Future Generation Computer Systems 145 (2023) 578–589

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Power and thermal-aware virtual machine scheduling optimization in cloud data center

Rui Chen [a], Bo Liu [a], WeiWei Lin [b,c,*], JianPeng Lin [b], HuiWen Cheng [a], KeQin Li [d]

[a] School of Computer Science, South China Normal University, Guangzhou, 510631, China
[b] School of Computer Science and Engineering, South China University of Technology, Guangzhou, 510006, China
[c] Peng Cheng Laboratory, Shenzhen, 518055, China
[d] Department of Computer Science, State University of New York, New Paltz, NY 12561, USA

ARTICLE INFO

Article history:
Received 15 April 2022
Received in revised form 9 February 2023
Accepted 31 March 2023
Available online 5 April 2023

Keywords:
Cloud computing
Data center cooling
VM consolidation
Ant colony optimization
Energy efficiency in a data center

ABSTRACT

Virtual machine (VM) consolidation technology is a commonly used energy-saving method in data centers. Most of the existing methods are committed to consolidating VMs to a small number of servers to improve the utilization of server resources and reduce the total energy consumption while preventing hotspots. However, energy and thermal-aware VM scheduling is a multi-objective optimization problem. Most of the existing related work cannot provide adequate means to adjust the impact of energy consumption and temperature on VM scheduling. Hence, this paper proposes a power and thermal-aware VM dynamic scheduling scheme (PTDS) for cloud data centers. The proposed PTDS dynamically adjusts the VM consolidation scheme by real-time detecting the server temperature and resource utilization rate under the premise of considering the thermal cycle effect of the data center computer room. Power and Thermal Objective Ant Colony Optimization (PTOACO) is proposed in the VM placement. PTOACO improves the defect that the ant colony algorithm easily falls into local optimization and adds control parameters to adjust the bias between sub-objectives. We performed extensive experiments by using real PlanetLab and random workloads. The performance results were compared with several advanced schemes regarding total energy consumption, hotspots, SLA violation rate, etc. The experimental results demonstrate that PTDS reduces energy consumption by 26.69% on average compared with other advanced schemes and ensures a meager SLA violation rate while avoiding hotspots.

1. Introduction

In recent years, the vast energy consumption generated by cloud data centers has attracted wide attention from society. Relevant survey data [1] show that electricity demand in US data centers has increased from 29 billion kWh in 2000 to nearly 73 billion kWh in 2020. The computing device will also increase the thermal load while using electric energy. The thermal load of a single rack will reach 50 kW in 2025 [2]. In addition, it also leads to the decline of quality of service (QoS) and environmental deterioration [3,4], which runs counter to the concept of green sustainable development. Therefore, achieving energy conservation and emission reduction is the key to developing the next generation of green data centers.

As shown in Fig. 1, information and communications technology equipment (ICT) and cooling equipment consume most of the energy in the data centers. The server is the most critical ICT equipment. In most cases, the average CPU utilization rate of the server is only 15% to 20%, and there are a large number of servers in an idle state which consumes much energy. Therefore, VMs can be relocated into as few physical machines (PM) as possible through VM consolidation technology, and the low-load PM can be switched off. No-load PM can be switched to a sleep state to reduce the number of PMs during activities and avoid energy waste [5]. For example, Hsieh et al. [6] solved the dynamic scheduling problem of VM by predicting CPU utilization based on the gray Markov model; similarly, Karmakar K et al. [7] expressed the VM placement problem as a multi-objective optimization problem and proposed a VM consolidation algorithm based on the ant colony algorithm. The above works are committed to improving the utilization of server resources and focusing the workload on a few servers to reduce the overhead of computing equipment. However, on the one hand, this strategy is easy to causes the active server to be running under high load for a long time, resulting in local hotspots. On the other hand, the increase in local temperature will trigger the cooling system to produce more cold air to avoid thermal risk [8], resulting in increased operating costs of the cooling system while reducing the server's

* Corresponding author at: School of Computer Science and Engineering, South China University of Technology, Guangzhou, 510006, China.
E-mail address: linww@scut.edu.cn (W. Lin).

https://doi.org/10.1016/j.future.2023.03.049
0167-739X/

reliability. In addition, upgrading computing equipment hardware and a good computer room layout can improve the thermal performance of the data center [9] so that the cloud data center can work at a higher ambient temperature. Therefore, reducing energy consumption and avoiding thermal risk are two conflicting objectives. In virtual machine consolidation, it is necessary to consider the operating costs of computing equipment and cooling equipment. It should provide cloud service providers with the right to adjust the priority relationship between reducing energy consumption and thermal risk.

