moving from one state to another depends on transition probabilities. In other words, the chosen action A depends on the probability that the MDP moves into each state $S(t+1)$. We assume that the decision-maker uses a long history of workload demand and renewable power generation to compute these probabilities. Sections 5.2 and 5.3 describe sample procedures we followed to compute these probabilities. So, the probability of attaining a specific workload level $W(t)$ and green power generation $G(t)$ at a time slot t are known in advance and are denoted by $P(W(t))$ and $P(G(t))$, respectively. Thus, the transition probability with a specific action A from a state $S(t)$ to $S(t+1)$ is computed as:

$$P_A(S(t), S(t+1)) = P(W(t+1)) \times P(G(t+1)) \quad (16)$$

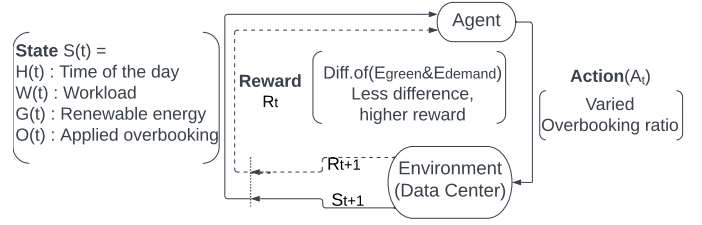


Fig. 3: The operation of the MDP.

3.3.5 Optimal Policy

The expected reward is maximized in the observation period for the optimal policy. This optimal policy π^* contains the best action for all the states in MDP and the equation for this policy is shown below:

$$\mathcal{O}^{\pi^*}(S(t)) = \max_A \left\{ R(S(t), A) + \gamma \sum_{S(t+1)} P_A(S(t), S(t+1)) \times \mathcal{O}^{\pi^*}(S(t+1)) \right\} \quad (17)$$

Note that, this equation is referred to as the *Bellman optimality equation*, where $\mathcal{O}^{\pi^*}(S(t))$ is the value function for the optimal policy and π^* refers to optimal policy. The algorithm to solve this problem is presented in the next section.

4 GREEN AWARE ALGORITHM

This section proposes an algorithm to address the problem formulated in Section 3.3 and explains how it works in a typical CDC scenario.

4.1 MDP-based Green aware Algorithm

Our green-aware MDP based algorithm, outlined in Algorithm 1, can determine the control actions for the optimal policy. The policy provides actions for selecting optimal overbooking levels to allocate resources in each time slot and consequently leads to match the supply of energy from renewable source with energy requirement of the data center.

Initializing system information: The algorithm starts by initializing the system with information about the maximum levels for workloads and green energy, as well as the number of observation slots. It then sets up the necessary variables to explore all possible states (Lines 1-2).

Identifying reachable states and the possible actions: The algorithm identifies all possible states based on the maximum levels of workload, green energy, and overbooking that can be applied. Only states with a non-zero probability are considered reachable (Lines 3-13). For these reachable states, the algorithm adds possible actions accordingly (Lines 14-16).

Updating transition probabilities and reward of states: The algorithm retrieves the transition probabilities (Equation (16)) of different levels of workload and green power. Then, it updates the reward (Equation (15)) of each state accordingly (Lines 17-18).

Computing the utility function: The algorithm calculates the optimal expected reward value for all reachable states using Algorithm 2 (Line 19).

Algorithm 1 Control Actions for the MDP Based Green aware Algorithm

Input: Observation time slots, Transition probabilities for workloads and green energy, System states

Output: Control actions for all reachable states.

```

1:  $T \leftarrow$  Number of observation slots,  $W_M \leftarrow$  Maximum
   levels of workload,  $G_M \leftarrow$  Maximum levels of green
   energy,  $P(w(t)), P(g(t)) \leftarrow$  Transition probabilities for
   workloads and green energy as described in section 5.2
   and 5.3
2:  $h_t, w_t, g_t, o_t \leftarrow 0, \mathbb{S} \leftarrow \emptyset$   $\triangleright$  Required variables for
   time slots, workload, green energy, overbooking levels,
   respectively, to build the set  $\mathbb{S}$ , of all possible states that
   our system can reach
3: while  $h(t) \leq T$  do
4:   while  $w(t) \leq W_M$  do
5:     while  $g(t) \leq G_M$  do
6:       if  $P(w(t)) \times P(g(t)) > 0$  then
7:         while  $o(t) \leq O_M$  do
8:            $\mathbb{S}.Append(S(t) \text{ as } (h(t), w(t), g(t), o(t)))$ 
9:         end while
10:        end if
11:      end while
12:    end while
13:  end while
14: for all reachable states in  $\mathbb{S}$  do
15:   Adding action  $A(a, b)$  into possible actions  $A(S(t))$ 
   for  $S(t)$ 
16: end for
17: Update transition probabilities from  $S(t)$  to  $S(t+1)$ 
18: Update the reward function from  $S(t)$  to  $S(t+1)$ 
19:  $\mathcal{O}^{\pi^*}(S(t)) \leftarrow$  Optimal expected reward by Algorithm 2
20: Deciding the control actions based on  $\mathcal{O}^{\pi^*}(S(t))$ .
21: Return best actions for states
  
```

Deciding upon the actions: Based on the optimized expected reward value, the algorithm selects the action that maximizes the objective function for each state, as outlined in Equation (17). It then returns the *best action* for each state.

Algorithm 2 determines the optimal expected reward by iterating through all possible actions. It utilizes value iteration to maximize the reward value by initializing the basic parameters of the MDP. The algorithm takes all the reachable states and the possible actions as inputs, and iterates to find the optimal expected reward value $\mathcal{O}^{\pi^*}(S(t))$ with optimal policy π^* . In each iteration, the optimal expected reward value is updated based on the expected reward from the previous state. The algorithm converges when the next iteration could not modify the expected reward value more than a pre-determined small value. Once the optimal expected reward value is obtained from Algorithm 2, Algorithm 1 can determine the optimal control action.

Complexity analysis: In our proposed algorithm, each parameter value was iterated over all possible combinations of the other parameters, thus forming the solution space of all possible states that our system can reach. This resulted in a polynomial time complexity of $O(\tau\kappa\lambda\mu)$, where τ , κ , λ , and μ are the maximum number of levels of observation-slots, green power, workload, and overbooking,

respectively. This is because the algorithm iterates over this solution space.

4.2 Working Mechanism with MDP-Based Algorithm

The overall workflow of our proposed system utilizing the MDP-based green-energy-aware algorithm is presented in Fig. 4. The cloud data center clients subscribe to resources from the data center to manage and run end-user applications. CDC clients and operators discussed and agreed with the *Green SLA* to promote environmentally friendly computing, which incentivizes both parties to use renewables. Workloads to serve the service demanded by end users are transmitted to be processed while complying with the *Green SLA* standards. The Controller component receives workload requests and checks resource utilization and energy profile information from the power system. Workloads are allocated to virtual machines (VMs) in the data center infrastructure. The Controller module scrutinizes and evaluates this information before processing the workloads. It checks for a suitable overbooking level, as suggested by Algorithm 1, to control the energy requirement to serve these workloads to match the available renewable energy.

5 PERFORMANCE EVALUATION

We conducted a performance evaluation with realistic settings to validate the effectiveness of our proposed MDP-based algorithm. We used historical solar data from the Australian Government Bureau of Meteorology along with three realistic workload traces. This approach allowed us to simulate different scenarios and measure the effectiveness of our algorithm in terms of reducing brown energy consumption and balancing energy consumption with the availability of green energy.

