energies

MDPI

Article

Energy-Aware Scheduling Based on Marginal Cost and Task Classification in Heterogeneous Data Centers †

Kaixuan Ji [1,2,‡], Ce Chi [1,2,‡], Fa Zhang [1,‡], Antonio Fernández Anta [3], Penglei Song [4,‡], Avinab Marahatta [5], Youshi Wang [6] and Zhiyong Liu [1,*,‡]

1 High Performance Computer Research Center, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100095, China; jikaixuan@ict.ac.cn (K.J.); chice18s@ict.ac.cn (C.C.); zhangfa@ict.ac.cn (F.Z.)
2 School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 101408, China
3 IMDEA Networks Institute, Avda. del Mar Mediterraneo, 22, 28918 Leganes, Spain; antonio.fernandez@imdea.org
4 Information Engineering College, Capital Normal University, Beijing 100048, China; 2191002021@cnu.edu.cn
5 Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China; avinab.marahatta@iie.ac.cn
6 Meituan-Dianping Group, Beijing 100102, China; wangyoushi@meituan.com
* Correspondence: zyliu@ict.ac.cn; Tel.: +86-13521629531
† This paper is an extended version of our paper published in the 8th International Workshop on Energy-Efficient Data Centers (E2DC2020) at the Eleventh ACM International Conference on Further Energy Systems (e-Energy' 20), Virtual Event, Melbourne, Australia, 22–26 June 2020; pp. 482–488.
‡ Current Address: No.6 South Kexueyuan Rd, Beijing 100190, China.

Citation: Ji, K.; Chi, C.; Zhang, F.; Anta, A.F.; Song, P.; Marahatta, A.; Wang, Y.; Liu, Z. Energy-Aware Scheduling Based on Marginal Cost and Task Classification in Heterogeneous Data Centers. *Energies* **2021**, *14*, 2382. https://doi.org/10.3390/en14092382

Academic Editor: Valentina Colla

Received: 22 March 2021
Accepted: 17 April 2021
Published: 22 April 2021

Abstract: The energy consumption problem has become a bottleneck hindering further development of data centers. However, the heterogeneity of servers, hybrid cooling modes, and extra energy caused by system state transitions increases the complexity of the energy optimization problem. To deal with such challenges, in this paper, an Energy Aware Task Scheduling strategy (EATS) utilizing marginal cost and task classification method is proposed that cooperatively improves the energy efficiency of servers and cooling systems. An energy consumption model for servers, cooling systems, and state transition is developed, and the energy optimization problem in data centers is formulated. The concept of marginal cost is introduced to guide the task scheduling process. The task classification method is incorporated with the idea of marginal cost to further improve resource utilization and reduce the total energy consumption of data centers. Experiments are conducted using real-world traces, and energy reduction results are compared. Results show that EATS achieves more energy-savings of servers, cooling systems, state transition in comparison to the other two techniques under a various number of servers, cooling modules and task arrival intensities. It is validated that EATS is effective at reducing total energy consumption and improving the resource utilization of data centers.

Keywords: data center; energy-aware; marginal cost; task scheduling; cooling system; task classification

1. Introduction

As crucial infrastructure, data centers have been exponentially developing with the rapid innovations in cloud computing technology [1]. The data center provides immense computing and storage resources for cloud users to meet their increasing demands. However, power-hungry and environmental footprint issues are impeding the further development of data centers [2]. Recent statistics indicate that the data center's power demands will increase more than 66% over the period 2011–2035 [1]. The electricity consumed by U.S. data centers was approximately 70 billion kilowatt-hours in 2014, accounting for approximately 1.8% of the total electricity consumed in the U.S [3]. Besides, according

to reports, the proportion of worldwide annual carbon emissions generated by data centers is 0.3% [4]. Rising energy consumption and carbon emissions in data centers are the prominent problems limiting the further expansion of data centers. The huge energy consumption and serious environmental problems of data centers have aroused extensive academic research. Therefore, to solve these problems, it is important to develop an effective energy-efficient strategy.

The energy consumption of data centers mainly comes from two components, information technology (IT) systems and cooling systems [5]; the energy consumption of servers is dominant among the IT devices (e.g., servers, network devices and security devices). Recent studies show that servers occupy about 56% of total data center energy, as idle servers still consume a vast amount of energy [6]. Thus, optimizing the high energy consumption of servers is necessary. Apart from server power consumption, cooling systems are another high energy consumer in data centers, which account for almost 30% of the data center's energy use [6]. Therefore, it is vital to reduce the energy consumption of cooling systems as well.

To solve the huge energy consumption issue of data centers, many works propose techniques to reduce the energy consumption by the IT systems and cooling systems. For the energy optimization of servers, optimizing the task scheduling strategy based on an energy consumption model of servers is an effective means. In addition, to optimize the energy consumption of cooling systems, it is useful to control the cooling systems dynamically according to the energy consumption model of cooling systems [6]. However, servers and cooling systems are related in a data center—that is, cooling systems are responsible for removing the heat generated by servers. Thus, optimizing the energy efficiency of servers and cooling systems independently easily results in additional energy waste and leads to a suboptimal solution [7]. Individually optimizing the energy efficiency of servers may produce local hot spots. Solely considering the energy efficiency optimization of the cooling system may lead to inadequate cooling in the data center. Therefore, it is necessary to improve the energy efficiency of servers and cooling systems cooperatively.

There exist four challenges in the joint energy optimization of data centers. First, to optimize the energy consumption of the IT systems and cooling systems cooperatively, it is necessary to consider the different energy consumption behaviors of the various kinds of IT devices and cooling systems, which increases the complexity of the optimization of the data center's energy consumption. The servers and cooling systems are treated as different energy-consuming devices with respect to energy, as they usually have different energy use behaviors according to task scheduling. Thus, it is difficult to evaluate the energy changes of IT systems and cooling systems. Second, the workloads in the cloud often vary over time, and the resource requirements, arrival rates and run times have large variations. This brings challenges to the energy optimization. Therefore, it is necessary to classify tasks according to their different characteristics. Third, to improve the resource utilization in data centers, dynamic power management together with server consolidation can be used to reduce power consumption. However, server consolidation and frequently switching system states will cause extra energy costs and time delays. Improving the resource utilization without leading to rising energy costs is challenging. Fourth, there are multiple cooling modes accompanied by various cooling capacities and levels of cooling effectiveness. Finding out how to apply numerous cooling modes collaboratively to maximize the energy efficiency of cooling systems is another crucial problem.

To deal with the aforementioned challenges, an energy aware task scheduling (EATS) strategy utilizing marginal cost and task classification is proposed in this paper, which aims at maximizing the energy efficiency of servers and cooling systems, and reducing the energy caused by state transitions; it also considers the heterogeneity of servers and workloads.

Herein, first, the joint energy consumption model of a data center, including servers, cooling systems and the energy caused by system state transitions is developed, by which hybrid cooling modes, including the outside air cooling mode and chilled water cooling mode, are jointly applied, and the optimal cooling allocation between the two cooling

modes is derived. Second, the concept of marginal cost is introduced to guide the task scheduling. Servers and cooling systems are different energy-consuming devices with different energy behaviors in the process of task scheduling. By using the concept of marginal cost, the difficulty of cooperatively optimizing the energy of servers and cooling systems due to the system heterogeneity can be solved. Third, marginal cost is incorporated into the task classification method to further improve the resource utilization and reduce the total energy consumption of data centers. Finally, for solving the data center energy consumption minimization problem, an energy-aware scheduling strategy based on the marginal cost and task classification method is provided. Experiments were conducted utilizing two datasets (Google cluster data [8] and Alibaba cluster data [9]), and the results indicate the validity of EATS for improving the energy efficiency of data centers in comparison with other algorithms. With the scheduling strategy, a task will be allocated to the most energy-efficient server resources and cooling system resources, so that the energy consumption of data centers is reduced and the resource utilization is improved.

The major contributions of this paper are shown as follows:

(1) Jointly considering the energy optimization of servers and cooling systems, and the system state transition cost. The total energy consumption model, including the server model, the state transition energy model and the cooling system model, was developed.
(2) A cooling model was developed which adopts two different cooling modes, and a strategy for dynamically adjusting various cooling modes based on real-time workload characteristics is proposed.
(3) The concept of marginal cost in data centers is introduced to guide the task scheduling process. The task classification method is combined with marginal cost evaluation to further improve resource utilization.
(4) An energy-aware task scheduling strategy using the marginal cost evaluation and the task classification method is proposed to solve the energy minimization problem of the data center and optimize the energy consumption caused by state switching.

The contents of each section are presented as follows. Section 2 introduces the related work about the data center energy efficiency optimization techniques. Section 3 shows the structure of a data center. The power models and energy minimization problem of the data center are developed in Section 4. Section 5 introduces the task scheduling strategy for solving the energy optimization problem. The experiment setup and experiment results are discussed in Section 6. Finally, Section 7 summarizes the main content of the paper and the future work.

2. Related Work

2.1. The Energy Optimization Techniques for IT Systems

There have been many works focusing on the optimization of IT system energy efficiency in data centers.

In [10], a scheduling algorithm considering multi-sleep modes of servers was proposed to minimize the energy consumption of servers while satisfying the QoS requirement. Reference [1] developed a joint energy consumption optimization scheme, taking into account servers' power, network power and workload migration cost; server heterogeneity was considered. In [11], a two-tier VM placement algorithm, including a queuing structure and a multi-objective virtual machine (VM) placement algorithm, was developed to improve the resource utilization and energy efficiency of the IT systems, and various migration techniques were discussed. A threshold-based dynamic algorithm DCABA was developed in [12] to optimize the operation cost and the active server number. Both load balance and server consolidation techniques were utilized to improve the effectiveness of the algorithm.

In [13], a harmony-inspired genetic task scheduling algorithm (HIGA) was proposed to reduce the total makespan and energy consumption of servers in a data center. In [14], the authors proposed a novel meta heuristic method to jointly optimize task scheduling and machine placement in cloud data centers. In [15], Medara et al. proposed a workflow

task scheduling algorithm with dynamic voltage and frequency scaling (DVFS), which focuses on optimizing energy efficiency and reliability. In [16], the authors formulated a mathematical model on the scheduling problem of optimizing energy under makespan and reliability constraints on heterogeneous multiprocessor systems. In [17], Liu et al. formulated a programming problem for minimizing the energy cost with the time constraint, which focuses on searching for the optimization to the scheduling problem using the Q-learning algorithm and reducing the long convergence time caused by a large amount of training considering heterogeneous and large-scale data centers. In [18], Deng et al. proposed a novel multi-objective optimization method that can jointly maximize the profit of the distributed green data centers (DGDCs) provider and minimize the average task loss possibility of tasks.

2.2. The Energy Optimization Techniques for Cooling Systems

A lot of research has also been done to improve the cooling energy efficiency in data centers.

