# RESEARCH

# **Open Access**

# A resource scheduling method for cloud data centers based on thermal management



Li Mao<sup>1</sup>, Rui Chen<sup>2</sup>, Huiwen Cheng<sup>3</sup>, Weiwei Lin<sup>2\*</sup>, Bo Liu<sup>3\*</sup> and James Z. Wang<sup>4</sup>

# Abstract

With the rapid growth of cloud computing services, the high energy consumption of cloud data centers has become a critical concern of the cloud computing society. While virtual machine (VM) consolidation is often used to reduce energy consumption, excessive VM consolidation may lead to local hot spots and increase the risk of equipment failure. One possible solution to this problem is to utilize thermal-aware scheduling, but existing approaches have trouble realizing the balance between SLA and energy consumption. This paper proposes a novel method to manage cloud data center resources based on thermal management (TM-VMC), which optimizes total energy consumption and proactively prevents hot spots from a global perspective. Its VM consolidation process includes four phases where the VMs scheduler uses an improved ant colony algorithm (UACO) to find appropriate target hosts for VMs based on server temperature and utilization status obtained in real-time. Experimental results show that the TM-VMC approach can proactively avoid data center hot spots and significantly reduce energy consumption while maintaining low Service Level Agreement (SLA) violation rates compared to existing mainstream VM consolidation algorithms with workloads from real-world data centers.

Keywords Cloud data center, Energy consumption, Thermal management, Virtual machine consolidation

# Introduction

In recent years, the rapid development of cloud computing, big data, and the Internet of Things has driven the widespread construction [1] of data centers that serve as the physical platform and infrastructure of cloud computing. According to the statistics of the 2022 Data Center White Paper, although the number of new servers put into use in the world from 2015 to 2021 is relatively stable, the scale of data centers is expected to continue to grow in the future [2]. Today, data centers consume about

\*Correspondence: Weiwei Lin linww@scut.edu.cn Bo Liu

liugubin530@126.com

<sup>1</sup> Guangdong Police College, Guangzhou, China

<sup>2</sup> School of Computer Science and Engineering, South China University

of Technology, Guangzhou, China

<sup>3</sup> School of Computer Science and Technology, South China Normal University, Guangzhou, China

2% of the world's energy [3] while producing more than 43 million tons of CO2 annually [4]. The rising energy consumption increases the operating cost of enterprises and pollutes the environment, which is not conducive to the development of green data centers. High energy consumption has become the main challenge of data center management since most data centers experience relatively low energy efficiency. Studies have shown that the effective resource utilization rate of worldwide data centers is only 20%-30% [5]. Therefore, Power Usage Effectiveness (PUE) is still an important research topic in data center management.

Most of the energy consumption of cloud data centers comes from computing and cooling systems. In recent years, cloud data centers have adopted virtual machine consolidation (VMC) techniques to periodically migrate and reallocate virtual machines (VMs) to reduce the number of active hosts and hence reduce energy consumption [6–9]. Although reducing the number of active hosts seems to reduce the energy consumption of



© The Author(s) 2023. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

<sup>&</sup>lt;sup>4</sup> School of Computing, Clemson University, Clemson, SC, USA

computing systems, excessive consolidation of VMs may overload the hosts, resulting in hot spots, thereby affecting system performance. Furthermore, hot spots cause an increase in the workload and energy consumption of air conditioners. To avoid the hot spots, researchers proposed thermal-aware scheduling methods for thermal management to optimize the workload distribution of computing systems and control data center temperature [10–14]. Thermal-aware scheduling balances the temperature of hosts through resource reallocation to reduce thermal gradients and hot spots within the data center. These methods usually model the problem of minimizing data center energy consumption into a nonlinear optimization problem with thermal constraints and propose a resource allocation algorithm to solve this problem. However, existing methods focus on algorithms for the virtual machine placement stage (VMP) in the VMC process and often fail to balance the quality of services (QoS) and energy consumption. There is an urgent need for a new energy-saving solution that can satisfy the objectives of both cloud providers and users.

To this end, we propose a thermal management-based resource scheduling method for cloud data centers (TM-VMC), which dynamically allocates virtual machines by modeling the data center with both temperature and resource utilization considered. This dynamic VMC process consists of four integrated parts: a host overload detection process (TU) based on CPU temperature and utilization, an underload host detection process (HOAVG) based on median average utilization, a virtual machine selection policy (MMR) based on maximum migrated memory, and a virtual machine placement policy (UACO) based on an improved ant colony algorithm. The goal of TM-VMC is not to minimize the number of hot spots in the data center but to actively avoid hot spots and reduce total energy consumption while ensuring service quality for users. In addition, we design different algorithms in various stages of the VMC process to form a complete virtual machine consolidation process scheme. The main contributions of this work include:

(1) A thermal management-based resource scheduling method (TM-VMC) for the optimization of the total energy consumption of the data center. This method includes four algorithms in the process of VMC. TM-VMC can proactively avoid data center hot spots and minimize the total energy consumption of the data center while meeting SLAs (Service Level Agreements).

(2) An overload detection strategy (TU) based on host temperature and utilization. It determines whether a host is overloaded by detecting host temperature and utilization status in real time. It relocates redundant VMs away from the overloaded hosts, proactively avoiding data center hot spots.

(3) An underload host detection method (HOAVG) based on median average utilization. It selects underloaded hosts based on median average utilization and shuts them down, which avoids the overload problem caused by greedy server shutdown policies.

(4) A virtual machine selection policy (MMR) based on the memory allocated to the VMs. It migrates the virtual machines with the largest allocated memory, which reduces the frequent migration activities in the data center.

(5) An improved ant colony algorithm (UACO) used in the virtual machine placement stage. It includes a novel state transition rule and fitness function suitable for the thermal management resource scheduling problem. It also uses an improved pheromone update method to avoid the problem of easily falling into local optima in traditional Ant Colony Optimization algorithm (ACO), allowing the algorithm to find the optimal virtual machine placement solution in a short time.

The remainder of this paper is organized as follows: After introducing recent studies on VMC and thermalaware scheduling for data centers in Sect. 2, we present the problem definition of VM resource scheduling based on thermal management in Sect. 3. We discuss the framework of the TM-VMC method and the related design of each stage in the consolidation process in Sect. 4. After conducting performance studies of our proposed TM-VMC method and several state-of-art VMC methods using the CloudSim simulation platform in Sect. 5, we present our conclusion and discuss the future work in Sect. 6. Table 1 lists the abbreviations used in this paper.

#### **Related work**

Virtual machine consolidation is a common method for optimizing computing power consumption. Dynamic virtual machine consolidation minimizes power consumption by distributing computing resources among fewer hosts and shutting down underutilized servers. There have been numerous studies addressing the problem of VMC [15–18]. Virtual machine placement (VMP) is the most important stage in the VMC process, that is, the process of reassigning *n* virtual machines to *m* hosts. In the early stage, researchers regarded the VMP problem as a box-packing problem with NP-hard properties, and usually used heuristic algorithms to solve its optimal allocation scheme, such as first-fit algorithm (FF), bestfit algorithm (BF), descending first-fit algorithm (FFD), descending best-fit algorithm (BFD), etc. [19]. Reference [7] proposed two new heuristic algorithms for the VMC

 Table 1
 Table of abbreviations used in this paper

abbreviation	Definition		
VM	Virtual machine		
VMC	Virtual machine consolidation		
TM-VMC	A Thermal Management-based Vir- tual Machine Consolidation Method		
VMP	Virtual machine placement		
MMR	A virtual machine selection policy based on maximum migrated memory		
TU	An overload detection strategy based on host temperature and utilization		
HOAVG	an underload host detection method based on median average utilization		
SLA	Service Level Agreement		
CRAC	Computer Room Air Conditioning		
COP	Coefficient of Performance		
ACO	Ant Colony Optimization algorithm		
UACO	A virtual machine placement policy based on improved ant colony algorithm		

process based on Beloglazov et al. [6]. The authors proposed a unique method for underloaded host detection: if the CPU utilization of two hosts is equal, the host with fewer VMs has a better chance to switch to sleep mode. The method effectively reduces SLA violation rates. In the virtual machine placement phase, they introduced an efficient QoS-aware algorithm based on the minimum correlation between host utilization and virtual machines. With the wide application of reinforcement learning (RL) in optimization problems, some researchers have proposed RL-based virtual machine consolidation methods [20–23].