5.1 Workload Traces

We employed three realistic data center workloads, derived from the *Google Cluster*, *Wikipedia*, and *Nectar* community cloud traces. To fit the workloads into states, we divided the workloads into 10 discrete levels. The *Google Cluster* and *Wikipedia* traces are publicly available and have been widely used in the literature [51], [52] as representative of real-world workloads. The *Google* workload represents activities in a cluster of approximately 12,500 compute cells managed by Google's *Borg* cluster management software. The *Google Cluster* trace includes job requests submitted to a cluster over a period of one month. *Wikipedia* traces contains 10% of all HTTP requests issued to *Wikipedia* (in all languages) available at [52], [53]. The workload follows a diurnal pattern with clear periods of different workload intensity largely matching the Solar traces. *Nectar* [54] is an Australia's national research cloud that provides services to Australian universities and researchers. The traces for the *Nectar Cloud* are extracted from the real usage of resources at the Monash University *Nectar* node. The *Nectar* cloud workload pattern is consistent throughout the year.

Since we needed VM requests for our experiments, we mapped Google's *job* requests to generate VM requests similar to [16], [32]. We used the first 504 hours of data to compute the workload level probabilities and the last 7 days

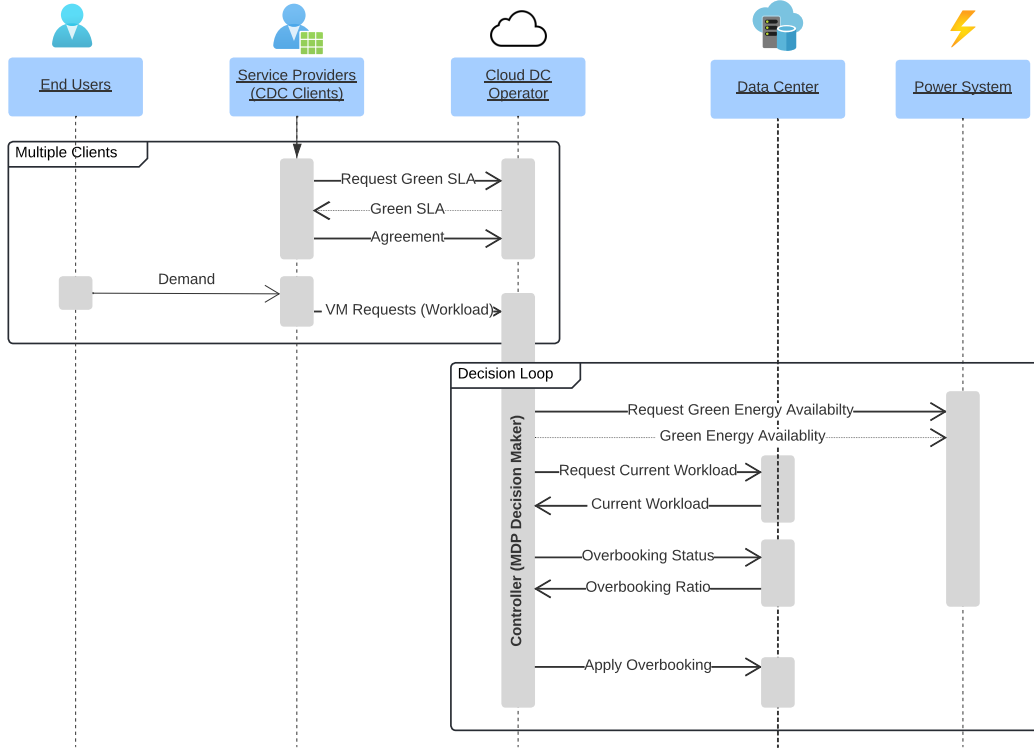


Fig. 4: The sequence diagram of the overall workflow of the system.

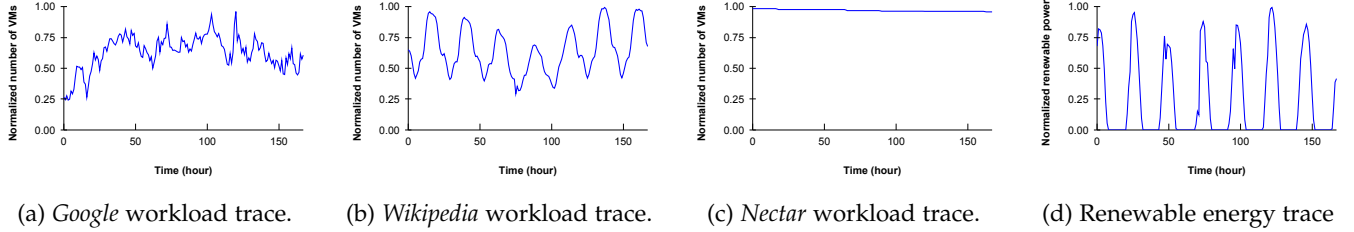


Fig. 5: Workloads and a renewable energy trace used in the experiment

of data for the experiment. Fig. 5a shows the normalized workload of VM requests generated based on the scheduling algorithm in [12], [32] used in our experiments. We used 4231 as the peak number of VMs and scaled the workload accordingly.

For *Wikipedia* workload, we used the requests from Oct-14, 2007, which totaled about 2119.6M requests. We used the first 504 hours of data to compute the workload level probabilities and the next 7 days of data for the experiment. We mapped these request traces to generate VM requests in a similar way as done in [32]. We normalized the *Wikipedia* workload shape to the peak and then matched its peak to the peak of the *Google* workload to standardize the experimental environment. Fig. 5b shows the normalized *Wikipedia* workload trace used in our experiments.

The *Nectar Cloud* workload was collected over a longer period of time, so we used the first 3600 hours of data to compute the workload level probabilities and the next 7 days of data for the experiment. We normalized the workload shape to its peak, and then matched it to the peak of the *Google* workload to standardize the experimental

environment. Fig. 5c shows the normalized *Nectar Cloud* workload trace used in this experiment.

5.2 Workload Level Probabilities

$P(W(t))$ represents the probability of achieving a specific workload level $W(t)$ at time slot t . It is calculated based on historical data on a daily cycle. So, for a specific time slot of the day (e.g., 10:00 to 11:00 am), it counts the number of occurrences of a certain workload level in that time slot by iterating over the historical data. The number of occurrences is then divided by the total number of counts in all the levels for that time slot to calculate the respective probability. This results in a 10×24 probability matrix, where the rows represent the different workload levels and columns represent the time slots of a day.

5.3 Solar Power Levels

We use hourly satellite data for solar irradiation falling on a horizontal surface collected by the Australian Bureau of Meteorology. We employ historical data for global

Algorithm 2 Optimal expected reward for Algorithm 1

Input: all reachable states S in \mathbb{S} , possible action $A(S)$, estimated transition probabilities at time interval t
Output: Optimal expected reward value

```

1:  $\varepsilon \leftarrow \text{error}, \gamma \leftarrow \text{Discount factor}$ 
2: repeat
3:    $\mathcal{O} \leftarrow \mathcal{O}', \delta \leftarrow 0,$ 
4:   for each state  $S \in \mathbb{S}$  do
5:      $\mathcal{O}'(S) = R(S, A) + \gamma \cdot \max_A \left\{ \sum_{S'} P_A(S(t), S') \times \right.$ 
        $\left. \mathcal{O}(S') \right\}$ 
6:     if  $|\mathcal{O}'(S) - \mathcal{O}(S)| > \delta$  then
7:        $\delta \leftarrow |\mathcal{O}'(S) - \mathcal{O}(S)|$ 
8:     end if
9:   end for
10: until  $\delta < \varepsilon(1 - \gamma)/\gamma$ 
11: Return  $\mathcal{O}'(S)$  as  $\mathcal{O}^{\pi^*}(S(t))$  for Algorithm 1  ▷ optimal
       expected reward
  
```

horizontal solar irradiance (GHI) on an hourly basis over 25 years (1990 to 2014). We assume that the solar system proportionately converts hourly GHI values to power. We use the same hour of the day to compute the likelihood of green power generation at a specific level $P(G(t))$. For example, to compute the probabilities at level 6 at hour 11:00 am on January 30, 2015, we look at the historical data over 25 years at 11:00 am on the same date. We count the number of occurrences of attaining a certain level in that time slot, and iterate over the historical data. We built our MDP-based model with 10 equally-sized levels of power generation, where the minimum level is 0 and the maximum is 9.