In [19], the energy optimization of the air-side economizer in the modular data center was considered. An energy optimization method was proposed to decide on the optimal supply air temperature settings for the cooling system. In [20], a deep reinforcement learning (DRL)-based cooling control algorithm was developed to achieve the cooling energy savings of the data center. It is based on deep deterministic policy gradient (DDPG), which includes an evaluation network to predict the energy cost and a policy network trained to predict optimized control settings. In [21], the authors explain that the settings of free cooling mode switchover temperature and cooling water approach temperature in free cooling systems are typically fixed. To further optimize the cooling energy efficiency, a systematic model-based methodology is proposed to optimize the two kinds of temperature. JCWM [22] is a joint cooling and workload management algorithm considering the thermal effects of servers workloads and cooling systems to improve the cooling efficiency. In [23], the authors focused on reducing the data center energy consumption by resolving the characteristic of airflow and temperature distributions.

2.3. Joint Energy Optimization Techniques

However, solely optimizing the IT systems or cooling systems is still inefficient for data center energy reduction. In the worst case, if the IT power is optimized through excessive workload consolidation, cooling power may even increase due to more hot spots. Therefore, in order to optimize the overall energy consumption of data centers, there have been some works optimizing the IT systems and cooling systems together.

PowerTrade-d [24] aims to reduce the total power consumption of data centers by trading-off the cooling power and idle power of servers. In this scheme, the cooling environment in a data center is divided into cold zones, warm zones and hot zones; a centralized workload distribution strategy is utilized in cold zones and a balanced workload distribution strategy is utilized in warm zones. In [25], a real-time task scheduling algorithm was proposed, called rTCS, which optimizes the energy efficiency of the data centers by jointly considering the energy consumption of the servers and cooling systems. The main difference between rTCS in [25] and the proposed strategy in this paper is the use of marginal cost and a combination of marginal cost and task classification for further energy efficiency improvement. In [7], a cross-layer algorithm JOINT was developed to minimize the entire energy consumption of the data center by collaboratively optimizing the chip layer, server layer and room layer. Reference [26] presents a deep learning algorithm that provides a strategy to place the servers in a suitable location considering the effect of power and temperature. DeepEE was proposed in [27], which introduces DRL techniques to joint energy optimization of IT systems and cooling systems.

Differently from previous works, we optimize the IT and cooling systems for data centers from a marginal cost perspective. The problem of optimizing IT and cooling power consumption and server state transition costs is formalized and solved by our marginal

energy approach. Moreover, improving the resource utilization is also one of our goals, for which task classification is used and combined with our marginal energy method. Part of the work was presented in the 8th International Workshop on Energy-Efficient Data Centres (E2DC2020) at the Eleventh ACM International Conference on Future Energy Systems (e-Energy'20), 22–26 June 2020, Virtual Event, Australia [28].

3. The Structure of a Data Center

In this section, the structure of the data center is introduced. The structure of our data center mainly involves three parts (users, task processing modules, and cooling modules) to support the users' request scheduling.

(1) Users: Cloud users submit service requests from anywhere globally to the data center, and the service requests mainly include web-applications, etc.
(2) Task processing modules: Serves as the interface between the data center infrastructure and users, which requires the iteration of the following components to support the energy-aware scheduling.
    (a) Task observer: Observes the arriving tasks and conveys the information of tasks to the recording component, energy monitor and classification controller.
    (b) Energy monitor: Observes energy consumption caused by tasks, servers and cooling systems, and provides this information to the real-time scheduler to calculate the marginal energy of data centers and make energy-efficient task scheduling decisions.
    (c) Classification controller: It is responsible for getting a category label according to the run times and end times of tasks predicted by the prediction controller.
    (d) Real-time scheduler: Assigns tasks to servers and determines the server and cooling resources for the allocated tasks. It also determines when servers are powered on or powered off to satisfy the demand.
    (e) Recording and prediction controller: Recording monitors the actual resource usage and run times of the submitted tasks. Records the ID, type, run time, category label and resource usage of historical tasks. Additionally, the historical data of the resource usage and run time are applied to predict the run times of other tasks in the future in the prediction controller.
(3) Cooling modules: Each cooling module includes independent servers and sub-cooling systems.
    (a) Servers: the underlying servers provide the hardware infrastructure for supporting virtualized resources to meet task demands.
    (b) Cooling systems: it provides the heat dissipation function to dissipate the heat generated by the server, and it operates in a hybrid cooling mode.

The structure of the data center is indicated in Figure 1. We assume that the data center has N independent modules, called cooling modules [29]. Each cooling module comprises independent information technology equipment and sub-cooling systems. The cooling support of servers is provided by the sub-cooling systems of each module alone.

Assume that each module $C_i (i \in [1, N])$ consists of $M_i (i \in [1, N])$ servers. Each server can be denoted by $S_{ij} (i \in [1, N], j \in [1, M_i])$, which represents the jth server in the ith module. The servers are heterogeneous and have different computing resources and power efficiencies. The sub-cooling system in each module operates in hybrid-cooled modes, consisting of the outside air cooling (oac) and chilled water cooling (cwc). The structure of each cooling module is shown in Figure 2.

The task scheduling process is described as follows: when users submit the task to the data center, the information of the arriving task will be observed by the task observer and recorded by the recording part. Through analyzing the stored and new observed information from the recording part, the run times of the task will be obtained through the prediction controller that uses a machine learning method to predict the run times or from the history data. Then with the obtained value of run time, the task will be classified

and receive a category label based on the classification controller. The marginal energy value will be calculated based on the energy monitor. Finally, the task is scheduled to a suitable server and cooling module in real-time with the classified value and marginal energy evaluation.

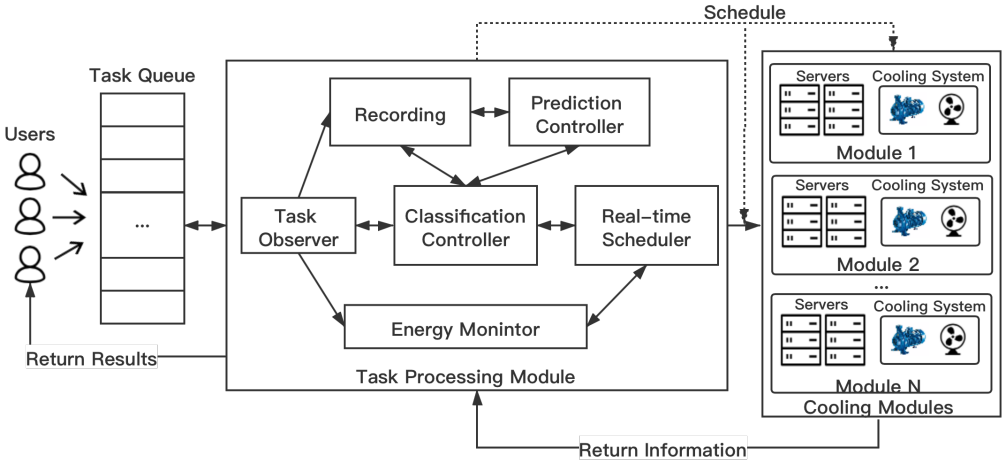


Figure 1. The structure of the data center.

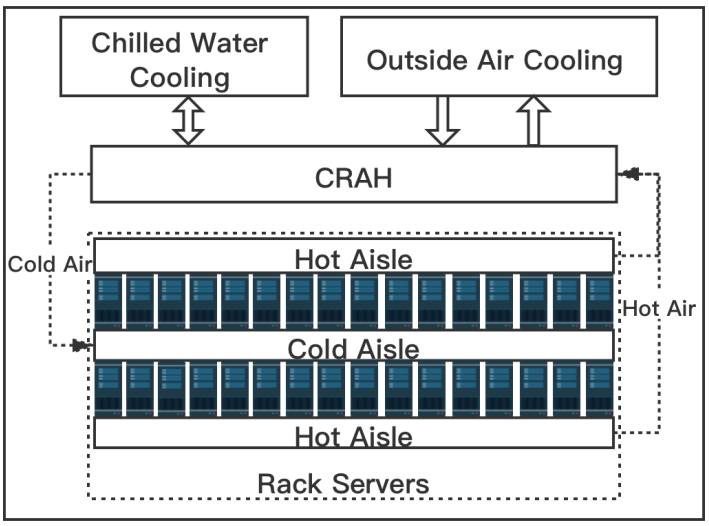


Figure 2. The structure of the cooling module.

4. Power Consumption Models in the Data Center

In this section, the power consumption models in a data center are developed. We first develop the power models consisting of servers, cooling systems and the state transition model. Then, the energy consumption model of the data center is modeled, and the data center energy minimization problem is formulated.

4.1. Workload and Server Power Consumption Model

Assume that each task can be denoted by $t_a (a \in 0, 1, 2, 3...)$. The run times and end times of a task are represented by r_{t_a} and e_{t_a}, respectively. It can be concluded that based on the historical data collected and analyzed in the recording and prediction controller, or using the machine learning techniques, the proximate run times of a task can be acquired [30]. Thus, r_{t_a} can be obtained through the recording and prediction controller. Predictions of the run time and end time of the task can be conducted in the prediction controller, and that is not the primary research focus of this paper. $R_{t_a}^{cpu}$ represents the CPU resource requirement of tasks t_a. Tasks submitted by cloud users to the data center will be sent to servers that

provide the hardware infrastructure and resources. Assume that $RT_{s_{ij}}^{cpu}$ and $RA_{s_{ij}}^{cpu}$ express the total CPU resource capacity and available CPU resources at the current time of servers. Thus, when tasks are scheduled to a server, the resource constraint Equation (1) should be satisfied.

$$RA_{s_{ij}}^{cpu} - R_{t_a}^{cpu} \geq 0, \tag{1}$$

The power consumption of servers is related to the idle power and the current CPU utilization of the servers. Based on [31], the power consumption of servers can be denoted by Equation (2)

$$P_{s_{ij}}^{t} = \begin{cases} 0 & r_{s_{ij}}^{t} = 0 \\ P_{s_{ij}}^{idle} + (P_{s_{ij}}^{full} - P_{s_{ij}}^{idle}) \times (2r_{s_{ij}}^{t} - (r_{s_{ij}}^{t})^{1.4}) & r_{s_{ij}}^{t} \neq 0 \end{cases} \tag{2}$$

where $r_{s_{ij}}^{t}$ represents the CPU resource utilization of the server S_{ij} at the time slot t, and $r_{s_{ij}}^{t} = (1 - RA_{s_{ij}}^{cpu})/RT_{s_{ij}}^{cpu}$. The idle power consumption of a server is denoted by $P_{s_{ij}}^{idle}$, and the peak power is represented by $P_{s_{ij}}^{full}$. Assume that servers only have two operating states, inactive and active. If the power consumption of a server is 0, the server is in an inactive state; otherwise, the server is in an active state.

4.2. Power Consumption Models of Cooling Systems

In the data center, there are two common cooling systems: one is the outside air cooling system; the other is the chilled water cooling system. We first introduce the two cooling modes, and then develop a final cooling system model.

4.2.1. The Outside Air Cooling System

In this cooling mode, the energy used is mainly consumed by the blowers, which can be approximated as a cubic function of the blower speed [32]. By referring to the basic thermal transfer theory and the general fan laws, we can find that the power consumption consumed by the outside air cooling (oac) system is a convex function of the server power consumption and shown as follows [25]:

$$P_{c_i,oac} = k(P_{s_i,oac})^3, \qquad k > 0 \tag{3}$$

where the parameter k is proportional to the temperature difference between the outside air temperature and the server exhausting air temperature. $P_{s_i,oac}$ represents the total power consumption of the servers whose cooling function is supported by the outside air cooling mode.