In this paper, an energy and thermal-aware dynamic VM scheduling scheme, namely PTDS, is proposed. This scheme focuses on the real-time detection of server temperature and resource utilization under the heat recirculation effect to dynamically adjust the VM consolidation. PTDS can adjust the priority relationship between the two sub-objectives of energy consumption and temperature to minimize the data center's computing and cooling energy consumption while reducing the thermal risk. We use the low complexity heat recirculation model proposed by Tang et al. [10]. The model calculates the device's power based on the data center's thermodynamic and physical characteristics and accurately describes the linear power model related to CPU utilization and the linear model of the server inlet temperature predicted under a given server utilization. We analyze the proposed algorithm with extensive simulation-based experiments using CloudSim [11] with real PlanetLab [12] and random workloads. Experiments show that the proposed scheme reduces the overall energy consumption of the data center and the SLA violation rate while preventing hotspots.

In brief, the major contributions of this paper can be summarized as follows:

•The paper proposes a VM scheduling scheme PTDS for the cloud data center to reduce the overall computing and cooling energy and proactively prevent hotspots.

•A novel host overload detection algorithm, called Average Median Deviation (AMD), determines the upper limit threshold of CPU utilization according to the average dispersion degree of CPU utilization history of the server.

•A novel VM placement algorithm based on an improved ant colony algorithm, called PTOACO, modifies the evaluation criteria of the solution of the ant colony algorithm and modifies the update rules of the ant colony algorithm to avoid premature falling into a locally optimal solution. Meanwhile, the influence of energy and temperature on the solution is adjusted by preset parameters.

•We implement the proposed scheme and validate its efficiency with extensive experiments using real workloads through simulation and demonstrate its superiority by comparing it to the several baseline schemes.

The rest of the paper is organized as follows: Section 2 provides a literature review of the proposed VM consolidation. Section 3 describes the scheme content and implementation details. Section 4 shows the experimental results. Finally, conclusions are made in Section 5.

2. Related work

A large number of researches are being carried out in cloud computing to reduce the operating cost of cloud data centers. The VM deployment and consolidation research can be divided into many aspects, and each aspect has a different focus. In recent years, considerable attention has been paid to the energy cost of the cooling system. How to reduce the impact of temperature by improving the scheduling of VMs has become a research hotspot. In this section, previous literature is discussed.

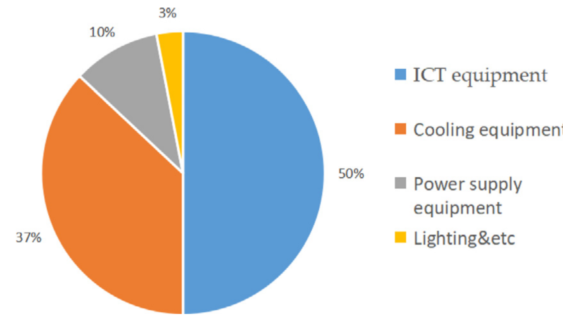


Fig. 1. The power consumption breakdown of the entire data center and cooling system [5].

Reducing the energy consumption of computing equipment from the hardware is a typical energy-saving method. For example, Intel has developed a dynamic voltage and frequency scaling (DVFS) technology for energy saving at the chip level [13], which can adaptively change the processor's frequency through the utilization rate of the processor. However, the disadvantage of this technology is that it can only be applied to a single node. Chen et al. [14] formally described the dynamic optimization problem of server configuration and DVFS control, including the response time SLA and the cost of server closure. In addition, it also includes GreenSwitch by detecting hardware workload dynamic management battery energy-saving method [15]. The above research provides an idea of energy-saving from the perspective of hardware and has achieved good results, but it does not consider the use of virtualization technology. The development of hardware virtualization technology makes VM consolidation, load balancing and other technologies widely used in data center saving [16].