In recent years, swarm intelligence algorithms have been widely used in virtual machine scheduling problems [24]. For example, Kansal et al. [8] first applied the Firefly algorithm (FA) to the energy-aware data center VM scheduling problem. Since the traditional particle swarm algorithm (PSO) is applicable to continuous problems, Ibrahim et al. [9] used a discrete version of particle swarm optimization algorithm (PAPSO) based on decimal coding to map the migrated VMs to the most suitable physical machines, which can reduce energy consumption without violating SLAs. In [25], a gene aggregation genetic based VM migration algorithm VMM-GAGA was designed by improving the genetic algorithm (GA) encoding. The authors synthesized a set of chromosomes as genes in two virtual machines with small footprint but high communication and migrated them to the same host with low utilization. Liu et al. [26] used a low complexity Extreme Learning Machine (ELM) prediction based Multiple Swarm Ant Colony System algorithm (ELM-MPACS) to schedule virtual machines. The algorithm first uses ELM to predict the host state and then moves the VMs on the overloaded hosts out of the host while the VMs on the underloaded hosts are merged to another under-utilized host with higher utilization. In [27], a multi-objective algorithm for VM placement was proposed. The algorithm is based on a flower pollinationbased nondominated sorting optimization (FP-NSO) algorithm and aims to improve resource utilization, reduce energy consumption, and lower carbon emissions.

In order to reduce the local hot spots problem caused by virtual machine consolidation, researchers have incorporated data center thermal effects into the scheduling method [28]. By developing and evaluating fine-grained models of Computer Room Air Conditioning (CRAC), room temperature, humidity, and servers, and integrating these models into scheduling algorithms, minimizing the total energy consumption using the characteristics of the data center is a basic solution for thermal-aware scheduling [29]. The method transforms the data center energy minimization problem into a nonlinear programming problem with thermal constraints, and usually uses heuristic algorithms to solve the suboptimal solution of the programming problem. Ilager et al. [10] formulated the energy minimization problem as an optimization problem with thermal constraints and proposed a thermal-aware scheduling algorithm (ETAS) that was able to control data center temperature and dynamically integrate virtual machines so that the total energy consumption is minimized. Similarly, Feng et al. [11] proposed a global energy-aware VMP strategy, but different from the literature [10], the energy consumption model established by the authors includes computing systems, cooling systems and network equipment. Arroba et al. [12] designed a meta-heuristic optimization strategy that relies on a simulated annealing algorithm (SA) to achieve joint optimization of IT and cooling energy consumption. To control server temperature, the authors defined a maximum cooling set point (maxCooling Set Point) for each host, and the final cooling setpoint is set as the minimum value among the maximum cooling setpoints of all servers. Reference [13] proposed a simulated annealing based algorithm (SABA) to solve the VM placement problem. SABA achieves an approximate optimal value with fewer iterations than SA and reduces energy consumption by considering thermal recirculation. Van et al. [14] proposed an optimized thermal-aware work scheduling and control model with the aim of finding the optimal setpoints for host temperature and workload

distribution to minimize data center energy consumption. In this model, the authors assumed that the servers in the data center are isomorphic, thereby transforming the complex energy minimization optimization problem into a simple equivalence optimization problem for a homogeneous data center, but the assumption of server isomorphism limits the practical scope of this approach.

For a long time, researchers have made many attempts to optimize energy consumption in data centers. Virtual machines are dynamically consolidated to save energy but violate thermal regulations, while thermal-aware scheduling is difficult to guarantee SLA and energy consumption at the same time. Some of the previous thermal -aware scheduling studies were only applicable to small data center scenarios, some optimized data center energy consumption from a local perspective, and some were based on homogeneous server applications and were not applicable to heterogeneous environments. To address the above issues, we model the total data center power consumption and propose a thermal management-based heterogeneous data center resource scheduling method (TM-VMC) for total energy consumption optimization. The proposed approach optimizes data center energy consumption from a global perspective and can prevent data centers from generating hot spots and reduce energy consumption on the basis of guaranteed SLAs.

### Overall architecture based on thermal management resource scheduling Overall framework

In this work, we first abstract a resource scheduling framework based on thermal management (Fig. 1). In a virtualized data center, tasks are encapsulated into one or more virtual machines, and after users submit load tasks, the virtual machine broker allocates resources to suitable hosts. The system monitors the host status in real time, selects overloaded hosts, and migrates some VMs in overloaded hosts and all VMs in underloaded hosts to other suitable hosts. The scheduling system minimizes energy consumption by dynamically consolidating virtual machines. The decision of the scheduling system is based on the models established in the paper, including the computing system power consumption model, cooling system power consumption model, and temperature model.

## **Problem definition**

To facilitate the definition of the problem, Table 2 gives the variables and the corresponding explanations commonly used in this paper. An efficient scheduling method should guarantee QoS while minimizing energy consumption. In the process of consolidating allocations, different deployment results will be obtained depending on the optimization objectives. The strategy of this paper is to avoid hot spots and simultaneously reduce the total energy consumption of the data center without compromising the quality of service. To this end, we add temperature constraints and utilization constraints to the virtual machine scheduling model, with the primary optimization goal of minimizing the total power consumption of the data center. In summary, the resource scheduling problem based on thermal management is summarized as follows:

$$MinimizeP_{total} = P_{IT} + P_{CRAC} = \left(1 + \frac{1}{COP(T_{sup})}\right)P_{IT}$$
(1)



Fig. 1 Overall framework of thermal management-based resource scheduling

#### Table 2 List of variables

Variable	Description			
P <sub>total</sub>	Total data center power consumption			
P <sub>IT</sub>	Computing system power consumption			
P <sub>CRAC</sub>	Cooling system power consumption			
U <sub>max</sub>	Upper threshold of the host CPU utilization			
T <sub>max</sub>	Upper threshold for the CPU temperature of the host			
T <sup>j</sup> Cpu	CPU temperature of host <i>j</i>			
R	Server heat resistance			
С	Server heat capacity			
R <sub>cpu</sub>	CPU capacity			
R <sub>mem</sub>	Memory resource capacity			
U <sup>j</sup> <sub>cpu</sub>	CPU utilization of host <i>j</i>			
R <sub>bw</sub>	Bandwidth resource capacity			
Pi	Power consumption of a single node <i>i</i>			
T <sub>sup</sub>	CRAC air supply temperature			

Among them, constraint 2 ensures that the host CPU utilization does not conflict with the threshold due to the increase of workload, and constraint 3 ensures that the server temperature does not exceed the safety threshold. Constraint 4 limits that the sum of VM resource requests cannot exceed the resource availability of physical machines, n is the number of virtual machines placed on *Host<sub>j</sub>*. Constraint 5 restricts a virtual machine to be placed on only one host,  $x_{i,j}$  is a binary variable that indicates whether the  $VM_i$  is placed on the *Host<sub>j</sub>*, m is the total number of hosts.