4.2.2. The Chilled Water Cooling System

The traditional chilled water cooling system mainly comprises chillers, cooling towers and pumps. It is responsible for cooling the hot air extracted from the raised floor and server inlet through the computer room air handler (CRAH) [33].

The power consumption calculation of the chiller water cooling is very complicated. The typical chilled water cooling is usually described as follows based on a literature search and example measurements [34]:

$$P_{c_i,cwc} = P_{s_i,cwc}/COP_{chiller}, \tag{4}$$

where $P_{s_i,cwc}$ represents the power consumption of all servers in the cooling module C_i whose heat dissipation is provided by the chilled water cooling system. $COP_{chiller}$ is the actual chilled water cooling parameter, which is decided by the chillers.

4.2.3. Computer Room Air Handler

CRAH is responsible for extracting the heat generated by the active server, and supporting the cold air. When the outside air temperature is cold enough, CRAH can provide the outside cold air to servers from the blowers directly, or when the outside air is not suitable, the cold air will be generated by the chiller system and cooling tower of the chilled water cooling system to cool the servers. The power consumption of CRAH mainly comprises fans and some other components, such as sensors and control systems; the power consumption of fans is the major concern. Thus, the power model of CRAH can be denoted by:

$$P_{crah} = p_{crah}^{idle} + p_{crah}^{peak} f^3, \tag{5}$$

where f represents the fan rate, which is set as a fixed value, p_{crah}^{idle} denotes the idle power consumption of the CRAH and p_{crah}^{peak} denotes the peak power of the CRAH.

4.2.4. The Final Power Model of a Cooling System

In a data center, operating in different cooling modes will save more cooling energy [34]. As such, we consider a data center composed of multiple cooling modes $cm_1, cm_2, ..., cm_i, ...$; thus, we can get the following cooling system power model:

$$P_{c_i} = u_1 P_{c_i,cm_1} + u_2 P_{c_i,cm_2} + ... + u_i P_{c_i,cm_i} + ... \tag{6}$$

where $u_1 + u_2 + ... + u_i + ... = 1$, and $u_i \in [0, 1]$.

In consideration that various cooling modes always have different cooling capacities and effectivenesses within workload changes [32], we apply two cooling modes (outside air cooling and chilled water cooling) to improve the energy efficiency of the cooling system. The cooling modes are adjusted based on different workload changes within various time slots. Thus, the final cooling system model is derived as:

$$P_{c_i} = \begin{cases} u_1 P_{c_i,oac} + (1-u_1) P_{c_i,cwc} + P_{crah} \\ k(u_1 P_{s_i})^3 + (1-u_1) P_{s_i} / COP_{chiller} + p_{crah}^{idle} + p_{crah}^{peak} f^3 \end{cases} \tag{7}$$

where $u_1 \in [0, 1]$, and the optimal cooling allocation condition is derived as $u_{opt} = \min(\sqrt{3kP_i^2} * COP_{chiller}, 1)$ [32]. When $u_1 = u_{opt}$, the cooling capacity allocation between the two cooling modes is the optimal; as a result, the cooling energy efficiency is maximized. Thus, the final cooling system model can be denoted in the following form:

$$P_{c_i}^t = \begin{cases} 0 & P_{s_i} = 0 \\ k(u_{opt} P_{s_i})^3 + (1-u_{opt}) P_{s_i} / COP_{chiller} + P_{crah} & P_{s_i} \geq 0 \end{cases} \tag{8}$$

4.3. A Power Model of State Transitions

There exist delays and extra energy consumption during the server and cooling systems' state transitions: when servers and the cooling system transition from an inactive state to an active state or from an active state to inactive state. Let $P_{on \leftarrow off}^{s_{ij}}$ and $P_{off \leftarrow on}^{s_{ij}}$ denote the server transition power, and $M_{on \leftarrow off}^t$, $M_{off \leftarrow on}^t$ represent the number of servers that need be turned off or turned on. Thus, the transition power consumption of servers in time slot t can be calculated as:

$$P_{trans}^{server} = P_{on \leftarrow off}^{s_{ij}} \cdot M_{on \leftarrow off}^t + P_{off \leftarrow on}^{s_{ij}} M_{off \leftarrow on}^t \tag{9}$$

Moreover, the transition power consumption of cooling systems in timeslot t can be calculated as Equation (10), where $N^t_{on\leftarrow off}$ and $N^t_{off\leftarrow on}$ represent the numbers of cooling systems that need be turned off and turned on, respectively.

$$P^{cooler}_{trans} = P^{c_i}_{on\leftarrow off} \cdot N^t_{on\leftarrow off} + P^{c_i}_{off\leftarrow on} N^t_{off\leftarrow on} \tag{10}$$

Thus, the energy of transition of servers and cooling systems can be represented by:

$$E_{trans} = \int_{t\in\tau} (P^{server}_{trans} + P^{cooler}_{trans}) dt \tag{11}$$

4.4. Problem Formulation

In this part, we formulate the energy consumption minimization problem. Our optimization problem takes as input the arriving task's resource information at the current time without knowing all arrival tasks in the future, and attempts to get the optimal task scheduling scheme for the arrived tasks.

In the data center, the total server energy consumption E_S in a period time τ can be denoted by:

$$E_S = \sum_{N^{active}} \left(\sum_{M_i} \int_{t\in\tau} \vartheta P^t_{s_{ij}} dt \right), \tag{12}$$

where N^{active} represents the number of active cooling modules, τ is the period of time and ϑ denotes the server's state. If a server is active, $\vartheta = 1$; otherwise, $\vartheta = 0$

For the cooling system, the energy consumption of all the cooling systems in every cooling module is represented as Equation (13), where E_C represents the total energy consumption of cooling systems in the data center.

$$E_C = \sum_{N^{active}} \left(\int_{t\in\tau} (P^t_{c_i}) dt \right), \tag{13}$$

In the formulation, the goal is to minimize the total energy consumption, which includes three parts: server energy consumption, cooling systems' energy consumption and the state transition energy consumption. Thus, the energy consumption minimization problem of a data center can be formulated as (14):

$$\min \quad E_C + E_S + E_{trans} \tag{14}$$

$$s.t. RA^{cpu}_{s_{ij}} - R^{cpu}_{t_a} \geq 0, \tag{15}$$

$$\vartheta \in 0, 1, \tag{16}$$

$$0 \leq N^{active} \leq N. \tag{17}$$

where in constraint (17) N^{active} represents the number of active cooling modules. Constraint (15) represents the resource constraint of servers, which means the amount of resources requested by the tasks cannot exceed the resource capacity of the server. Constraint (16) indicates the state of a server; if the server is turned on and working, $\vartheta = 1$, and otherwise, $\vartheta = 0$. Constraint (17) means the number of active cooling modules can not be larger than the total number of modules.

5. Energy-Aware Scheduling Strategy

In this section, to solve the energy consumption minimization problem (14) in data centers stated in the previous section, an energy-aware task scheduling strategy utilizing marginal cost and task classification method is introduced, incorporating a task classification method and a scheduling algorithm. The proposed strategy first applies the task classification method to classify tasks and servers. Within the task classification method, the arriving task will obtain a label. Additionally, we allocate tasks to the servers with

the same label. The definitions of marginal energy in homogeneous and heterogeneous data centers are discussed and provided to manage the task scheduling algorithm. Finally, consolidated with the task classification method and marginal cost evaluation, the task scheduling algorithm is proposed, and the time complexity is analyzed. The data used in the strategy were from real word traces, including Google cluster trace and Alibaba cluster trace. Parts of the datasets (one day) were selected. We mainly utilized the information in the file batch_task.csv of Alibaba cluster data, which includes task name, task_type, start_time, end_time, plan_cpu, plan_mem, etc. For the Google cluster trace, we used the file task_events_table.csv, which includes a timestamp, machine ID, a resource request for CPU cores, etc. The task_type, start times, end times and resource requests for CPU cores were extracted from the two datasets as the inputs of the classification method and the scheduling algorithm.

5.1. Task Classification Method

We introduce how to classify the servers and tasks utilizing the task classification method. When tasks arrive, the information of the arriving task is observed and recorded by the task observer. Based on the stored and newly observed information, the run times of a task can be acquired through analyzing and using the historical data or predicted by the machine learning methods [30]. It is assumed that r_{t_a} denotes the run times, and e_{t_a} represents the ending time of the task t_a. As the task is scheduled in real-time, the start times s_{t_a} can be calculated through $s_{t_a} + r_{t_a} = e_{t_a}$ and $s_{t_a} \geq 0$. Through the recording and prediction controller part, the historical data are collected and analyzed. Then the proximate run times are predicted by machine learning methods [30].

As presented in [25], the arriving tasks will be classified and obtain labels according to the running and ending times of tasks. The method to calculate the classification labels of tasks $C(\alpha, \beta)$ is depicted in Equation (18). It is calculated in real-time such that the time requirements can be fulfilled. Besides, the space requirements are met too. One can refer to [25] to check the theory features and why the two requirements can be satisfied. The classification method aims at scheduling tasks to the same classified servers. As a result, the tasks with relevant run times and ending times will be scheduled to servers with the same classification labels, which will provide more chances to power off idle servers, and the utilization of servers is increased.

Finally, the time complexity is analyzed. We need $\mathcal{O}(1)$ to calculate the classification label. The classification label of servers $S(\alpha, \beta)$ is decided on by the tasks running on it. If the server is in an active state, the label is the same as the tasks operating in it; otherwise, we set the label of inactive servers as $S(0, -1)$. After the task is classified with the method in Equation (18), it is sufficient to improve the utilization of servers, and the energy consumption through scheduling tasks to the same classification servers is reduced.

$$\begin{cases} \alpha = \max\{0, \lceil \log_2 r_{t_a} \rceil\} \\ \beta = \max\{1, \left\lfloor \frac{2e_{t_a}}{2^{\lceil \log_2 r_{t_a} \rceil}} \right\rfloor\} \end{cases} \tag{18}$$

5.2. Marginal Cost in Data Centers

The concept of marginal cost means the influence of incremental increase of newly produced product on the total cost in economics and finance. This concept indicates the change in the dependent variable when the independent variable in the functional relationship changes slightly. In the study of mathematical theory, the marginal cost is expressed by the partial derivative of total cost TC and quantity Q. Thus, the equation of marginal cost can be represented as:

$$MC = \frac{\partial TC}{\partial Q} \tag{19}$$

In this paper, we introduce the marginal energy in data centers based on the concept of marginal cost to minimize the total energy consumption of the data center, namely, to optimize the value of E_{DC}, where $E_{DC} = E_S + E_C + E_{trans}$. Next, we will discuss the definition of the marginal energy in two kinds of data center cluster environments based on marginal cost. The concept of marginal cost can be applied to both the homogeneous and heterogeneous data center cluster environments. Finally the optimal scheduling condition with marginal cost of data centers is derived.