In addition, some VM consolidation schemes focus on balancing the relationship between reducing energy consumption and maintaining performance. For example, Beloglazov et al. [17] consolidated the VM into as few hosts as possible according to the CPU utilization rate and used the improved best fit decreasing algorithm PABFD to find the appropriate target host for the VM. Luo et al. [18] proposed a network-aware VM rescheduling algorithm by combining network requirements. Zhao et al. [19] proposed an energy-saving scheduling technology based on model predictive control by combining DVFS technology. Gaggero et al. [20] proposed a predictive control model, where VM migration is transformed into an optimal control problem in a finite range. Kansal et al. [21] first proposed to apply the firefly algorithm to energy-aware data center virtual machine real-time migration. Li et al. [22] designed a dual-threshold method using multiple resources to trigger the migration of VMs, and proposed an algorithm MPSO based on the improved particle swarm optimization method. Li L et al. [23] also designed a VM scheduling method based on SLA and energy consumption perception by improving the HS3MC model. Ding et al. [24] proposed a VM consolidation framework based on resource utilization and PPR of the heterogeneous host. The above literature saves the data center's computing system overhead by optimizing the migration of VMs. However, they do not consider the cooling system overhead and potential thermal risks of cloud data centers.

Heat-aware resource management technology has lately received significant attention for research due to the increasing energy consumption of cooling systems in data centers. Literature [25] pointed out that the unreasonable design of the cooling system in the data center will lead to the heat recirculation phenomena, which is an alarming phenomenon that the cooling efficiency is reduced due to the mixing of old and hot air.

The following literature proposed different management schemes considering the cost of cloud data center computing and cooling system. Sun et al. [26] introduced the thermal-aware load and proposed an online scheduling heuristic algorithm for task scheduling and thermal management. Wang L et al. [27] proposed the thermal-aware task scheduling algorithm TASA to reduce temperatures and cooling system power consumption. Li Xiang et al. [28] used CFD to design a cooling model that considers the thermal characteristics of CRAC, air and servers in the data center, and designed a VM placement and migration algorithm GRANITE to minimize computing and cooling resources. However, they did not consider managing underloaded hosts. In addition, Lee et al. [29] introduced the heat imbalance model and proposed a thermal-aware VM allocation scheme but did not consider the effect of heat recirculation. Ilager et al. [30] proposed energy consumption and heat-aware scheduling algorithm ETAS, a meta-heuristic algorithm based on GRASP, and can dynamically consolidate VMs in the case of active prevention of hotspots. However, this work greedily selects the target host for each VM during the placement of VMs and does not reduce the total cost from global considerations. Similarly, Abbas Akbari et al. [31] provided a heuristic algorithm for thermal-aware VM allocation based on a genetic algorithm, but it belongs to an offline algorithm and cannot realize online thermal-aware VM allocation. Feng et al. [32] proposed a two-step algorithm to reduce the overhead of data centers from three aspects: cooling system, computing system and network. A simulated annealing algorithm was used to minimize the computational and cooling overhead in the first step. In the second step, VMs with high traffic costs were placed on servers close to the location to reduce network overhead. However, the algorithm did not fully consider how to define the threshold of server utilization and the VM selection policy. Aghasi A et al. [33] proposed a BGSA-based VM placement algorithm to minimize computational and cooling overhead and designed an adaptive fuzzy mechanism to enhance the algorithm. M. Hasan Jamal et al. [34] proposed hotspot adaptive workload deployment algorithm and hotspot aware server relocation algorithm based on thermal profiling regarding outlet temperature prediction. However, it is hard to move a large number of hosts in the real data center environment. Because of the uncertainty of computing resources and the complexity of temperature changes, the computing system and cooling system should be considered together. Unfortunately, the above work cannot provide cloud service providers with the right to prioritize reducing operating costs or avoiding thermal risks. Therefore, this paper focuses on the dynamic adjustment of the VM consolidation scheme based on server temperature and resource utilization under the heat recirculation effect and provides a solution for balancing the influence of energy consumption and temperature.