#### Data center modeling

#### IT energy consumption model

A large number of scholars have studied data center computing device energy consumption modeling methods [30-32]. Unlike these works, instead of using a server power consumption model based on a simple linear model, we use actual power consumption data provided by SPECpower[33] benchmarking results for server energy consumption modeling. SPECpower provides researchers with data on the power consumption of servers under different load levels, which can be used to calculate real-time server power. The real-time power  $P_i(t)$ of a single server can be obtained according to the CPU utilization in different ranges, and the piecewise function related to time *t* can be obtained by using the linear interpolation method for the server data of SPECpower:

$$P_{i}(t) = \begin{cases} \alpha_{1}u_{i}(t) + b_{1}, 0 \leq u_{i}(t) < 0.1\\ \alpha_{2}u_{i}(t) + b_{2}, 0.1 \leq u_{i}(t) < 0.2\\ \alpha_{3}u_{i}(t) + b_{3}, 0.2 \leq u_{i}(t) < 0.3\\ \dots\\ \alpha_{10}u_{i}(t) + b_{10}, 0.9 \leq u_{i}(t) \leq 1 \end{cases}$$

$$(2)$$

In this formula,  $u_i(t)$  is the CPU utilization of *host*<sub>i</sub> at time *t*, { $\alpha_1$ ,  $\alpha_2$ ,...,  $\alpha_{10}$ } and { $b_1$ ,  $b_2$ ,...,  $b_{10}$ } are the slopes and intercepts of each line in the segment function, respectively, and they are obtained by segmented linear interpolation. Integrating  $P_i(t)$  can obtain the energy consumption of *host*<sub>i</sub> in the time period  $t_1 \sim t_2$ , as shown in formula (3). The total energy consumption of IT equipment in the data center is the sum of the energy consumption of all servers, as shown in Eq. (4).

$$E_i = \int_{t_1}^{t_2} P_i(t) dt \tag{3}$$

$$E = \sum_{i=1}^{m} E_i \tag{4}$$

#### Cooling system energy consumption model

The refrigeration control system is used to control the humidity and temperature of the data center, and it also consumes energy during operation, including air conditioners and fans. CRAC is the workhorse of the refrigeration system, optimizing the control of the CRAC system is crucial to the optimization of the energy consumption of the cooling system of the data center [34]. The efficiency of a data center cooling system is often measured using the coefficient of performance (COP), which depends on the physical layout and thermodynamic characteristics of the data center. It is generally defined as the ratio of the total power consumed by the cooling system to extract heat, for which the CRAC power consumption can be modeled as:

$$P_{CRAC} = \frac{P_{IT}}{COP(T_{sup})}$$
(5)

Regarding the COP, we use the empirical model proposed by Zhan et al. [35]:

$$COP = 0.0068 T_{sup}^{2} + 0.0008 T_{sup} + 0.458$$
(6)

Therefore, the power consumption of the cooling system is defined as:

$$P_{CRAC} = \frac{P_{IT}}{0.0068T_{sup}^2 + 0.0008T_{sup} + 0.458}$$
(7)

#### Temperature model

In a traditional air-cooling system, the cold air from the data center air conditioner enters the rack, flows through the server, and exits the rear of the rack. In such a data center, it is necessary to focus on the airflow inlet temperature  $T_{in}$  and the server CPU temperature  $T_{cpu}$ . For both temperature metrics, we use existing methods to model and incorporate them into our temperature model. The host inlet temperature is defined as a linear combination of the air conditioning supply air temperature  $T_{sup}$  and the temperature rise due to the heat recirculation [36].

$$T_{in}^{i}(t) = T_{sup} + \sum_{j=1}^{N} d_{i,j} \times P_{j}(t)$$
(8)

Considering the heat recirculation effect in a specific region based on the current physical layout of the data center, the recirculation effect can be quantified as the heat distribution matrix  $d_{i,j}$ . It represents the degree to which the inlet temperature of host *i* is affected by host *j*, and *N* is the number of hosts in the recirculation zone. The heat distribution matrix of this experiment comes from the literature [10]. According to the suggestion of ASHRAE [37], we set the supply temperature  $T_{sup}$  of the CRAC to 25 °C.

The server inlet temperature cannot represent the server's operating state. The ultimate purpose of data center cooling is to control the server CPU temperature. We use the RC model [38] to define it as:

$$T = PR + T_{in} + (T_0 - PR - T_{in}) \bullet e^{\frac{-t}{RC}}$$
(9)

where *T* is the server CPU temperature, *P* is the server power, R and C are the thermal resistance and thermal capacity of the server, respectively,  $T_0$  is the initial temperature of the system, and  $T_{in}$  is the server inlet temperature.

# Thermal management-based virtual machine consolidation

## VM consolidation process

Dynamic virtual machine consolidation minimizes power consumption by distributing computing resources among a small number of hosts and shutting down underutilized servers. In virtual machine consolidation, there are several key issues to address. For example, when to trigger a VM migration (host overload or underload), which VMs are selected to perform the migration task (select VMs to migrate), which target hosts should be selected to redeploy the selected VMs (target host selection).

To solve the above problems, we propose a thermal management-based virtual machine scheduling method (TM-VMC). It includes: a host overload detection

method (TU) based on CPU temperature and utilization, an underload host detection method (HOAVG) based on median average utilization, a VM selection policy (MMR) based on maximum migrated memory, and a VM placement policy (UACO) based on improved ant colony algorithm. As shown in Figs. 2 and 3, all hosts in the data center are divided into three groups: the overloaded host set  $H_{al}$ , the underloaded host set  $H_{ul}$ , and the normal host set  $H_n$ . Specifically, the TM-VMC policy first uses the TU algorithm to consider hosts with high utilization or temperature as overloaded and add them to the  $H_{ol}$  set. Then, MMR is used to select VMs to be migrated from the  $H_{ol}$  set, and together with all VMs in the underloaded host  $H_{ul}$ , they form the list of VMs to be migrated  $VM_{M list}$ , and the underloaded host detection is based on the HOAVG method. Finally, the improved ant colony algorithm (UACO) proposed in this paper is used to find suitable target hosts in the normal host  $H_n$  for each VM to be migrated.

#### Host overload detection policy

Most existing virtual machine consolidation methods rely solely on CPU utilization to detect whether a host is overloaded. By establishing a thermal analysis model, we design an overloaded host detection strategy (TU) based on CPU temperature and utilization, which utilizes the combination of host utilization and server CPU temperature to achieve host overload detection. The key to the overload detection algorithm is to determine the reasonable upper threshold of the host. We propose a method for AMD to determine the upper utilization threshold, which is expressed as follows:

$$AMD = Avg(|U_{history_i} - Median(U_{history})|)$$
(10)

$$U_{max} = 1 - s * AMD \tag{11}$$

Among them, AMD is the average difference between each record and the median in the historical server utilization records, representing the average dispersion of server utilization history.  $U_{history_i}$  represents the *i*-th item of the host utilization history, and *s* is the safety factor, which is used to adjust the evaluation level.