5.2.1. The Definition of Marginal Energy Based on Marginal Cost

The definition of marginal energy is developed based on marginal cost. To determine which server resources and cooling module resources are chosen for the task, we will calculate the energy changes caused by scheduling the task to different cooling modules and servers. We calculated the sum of the power changes brought about after all the task assigned in the chosen module finish instead of only considering the power change value at the current time slot. The marginal energy based on the concept of marginal cost for servers and cooling modules is defined as $M_{s_{pq}}^{server}$ and $M_{c_p}^{cooler}$, respectively. Equation (21) represents the energy changes from time slot ξ^{init} to time slot ξ^{end} after the task scheduled to the server S_{pq}. Equation (20) calculates the energy changes after the task choose the cooling module resources C_p. Where ξ^{init} denotes the time when the task is allocated to the servers and cooling modules. ξ^{init} represents the time all the task in the allocated cooling modules finish.

$$M_{c_p}^{cooler} = \frac{\partial(\sum_{i=1}^{i=N}\left(\sum_{j=1}^{j=M}\int_{t=\xi^{init}}^{t=\xi^{end}} P_{s_{ij}}^{t}dt\right) + \sum_{i=1}^{i=N}\left(\int_{t=\xi^{init}}^{t=\xi^{end}} P_{c_i}^{t}dt\right) + \int_{t=\xi^{init}}^{t=\xi^{end}}(P_{trans}^{server} + P_{trans}^{cooler})dt)}{\partial P_{c_p}} \tag{20}$$

With the Equation (20), the suitable cooling module resources will be determined.

$$M_{s_{pq}}^{server} = \frac{\partial(\sum_{i=1}^{i=N}\left(\sum_{j=1}^{j=M}\int_{t=\xi^{init}}^{t=\xi^{end}} P_{s_{ij}}^{t}dt\right) + \sum_{i=1}^{i=N}\left(\int_{t=\xi^{init}}^{t=\xi^{end}} P_{c_i}^{t}dt\right) + \int_{t=\xi^{init}}^{t=\xi^{end}}(P_{trans}^{server} + P_{trans}^{cooler})dt)}{\partial P_{s_{pq}}} \tag{21}$$

Based on the Equation (21), the suitable server resources will be decided for the task.

5.2.2. Homogeneous Data Center Cluster

In the homogeneous data center cluster, the hardware configurations of all servers are the same, and the peak power, idle power, and energy efficiency of each server running different workloads are the same. When scheduling tasks, we only need to consider whether to power on a new idle server or schedule tasks to a running server. The cooling efficiency of different cooling modules may be different, so it is also necessary to determine which cooling module is allocated for the task. To determine which server resources and cooling module resources are chosen, we need to calculate the energy changes caused by scheduling the task to different cooling modules. We calculated the sum of the power changes brought about after all the tasks assigned in the chosen module finish instead of only considering the power change value at the current time slot. We can apply the marginal cost evaluation to guide the task scheduling process to choose the suitable server and cooling system resources with the minimum energy cost. Using the Equation (20) the cooling module resource is chosen, and with the Equation (21), we can decide whether to choose a running server or an idle server.

5.2.3. Heterogeneous Data Center Cluster

In the heterogeneous data center cluster, we assume that the servers are heterogeneous. Different servers have different peak power consumption, idle power consumption and energy efficiency. Different cooling modules have numerous cooling effectiveness and cooling parameters. Thus, when the tasks arrive dynamically, scheduling tasks to which server under which cooling module will cause varying energy consumption costs. It can

be decided by the marginal cost evaluation to choose suitable server and cooling system resources too. We need to determine which cooling module resource to choose, whether to choose a running server or a static server and decide the type of servers. With the Equations (21) and (20), the most energy-efficient server and cooling resources will be chosen. We need to calculate the energy changes after all the tasks in the current servers of the data center finish if the task is scheduled to any servers, such that the server chosen for the arriving task will lead to the minimum energy consumption.

Thus, based on Sections 5.2.2 and 5.2.3, we can conclude that the developed definition of marginal energy can be applied in homogeneous and heterogeneous data center clusters to guide task scheduling.

5.2.4. Optimal Scheduling Condition with Marginal Cost Evaluation

The task scheduling strategy needs to determine which server and cooling module resources for the arrived tasks to choose. (1) Which server and cooling module resources in the off state should be powered on in each time slot; (2) turning servers and cooling systems on/off; (3) deciding on the specific server and cooling system resources to satisfy the request of tasks; (4) determining the minimum state transition. With the definitions of the marginal energy of servers and cooling modules in data centers, the optimal task scheduling condition is developed. The basic idea is scheduling tasks to the servers with the minimum marginal energy value, which will lead to a minimum increase in overall energy after allocation and ensure the power-efficient optimal allocation. We need to calculate the energy changes after all the tasks in the current servers of the data center finish if the task is scheduled to any servers, such that the server chosen for the arriving task will bring the minimum energy consumption. The optimal scheduling condition is shown in Equations (22) and (23)

(1) Choosing the server resources with the minimum marginal energy value for the arrived task.

$$\forall s_{ik}(i \in N|j, p \in M_i), \frac{\partial E_{DC}^{\xi}}{\partial P_{s_{ip}}} \leq \frac{\partial E_{DC}^{\xi}}{\partial P_{s_{ij}}} \tag{22}$$

(2) Determining which cooling module resource to choose.

$$\forall c_q(q \in N), \frac{\partial E_{DC}^{\xi}}{\partial P_{s_i}} \leq \frac{\partial E_{DC}^{\xi}}{\partial P_{s_q}} \tag{23}$$

where $E_{DC}^{\xi} = \sum_{i=1}^{i=N}\left(\sum_{j=1}^{j=M}\int_{t=\xi^{init}}^{t=\xi^{end}} P_{s_{ij}}^{t}dt\right) + \sum_{i=1}^{i=N}\left(\int_{t=\xi^{init}}^{t=\xi^{end}} P_{c_i}^{t}dt\right) + \int_{t=\xi^{init}}^{t=\xi^{end}}(P_{trans}^{server} + P_{trans}^{cooler})dt$.

5.3. Scheduling Strategy Using the Marginal Cost and Task Classification Method

The task classification method, definitions of marginal energy and optimal scheduling condition with marginal cost evaluation of the data center have been described above. In this part, the task scheduling strategy with the marginal cost evaluation and task classification method is proposed to minimize the entire energy consumption. The scheduling strategy will (1) determine which server and cooling module resources in the off state should be powered on in each time slot; (2) determine various servers and cooling systems to turn on/off; (3) decide on the specific server and cooling system resources to (4) satisfy the request of tasks; (4) determine the minimum state transition.

To optimize the total energy consumption in data centers, there exist two ideas. Firstly, we should classify servers according to the tasks run times and ending time such that the tasks can be scheduled to the same classified servers with the same label. Secondly, we should choose the servers and cooling resources with minimum marginal energy value. The energy-aware task scheduling strategy is presented in Algorithms 1 and 2.

Algorithm 1 Energy-aware task scheduling.

Require:
Task t_a and the label $C(\alpha, \beta)$.
Ensure:
The Server S_{ij} allocated to the task.
1: **for** $i \in [1, N]$, $j \in [1, M_i]$ **do**
2: **if** $(RA_{s_{ij}}^{cpu} - R_{t_a}^{cpu} \geq 0) \&\& S(\alpha, \beta) == C(\alpha, \beta)$ **then**
3: Put S_{ij} into the set $\theta_{(\alpha,\beta)}$;
4: **end if**
5: **end for**
6: **if** $\theta_{(\alpha,\beta)} \neq \emptyset$ **then**
7: Calculate the marginal energy value with CalMarC($\theta_{(\alpha,\beta)}$)
8: Calculate the marginal energy value with CalMarC($\theta_{(0,-1)}$)
9: **return** the server S_{ij} with minimum marginal energy value
10: Classify the server and set the label of the server as $S(\alpha, \beta)$;
11: **end if**
12: **if** $\theta_{(\alpha,\beta)} = \emptyset$ **then**
13: **for** $C_i (i \in [1, N])$ **do**
14: **for** S_{ij} with label $C(0, -1)$ **do**
15: **return** the server S_{ij} with minimum marginal cost value using CalMarC($\theta_{(0,-1)}$)
16: Classify the server and set the label of the server as $S(\alpha, \beta)$;
17: **end for**
18: **end for**
19: **end if**

When a task is submitted by users to the data center, through the classification controller, the task will obtain a category label $C(\alpha, \beta)$. All servers are traversed to find the servers whose available CPU resource is larger than the resource requested by the task and with the same category label denoted by $S(\alpha, \beta)$. Servers that satisfy the above conditions will be put into a set $\theta_{(\alpha,\beta)}$. If $\theta_{(\alpha,\beta)} \neq \emptyset$, we will select an active server from the set $\theta_{(\alpha,\beta)}$ for the arriving task as follows:

- Due to the heterogeneous characteristics of servers, scheduling tasks to different servers will cause various amounts of energy consumption; thus, we utilize the concept of marginal cost to decide which server resource is suitable. For every active server in the set $\theta_{(\alpha,\beta)}$, the marginal energy $M_{s_{pq}}^{server}$ is calculated with the Equation (21). The process of calculating the marginal energy value is described in Algorithm 2, which is a sub-algorithm of Algorithm 1. Lines 11–14 in Algorithm 2 calculate the energy changes after all the tasks in the current servers of the data center finish if the task t_a is scheduled to any servers. Besides, it calculates the sum of the server power consumption changes, the cooling system power consumption changes and the state transition power consumption changes at each moment before the end times of task t_a. The sum of the power changes is set as the marginal energy value.
- Considering the transition power consumption, we need to decide if we should to turn on a new server or just choosing a running server with the same label. Thus, for each idle server in the set $\theta_{(0,-1)}$, we calculate the marginal energy value using the Equation (20) and Algorithm 2 to decide the cooling resource and with the Equation (21) to choose the server resource.
- Scheduling the submitted task to a server that satisfies the optimal scheduling condition based on the Equation (22).

If $\theta_{(\alpha,\beta)} = \emptyset$, we need to choose an idle server whose label is $S(0, -1)$ from all cooling modules for the submitted task as follows:

- For each cooling module, the marginal energy value of cooling modules is calculated to decide on the cooling resources using Equation (20)
- Due to the data center being heterogeneous, the idle servers in different modules may be different and have different power consumption and energy efficiency. Thus, in this heterogeneous cluster, it is necessary to determine the server resource. The marginal energy value of servers is calculated using Equation (21) and Algorithm 2.
- Finally, we schedule task t_a to an idle server with the minimum marginal energy value according to the Equations (22) and (23) and provide a classification label for the idle server.

Algorithm 2 CalMarC(server_sets).