5.4 Baseline Algorithms

Load Balanced (LB): This naive algorithm aims to distribute workloads evenly among the available hosts. The algorithm selects the least loaded host to accommodate a newly requested VM and does not take overbooking into consideration. We use this widely used policy as a baseline algorithm for comparison.

Best Fit (BF): This algorithm follows the best fit policy and places VMs into the most loaded host that still has the capacity to accommodate the VM. Like LB algorithm, it applies zero overbooking and does not consider any VM migration. This policy, along with additional constraints, is also commonly used as a baseline for comparing with modified strategies.

BF75 Algorithm: This policy is similar to the BF baseline policy, but with a constant overbooking level of 75% for all hosts. However, unlike the MDP-based policy, this one overbooks the resources all the time, even when enough renewable energy is available and hence not caring for *Green SLA*. We will compare the results of this updated policy to our dynamic overbooking algorithm to understand the differences.

BF150 Algorithm: It follows a similar policy to the previous baseline policy, but with a higher constant overbooking level of 150% similar to that used in the MDP-based policy. We considered this policy as a lower-bound baseline. We

aim to compare the performance of our dynamically overbooked algorithm to this lower-bound model that always overbooks at the highest level.

Reactive based Green-energy-aware Best Fit with Migration (BFMig) Algorithm: This policy launches the VMs similar to BF and then applies VM migration to reduce brown energy usage when not enough renewable is available. A recent work [13] utilized a modified version of this policy to compare their scheme. We modified this policy as a renewable-energy-aware exploitation of VM migration and consolidation to reduce brown energy use. This policy measures the power use at each prefix time-slot and triggers migrations to consolidate VMs that results in reducing the power usage when not enough renewable is supplied. We use this modified renewable-energy-aware policy for comparison.

In our research, we have considered discussed strategies to compare with our proposed method. Among these, the Load Balanced (LB) and Best Fit (BF) algorithms are commonly reported and utilized VM placement strategies for managing cloud workloads, which we have used as baseline algorithms to measure the efficacy of our proposed method. Additionally, we have introduced two improvised baselines that overbook without any consideration of renewable energy availability. It was necessary to understand the effects with and without this awareness to evaluate the impact of our proposed strategy. Therefore, we have incorporated these two baselines, namely BF75 and BF150, in this section for comprehensive analysis.

Furthermore, the current contribution presents a dynamic management of workloads optimizing the renewable energy utilization searching over the entire solution space rather than any updated dynamic strategy based on reactive nature of changing the workloads and renewable energy availability with the time. So, an updated strategy which can *reactively* manage the resources with the presence of awareness of renewable energy availability should be a prominent test criteria. Therefore, we consider another updated algorithm *BFMig* for comparison: a renewable energy-aware, reactive, and dynamic strategy considering initial VM placements, VM consolidation, and migration applied with similar objectives. Recent work like [13] also utilized a modified version of this policy to compare their scheme. We modified this policy as a renewable-energy-aware synergistic exploitation of initial VM placements, VM migration and consolidation all together to reduce brown energy use.

5.5 Experimental Setup

The proposed algorithm are implemented in a simulation environment, as described in [43], to evaluate their performance. The simulation includes all the monitoring components to handle variable renewable energy availability, multiple energy supply sources, energy consumption for *Compute*, *Network*, and *Cooling* subsystems, energy overhead during migration activities, and dynamic overbooking methods. The data center used in the simulation has 128 hosts connected by an 8-port fat-tree network topology with 16 hosts per pod and 25°C cold air supply for the cooling system. The energy consumption in the data center is derived by the consumption of servers (260W on-peak),

TABLE 3: Experimental setup table

Parameter description	Value
The total number of hosts in the DC	128
Number of hosts under edge switch	4
Power consumption of host servers	260W
Power consumption of switch component	60W
Cold air supply temperature	25°C
Memory size of VMs	4GB
Number of VMs in a host	34
Number of levels in workload traces	10
Number of levels in solar energy traces	10
Number of levels in employed overbooking degree	6
Overbooking degree employed in each level	30%
Frequency of workload and energy traces in a day	24
Required model parameters as adopted from [45]	a=0.0068 b = 0.0008 c = 0.458
Required model parameters as adopted from [50]	a=0.512 b = 20.165
ρ value explained in Sec. 3.3.3 for general results	0.5
ρ value explained in Sec. 3.3.3 for results in Fig. 11	0.0 to 1.0
Overbooking ratio used in BF75 algorithm	75%
Overbooking ratio used in BF150 algorithm	150%
δ_t duration in time slots	1 hour

cooling components, switches (60W), VM migration activities using the energy models explained in Section 3.2.

Physical servers and VMs are assumed to be homogeneous in the setup. Each physical server has the capacity to host up to 34 VMs at peak load without overbooking. The energy overhead for migration activity is modeled in Section 3.2.4 according to [49] and 4GB of memory, similar to a popular medium-sized VM in OpenStack, is set as the amount of data to be transferred over the network for a single VM migration.

The experiments analyze the impact of 30% overbooking for each level, with five such levels. The value of ρ is set to 0.5 for general results and the impact of varying ρ values is presented in the Section 5.7.6. All simulations are run on an Intel Core i7-8850H 2.6GHz \times 12 core CPU 64-bit computer with 32 GB of DRAM, running Ubuntu 18.04.3. For experiments dealing with migration policy, the time window to run a migration is set to 1 hour, so migration is attempted as per our algorithm once every hour.

5.6 Performance Metrics

Absolute energy difference: This metric is used to measure the difference between the consumed energy and generated renewable energy on an hourly basis. It is used to evaluate how well the energy requirements are matched to the supply of renewable energy. In the results section, the summation of this metric over all days is reported to provide a total difference. The value is computed thus:

$$\sum_{t=0}^T \delta_t \cdot \left| E_t^C - E_t^G \right| \quad (18)$$

where, E_t^C is the consumed energy to serve the workload at time-slot t , E_t^G is the generated green energy at time-slot t , and δ_t is the duration of the time-slot t , which is an hour in this case. The goal is to minimize this metric.

Brown energy usage: Another main objective is to analyze the consumption of brown energy. To understand this factor, we measure the consumed energy required from the

brown energy source (grid) for all considered algorithms. This metric represents the total amount of brown energy (E_B) usage over all hours, as described in the following equation:

$$E_B = \sum_{t=0}^T \delta_t \cdot \text{MAX} \left\{ 0, (E_t^C - E_t^G) \right\} \quad (19)$$

where, E_t^C , E_t^G , and δ_t are similar to (18). This metric provides the total amount of brown energy consumption over all hours. The goal is to minimize this consumption.

Percentage of green energy usage: Another objective is to analyze whether our MDP-based approach can utilize a higher fraction of renewable energy. To evaluate this, we use the metric “percentage of renewable energy usage” (E_G^p). The percentage of renewable energy usage out of the total energy consumed to serve the workload is measured by the following equation. The denominator of (20) measures the total consumed energy to serve the workload.

$$E_G^p = \frac{\left(\sum_{t=0}^T \delta_t \cdot E_t^C \right) - E_B}{\sum_{t=0}^T \delta_t \cdot E_t^C} \times 100\% \quad (20)$$

where, E_t^C , E_B , and δ_t are similar to (18) and (19).

This metric provides the percentage of renewable energy usage out of the total energy consumed to serve the workload. The goal is to increase this percentage, which means to utilize more renewable energy.