In a real physical machine, server status is not only determined by CPU utilization, but host temperature is also an important factor. Excessive host temperature will affect system performance and increase the risk of machine failure. Therefore, we detect the host status by considering both utilization and temperature. The



Fig. 2 TM-VMC dynamic virtual machine consolidation process



Cluster of data center servers

Fig. 3 Underload server processing method HOAVG

temperature measurement in this paper is based on the server CPU temperature model (Eq. 13) described in Sect. 3.3.3, and the upper temperature threshold is set to the static threshold  $T_{max}$ . We set this value to 95 °C according to the recommendation of the literature [10]. When the utilization rate of the host to be detected is

greater than  $U_{max}$  or the temperature exceeds the static temperature upper threshold  $T_{max}$ , the host is considered to be overloaded and added to the list of overloaded hosts. Algorithm 1 shows the pseudocode of the algorithm.

Input: hostList
-----------------

Output:	overloadedHosts
1: <b>for</b> he	ost in hostList <b>do</b>
2: if	f (isHostOverUtilized (host)    isOverTemprature (host)) then
3:	overLoadedHosts←host
4: en	d if
5: end f	or
6: retur	n overloadedHosts
7: funct	ion isHostOverUtilized (host)
8: L	$U_{max} \leftarrow 1 - s * Avg( U_{history_i} - Median(U_{history}) )$
9: u	tilization←host.getUtilization()
10: <b>retu</b>	<b>rn</b> utilization> <i>U</i> <sub>max</sub>
11: <b>end</b>	function
12: <b>func</b>	etion isOverTemprature (host)
13:	$T_{max} \leftarrow 95^{\circ}\mathrm{C}$
14:	calculate $T_{in}$ according to equation (8)
15:	calculate $T_{cpu}$ according to equation (9)
16:	temperature←getHostTemperature(host)
17: <b>retu</b>	<b>rn</b> temperature > <i>T</i> <sub>max</sub>
18: end	function

Algorithm 1. TU Host Overload Detection

#### Host underload detection policy

In an enterprise-level data center, even a host with zero load still consumes about 70% of its peak power consumption. Research shows that the effective resource utilization rate of the data center is only 20%-30% [5], which leads to a large amount of unnecessary energy power consumption, turning off these low-load hosts can save power. The commonly used method to determine underloaded hosts is the greedy method based on minimum utilization. Unlike previous work, we propose an underloaded host detection method (HOAVG) based on median average utilization.

As shown in Fig. 3, HOAVG first divides all active hosts into two parts, with the underutilized half as the pending host  $H_p$  and the remaining hosts as the higher-utilized part  $H_{high}$ . Then calculate the average utilization value  $U_{avg}$  of all  $H_p$ , take the part of hosts with CPU utilization less than  $U_{avg}$  as underloaded hosts, and the part of hosts with CPU utilization greater than  $U_{avg}$  together with  $H_{high}$  to form the target host  $H_{target}$ , as the target host for placing the virtual machines to be migrated.

#### VM selection policy

Once a host is detected as being overloaded, the VM selector will select specific virtual machines from this host and migrate them to other suitable hosts. An efficient VM selection strategy can minimize the cost of migration and SLA violations. We design a maximum migrated

Page 8 of 18

memory-based policy (MMR) that always selects the VM with the largest memory among the candidate VMs for migration. MMR first obtains the list of virtual machines from the overloaded host, and then greedily selects the virtual machine with the largest memory and puts it into the list of virtual machines to be migrated. After the selected virtual machine is migrated, it is necessary to redetect whether the host is overloaded. If the host is still overloaded, the MMR migration policy will be applied to the host again to select another virtual machine until the host is considered not to be overloaded. This strategy can effectively reduce the number of virtual machine migrations, thereby reducing energy consumption.

#### VM placement policy

The virtual machine placement strategy in this paper is based on the improved ant colony algorithm UACO. Ant Colony Optimization algorithm (ACO) is a heuristic optimization algorithm proposed by Marco Dorigo [39] in 1996 to simulate the behavior of ant colonies seeking the optimal path in the foraging process, which has been widely used in the virtual machine scheduling problem.

#### Initial pheromone and heuristic information

As with any ACO implementation, the definition of pheromone and heuristic information is critical to building high-quality solutions. In the VMP problem, if there are *n* virtual machines to be assigned to *m* normally running hosts, the pheromone  $\tau_{ij}$  represents the support rate of  $VM_i$  placed on the specified  $PM_j$ . Each time a path is selected by an ant, the pheromone concentration of that path is increased. In the initial stage of the algorithm, the initial pheromone is set in this paper as follows:

$$\tau_0 = (\frac{1}{H_a \bullet M_{\nu m}}) \tag{12}$$

where  $H_a$  represents the number of active hosts in the data center and  $M_{vm}$  represents the number of virtual machines to be migrated.

The heuristic information is a problem-specific value that represents the expectation of assigning a virtual machine  $VM_i$  to a host  $Host_j$ . In general, the heuristic function is determined based on different objectives for different problems, and heuristic information may vary significantly even in ACO implementations of the same problem [40]. The definition of heuristic information in this paper aims to minimize the total energy consumption of the data center after each virtual machine is assigned to a host by considering the resource utilization balance and the total resource utilization. Therefore, we define the heuristic information as:

$$\eta_{ij} = \frac{1}{P_{j \ after}} \tag{13}$$

where  $P_{j\_after}$  indicates the power consumption after *Host<sub>i</sub>* is put into the virtual machine.

#### State transition rule

The ants decide the computational node to be selected by the current VM through the state transfer rule, and we adopt the dynamic pseudo-random proportion rule to design the state transfer probability formula of the VM. When choosing a target host for the virtual machine  $VM_i$ , each ant makes a decision in the following way:

$$A_{i} = \begin{cases} argmax_{j \in J_{k(i)}} (\alpha \tau_{ij} + \beta \eta_{ij}) ifc \le c_{0} \\ P_{ij}^{k} otherwise \end{cases}$$
(14)

In the above formula, each ant generates a random number c in the [0,1] interval before selecting the target host for the virtual machine.  $c_0$  is a fixed parameter whose value is between 0 and 1. If the value is too large, the algorithm will easily converge to a local optimal value prematurely, resulting in stagnation. If the value is too small, the solution time of the algorithm will increase. In each iteration, if the generated random number is less than or equal to  $c_0$ , the virtual machine is assigned to the host with the highest pheromone concentration and heuristic information in the target host set, which helps the ants to quickly converge to a high-quality solution, that is, the ant selects the target as follows:

$$argmax_{j\in J_{k(i)}}(\alpha\tau_{ij}+\beta\eta_{ij}) \tag{15}$$

Among them,  $J_{k(i)}$  is the set of possible target hosts for  $VM_i$ ,  $\alpha$  is the pheromone influence coefficient, which represents the importance of the pheromone in the process of placing the virtual machine, and  $\beta$  is the influence coefficient of the heuristic information. When *c* is greater than  $c_0$ , the virtual machine selects the target physical machine according to the roulette, and calculates the probability of the roulette according to the following formula. At this time, the ants will conduct a wider search to avoid the premature stagnation of the algorithm.

$$p_{ij}^{k}(t) = \begin{cases} \frac{[\tau_{ij}(t)^{\alpha}] \bullet [\eta_{ij}(t)^{\beta}]}{\sum_{s \in J_{k(i)}} [\tau_{is}(t)^{\alpha}] \bullet [\eta_{is}(t)^{\beta}]}, & if j \in J_{k(i)} \\ 0, otherwise \end{cases}$$
(16)

#### Pheromone update rule

For the pheromone update of UACO algorithm, the local pheromone update rule and global pheromone update rule are proposed in this paper. Local pheromone update is used to evaporate pheromones to reduce the impact of low-quality solutions. In this paper, ants use the following local update rule to update pheromones:

$$\tau_{i,j} = (1 - \rho) \bullet \tau_{i,j} + \rho \bullet \tau_0 \tag{17}$$

Among them,  $\rho$  is the local pheromone volatilization factor, and its value is between 0 and 1, indicating the degree of pheromone evaporation. The larger the  $\rho$ , the less pheromone remains.  $\tau_0$  is the initial pheromone amount, in this paper, it is calculated according to formula (12).