Require:
Server sets $\theta_{(\alpha,\beta)}$ or $\theta_{(0,-1)}$, and Task t_a
Ensure:
The marginal energy value of data centers.
1: currentTime ← Get the current time slot
2: Tasklist ← Get all running tasks list
3: TaskNum ← Get total task number of server S_{ij}
4: $\lambda_{ij} = 0.0$ and $p = 0.0$
5: Get the current utilization $r_{s_{ij}}$ of the server s_{ij}
6: **if** TaskNum >= 1 **then**
7: $p = (P_{s_{ij}}^{full} - P_{s_{ij}}^{idle}) \times (2R_{t_a}^{cpu}/RT_{s_{ij}}^{cpu} - (r_{s_{ij}} + R_{t_a}^{cpu}/RT_{s_{ij}}^{cpu})^{1.4} + {r_{s_{ij}}}^{1.4})$
8: **else**
9: $p = P_{s_{ij}}^{idle} + (P_{s_{ij}}^{full} - P_{s_{ij}}^{idle}) \times (2R_{t_a}^{cpu}/RT_{s_{ij}}^{cpu} - (r_{s_{ij}} + R_{t_a}^{cpu}/RT_{s_{ij}}^{cpu})^{1.4} + {r_{s_{ij}}}^{1.4})$
10: **end if**
11: **while** Tasklist $\neq$ *null* and $currentTime < rt_a + st_a$ **do**
12: $\Delta = \lambda_{ij} + (f(P_{s_i} + p) - f(P_{s_i})) \times (et_a - currentTime) + p$
13: $\lambda_{ij} = \Delta + (P_{trans}^{server} + P_{trans}^{cooler})$
14: **end while**
15: **return** λ_{ij}

When the task is submitted dynamically, within the proposed scheduling strategy EATS, the task will be assigned to the suitable server resources and cooling resource with the minimum marginal energy value. Finally, the time complexity of the EATS will be analyzed.

From step (1) to step (4), the active and available server set whose label is the same as the arriving task is acquired. Assume that there exist z servers in the data center, thus the time complexity is $\mathcal{O}(z)$.

If $\theta_{(\alpha,\beta)} \neq \emptyset$, it depicts how to choose a suitable server resource with the minimum marginal energy value from step (7) to step (10). Step (7) and step (8) represent calculating the marginal energy value of the server in the active and available server set $\theta_{(\alpha,\beta)}$ and idle server set $\theta_{(0,-1)}$ using Algorithm 2. Steps (6) to (15) in Algorithm 2 introduce how to calculate the marginal energy value of data centers. We need to calculate the energy changes after all the tasks in the current servers of the data center finish if the task is scheduled to any servers, such that the server chosen for the arriving task will bring the minimum energy consumption. Assume that the length of running task list is denoted by ω, the time complexity is $\mathcal{O}(2 * \omega)$. Step (9) and step (10) in Algorithm 1 represent choosing the server resource with the minimum marginal energy value with the time complexity $\mathcal{O}(|\theta_{(0,-1)}| + |\theta_{(\alpha,\beta)}|)$.

If $\theta_{[\beta,\eta]} = \emptyset$, we need choose an idle server with the minimum marginal energy value using Algorithm 2 which is shown from step (13) to step (19) in Algorithm 1. The time complexity is $\mathcal{O}(N * |\theta_{(0,-1)}| * \omega)$. Thus, the overall time complexity of EATS is $\mathcal{O}(z) + \mathcal{O}(2 * \omega * (|\theta_{(0,-1)}| + |\theta_{(\alpha,\beta)}|)) + \mathcal{O}(N * |\theta_{(0,-1)}| * \omega) = \mathcal{O}(z + \omega * (|\theta_{(0,-1)}| + |\theta_{(\alpha,\beta)}|) + N * |\theta_{(0,-1)}| * \omega)$.

6. Experiments

In the section, to appraise the proposed strategy that can effectively reduce the whole energy consumption of the data center, we built an experiment system, and the details of our experiment setup and results are presented.

We built our simulation platform based on JAVA with OpenJDK 11.0.10 (Oracle, USA). The experiment was conducted on a machine equipped with windows 10 operating system, Intel(R) Core(TM) i5-9500 CPU, 3.00GHz 3.00GHz, 8GB RAM, 2TB Disk storage. During the simulation, most information, including the server power and cooling power, the transition power of servers and the total data center power at each time slot, was saved into files. After the information collection and data collation, the figures were plotted through Matlab software.

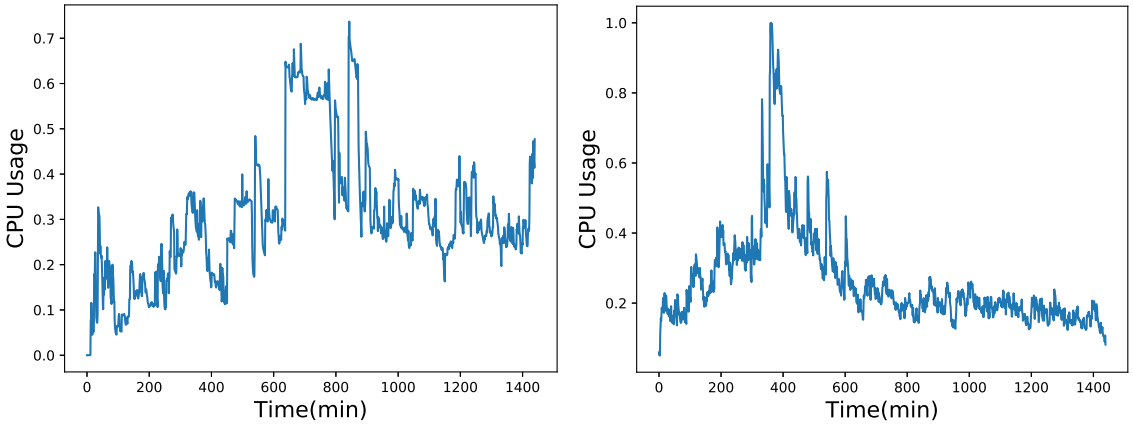
6.1. Experimental Setup

Assume that a data center consists of 5 modules, each module containing 600 servers. Therefore, there are a total of 3000 servers in the data center. For the cooling system, the cooling parameter k in Equation (8) is set to 5×10^{-4} [25], and the parameter of chiller system $COP_{chiller}$ is set to 5 [32]. The fan rate of CRAH is set to 0.66 [35]. The idle power and peak power of CRAH are 100 and 3000 W, respectively. The transition power of servers from on state to off state is denoted as 118.52 W [36], and the power consumed by the transition from off state to on state is 128.21 W [36]. The settings of the experiments are shown in Table 1.

Table 1. Experimental settings.

Parameter	Value
Number of servers	3000
Number of cooling modules	5
k	5×10^{-4}
$COP_{chiller}$	5
p_{crah}^{idle}	100 W
p_{crah}^{peak}	3000 W
f	0.66
$P_{on \leftarrow off}^{s_{ij}}$	118.52 W
$P_{off \leftarrow on}^{s_{ij}}$	128.21 W

We utilized two types of datasets to evaluate the performance of the proposed strategy EATS, both of which come from real-world traces. The two datasets are Google cluster data [8] and Alibaba Cluster Data V2018 [9]. We randomly selected some resource data over 1 day from the two datasets and extracted the task traces from files (batch_task.csv of Alibaba cluster data and task_events_table.csv of Google cluster data), including task arrival times, task end times and the CPU resource requirements of each task. The Google cluster dataset presents the resource utilization data of servers during a month and Alibaba cluster dataset provides the resources usage data of 4000 machines in 8 days. The per-minute workload CPU utilization values of the two datasets are presented in Figure 3a,b, respectively.



(**a**) Per-minuter workloads of Google cluster Data. (**b**) Per-minuter workloads of Alibaba cluster Data.

Figure 3. The highly random workloads in data centers.

In the data center cluster, it is assumed that the servers are heterogeneous. Different servers have different peak power consumption, idle power consumption and energy efficiency. In a cooling module, there exist six types of servers. Table 2 provides the power parameters of the six types of servers in the data center [25,26,37].

6.2. The Baseline Algorithm

In the experiment, the proposed strategy EATS is compared with two techniques (Tech-1 and Tech-2).

- Tech-1: In Tech-1, it schedules tasks between the various cooling modules based on the load balancing practice. For the servers, it gives preference to scheduling tasks to active servers.
- Tech-2: Tech-2 is based on the algorithm rTCS in [25]. rTCS schedules the arriving task in real-time. Once a task arrives, rTCS will firstly classify the task and servers using a task classification algorithm. After the tasks are labeled, the tasks will be scheduled to the servers with the same label. As a result, high utilization of servers will be guaranteed. As for the quadratic characteristic of the cooling function, to further reduce the energy consumption of cooling systems, tasks with the same label will be allocated to different cooling modules evenly.

Table 2. Server parameters in the experiment.

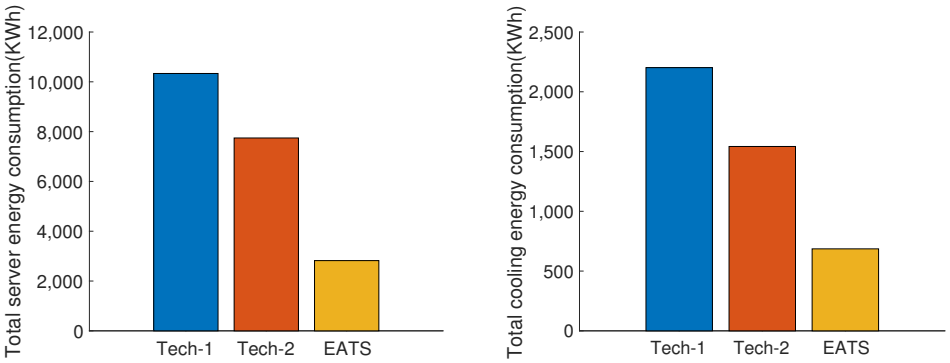
Server Type	Server Power Parameters			
	Idle Power (W)	Peak Power (W)	CPU Cores	Total Number
1	200	500	16	600
2	200	300	32	300
3	100	200	8	600
4	110	300	96	600
5	430	1000	64	450
6	1590	2490	16	450

6.3. Results

6.3.1. Energy Consumption of Servers and Cooling System

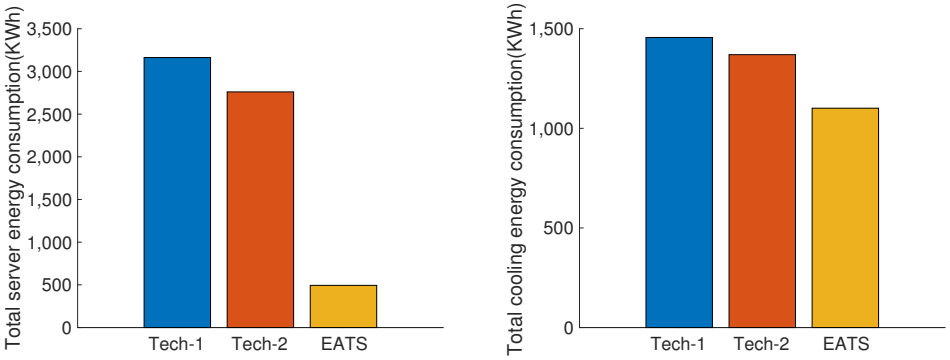
In a heterogeneous data center, the servers are heterogeneous and have different energy efficiencies and levels of power consumption. First, the energy consumption of servers and cooling systems of the proposed strategy EATS is compared with that in Tech-1

and Tech-2, respectively. As different datasets have different characteristics, it is necessary to compare the results for energy reduction based on various datasets. Therefore, we utilized two real world datasets (Alibaba cluster data and Google cluster data) to verify the performance of EATS regarding improving the energy efficiency of servers and cooling systems, respectively. The server energy consumption results using Alibaba cluster data are illustrated in Figure 4a. As shown in Figure 4a, the server energy consumption of EATS was the smallest compared with other two algorithms, which verifies the effectiveness of the proposed EATS at reducing server energy consumption. Apart from the Alibaba cluster data, we also used the Google cluster data to validate the performance of EATS. The server energy consumption results using Google cluster data are indicated in Figure 5a. With Figure 5a, it can be concluded that EATS can save more energy of servers than Tech-1 and Tech-2. Thus, based on Figures 4a and 5a, we can conclude that EATS is effective at reducing server energy consumption in comparison to Tech-1 and Tech-2.