Green SLA violations: We use overbooking to favor the use of renewable energy, but our objective is not to use it when sufficient renewable energy is available. Therefore, instances where we employed overbooking while there was sufficient renewable energy present as instances of a *Green SLA violation*. We measure this metric as *Green SLA Violation Ratio* (SLA_{Green}), which is quantified as the ratio of the total instance-time (I_t) that the system remains overbooked while there is enough renewable energy available and the total instance-time. It is described in the following equation:

$$SLA_{Green} = \frac{\sum_{\text{Violated Time Slots}} O_t^d \times I_t}{\sum_{\text{All Time Slots}} I_t} \quad (21)$$

where, *Instance-time* I_t measures the sum of the products of duration and number of VM instances in the time slot t and O_t^d is the degree of overbooking level applied at the time slot t .

This metric provides the ratio of the instances where the system remains overbooked while there is enough renewable energy available. The goal is to minimize this ratio, which means to avoid overbooking when there is enough renewable energy available.

Our research is focused on the utilization of renewable energy sources within data center infrastructure. We specifically explore the foundational aspects of energy utilization and environmental impact within data center operations. Thus, our evaluation metrics are tailored to reflect this specialized focus, with a primary emphasis on energy metrics, which inherently encompass carbon footprint and cost considerations. For instance, the utilization of brown energy, a widely accepted metric in green or sustainable computing, serves as a straightforward indicator of the efficiency of

approaches addressing this problem. Additionally, we introduce a novel metric, *Green SLA Violations*, which addresses the intersection of environmental sustainability and service level agreements, an area we believe deserves attention in modern data center management.

5.7 Experimental Results and Analysis

We describe experimental results in this section. To evaluate the performance, we performed our experiment on our simulation setup (see Section 5.5) with the VM traces of three workloads (see Section 5.1), each of 24×7 hours. Renewable energy traces utilized for the experiment are described in Section 5.3.

5.7.1 Brown Energy Consumption

The total brown energy purchased from the grid (brown sources) in seven days for different algorithms are shown in Fig. 6a, Fig. 6b, and Fig. 6c. Our proposed MDP-based approach results in about 2323kWh, 1379kWh, and 1688kWh of brown energy consumed to serve *Nectar*, *Google* and *Wikipedia* workloads, respectively. This is the closest to the BF150 approach (about 1915kWh, 1062kWh, and 1506kWh for *Nectar*, *Google* and *Wikipedia* workloads, respectively), which achieves the best results by always overbooking the system at the highest degree possible. Note that our proposed MDP based approach applied overbooking in different degrees as per the availability of green energy. However, it still achieves substantially close results to the strategy that always overbooks to the highest level, while outperforming it in other metrics as discussed in the following.

5.7.2 Matching against Green Energy availability

This section demonstrates how the resource allocation strategies of the algorithms perform in matching the renewable energy supply to serve the different workloads. Our proposed MDP-based approach can allocate resources in a way that can achieve a better match with the supply of renewable energy. We measure the total absolute energy difference on an hourly basis, as explained in Equation (18) for seven days using three years of solar data for different algorithms. Figures 7a, 7b, and 7c show the average of total difference for seven days for different algorithms. Our MDP-based approach results in the lowest difference (about 2395kWh, 1731kWh, and 1967kWh for *Nectar*, *Google* and *Wikipedia* workloads, respectively) and therefore outperforms all other algorithms, including the BF150 approach (about 2490kWh, 2014kWh, and 2327kWh for *Nectar*, *Google* and *Wikipedia* workloads, respectively), which always overbooks the system at the highest degree possible. This trend is consistent for all three types of workloads.

5.7.3 Utilization of Green Energy

One of the objectives was to analyze whether our MDP-based approach can utilize a higher proportion of green energy. We measured the percentage of green energy usage out of the total energy usage to serve the workload based on Equation (20). The average of green energy usage percentage in seven days for different algorithms are shown in Figures 8a, 8b, and 8c. Our proposed MDP-based approach (about 41.3%, 49.5%, and 46% for *Nectar*,

Google and *Wikipedia* workloads, respectively) outperforms the BF150 approach (about 37.1%, 41.5%, and 37% for *Nectar*, *Google* and *Wikipedia* workloads, respectively) through its varying levels of overbooking. This trend is consistent for all three types of workload. More detailed comparisons are presented in the next section for all three workload patterns.

5.7.4 Green SLA Violation Ratios

We analyzed the *Green SLA violation ratios* of algorithms that employed overbooking in order to exploit its benefits for the utilization of clean energy. We used Equation (21) to assess the performance of these algorithms. The lower the ratio, the less overbooking occurs when there is enough renewable energy. We found that the violation ratio is the highest for BF150 as this algorithm is unaware of green energy availability and applied the highest level of overbooking while sufficient renewable energy is available to serve the workload. This trend is the same for all three workload types. The average of the sum of *Green SLA* violations ratios for the selected seven days from three consecutive years shows that our proposed MDP-based algorithm results in fewer violations than the overbooking-based baselines for all three considered *Nectar*, *Google* and *Wikipedia* workloads. Our proposed MDP-based algorithm always results in very low violation as it decides and triggers the overbooking ratio more accurately. The average sum of the seven days is shown in Figures 10a, 10b, and 10c for *Nectar*, *Google* and *Wikipedia* workloads, respectively.

It is worth mentioning that overbooking of resources may lead to QoS degradation for end-users, such as delayed response times. QoS degradation in general may increase with higher overbooking levels, but the impact may vary with the varying application types or utilization levels. An experiment conducted in [55] measured how overbooking may affect response time, and one service showed about 2.5 times increment (from about 31 ms to about 79 ms) in 75% overbooking compared to no overbooking. Another service shows nearly two times increment (from about 22 ms to about 47 ms) from no overbooking to 75% overbooking. This work does not report on the QoS degradation in essence, as it is up to cloud clients (service providers) to decide how to manage QoS for end users, determine acceptable levels of overbooking, and whether the *Green SLA* is appropriate to apply in a specific context.

5.7.5 Comparison of Improvements

We compared the performance of various algorithms across three types of workloads and measured their relative performance improvement over the naive LB algorithm in terms of brown energy usage, absolute energy difference, percentage of green energy usage, as shown in Fig. 9. Our proposed MDP-based approach implements varying degrees of overbooking based on the availability of renewable energy and manages to reduce brown energy usage significantly (59.5%, 67.7%, 63.6% for *Nectar*, *Google* and *Wikipedia*, respectively). BF150 achieve the best performance across all algorithms in terms of brown energy usage due to its always overbooking strategy at the highest level possible. In comparison, our proposed MDP-based approach results in slightly higher brown energy usage (7.1%, 7.4%, 6.9% for *Nectar*, *Google* and *Wikipedia*, respectively) with significantly lower *Green SLA*

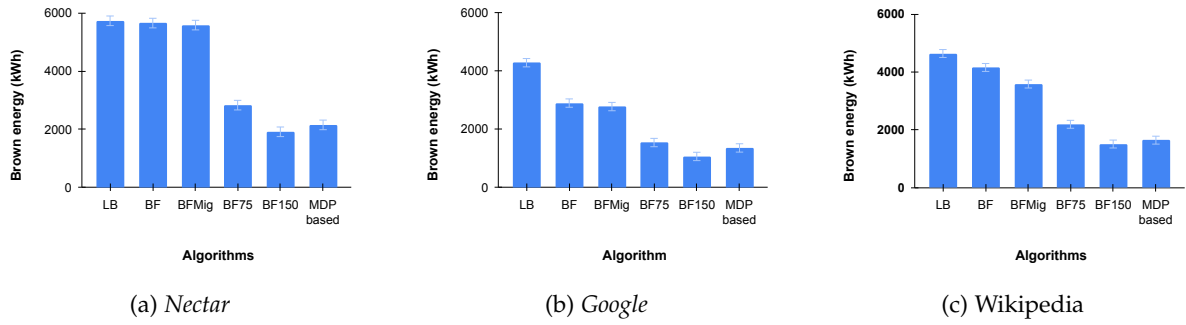


Fig. 6: The average of total brown energy usage for various algorithms for all three workloads.