ACO is easy to fall into local optimum due to its positive feedback characteristics. To address this issue, we have designed a global pheromone update method for the ants, wherein the pheromone concentration increment of each ant is modified based on its fitness function value. The higher the fitness value, the larger the pheromone increment. This method makes the search of the algorithm have a certain direction and gradually approach the optimal solution. In order to evaluate the quality of the solution, we define the fitness function as:

$$F_{c} = \varepsilon * \frac{P_{ant\_max}}{\sum_{1}^{N_{ant}} P_{ant\_max}} + (1 - \varepsilon) * \frac{Temp_{ant\_sum}}{\sum_{1}^{N_{ant}} Temp_{ant\_sum}}$$
(18)

Among them, the parameter  $\varepsilon$  is a number between 0 and 1, which is used to adjust the weight of the solution.  $P_{ant\_max}$  represents the maximum power consumption that may be generated by the target host in the allocation scheme generated by the ant.  $Temp_{ant\_sum}$  represents the sum of all target host temperatures in the allocation scheme generated by this ant.

Specifically, in our method, after completing one iteration, the ants first calculate their respective fitness values and then sort all ants in ascending order according to the fitness value  $F_c$ . Then, the top 20% ants were selected to increase the pheromone concentration and recorded as the better ants Ant<sub>better</sub>. The remaining ants didn't increase the pheromone concentration, they only had the effect of pheromone volatilization. In order to prevent the optimal solution from being forgotten due to insufficient initial advantages, our method adaptively adjusts the pheromone concentration of the ants in this part of Ant<sub>better</sub> according to their own fitness function values. Specifically, according to the position of each ant in the ranking, the weight of increasing pheromone concentration is set. The global pheromone update rule is as follows:

$$\tau_{ij}(t+n) = \begin{cases} \tau_{ij}(t) \cdot \left(\Delta \tau_{max} - l \cdot (rank_k - 1)\right) ifAnt \in Ant_{better} \\ (1 - \sigma) \cdot \tau_{ij}(t) otherwise \end{cases}$$
(19)

where  $\sigma$  is the global pheromone volatility factor,  $\Delta \tau_{max}$  is the maximum proportion of pheromone concentration

that can be increased,  $rank_k$  denotes the ant ranked *k*-th, and *l* is defined as:

$$l = \frac{\Delta \tau_{max} - 1}{K} \tag{20}$$

Here, *K* is the number of  $Ant_{better}$  ants. According to this rule, the first ant updates the pheromone concentration to the original  $\Delta \tau_{max}$ , while the last ant, that is, the ant at the 20% edge, the proportion of increase in pheromone concentration of its corresponding path tends to be close to 0. The following algorithm 2 shows the pseudocode of the UACO algorithm.

Ostrata Missing Mar
Output: Migration Map
1: initialize all the parameters
2: initialize PowerMatrix() and PheromoneMatrix()
3: for $c \in [1, N_c]$ do
4: for $k \in [1, N_{ant}]$ do
5: for $i \in [1, N]$ do
6: compute heuristic using equation (13)
7: generate a random variable c between 0 and 1
8: if $c \le c_0$ then
9: choose a destination host using equation (15)
10: else
11: choose a destination host using equation (16)
12: end if
13: end for
14: update local pheromone using equation (17)
15: end for
16: calculate Fitness value according equation (18)
17: sort Fitness in ascending order
18: AntBetterList←the first 20% ants
19: update global pheromone using equation (19)
20: end for

Algorithm 2. UACO VM Placement

# Experiment setup and analysis of results

#### Experiment setup

In this paper, we use CloudSim 3.0.3 [41] as the simulation experiment platform to verify the performance of the TM-VMC method. Experiments build cloud data center simulations to evaluate the feasibility and effectiveness of the TM-VMC strategy. The data center set in this experiment contains 10 areas, each area consists of 10 racks, the racks are arranged in a  $5 \times 2$  layout, and each rack is placed with 10 servers. The experiment consists of 1000 heterogeneous servers, half of which are Intel Xeon X5670 and the other half are Intel Xeon X5675. The host configuration details are shown in Table 3. Four types of single-core virtual machine instances were selected in the experiment, as shown in Table 4. The workload used in the experiment is derived from the CoMon project, which monitors the operation of the PlanetLab infrastructure [42]. The data of the project includes the CPU utilization of more than 1000 virtual machines in 500 different regions around the world, and the data recording period is five minutes. Table 5 lists the workload data characteristics.

To evaluate the performance of the TM-VMC method, the experiment is conducted in two parts. Firstly, the TU-MMR-HOAVG combination strategy formed by the combination of the overloaded host detection strategy TU, the virtual machine selection strategy MMR and the underloaded host detection strategy HOAVG in the TM-VMC method is compared and tested to verify the performance of the proposed algorithm. Subsequently, the UACO method is compared with other virtual machine placement methods to verify the performance of UACO. At the same time, in order to evaluate the performance of the method, we set the following evaluation indicators:

(1) Total Energy Consumption: reducing the total energy consumption of the data center is the primary goal of this paper. In this experiment, the total energy consumption is expressed as the sum of the energy consumption of the computing system and the cooling system.

(2) Number of Hot Spots: high server temperatures may affect the overall operational implementation of the data center, which can lead to server downtime with serious consequences. This metric indicates the number of times the host exceeds the temperature threshold throughout the simulation.

(3) SLA Violation Rate: this metric captures the system performance overhead due to dynamic consolidation, which mainly includes performance degradation due to full load in the data center  $SLA_{TAH}$  and

 Table 4
 Virtual machine instance types

Туре	CPU frequency (MIPS)	RAM (MB)
Extra large instance	2500	900
Large instance	2000	1700
Small instance	1000	1400
Micro instance	500	600

#### Table 3 Server characteristics

Туре	СРՍ Туре	Frequency (GHz)	Core	RAM (GB)
IBM × 3550 M3	Intel Xeon X5670	2.93	6	4
IBM×3550 M3	Intel Xeon X5675	3.07	6	4

Load instance	Date         Number of VMs         Average utilization rate(%)		The standard deviation(%)	
P1	2011/03/03	1052	12.31	17.09
P2	2011/03/06	898	11.44	16.83
Р3	2011/03/09	1061	10.70	15.57
P4	2011/03/22	1516	9.26	12.78
P5	2011/03/25	1078	10.56	14.14
P6	2011/04/03	1463	12.39	16.55
P7	2011/04/09	1358	11.12	15.09
P8	2011/04/11	1233	11.56	15.07
P9	2011/04/12	1054	11.54	15.15
P10	2011/04/20	1033	10.43	15.21

Table 5 Workload characteristics

performance degradation due to virtual machine migration *PDM*, and is calculated by the following equation:

$$SLA_{violation} = SLA_{TAH} \times PDM$$
 (21)

Here, *SLA*<sub>TAH</sub> indicates the SLA violation time per host, calculated according to the following equation:

$$SLA_{TAH} = \frac{1}{N} \sum_{i=1}^{N} \frac{t_{max}^{i}}{t_{active}^{i}}$$
(22)

where *N* is the total number of hosts,  $t_{max}^i$  is the total time for *Host<sub>i</sub>* to reach full load state, and  $t_{active}^i$  is the total active time of *Host<sub>i</sub>*. In addition, the performance overhead *PDM* generated by VM migration is defined as:

$$PDM = \frac{1}{M} \sum_{j=1}^{M} \frac{pdm_j}{C_{demand_j}}$$
(23)

Among them, M is the total number of VMs,  $pdm_j$  is the performance degradation due to the dynamic migration of  $VM_j$ , which is set to 10% in this experiment with reference to the suggestion of literature [43]. *demand*<sub>j</sub> indicates the total amount of CPU resources requested by  $VM_j$  during its lifetime.