(**a**) Comparison of server energy consumption. (**b**) Comparison of cooling energy consumption.

Figure 4. Comparison of cooling and server energy consumption for Alibaba cluster data.



(**a**) Comparison of the server energy consumption. (**b**) Comparison results of the cooling system energy consumption.

Figure 5. Comparison of cooling and server energy consumption for Google cluster data.

We compare cooling system energy-saving results with Tech-1 and Tech-2 using the two datasets next, and the results are shown in Figures 4b and 5b, respectively. The results shown in Figure 4b are based on Alibaba cluster data. Figure 4b indicates the energy-saving results using Google cluster data. From the two figures, we can find that the energy consumption of cooling systems caused by EATS is the least compared to Tech-1 and Tech-2, which shows that EATS can also effectively improve the energy efficiency of cooling systems.

As illustrated in Figures 4 and 5, the effectiveness of EATS at jointly reducing the energy consumption of servers and cooling systems is validated.

6.3.2. Transition Energy Consumption

In this part, we compare the energy caused by the server and cooling system state transition with other algorithms. As frequent system state transitions will cause more energy consumption and extra cost, it is vital to devise an energy-efficient scheduling strategy that can optimize the energy caused by the state transitions. EATS aims at jointly improving the energy efficiency of servers, cooling systems and the energy caused by state transitions. To verify the effectiveness of the proposed algorithm, we compare the transition energy consumption with Tech-1 and Tech-2, and the comparative results using Alibaba cluster data are depicted in Figure 6. As shown in Figure 6, the transition energy consumption achieved by the proposed strategy EATS was the smallest; this is because these two algorithms (Tech-1 and Tech-2) only consider the energy consumption optimization of servers and cooling systems without considering the energy consumption caused by state transitions. Therefore, the results in Figure 6 display that the proposed strategy EATS can effectively save data center transition energy consumption compared with Tech-2 and Tech-1. They also verify that the proposed strategy EATS is efficient in cooperatively saving the total energy consumption of servers, cooling systems and state transition in the data center compared with Tech-1 and Tech-2.

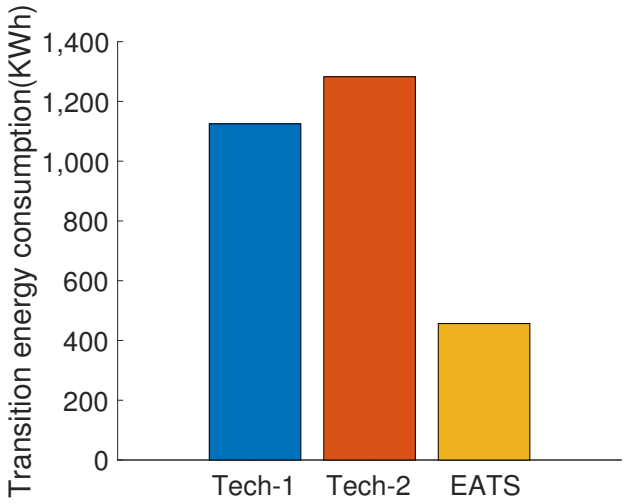


Figure 6. The comparison results of transition energy consumption for Alibaba cluster data.

6.3.3. Total Energy Consumption

The core idea of the paper is to optimize the total energy consumption consisting of servers, cooling systems and state transitions energy. Thus, we collated the output data and summarize the total energy consumption of the three algorithms. The comparison results utilizing the Alibaba cluster data are depicted in Figure 7. Figure 8 shows the results of total energy consumption based on Google cluster data compared with other two algorithms. As we can see in these two figures, the energy consumption achieved by EATS was the smallest in comparison to Tech-1 and Tech-2. As the proposed strategy EATS focuses on cooperatively improving the energy efficiency of servers, cooling systems and reducing the energy costs caused by system state transitions, EATS achieved the smallest energy consumption of a data center, which indicates that jointly optimizing the energy consumption of servers, cooling systems and state transition energy is efficient. With the results of Figures 7 and 8, we can validate the better performance of EATS regarding energy savings comparedd with Tech-1 and Tech-2 using various datasets.

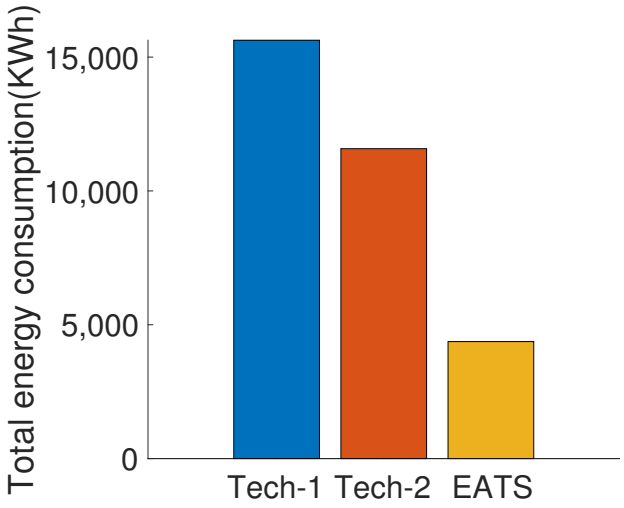


Figure 7. Comparison of data center energy consumption for Alibaba cluster data.

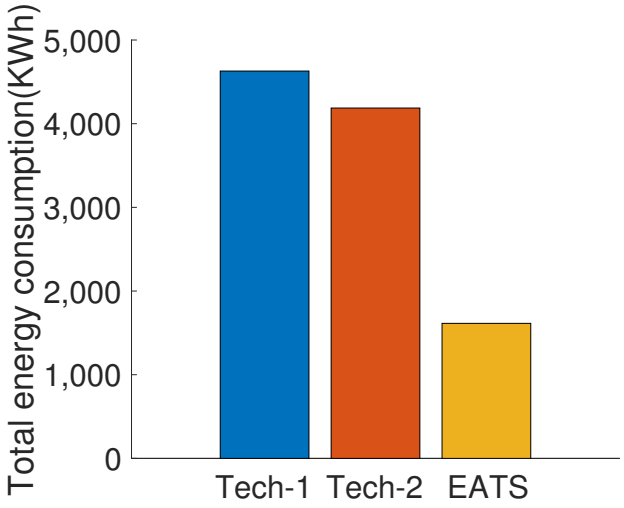


Figure 8. Comparison of data center energy consumption for Google cluster data.

6.3.4. Various Scenarios

To analyze the energy-saving results of EATS under various data center scenarios, we add three scenarios with the same total number of servers, but the number of cooling modules and the number of servers in each module are different. We conducted the experiments under each data center scenario and compared the total energy consumption of EATS with that of the other two algorithms. The comparison results are shown in Figure 9. In Figure 9, scenario 1 represents that a data center consisted of one cooling module and the module had 3000 servers. Scenario 2 represents that a data center included three cooling modules and each module included 1000 servers. Scenario 3 represents that a data center included five cooling modules and each module included 600 servers, which is the same as in the base assumption of the paper. Scenario 4 represents a data center that has 10 cooling modules, and every module has 300 servers. Based on the results concluded from Figure 9, we found that under different data center scenarios, the total energy consumption of EATS was always the smallest compared with Tech-1 and Tech-2. This validates that under different data center scenarios, the proposed algorithm EATS is effective at reducing the total data center energy consumption.

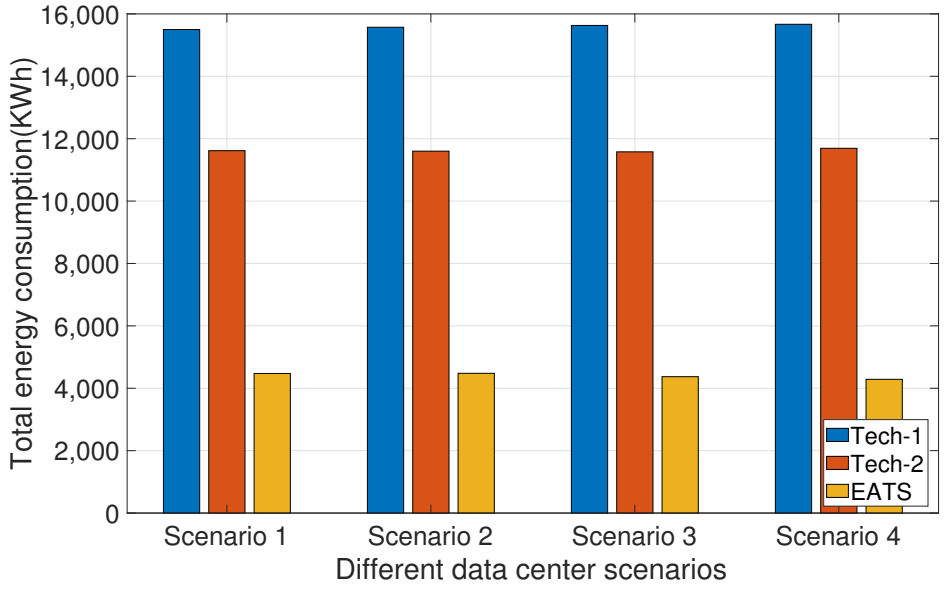


Figure 9. Comparison results of the total energy consumption with different data center scenarios.

6.3.5. Task Arrival Intensity

In this part, first, we compare the total energy consumption of EATS with Tech-1 and Tech-2 under various task arrival intensities in Figure 10. The task arrival intensity means the number of arriving task per minute. Data 1, Data 2, Data3, and Data 4 in Figure 10 represent different datasets with various task arrival intensities. From Data 1 to Data 4, the task arrival intensity value is gradually increasing. The four kinds of data are all from Alibaba cluster data. Data 2, Data 3, and Data 4 are obtained by expanding 2, 3, and 4 times on the basis of data 1, respectively. As we can see from Figure 10, the energy consumption of a data center reduced by EATS is largest compared with Tech-1 and Tech-2. Besides, with the increase of task arrival intensities, the performance of EATS on reducing the energy consumption become more better compared with the algorithm Tech-1. The results indicate that the performance of EATS is better than the other two techniques in reducing the total energy consumption under different task arrival intensities.

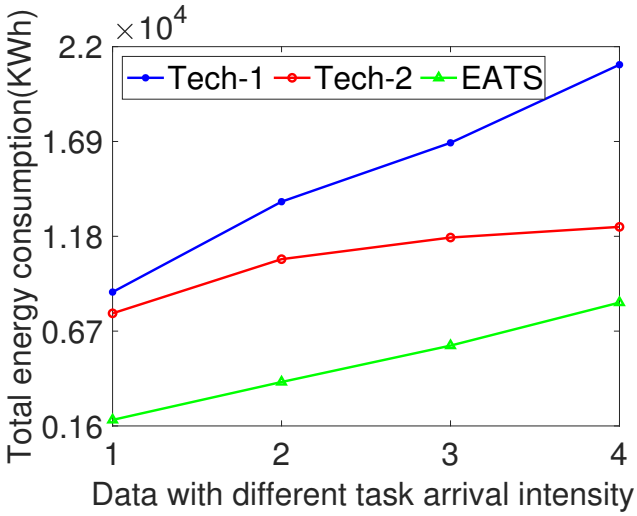


Figure 10. The influence of various task arrival intensities on the energy consumption.