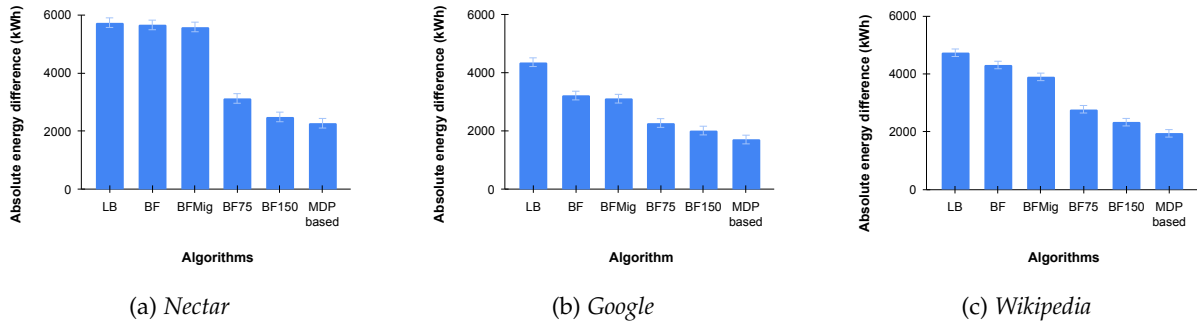


Fig. 7: The average of total absolute energy difference (measured in hourly basis) for various algorithms for all three workloads.

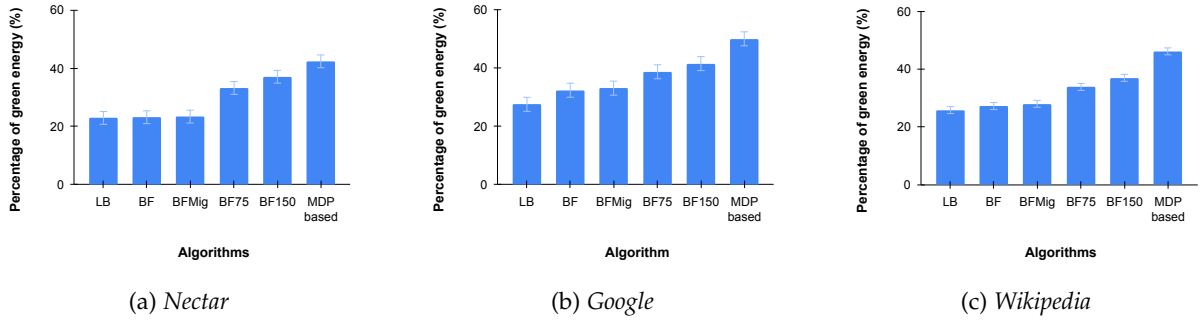


Fig. 8: The average of green energy usage percentage for various algorithms for all three workloads.

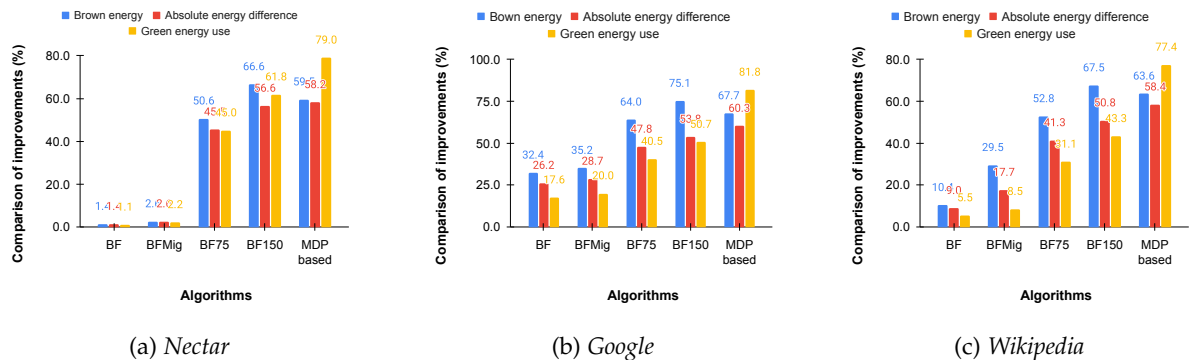


Fig. 9: Improvements in brown energy consumption, absolute energy difference, and green energy use normalized to the naive *LB* algorithm for various algorithms for all three workloads.

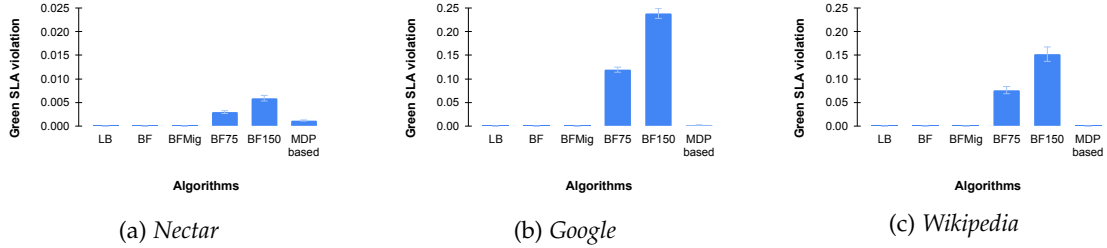


Fig. 10: *Green SLA violation ratio of various algorithms for all three workloads. No Green SLA violation is reported for the LB and BF approaches as they do not apply overbooking.*

Workload Type	Brown energy	Absolute difference	Green energy
<i>Nectar</i>	59.5	58.2	79.0
<i>Google</i>	67.7	60.3	81.8
<i>Wikipedia</i>	63.6	58.4	77.4

TABLE 4: Comparison of improvements (%) in *MDP based* algorithm for all three workloads compared to the naive *LB* algorithm. *MDP based* algorithm shows promising improvements in reducing the power use from the brown sources, absolute energy difference (measure of how energy consumption matched with renewable energy supply, bigger value means better matching), and the proportion of green energy usage to the total energy use, for all three workloads.

violations (As shown in Fig. 10). This performance is 58.1%, 35.3%, and 53.2% better than the baseline BF for *Nectar*, *Google* and *Wikipedia*, respectively. The *MDP-based* approach also results in the best match with the available renewable energy by 58.2%, 60.3%, and 58.4% for *Nectar*, *Google* and *Wikipedia* workloads, respectively. This is 56.8%, 34.1%, and 49.4% better than BF, and 1.6%, 6.5%, and 7.6% better than BF150 for *Nectar*, *Google* and *Wikipedia* workloads.

The percentage of renewable energy use is significantly higher for the *MDP-based* strategy (79.0%, 81.8%, 77.4% for various workloads, respectively). It outperforms the always most heavily overbooked *BF150* strategy by 17.2%, 31.1%, and 34.1%, respectively. This improvement is more significant when we compared our strategy with the *BF* non-overbooking strategy (77.9%, 64.2%, and 71.9% respectively). Our results demonstrate that the renewable energy usage improvement is higher when the overall workload is higher in the system. Hence, the *Nectar* workload, which generates a consistently higher workload demand, produces the best improvement in this case. A summary of these comparisons is also shown in Table 4.

5.7.6 Impact of ρ

The trend lines displayed in Fig. 11a illustrate how power consumption from brown energy sources varies with changes in the ρ factor. The trend lines for all three workloads demonstrate that brown energy consumption gradually decreases as the value of ρ increases. This is due to the fact that higher rewards are attained by using less brown energy. However, achieving these higher rewards involves adopting a higher degree of overbooking, which comes with a risk of higher *Green SLA* violations. Upon analyzing the results, we discovered that our algorithm was able to reduce brown energy consumption to levels as low as *BF150*.

Fig. 11b shows how absolute energy difference (computed by Equation 18) changes with varying ρ factor in the

MDP-based algorithm. The trend lines for all three workloads indicate that this difference decreases slightly after a certain value of ρ , as higher rewards are achieved when energy consumption is closer to, though still higher than, the available green power.