(4) Number of VM Migrations: it indicates the total number of migrations per scheduling cycle throughout the runtime. In enterprise-class data centers, VM migrations take a certain amount of time, consume a lot of resources, and lead to system performance degradation, so it is necessary to reduce the number of data center migrations.

(5) Number of Hosts Shut Down: the VMC method runs more workloads in a small number of physical

machines to shut down underloaded hosts. The virtual machine scheduling method should bring more low-load hosts off to reduce energy consumption.

#### **Results analysis**

For the evaluation of the combined TU-MMR-HOAVG policy, the IQR, MAD, and THR overload server detection policies, MC and MU virtual machine selection policies, and greedy underload host policy proposed in the literature [6] are selected in the paper and combined into six different VMC methods for comparison experiments. We select three days of data in load as the test (2011/03/06, 2011/04/11, 2011/04/03). The data from these days are chosen because they can represent different data center cluster sizes respectively. The VM allocation policies used are all PABFD algorithms as a way to ensure the consistency of the placement algorithm.

The ultimate goal of the approach in this paper is to make the total energy consumption of the data center the least in order to achieve a balanced state of the computing and cooling systems. As shown in Fig. 4, the combined TU-MMR-HOAVG strategy has a clear advantage in energy saving, and the TU-MMR-HOAVG strategy is always the best in terms of energy consumption, regardless of the cluster size. Compared with other methods, TU-MMR-HOAVG reduces the total data center energy consumption by 7.87% on average, 2.73% on average compared to the best method MAD-MC-GREEDY, and 14.61% on average compared to the worst method THR-MU-GREEDY.

As seen in Fig. 5(a), TU-MMR-HOAVG is able to shut down more underloaded hosts compared to other methods. HOAVG selects underloaded hosts based on the median average utilization and shuts them down, which avoids the overload problem caused by the greedy server



Fig. 4 Energy consumption comparison of TU-MMR-HOAVG combination strategy

shutdown policy that over shuts down servers. Among all approaches, the combined TU-MMR-HOAVG policy consistently produces the lowest number of VM migrations for different sizes of workload data (Fig. 5(b)), which is due to the fact that the MMR policy selects VMs with the largest memory to migrate out. In other words, the MMR policy tends to select one VM with large requested resources rather than several VMs with small requested resources, reducing unnecessary migration activities in the data center by this way.

Another objective of the combined TU-MMR-HOAVG strategy in this paper is to avoid hot spots in the data center. For this purpose, we counted the number of hot spots generated by various policies. According to Fig. 5(c), the TU-MMR-HOAVG approach has no hot spots and ensures that the temperatures of all hosts are within the safety threshold, avoiding the thermal potential danger in the data center. All other policies in the platform have multiple hot spots due to high load on some hosts caused by considering only host utilization without considering temperature. On the other hand, while the TU-MMR-HOAVG policy has a higher SLA violation rate than the other policies (Fig. 5(d)), all other policies have higher energy consumption values than TU-MMR-HOAVG, and they generate more hot spots and thermal violations. The combination of TU-MMR-HOAVG policy results in the lowest energy consumption and avoids hot spots in the data center. Overall, the combined TU-MMR-HOAVG strategy can keep the SLA violation rate below 0.005%, which can guarantee the quality of service for users.

In terms of VM placement algorithms, to verify the effectiveness of UACO algorithm, we compare UACO with the six other common VM placement algorithms (FFD, MBFD, TAS, GRANITE [44], ACS\_VMC [45], EVMCACS [46]). Among them, FFD, MBFD are heuristic algorithms which place VMs based on greedy heuristic policy. TAS is a thermal-aware scheduling policy which selects hosts with low temperature as the target host. GRANITE method predicts the temperature of future hosts and moves VMs out of the hosts which are most likely to exceed the temperature threshold and selects target hosts based on greedy policy. ACS\_VMC and EVMCACS are both intelligent algorithms based on ant colony algorithm. These swarm intelligence algorithms contain more parameters, and the settings of the algorithm parameters are shown in Table 6.

Figure 6 shows the energy consumption trend of each VMP algorithm under different workloads. It can be seen that as the number of virtual machines increases, the search space of the solution increases, and the energy consumption of all algorithms increases. In all load instances, the UACO algorithm excels in energy consumption. In general, the swarm intelligence algorithm is generally more energy-efficient than the traditional heuristic VMP algorithms (FFD, MBFD), and the thermalaware scheduling method is not much different from the traditional heuristic algorithm. Compared with the worst FFD algorithm, UACO reduces energy consumption by an average of 28.93%, compared with the best EVMCACS algorithm by an average of 16.53%, and compared with all other virtual machine placement methods by an average of 24.05%. This is due to UACO's preference for host



(c) Comparison of the number of hotspots Fig. 5 Comparison of different VMC strategies under different loads



(d) SLA violation rate comparison

	Table 6	Algorithm	parameter	settings
--	---------	-----------	-----------	----------

Algorithm	UACO	ACS_VMC	EVMCACS
Number of ants:N <sub>ant</sub>	15	15	15
Number of iterations: $N_c$	10	10	10
Critical number:c0	0.7	0.7	0.7
Local pheromone volatile factor: $ ho$	0.3	0.3	-
Global pheromone volatile factor: $\sigma$	0.4	0.4	0.4
Pheromone importance factor: $\alpha$	0.9	-	-
Importance factor of heuristic information: $meta$	0.9	0.9	0.9
The maximum proportion of increase in pheromone concentration: $\Delta  au_{max}$	1.5	-	-
Weight of power consumption in fitness function $\!x$	0.5	-	-
Weight of the number of hosts closed in fitness function: $\gamma$	-	5	5



Fig. 6 Comparison of the total energy consumption of the data center under different loads

physical machines with balanced energy and temperature after migration, and has improved the method of updating pheromones, which allows the algorithm to find the global optimal solution. In addition, UACO is able to keep the SLA violation rate within 0.002%, which is a relative balance between energy consumption and QoS compared to other methods.

In terms of the number of virtual machine migrations, as shown in Fig. 7, UACO's performance is not very outstanding, inferior to the traditional heuristic algorithm, but better than other swarm intelligence algorithms. It is because the algorithm has blindness in searching at the primary stage, which makes the VMs migrate frequently in the preliminary stage, while at the later stage, due to the improved pheromone update method of ACO, the algorithm can eventually jump out of the local optimal solution and search for the global optimal solution of energy consumption.



Fig. 7 Comparison of virtual machine migration counts under different loads

the FFD and MBFD greedily choose the hosts with less incremental energy consumption, which avoids unnecessary migration of data centers. the TAS algorithm aims to reduce data center hot spots while ignoring the energy consumption problem, the frequent migration of virtual machines leads to an increase in energy consumption.