Next, the transition energy consumption under different task arrival intensities is compared. Figure 11 presents the transition energy-saving results in comparison to Tech-1 and Tech-2 with various task arrival intensities. In Figure 11, Data 1, Data 2, Data 3,

and Data 4 indicate four kinds of datasets from Alibaba cluster data with different task arrival intensities. From Data 1 to Data 4, the task arrival intensities of each dataset are gradually increasing. We found that with the changes of task arrival intensities, the energy-saving results of EATS were always better than other two algorithms, which verifies that EATS outperforms the others in terms of reducing the transition energy consumption of the data center compared with Tech-1 and Tech-2. The transition energy consumption caused by the algorithm Tech-1 was less than that of algorithm Tech-2, but as shown in Figure 10, the total data center energy consumption of Tech-1 was larger than the consumption of Tech-2 and EATS. Tech-2 and EATS both consider the joint energy optimization of servers and cooling systems, which verifies the effectiveness of jointly optimizing the energy consumption of data centers. The proposed strategy EATS considers the energy optimization of servers, cooling systems and system state transition cost; thus, the state transition and total data center energy consumption of proposed strategy EATS was smallest compared with Tech-1 and Tech-2. This verifies that the proposed strategy EATS is effective in cooperatively saving the total energy consumption of servers, cooling systems and state transition in the data center compared with Tech-1 and Tech-2.

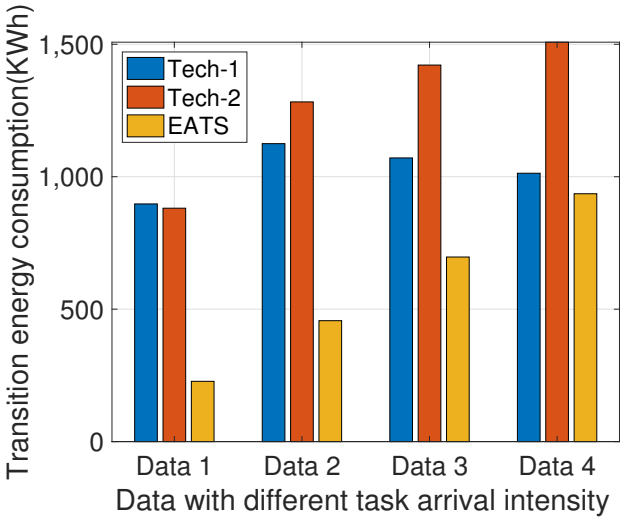


Figure 11. The comparison results of various task arrival intensities—the transition energy consumption.

6.3.6. Different Numbers of Cooling Modules

In the paper, we assumed that a data center includes five cooling modules, and each module had 600 servers. Thus, there were a total of 3000 servers in the data center. To verify the performance of the proposed strategy, it was necessary to analyze the influence of the number of cooling modules on the data center energy consumption. Thus, we made other assumptions about the number of cooling modules and compared the total energy consumption with Tech-1 and Tech-2. The comparison results are shown in Figure 12 using Alibaba cluster data. As shown in Figure 12, 3, 4, 5, 6, 7, 8, 9 and 10 were the numbers of cooling modules, respectively. Each module included 500 servers. As such, we assumed that a data center consisted of three cooling modules, and each module included 500 servers. Thus there were 1500 servers in the data center. We assumed that a data center consisted of four cooling modules, and each module included 500 servers, etc. From the figure, we find that although the number of cooling modules was changed, the energy-saving results of EATS were always better than those of Tech-1 and Tech-2. Besides, with the increase in the number of cooling modules, the energy consumption by Tech-1 and Tech-2 increased. However, there was basically no change in the energy consumption value of our strategy, which indicates that the idea of marginal cost we introduced in the strategy can well determine the most energy-efficient server resources and cooling resources. Thus, the effectiveness of reducing the total energy consumption of the proposed strategy is validated.

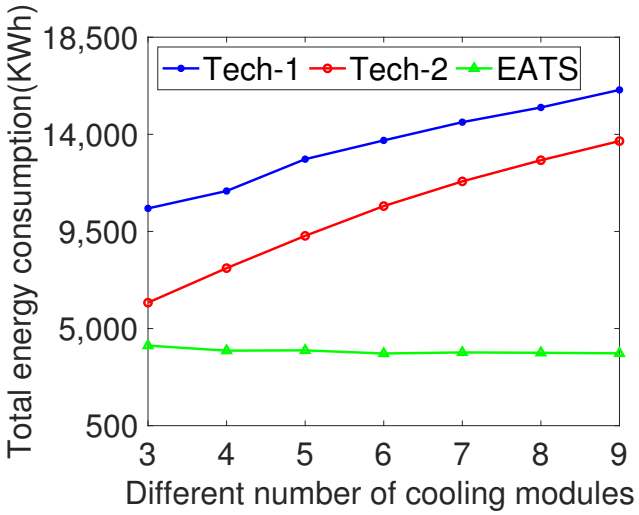


Figure 12. Comparison of total energy consumption with different numbers of cooling modules for Alibaba cluster data.

6.3.7. Different Numbers of Servers

The experiments were based on a data center scenario wherein the data center has 3000 servers and the servers are divided up by five cooling modules, such that each module contains 600 servers to support the resource needs. We also conducted the experiments in other scenarios where in a data center there were different numbers of servers. Figure 13 shows the comparison results of total energy consumption with different numbers of servers. We compare the performances of EATS with various numbers of servers in a data center (1800, 3000, 4200, 5400), respectively, and the servers were divided into six cooling modules. As such, in a data center, there was a total number of 1800 servers that were divided into six cooling modules, so each module had 300 servers; there was a total of 4200 servers that were divided into six cooling modules, so each module had 700 servers, etc. With the energy reduction results in Figure 13, we found that with the changes of server number, the energy consumption of the proposed strategy EATS was stable. However, the energy consumption results of Tech-1 and Tech-2 became more lager with the increase in the number of servers. The results in Figure 13 indicate the effectiveness of the concept of marginal cost in the evaluation of the energy behaviors in the data center. With the marginal cost evaluation, the strategy will schedule the task to the suitable servers with less energy consumption. This verifies that EATS is effective at optimizing the total energy consumption of a data center under various numbers of servers.

6.3.8. Energy Consumption in One Module

We compared the energy reduction results under various numbers of cooling modules and servers; see Sections 6.3.6 and 6.3.7. It is vital to compare the energy-saving results when there is only one cooling module in the data center, which means the data center consists of one cooling module and the module has a total number of 3000 servers. The comparison results of total energy consumption with one module are shown in Figure 14 using Alibaba cluster data.

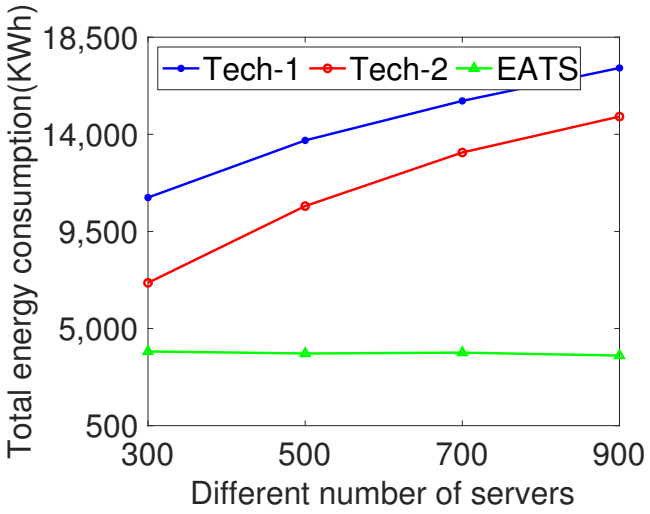


Figure 13. Comparison of total energy consumption with different number of servers for Alibaba cluster data.

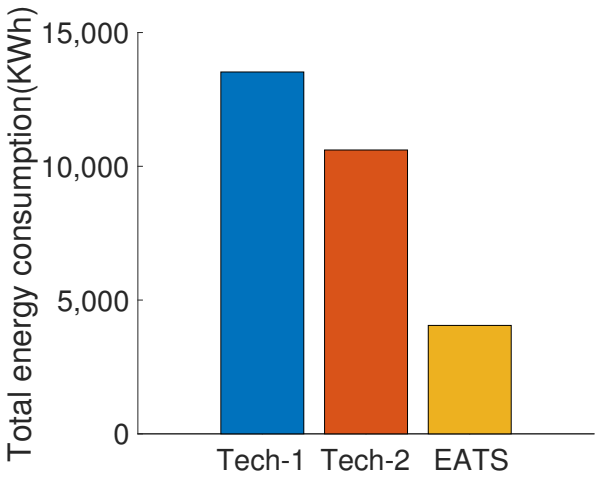


Figure 14. Comparison of total energy consumption with one module for Alibaba cluster data.

From Figure 14, it can be concluded that under one cooling module, the energy consumption of the data center achieved by EATS is the least among the three algorithms. It is hence validated that the proposed strategy EATS performs better than Tech-1 and Tech-2 in energy consumption.

Figure 15 indicates the energy-saving results under different datasets regarding total data center energy consumption with one module. The four datasets were all extracted from Alibaba cluster data, and the task arrival intensity of the datasets gradually decreased from dataset 1 to dataset 4. As indicated in Figure 15, EATS can effectively reduce energy consumption compared with the other two algorithms under various arrival intensities.

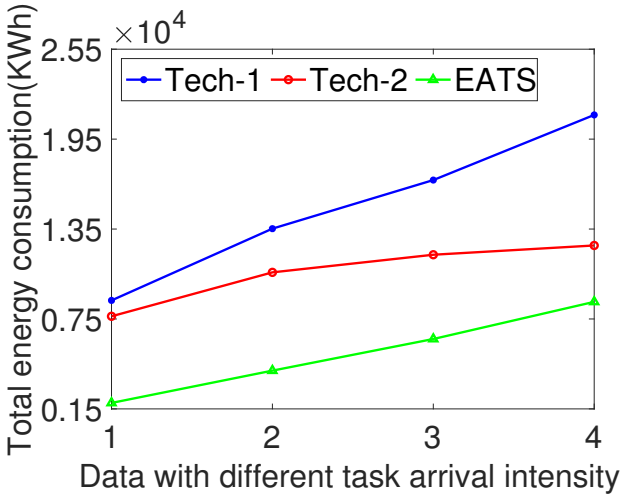


Figure 15. The comparison results of the energy consumption with one module and various datasets.