Fig. 11c illustrates how the percentage of energy consumption from green sources varies with ρ factor using the *MDP-based* algorithm. The trend lines for all three workloads indicate that the energy utilization from green sources slightly increases with higher values of ρ . This is because the algorithm rewards situations that are more environmentally friendly, with lower brown energy usage and higher green energy utilization. Thus, a higher ρ value encourages greater use of green energy sources.

Our findings regarding *Green SLA* violations align with our other observations. As depicted in Fig. 11d, we observed a direct relationship between increasing values of ρ and higher levels of *Green SLA* violations, which can be attributed to the increased levels of overbooking that are encouraged by higher ρ values. This trend was consistent across all three workloads, except for the *Nectar* workload. This occurs because the energy consumption from *Nectar* consistently exceeds the input of renewable energy generated, allowing for permissible overbooking at each time slot. Consequently, there are fewer instances of violations resulting from unnecessary overbooking.

5.8 Discussions

This section summarizes the notable advantages and limitations of the proposed method compared to alternative methods and discusses the challenges and difficulties in applying it in real-world environments.

Key Advantages: The proposed method presents several significant advantages over existing approaches that focus on adapting data center loads to match the renewable energy supply. Firstly, our methodology is distinguished by dynamically adjusting the degree of overbooking based on the availability of renewable energy, eliminating the need to reject any workload demand. This unique contribution has not been previously explored in the literature.

Secondly, our proposed approach reduces overbooking and over-subscription of resources when local renewable energy generation is abundant, and it provides a framework for flexible service level agreements based on the availability of renewable energy.

Lastly, heuristic methods have limitations, such as their inability to analyze the entire solution space effectively, and their effectiveness heavily relies on the strength of the heuristic within the context of the problem being solved.

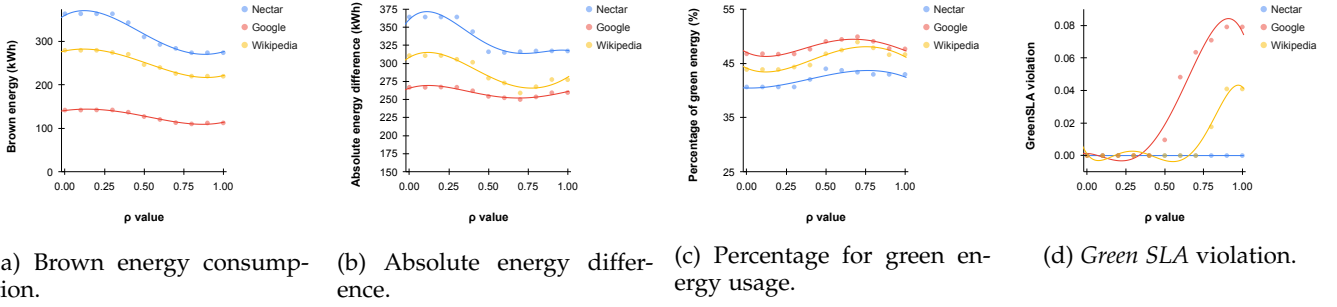


Fig. 11: Trend line of power consumption from brown energy sources, absolute energy difference, green energy use, and violating *Green SLA* when serving Google, Wikipedia, and *Nectar* workloads with varying ρ value applied in reward mechanism through MDP based algorithm.

Consequently, the performance of heuristic methods cannot be guaranteed. In response to the non-deterministic variability of workload demand and renewable energy supply, we propose using stochastic processes to model these complex phenomena accurately. By doing so, our approach offers a more robust and effective solution than traditional heuristic methods.

Challenges: This work promotes the use of on-site renewable generation to accelerate decarbonization and enable greener services, without the need for local energy storage. This approach is particularly suited for small-scale (micro) data centers located at edge locations, a market that is rapidly growing [56]. This, however, is not without challenges when putting the proposed method into practice, including: a) dealing with the heterogeneity of resources and applications in large-scale data centers, b) the “curse of dimensionality” that our MDP-based solution may confront, when the number of states and actions to be considered grows exponentially with the increasing number of dimensions or variables [57], c) the impact of aggressive consolidation on the wear-and-tear, and lifespan of physical servers due to frequent on-off cycles, and d) the MDP approach necessitates prior knowledge of the probability distribution related to the availability of renewable energy and workload variations, which can be challenging to obtain in some cases. Reinforcement learning methods such as those mentioned in [28], which can operate effectively in unknown environments, can address this limitation.

While large-scale data centers may comprise a variety of heterogeneous servers and switches, devices in small-scale data centers are more likely to be homogeneous. Therefore, we focused on a homogeneous configuration and developed our strategies for this simplified setting. However, our work can be readily extended to model heterogeneous scenarios as a finite set of homogeneous subsets, without compromising the general structure of the solution.

Our proposed MDP-based solution presents challenges associated with the “curse of dimensionality” [57] [34], as noted above. While our solution only has three dimensions in states and one dimension in action, we discretize the workload and energy into a few coarsely grained levels (e.g., 10) to reduce the number of possible states and manage complexity. As a result, the optimality of the decision-making process is impacted by introducing bounded errors. However, it is still important to carefully select variables and use techniques such as dimensionality reduction to

ensure reasonable computational complexity when seeking an optimal solution.

We conducted experiments using seven days of renewable energy data collected over three consecutive years and real-world workloads. We varied the strategies while maintaining the same experimental environment and achieved promising results with our algorithm compared to alternatives for all tested workloads in the proposed experimental settings, which strengthens the validity of our conclusions.

Although our experiments were conducted in specific settings, the overall trends remain consistent. We used a realistic computing environment, including devices and power consumption as described, and adopted all setting values from highly cited publications using experimental configurations from relevant domains and common practices, ensuring that our approach performs well in a real-world scenario. Additionally, we compared our experimental results with baseline algorithms to validate that our proposed algorithm’s performance was within a viable order of magnitude. However, since the objective of an MDP algorithm is to optimize performance over longer time spans of operation, such a comparison cannot be considered as a general measure of relative performance and may vary based on the experimental design, experimental settings and time span of experiments. To identify conditions where MDP-based algorithms provide decisive performance advantages over greedy algorithms, a specific experimental design would be necessary that falls outside the scope of this work.

6 CONCLUSIONS AND FUTURE DIRECTIONS

This paper addresses the challenge of powering cloud data centers with renewable energy and proposes a framework that dynamically manages resources and leverages workload and energy source information. The main issue we address is determining when and in what capacity overbooking of resources should be applied to better match the power demand of the workload with the available renewable energy supply. Our proposed approach, based on a Markov Decision Process (MDP), effectively addresses this issue. The proposed method stands out as unique by dynamically adjusting the overbooking degree based on renewable energy availability. It uses stochastic processes to model non-deterministic variability, offering a more robust solution than traditional heuristic methods. Our findings

suggest that the proposed approach can significantly improve the utilization of green energy and reduce the usage of non-renewable energy. In the future, we will investigate Reinforcement Learning-based approaches to dynamically set optimal overbooking levels, consider the addition of battery storage, and also investigate prediction-based decision approaches.

ACKNOWLEDGMENT

The authors would like to thank the *Nectar Cloud* team at Monash University for sharing their workload data for our experiment, *Monash Graduate Research* for funding, and *Grammarly.com* for helping to fine-tune the language and grammar.