To further validate the performance of the UACO algorithm, the experiments counted the number of active hosts per hour of various placement algorithms under the PlanetLab load data (2011/03/03), as shown in Fig. 8 below, the number of active hosts in the cluster is basically maintained between 20-40 under the UACO algorithm, which is smaller than any other placement algorithms. GRANITE and FFD algorithms exhibit a high number of active hosts, around 80. FFD and MBFD tend to place VMs on the optimal server under the heuristic rule in the VM placement process. However, when the number of VMs increases, the order of VM placement affects the resource status of the server and the determination of the heuristic rule, which tends to make the algorithm fall into a local optimum, thus making it difficult to further reduce the number of active servers. ACS\_VMC can keep fewer active hosts at the beginning of the algorithm, but as time increases, the ants select hosts with sufficient resource capacity when determining the target hosts, which may start the dormant hosts, resulting in more active hosts than UACO. The UACO algorithm tends to assign the newly arrived workload to a host that is not idle and not overloaded after migration, and try not to start a new host, which reduces the number of active servers.

To deeply analyze the efficiency of UACO, we further compared the execution time of seven VM placement algorithms. Table 7 and Fig. 9 below show the performance of the various algorithms in terms of execution time. Since the swarm intelligence algorithm requires multiple iterations to update the population and has multiple individuals to traverse, the UACO, ACS\_VMC, and EVMCACS algorithms are higher than the other algorithms in terms of time overhead. The thermal-aware scheduling algorithm needs to iterate over the host temperature and sort the temperature, which increases the running time of the algorithm. According to Fig. 9, it can be seen that the improved ant colony algorithm (UACO) proposed in this paper has a slightly lower time overhead than the other two swarm intelligence algorithms, which is due to the improved pheromone update method of UACO, making the algorithm have certain directionality in the later solving process, thus accelerating the algorithm solving speed.

#### **Conclusion and future work**

In this paper, we focus on the optimization problem of data center resource scheduling, from power consumption modeling to virtual machine consolidation method, and propose a resource scheduling method based on thermal management (TM-VMC). This method includes a host overload detection method based on CPU temperature and utilization (TU), an underload host detection method based on median average utilization (HOAVG), a virtual machine selection policy based on maximum migration memory (MMR), and a virtual machine placement policy based on an improved ant colony algorithm



Fig. 8 Comparison of the number of active hosts per hour under the PlanetLab load data (2011/03/03) for the VMP algorithms

Load	UACO	TAS	GRANITE	FFD	MBFD	ACS_VMC	EVMCACS
P1	774.12	348.19	25.86	24.27	22.65	936.58	799.90
P2	645.59	251.18	21.76	21.45	21.46	725.51	769.91
P3	822.37	284.53	28.59	25.14	24.16	859.57	852.55
P4	2502.77	610.95	30.92	33.78	27.07	2476.84	2275.82
P5	915.24	334.36	29.92	26.20	22.53	926.88	958.76
P6	1345.73	937.59	39.30	34.58	24.56	1538.69	1565.53
P7	2099.84	715.92	37.92	38.85	27.65	1940.80	1987.04
P8	2060.73	874.82	44.41	34.35	29.98	2039.21	2110.68
P9	1229.14	745.14	28.81	32.50	23.72	1112.72	1164.21
P10	997.84	746.38	27.73	31.89	36.76	1013.06	1006.03

Table 7 Comparison of execution time (ms) of VMP algorithms



Fig. 9 Comparison of average execution time of VMP algorithms

(UACO). TU determines whether a host is overloaded by detecting host utilization and temperature status in real-time. It proactively relocates a few VMs off the overloaded hosts to avoid the data center hot spots. The MMR policy migrates VMs having the most memory to minimize the number of VMs selected for migration in the data center. HOAVG detects underloaded hosts based on server CPU utilization and shuts them down, which reduces the number of active servers. UACO places VMs on hosts with a balanced temperature and energy consumption using a fitness function based on the temperature and energy consumption after placement. Extensive performance studies show that TM-VMC can avoid data center hot spots and minimize the data center energy consumption while meeting SLAs.

In the future, we plan to build more refined temperature models to quantify the impact of temperature on data center energy consumption and collect relevant data from the real world through physical inspection equipment to better fit the model parameters to obtain more accurate temperature models that can better guide the resource scheduling experiments. In addition, the proposed method should be further validated and tested in an existing data center environment.

#### Authors' contributions

Li Mao: Methodology, Software, Writing—original draft. Rui Chen: Investigation, Writing—review & editing. Huiwen Cheng: Writing—review & editing. Weiwei Lin: Methodology, Writing—review & editing, Funding acquisition. Bo Liu: Supervision, Writing—review & editing. James Z. Wang: Writing—review & editing. The author(s) read and approved the final manuscript.

#### Funding

This work is supported by National Natural Science Foundation of China (62072187), Guangdong Marine Economic Development Special Fund Project (GDNRC[2022]17), Guangdong Major Project of Basic and Applied Basic Research (2019B030302002), Characteristic Innovation Projects of Ordinary Colleges and Universities in Guangdong Province (2020KTSCX087), Guangdong Science and Technology Plan Project (2018KJYZ009). James Z. Wang's work was supported (in part) by the National Institutes of Health (NIH)/Eunice Kennedy Shriver National Institute of Child Health and Human Development (R01 HD 069374) and the National Science Foundation (NSF) DBI (1759856).

#### Availability of data and materials

The workload data used in this work is based on the CoMon project [42].

#### Declarations

**Ethics approval and consent to participate** Not applicable.

#### **Competing interests**

There are no competing interests.

Received: 12 February 2023 Accepted: 23 May 2023 Published online: 10 June 2023

#### References

- 1. Cheng H, Liu B, Lin W, Ma Z, Li K, Hsu C (2021) A survey of energy-saving technologies in cloud data centers. J Supercomput 77(11):13385–13420
- China Academy of Information and Communications (2022) Data center white paper. http://www.ctiforum.com/uploadfile/2022/0428/20220 428104230327.pdf. Accessed 11 Feb 2023.
- Fernández-Cerero D, Fernández-Montes A, Jakóbik A (2020) Limiting global warming by improving data-centre software. IEEE ACCESS 8:44048–44062
- Koomey JG (2011) Growth in data center electricity use 2005 to 2010. https://alejandrobarros.com/wp-content/uploads/old/Growth\_in\_Data\_ Center\_Electricity\_use\_2005\_to\_2010.pdf. Accessed 11 Feb 2023.
- Ding W, Luo F, Han L, Gu C, Lu H, Fuentes J (2020) Adaptive virtual machine consolidation framework based on performance-to-power ratio in cloud data centers. Futur Gener Comput Syst 111:254–270
- 6. Beloglazov A, Buyya R (2012) Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers. Concurrency and Computation: Practice and Experience 24(13):1397–1420
- Horri A, Mozafari MS, Dastghaibyfard G (2014) Novel resource allocation algorithms to performance and energy efficiency in cloud computing. J Supercomput 69:1445–1461
- Kansal NJ, Chana I (2016) Energy-aware virtual machine migration for cloud computing-a firefly optimization approach. Journal of Grid Computing 14:327–345
- Ibrahim A, Noshy M, Ali HA, Badawy M (2020) PAPSO: a power-aware VM placement technique based on particle swarm optimization. IEEE ACCESS 8:81747–81764
- Ilager S, Ramamohanarao K, Buyya R (2019) ETAS: Energy and thermal aware dynamic virtual machine consolidation in cloud data center with proactive hotspot mitigation. Concurrency and Computation: Practice and Experience 31(17):e5221
- 11. Feng H, Deng Y, Li J (2021) A global-energy-aware virtual machine placement strategy for cloud data centers. J Syst Architect 116:102048
- Arroba P, Risco-Martín JL, Moya JM, Ayala JL (2018) Heuristics and metaheuristics for dynamic management of computing and cooling energy in cloud data centers. Software: Practice and Experience 48(10):1775–1804
- Feng H, Deng Y, Zhou Y, Min G (2022) Towards heat-recirculation-aware virtual machine placement in data centers. IEEE Trans Netw Serv Manage 19(1):256–270
- Van Damme T, De Persis C, Tesi P (2018) Optimized thermal-aware job scheduling and control of data centers. IEEE Trans Control Syst Technol 27(2):760–771
- Hsieh S, Liu C, Buyya R, Zomaya AY (2020) Utilization-prediction-aware virtual machine consolidation approach for energy-efficient cloud data centers. J PARALLEL DISTR COM 139:99–109
- Saadi Y, El Kafhali S (2020) Energy-efficient strategy for virtual machine consolidation in cloud environment. SOFT COMPUT 24(19):14845–14859
- 17. Wang H, Tianfield H (2018) Energy-aware dynamic virtual machine consolidation for cloud datacenters. IEEE Access 6:15259–15273
- Wang J, Gu H, Yu J, Song Y, He X, Song Y (2022) Research on virtual machine consolidation strategy based on combined prediction and