7. Conclusions and Future Work

Data centers are rapidly multiplying and becoming widespread, which is resulting in high energy consumption and inefficient resource utilization. This inefficiency of data centers wastes resources and energy, which may hinder the further development and usage of data centers. In this paper, an energy-aware task scheduling strategy based on the marginal cost and task classification method was proposed to reduce the energy consumption of servers and cooling systems cooperatively so that the total energy consumption of data centers is minimized see Supplementary. Firstly, joint energy consumption models, including the server, cooling system energy and state transition models were developed. The energy consumption optimization problem in data centers was formulated. Two cooling modes, including outside air cooling and chilled water cooling, were applied, and a strategy was developed to choose the optimal utilization of these two cooling systems in different time slots. Secondly, the concept of marginal cost in data centers was introduced to guide the task scheduling. The task classification method was used to classify tasks and servers to improve resource utilization combined with the marginal cost concept. A task scheduling algorithm using the marginal cost and task classification method was developed to solve the data center energy minimization problem to optimize the server energy consumption, the cooling system energy consumption and the state transition cost, collaboratively, such that the total data center energy consumption is reduced. Finally, experiments were conducted using two real-world datasets (Google cluster data and Alibaba cluster data). The experiment results indicate that the proposed algorithm EATS is effective at optimizing data center energy consumption and improving resource utilization. The workloads in the cloud often vary over time, and the resource requirements, arrival rates and running time have large variations. The predictions for the workload resources and workload running time are important and worthy of further investigation. Besides, task migration is also a key technology in data centers, and the introduction of marginal cost into task migration and resource management is also a problem worthy of study.

Supplementary Materials: Supplementary materials are available online at https://dl.acm.org/doi/abs/10.1145/3396851.3402657.

Author Contributions: Conceptualization, Z.L. and K.J.; methodology, K.J., F.Z. and A.F.A.; software, K.J. and C.C.; validation, K.J., P.S. and Y.W.; formal analysis, K.J. and C.C.; investigation, K.J., C.C. and A.M.; resources, A.M., P.S. and Y.W.; data curation, K.J., P.S. and A.M.; writing—original draft preparation, K.J.; writing—review and editing, F.Z., Z.L. and A.F.A.; visualization, C.C. and Y.W.; supervision, Z.L.; project administration, F.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the National Key Research and Development Program of China (grant number 2017YFB1010001) and the National Natural Science Foundation of China (grant numbers 61520106005 and 61761136014).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable. No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The author declare no conflict of interest.

References

1. Vakilinia, S. Energy efficient temporal load aware resource allocation in cloud computing datacenters. *J. Cloud Comput.* **2018**, *7*, 2. [CrossRef]
2. Reddy, V.D.; Gangadharan, G.; Rao, G.S.V. Energy-aware virtual machine allocation and selection in cloud data centers. *Soft Comput.* **2019**, *23*, 1917–1932. [CrossRef]
3. Shehabi, A.; Smith, S.; Sartor, D.; Brown, R.; Herrlin, M.; Koomey, J.; Masanet, E.; Horner, N.; Azevedo, I.; Lintner, W. United States Data Center Energy Usage Report. 2016. Available online: https://escholarship.org/content/qt84p772fc/qt84p772fc.pdf (accessed on 1 January 2021).
4. Jones, N. How to stop data centres from gobbling up the world's electricity. *Nature* **2018**, *561*, 163–167. [CrossRef]
5. Dayarathna, M.; Wen, Y.; Fan, R. Data center energy consumption modeling: A survey. *IEEE Commun. Surv. Tutorials* **2015**, *18*, 732–794. [CrossRef]
6. Zhang, W.; Wen, Y.; Wong, Y.W.; Toh, K.C.; Chen, C.H. Towards joint optimization over ICT and cooling systems in data centre: A survey. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1596–1616. [CrossRef]
7. Wan, J.; Gui, X.; Zhang, R.; Fu, L. Joint cooling and server control in data centers: A cross-layer framework for holistic energy minimization. *IEEE Syst. J.* **2017**, *12*, 2461–2472. [CrossRef]
8. Google ClusterData 2019 Traces. Available online: https://github.com/google/cluster-data/blob/master/ClusterData2019.md/ (accessed on 20 October 2020).
9. Alibaba Cluster-Trace-v2018. Available online: https://github.com/alibaba/clusterdata/blob/v2018/cluster-trace-v2018/trace_2018.md/ (accessed on 20 October 2020).
10. Gu, C.; Li, Z.; Huang, H.; Jia, X. Energy efficient scheduling of servers with multi-sleep modes for cloud data center. *IEEE Trans. Cloud Comput.* **2018**, *8*, 833–846. [CrossRef]
11. Satpathy, A.; Addya, S.K.; Turuk, A.K.; Majhi, B.; Sahoo, G. Crow search based virtual machine placement strategy in cloud data centers with live migration. *Comput. Electr. Eng.* **2018**, *69*, 334–350. [CrossRef]
12. Sahu, Y.; Pateriya, R.; Gupta, R.K. Cloud server optimization with load balancing and green computing techniques using dynamic compare and balance algorithm. In Proceedings of the 5th International Conference and Computational Intelligence and Communication Networks, Mathura, India, 27–29 September 2013; pp. 527–531.
13. Sharma, M.; Garg, R. HIGA: Harmony-inspired genetic algorithm for rack-aware energy-efficient task scheduling in cloud data centers. *Eng. Sci. Technol. Int. J.* **2020**, *23*, 211–224. [CrossRef]
14. Alboaneen, D.; Tianfield, H.; Zhang, Y.; Pranggono, B. A metaheuristic method for joint task scheduling and virtual machine placement in cloud data centers. *Future Gener. Comput. Syst.* **2021**, *115*, 201–212. [CrossRef]
15. Medara, R.; Singh, R.S. Energy Efficient and Reliability Aware Workflow Task Scheduling in Cloud Environment. *Wirel. Pers. Commun.* **2021**, 1–20. [CrossRef]
16. Deng, Z.; Cao, D.; Shen, H.; Yan, Z.; Huang, H. Reliability-aware task scheduling for energy efficiency on heterogeneous multiprocessor systems. *J. Supercomput.* **2021**, 1–39. [CrossRef]
17. Liu, X.; Liu, P.; Hu, L.; Zou, C.; Cheng, Z. Energy-aware task scheduling with time constraint for heterogeneous cloud datacenters. *Concurr. Comput. Pract. Exp.* **2020**, *32*, e5437. [CrossRef]
18. Yuan, H.; Bi, J.; Zhou, M.; Liu, Q.; Ammari, A.C. Biobjective task scheduling for distributed green data centers. *IEEE Trans. Autom. Sci. Eng.* **2020**, *18*, 731–742. [CrossRef]
19. Ham, S.W.; Park, J.S.; Jeong, J.W. Optimum supply air temperature ranges of various air-side economizers in a modular data center. *Appl. Therm. Eng.* **2015**, *77*, 163–179. [CrossRef]
20. Li, Y.; Wen, Y.; Tao, D.; Guan, K. Transforming cooling optimization for green data center via deep reinforcement learning. *IEEE Trans. Cybern.* **2019**, *50*, 2002–2013. [CrossRef] [PubMed]
21. Li, J.; Li, Z. Model-based optimization of free cooling switchover temperature and cooling tower approach temperature for data center cooling system with water-side economizer. *Energy Build.* **2020**, *227*, 110407. [CrossRef]
22. MirhoseiniNejad, S.; Moazamigoodarzi, H.; Badawy, G.; Down, D.G. Joint data center cooling and workload management: A thermal-aware approach. *Future Gener. Comput. Syst.* **2020**, *104*, 174–186. [CrossRef]
23. Moazamigoodarzi, H.; Tsai, P.J.; Pal, S.; Ghosh, S.; Puri, I.K. Influence of cooling architecture on data center power consumption. *Energy* **2019**, *183*, 525–535. [CrossRef]

24. Ahmad, F.; Vijaykumar, T. Joint optimization of idle and cooling power in data centers while maintaining response time. *ACM Sigplan Not.* **2010**, *45*, 243–256. [CrossRef]
25. Wang, Y.; Zhang, F.; Wang, R.; Shi, Y.; Guo, H.; Liu, Z. Real-time Task Scheduling for joint energy efficiency optimization in data centers. In Proceedings of the 2017 IEEE Symposium on Computers and Communications (ISCC), Heraklion, Greece, 3–6 July 2017; pp. 838–843.
26. Yan, L.; Liu, W.; Bai, D. Temperature and power aware server placement optimization for enterprise data center. In Proceedings of the 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS), Singapore, 11–13 December 2018; pp. 433–440.
27. Ran, Y.; Hu, H.; Zhou, X.; Wen, Y. Deepee: Joint optimization of job scheduling and cooling control for data center energy efficiency using deep reinforcement learning. In Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, 7–10 July 2019; pp. 645–655.
28. Ji, K.; Chi, C.; Marahatta, A.; Zhang, F.; Liu, Z. Energy Efficient Scheduling Based on Marginal Cost and Task Grouping in Data Centers. In Proceedings of the Eleventh ACM International Conference on Future Energy Systems, Melbourne, Australia, 22–26 June 2020; pp. 482–488.
29. Hamilton, J. Architecture for modular data centers. *arXiv* **2006**, arXiv:0612110.
30. Hilman, M.H.; Rodriguez, M.A.; Buyya, R. Task runtime prediction in scientific workflows using an online incremental learning approach. In Proceedings of the 2018 IEEE/ACM 11th International Conference on Utility and Cloud Computing (UCC), Zurich, Switzerland, 17–20 December 2018; pp. 93–102.
31. Liu, N.; Li, Z.; Xu, J.; Xu, Z.; Lin, S.; Qiu, Q.; Tang, J.; Wang, Y. A hierarchical framework of cloud resource allocation and power management using deep reinforcement learning. In Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, 5–8 June 2017; pp. 372–382.
32. Liu, Z.; Chen, Y.; Bash, C.; Wierman, A.; Gmach, D.; Wang, Z.; Marwah, M.; Hyser, C. Renewable and cooling aware workload management for sustainable data centers. In Proceedings of the 12th ACM Sigmetrics/Performance Joint International Conference on Measurement and Modeling of Computer Systems, London, UK, 11–15 June 2012; pp. 175–186.
33. Huang, W.; Allen-Ware, M.; Carter, J.B.; Elnozahy, E.; Hamann, H.; Keller, T.; Lefurgy, C.; Li, J.; Rajamani, K.; Rubio, J. TAPO: Thermal-aware power optimization techniques for servers and data centers. In Proceedings of the 2011 International Green Computing Conference and Workshops, Orlando, FL, USA, 25–28 July 2011; pp. 1–8.
34. Chen, T.; Wang, X.; Giannakis, G.B. Cooling-aware energy and workload management in data centers via stochastic optimization. *IEEE J. Sel. Top. Signal Process.* **2015**, *10*, 402–415. [CrossRef]
35. David, M.P.; Schmidt, R.R. Impact of ASHRAE environmental classes on data centers. In Proceedings of the Fourteenth Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm), Orlando, FL, USA, 27–30 May 2014; pp. 1092–1099.
36. Lang, W.; Patel, J.M. Energy management for mapreduce clusters. *Proc. VLDB Endow.* **2010**, *3*, 129–139. [CrossRef]
37. Akbari, A.; Khonsari, A.; Ghoreyshi, S.M. Thermal-aware virtual machine allocation for heterogeneous cloud data centers. *Energies* **2020**, *13*, 2880. [CrossRef]