REFERENCES

- [1] K. Zheng, X. Wang, L. Li, and X. Wang, "Joint power optimization of data center network and servers with correlation analysis," in *INFOCOM 2014-IEEE Conference on Computer Communications*, pp. 2598–2606.
- [2] L. A. Barroso, J. Clidaras, and U. Hözl, "The datacenter as a computer: An introduction to the design of warehouse-scale machines," *Synthesis lectures on computer architecture*, vol. 8, no. 3, pp. 1–154, 2013.
- [3] L. Tomás and J. Tordsson, "Improving cloud infrastructure utilization through overbooking," in *Proceedings of the 2013 ACM Cloud and Autonomic Computing conference*, 2013, pp. 1–10.
- [4] J. J. Son, A. V. Dastjerdi, R. N. Calheiros, and R. Buyya, "Sla-aware and energy-efficient dynamic overbooking in sdn-based cloud data centers," *IEEE Transactions on Sustainable Computing*, vol. 2, no. 2, pp. 76–89, 2017.
- [5] T. Chakraborty, A. N. Toosi, and C. Kopp, "Elastic power utilization in sustainable micro cloud data centers," *IEEE Transactions on Sustainable Computing*, 2023.
- [6] Z. Cao and S. Dong, "An energy-aware heuristic framework for virtual machine consolidation in cloud computing," *The Journal of Supercomputing*, vol. 69, no. 1, pp. 429–451, 2014.
- [7] M. Sabbir Hasan and E.-N. Huh, "Heuristic based energy-aware resource allocation by dynamic consolidation of virtual machines in cloud data center," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 7, no. 8, pp. 1825–1842, 2013.
- [8] M. Abdullah, K. Lu, P. Wieder, and R. Yahyapour, "A heuristic-based approach for dynamic vms consolidation in cloud data centers," *Arabian Journal for Science and Engineering*, vol. 42, no. 8, pp. 3535–3549, 2017.
- [9] K. Le, R. Bianchini, T. D. Nguyen, O. Bilgir, and M. Martonosi, "Capping the brown energy consumption of internet services at low cost," in *Int. Conf. on Green Computing*. IEEE, 2010, pp. 3–14.
- [10] A. Celesti, A. Puliafito, F. Tusa, and M. Villari, "Energy sustainability in cooperating clouds," in *CLOSER*, 2013, pp. 83–89.
- [11] W. Li, T. Yang, F. C. Delicato, P. F. Pires, Z. Tari, S. U. Khan, and A. Y. Zomaya, "On enabling sustainable edge computing with renewable energy resources," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 94–101, 2018.
- [12] A. N. Toosi, C. Qu, M. D. de Assunção, and R. Buyya, "Renewable-aware geographical load balancing of web applications for sustainable data centers," *Journal of Network and Computer Applications*, vol. 83, pp. 155–168, 2017.
- [13] F. Farahnakian, T. Pahlkalla, P. Liljeberg, J. Plosila, N. T. Hieu, and H. Tenhunen, "Energy-aware vm consolidation in cloud data centers using utilization prediction model," *IEEE Transactions on Cloud Computing*, vol. 7, no. 2, pp. 524–536, 2016.
- [14] Í. Goiri, K. Le, M. E. Haque, R. Beauchea, T. D. Nguyen, J. Guittart, J. Torres, and R. Bianchini, "Greenslot: scheduling energy consumption in green datacenters," in *Proceedings of Int. Conf. for High Performance Comp., Networking, Storage and Analysis*, 2011, pp. 1–11.
- [15] M. S. Hasan, F. A. de Oliveira, T. Ledoux, and J.-L. Pazat, "Enabling green energy awareness in interactive cloud application," in *Int. Conf. on Cloud Computing Technology and Science*. IEEE, 2016, pp. 414–422.
- [16] A. Khosravi, A. Nadjaran Toosi, and R. Buyya, "Online virtual machine migration for renewable energy usage maximization in geographically distributed cloud data centers," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 18, p. e4125, 2017.
- [17] X. Wang, Y. Yao, X. Wang, K. Lu, and Q. Cao, "Carpo: Correlation-aware power optimization in data center networks," in *2012 Proceedings IEEE INFOCOM*. IEEE, 2012, pp. 1125–1133.
- [18] B. Heller et al., "Elastictree: Saving energy in data center networks," in *Nsdi*, vol. 10, 2010, pp. 249–264.
- [19] R. Cziva, S. Jouët, D. Stapleton, F. P. Tso, and D. P. Pezaros, "Sdn-based virtual machine management for cloud data centers," *IEEE Transactions on Network and Service Management*, vol. 13, no. 2, pp. 212–225, 2016.
- [20] A. Arias Maestro, Ó. Sanjuán Martínez, A. M. Teredesai, and V. García-Díaz, "Blockchain based cloud management architecture for maximum availability," 2023.
- [21] H. Jin, T. Cheocherngngarn, D. Levy, A. Smith, D. Pan, J. Liu, and N. Pissinou, "Joint host-network optimization for energy-efficient data center networking," in *2013 IEEE 27th International Symposium on Parallel and Distributed Processing*. IEEE, 2013, pp. 623–634.
- [22] W. Fang, X. Liang, S. Li, L. Chiaraviglio, and N. Xiong, "Vm-planner: Optimizing virtual machine placement and traffic flow routing to reduce network power costs in cloud data centers," *Computer Networks*, vol. 57, no. 1, pp. 179–196, 2013.
- [23] A. Beloglazov and R. Buyya, "Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 7, pp. 1366–1379, 2012.
- [24] A. C. Ammari, W. Labidi, F. Mnif, H. Yuan, M. Zhou, and M. Sarrah, "Firefly algorithm and learning-based geographical task scheduling for operational cost minimization in distributed green data centers," *Neurocomputing*, vol. 490, pp. 146–162, 2022.
- [25] H. Yuan, J. Bi, and M. Zhou, "Geography-aware task scheduling for profit maximization in distributed green data centers," *IEEE Transactions on Cloud Computing*, vol. 10, no. 3, pp. 1864–1874, 2020.
- [26] H. Yuan, J. Bi, M. Zhou, Q. Liu, and A. C. Ammari, "Biobjective task scheduling for distributed green data centers," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 2, pp. 731–742, 2020.
- [27] W. Li, Y. Xia, M. Zhou, X. Sun, and Q. Zhu, "Fluctuation-aware and predictive workflow scheduling in cost-effective infrastructure-as-a-service clouds," *IEEE Access*, vol. 6, pp. 61 488–61 502, 2018.
- [28] G. Zhou, W. Tian, R. Buyya, R. Xue, and L. Song, "Deep reinforcement learning-based methods for resource scheduling in cloud computing: A review and future directions," *Artificial Intelligence Review*, vol. 57, no. 5, p. 124, 2024.
- [29] Y. Zhang, J. Yao, and H. Guan, "Intelligent cloud resource management with deep reinforcement learning," *IEEE Cloud Computing*, vol. 4, no. 6, pp. 60–69, 2017.
- [30] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. Andrew, "Geographical load balancing with renewables," *SIGMETRICS Performance Evaluation Review*, vol. 39, no. 3, pp. 62–66, 2011.
- [31] M. Lin, Z. Liu, A. Wierman, and L. L. Andrew, "Online algorithms for geographical load balancing," in *International green computing conference*. IEEE, 2012, pp. 1–10.
- [32] A. N. Toosi and R. Buyya, "A fuzzy logic-based controller for cost and energy efficient load balancing in geo-distributed data centers," in *Int. Conf. on Utility and Cloud Computing*. IEEE, 2015, pp. 186–194.
- [33] M. Xu, A. N. Toosi, B. Bahrani, R. Razzaghi, and M. Singh, "Optimized renewable energy use in green cloud data centers," in *Service-Oriented Computing: 17th International Conference, IC-SOC 2019, Toulouse, France, October 28–31, 2019, Proceedings 17*. Springer, 2019, pp. 314–330.
- [34] O. Dogru, J. Xie, O. Prakash, R. Chiuplunkar, J. Soesanto, H. Chen, K. Velswamy, F. Ibrahim, and B. Huang, "Reinforcement learning in process industries: Review and perspective," *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 2, pp. 283–300, 2024.
- [35] Z. Wu and Y. Wang, "Process industry scheduling based on graph neural network and reinforcement learning," in *2024 39th Youth Academic Annual Conference of Chinese Association of Automation (YAC)*. IEEE, 2024, pp. 1591–1596.
- [36] VMware, "Performance best practices for vmware vsphere 6.7," *VMware Inc.*, pp. 1–88, 2018.
- [37] AWS, "Spot instances," 2022, <https://docs.aws.amazon.com/-/AWSEC2/latest/UserGuide/using-spot-instances.html/>.