energy-aware in cloud computing platform. Journal of Cloud Computing 11(1):1–18  $\,$ 

- Coffman EG, Csirik J, Woeginger GJ (2002) Approximate solutions to bin packing problems. Handbook of applied optimization, 1st edn. Oxford University Press, New York, pp 607–615
- 20. Shaw R, Howley E, Barrett E (2022) Applying reinforcement learning towards automating energy efficient virtual machine consolidation in cloud data centers. Inf Syst 107:101722
- Zeng J, Ding D, Kang K, Xie H, Yin Q (2022) Adaptive DRL-based virtual machine consolidation in energy-efficient cloud data center. IEEE Trans Parallel Distrib Syst 33(11):2991–3002
- Rezakhani M, Sarrafzadeh-Ghadimi N, Entezari-Maleki R, Sousa L, Movaghar A (2023) Energy-aware QoS-based dynamic virtual machine consolidation approach based on RL and ANN. Cluster Computing 1–17.
- Aghasi A, Jamshidi K, Bohlooli A, Javadi B (2023) A decentralized adaptation of model-free Q-learning for thermal-aware energy-efficient virtual machine placement in cloud data centers. Comput Netw 224:109624
- Pourghebleh B, Aghaei Anvigh A, Ramtin AR, Mohammadi B (2021) The importance of nature-inspired meta-heuristic algorithms for solving virtual machine consolidation problem in cloud environments. Clust Comput 24(3):2673–2696
- Jiang Y, Wang J, Shi J, Zhu J, Teng L (2020) Network-aware virtual machine migration based on gene aggregation genetic algorithm. Mobile Networks and Applications 25:1457–1468
- Liu F, Ma Z, Wang B, Lin W (2020) A virtual machine consolidation algorithm based on ant colony system and extreme learning machine for cloud data center. IEEE ACCESS 8:53–67
- 27. Singh AK, Swain SR, Saxena D, Lee CN (2023) A bio-inspired virtual machine placement toward sustainable cloud resource management. IEEE Systems Journal.
- Saridou B, Bendiab G, Shiaeles SN, Papadopoulos BK, Thampi SM, Wang G, Rawat DB, Ko R, Fan CI (2021) Security in Computing and Communications 8th International Symposium SSCC 2020 Chennai India October 14–17 2020 Revised Selected Papers Thermal Management in Large Data Centres: Security Threats and Mitigation Springer Singapore Singapore pp. 165–179
- MirhoseiniNejad S, Moazamigoodarzi H, Badawy G, Down DG (2020) Joint data center cooling and workload management: a thermal-aware approach. Futur Gener Comput Syst 104:174–186
- Cheung H, Wang S, Zhuang C, Gu J (2018) A simplified power consumption model of information technology (IT) equipment in data centers for energy system real-time dynamic simulation. APPL ENERG 222:329–342
- Horvath T, Skadron K (2008) Multi-mode energy management for multi-tier server clusters. In: 2008 International Conference on Parallel Architectures and Compilation Techniques (PACT), Toronto, ON, Canada, 25–29 October 2008.
- 32. Gandhi A, Harchol-Balter M, Adan I (2010) Server farms with setup costs. PERFORM EVALUATION 67:1123–1138
- SPEC (2022) SPECpower\_ssj2008. http://www.spec.org/power\_ssj2008/ results/power\_ssj2008.html. Accessed 11 Feb 2023.
- Zhang Q, Meng Z, Hong X, Zhan Y, Liu J, Dong J, Bai T, Niu J, Deen MJ (2021) A survey on data center cooling systems: Technology, power consumption modeling and control strategy optimization. J SYST ARCHITECT 119:102253
- Zhan X, Reda S (2013) Techniques for energy-efficient power budgeting in data centers. In: 2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, TX, 29 May 2013.
- Tang Q, Mukherjee T, Gupta SKS, Cayton P (2006) Sensor-Based Fast Thermal Evaluation Model for Energy Efficient High-Performance Datacenters. In: 2006 Fourth International Conference on Intelligent Sensing and Information Processing, Bangalore, India, 15 October 2006.
- Harris AD (2018) The American Society of Heating, Refrigerating and Air-Conditioning Engineers. http://tc0909.ashraetcs.org/2018. Accessed 11 Feb 2023.
- Zhang S, Chatha KS (2007) Approximation algorithm for the temperatureaware scheduling problem. In: 2007 IEEE/ACM International Conference on Computer-Aided Design, San Jose, CA, USA, 04–08 November 2007.
- Dorigo M, Maniezzo V (1996) Ant system: optimization by a colony of cooperating agents. IEEE Trans on SMC-Part B 26(1):29–41
- Karmakar K, Das RK, Khatua S (2022) An ACO-based multi-objective optimization for cooperating VM placement in cloud data center. J Supercomput 78:3093–3121

- Calheiros RN, Ranjan R, Beloglazov A, De Rose CAF, Buyya R (2011) Cloud-Sim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience 41(1):23–50
- 42. Park KS, Pai VS (2006) CoMon: a mostly-scalable monitoring system for PlanetLab. ACM SIGOPS Operating Systems Review 40(1):65–74
- Beloglazov A, Abawajy J, Buyya R (2012) Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Futur Gener Comput Syst 28(5):755–768
- Li X, Garraghan P, Jiang X, Wu Z, Xu J (2017) Holistic virtual machine scheduling in cloud datacenters towards minimizing total energy. IEEE Trans Parallel Distrib Syst 29(6):1317–1331
- 45. Farahnakian F, Ashraf A, Pahikkala T, Liljeberg P, Plosila T, Porres I, Tenhunen H (2015) Using ant colony system to consolidate VMs for green cloud computing. IEEE T SERV COMPUT 8(2):187–198
- Aryania A, Aghdasi HS, Khanli LM (2018) Energy-aware virtual machine consolidation algorithm based on ant colony system. Journal of Grid Computing 16:477–491

#### **Publisher's Note**

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

# Submit your manuscript to a SpringerOpen<sup>™</sup> journal and benefit from:

- Convenient online submission
- ► Rigorous peer review
- Open access: articles freely available online
- ► High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com