- [38] A. A. Chien, "Good, better, best: how sustainable should computing be?" *Communications of the ACM*, vol. 64, no. 12, pp. 6–7, 2021.
- [39] A. Karimafshar et al., "A request dispatching method for efficient use of renewable energy in fog computing environments," *Future Generation Computer Systems*, vol. 114, pp. 631–646, 2021.
- [40] A. N. Toosi, C. Agarwal, L. Mashayekhy, S. K. Moghaddam, R. Mahmud, and Z. Tari, "Greenfog: A framework for sustainable fog computing," in *In Proceedings of the 17th International Conference on Service-Oriented Computing (ICSOC' 22)*. Springer, 2022.
- [41] S. Pelley, D. Meisner, T. F. Wenisch, and J. W. VanGilder, "Understanding and abstracting total data center power," in *Workshop on Energy-Efficient Design*, vol. 11, 2009, pp. 1–6.
- [42] A. Jayanetti and R. Buyya, "J-opt: A joint host and network optimization algorithm for energy-efficient workflow scheduling in cloud data centers," in *Proceedings of the IEEE/ACM UCC*, 2019, pp. 199–208.
- [43] T. Chakraborty, A. N. Toosi, C. Kopp, P. Stuckey, and J. Mahet, "Joint host-network power scaling with minimizing vm migration in sdn-enabled cloud data centers," in *IEEE/ACM International Conference on Utility and Cloud Computing*, 2020, pp. 1–12.
- [44] A. Shehabi, S. Smith, D. Sartor, R. Brown, M. Herrlin, J. Koomey, E. Masanet, N. Horner, I. Azevedo, and W. Lintner, "United states data center energy usage report," 2016.
- [45] J. D. Moore, J. S. Chase, P. Ranganathan, and R. K. Sharma, "Making scheduling "cool": Temperature-aware workload placement in data centers," in *USENIX annual technical conference*, 2005, pp. 61–75.
- [46] S. Ilager, K. Ramamohanarao, and R. Buyya, "Etas: Energy and thermal-aware dynamic virtual machine consolidation in cloud data center with proactive hotspot mitigation," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 17, p. 5221, 2019.
- [47] T. He, A. N. Toosi, and R. Buyya, "Camig: Concurrency-aware live migration management of multiple virtual machines in sdn-enabled clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 10, pp. 2318–2331, 2022.
- [48] H. Liu, H. Jin, X. Liao, L. Hu, and C. Yu, "Live migration of virtual machine based on full system trace and replay," in *Proceedings of the 18th ACM International Symposium on High Performance Distributed Computing*, ser. HPDC '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 101–110. [Online]. Available: <https://doi.org/10.1145/1551609.1551630>
- [49] M. Nelson, B.-H. Lim, G. Hutchins et al., "Fast transparent migration for virtual machines," in *USENIX Annual technical conference, general track*, 2005, pp. 391–394.
- [50] H. Liu, C.-Z. Xu, H. Jin, J. Gong, and X. Liao, "Performance and energy modeling for live migration of virtual machines," in *Proceedings of the 20th International Symposium on High Performance Distributed Computing*, ser. HPDC '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 171–182. [Online]. Available: <https://doi.org/10.1145/1996130.1996154>
- [51] C. Reiss, J. Wilkes, and J. L. Hellerstein, "Google cluster-usage traces: format+ schema," *Google Inc., White Paper*, pp. 1–14, 2011.
- [52] WikiBench, "Wikipedia access traces," 2021, [www.wikibench.eu/\\$?page_id=60](http://www.wikibench.eu/$?page_id=60), Accessed: June '21.
- [53] G. Urdaneta, G. Pierre, and M. Van Steen, "Wikipedia workload analysis for decentralized hosting," *Computer Networks*, vol. 53, no. 11, pp. 1830–1845, 2009.
- [54] "Nectar," <https://ardc.edu.au/services/nectar-research-cloud/>.
- [55] L. Tomàs and J. Tordsson, "An autonomic approach to risk-aware data center overbooking," *IEEE Transactions on Cloud Computing*, vol. 2, no. 3, pp. 292–305, 2014.
- [56] "Micro data center market is going macro," <https://www.vertiv.com/en-emea/about/news-and-insights/articles/blog-posts/why-the-micro-data-centre-market-is-going-macro/>, Aug. 05 2021.
- [57] R. Bellman, "Dynamic programming princeton university press princeton," *New Jersey Google Scholar*, 1957.



Tuhin Chakraborty received his PhD degree from the Department of Software Systems and Cybersecurity, Faculty of Information Technology, Monash University, Australia. The main area of his doctoral research was Sustainable Cloud Computing. His research interests include Green computing, Accessible computing, and Human-computer interaction. He has published in, and been a peer reviewer for, upper tier venues such as ACM SIGCHI, ACM Transaction on Accessible Computing, IEEE Transactions on Sustainable Computing, Mobile HCI, and ACM DIS.



Carlo Kopp (M'98—SM'10) received a BE with First Class Honours in Electrical Engineering from the University of Western Australia, in 1984, and MSc and PhD degrees in Computer Science from Monash University, in 1996 and 2000, respectively. He is with the Department of Software Systems and Cybersecurity at Monash University in Melbourne, Australia. His doctoral work was on the adaptation of active phased arrays for Gigabit/s data rate airborne networking applications, and related low elevation angle tropospheric microwave propagation problems. He has previously conducted research in operating systems, ad hoc networks and radio-frequency propagation, radar signature computational modelling, optical communications, and satellite navigation support protocols. Dr Kopp has a total of around 670 publications in multiple categories across the full breadth of his research interests, including 67 peer reviewed research papers, and he co-authored Chapter 5 in the third edition of Skolnik's *Radar Handbook*. Dr Kopp has extensive industry experience as a designer, developer and integrator of hard real-time embedded software, operating systems internals software, especially Unix device drivers, communications protocols, high speed digital hardware including Emitter Coupled Logic, SPARC motherboards, graphics adaptors, and analogue electronic hardware including wideband optical fibre receivers and transmitters. Dr Kopp is an Associate Fellow of the AIAA and a Senior Member of the IEEE.



Adel N. Toosi (Member, IEEE) received his PhD from the School of Computing and Information Systems (CIS) at the University of Melbourne in 2015. He is currently an Associate Professor in CIS within the Faculty of Engineering and Information Technology at the University of Melbourne, Australia. From 2018 to 2024, he was affiliated with Monash University as Lecturer and Senior Lecturer. Dr. Toosi has authored over 80 peer-reviewed papers in prestigious journals and conferences, including IEEE Transactions on Cloud Computing, IEEE Transactions on Services Computing, and IEEE Transactions on Sustainable Computing, among others. His work has garnered over 5,000 citations, with a current h-index of 30 (Google Scholar). He has received several notable awards and honors, including the AusPDC'21 and UCC'23 Best Paper Awards and a Best Paper Candidate recognition at ICSOC'21. He has secured funding for numerous projects, totaling over \$1.5 million, including grants from the Australian Research Council (ARC) Discovery and Linkage programs. Dr. Toosi serves on the editorial board of Future Generation Computer Systems (FGCS) and has guest-edited several special issues. He has also organized multiple workshops in his field, including the Starless workshops at PerCom 2022 to 2024. His research interests span cloud computing, edge computing, software-defined networking, serverless computing, green computing, and energy efficiency. Currently, he is focused on developing sustainable edge/fog computing environments. For more information, please visit his homepage at <http://adelnadarantoosi.info>.