3. Scheme design and implementation

3.1. System overview

The system model of this paper is shown in Fig. 2. The system consists of cloud users, a scheduling system and infrastructure. Users submit tasks to the deployed VMs. The scheduling system is responsible for receiving user requests, allocating computing resources for requests, and monitoring the CPU utilization and temperature of the hosts. Infrastructure includes physical equipment such as computing equipment and cooling equipment.

The VM consolidation scheme includes host load detection, VM selection, and VM placement.

(1) In the host load detection phase, the host overload detection algorithm is proposed by setting a fixed temperature threshold for the host and using the proposed AMD algorithm as the utilization threshold.

(2) The VM selection phase selects the VM to be migrated from the overload through the minimization algorithm (MM) [17].

(3) In the VM placement phase, the PTOACO algorithm is used to allocate the target host for the VM to be migrated under the consideration of energy consumption and temperature constraints.

In the process of VM consolidation, it is committed to reducing the running cost of computing equipment while maintaining the temperature detection of the host to avoid the loss of the host caused by high temperature, which will cause the cooling system to consume much extra energy to stabilize the temperature of the computer room.

The system model consists of three sub-models: (1) computing system power model, which describes the linear relationship between the power consumption of the host and the time change. (2) cooling system power model analyzes the use of cooling energy. (3) server temperature model uses CPU temperature to obtain the relationship between server utilization, CRAC cooling capacity, and thermal characteristics.

3.2. Energy consumption and temperature model

The models in the system are described below.

3.2.1. Computing system power model

A Cloud data center comprises heterogeneous servers with different physical capacities, power and processing capabilities. The power consumption of the host is mainly determined by its CPU utilization. This paper uses the following energy consumption model to represent the power consumption of a single active host [35]:

$$P_i(t) = \begin{cases} P_i^{idle} + \sum_{j=1}^{V_i} U(VM_{i,k}(t)) \times P_i^{dynamic} & (V_i > 0) \\ 0 & (V_i = 0) \end{cases} \tag{1}$$

In Eq. (1), the $\mathrm{P_i(t)}$ is the energy consumption of $\mathrm{h_i}$ at time t. The P_i^{idle} is the power of $\mathrm{h_i}$ without load; $P_i^{dynamic}$ is the dynamic power of $\mathrm{h_i}$. $U(VM_{i,k}(t))$ represents the utilization rate of computing resources when VM_k runs in $\mathrm{h_i}$ at time t; V_i is the total number of VMs running in $\mathrm{h_i}$; The idle host ($V_i = 0$) should be switched off to save unnecessary energy consumption.

3.2.2. Cooling system power model

CRAC is the central cooling equipment in cloud data centers, which takes up most cooling energy overhead [5]. The efficiency of CRAC is usually measured by calculating the energy consumption ratio between the computing system and cooling system, also known as Coefficient of Performance (CoP). So we use CoP to determine the power consumption model of the cooling system [36]:

$$CoP(T_{sup}) = \frac{P_{IT}}{P_{cooling}} \tag{2}$$

In Eq. (2), the $P_{cooling}$ is the total energy consumption of the cooling system; P_{IT} is the total energy consumption of the computing system; T_{sup} is the CRAC cold air supply temperature.

The higher value of CoP indicates higher cooling efficiency. CoP can be modeled using the regression techniques with multiple experiments using the different workloads and supply air temperature. Studies have shown a positive correlation between CoP and supply air temperature. In this paper, CoP measured by HP laboratory is used [36]:

$$CoP(T_{sup}) = 0.0068{T_{sup}}^2 + 0.0008T_{sup} + 0.458 \tag{3}$$

In Eq. (3), the T_{sup} is CRAC cold air supply temperature. Eq. (3) shows that increasing the supply air temperature can increase the cooling system's efficiency.

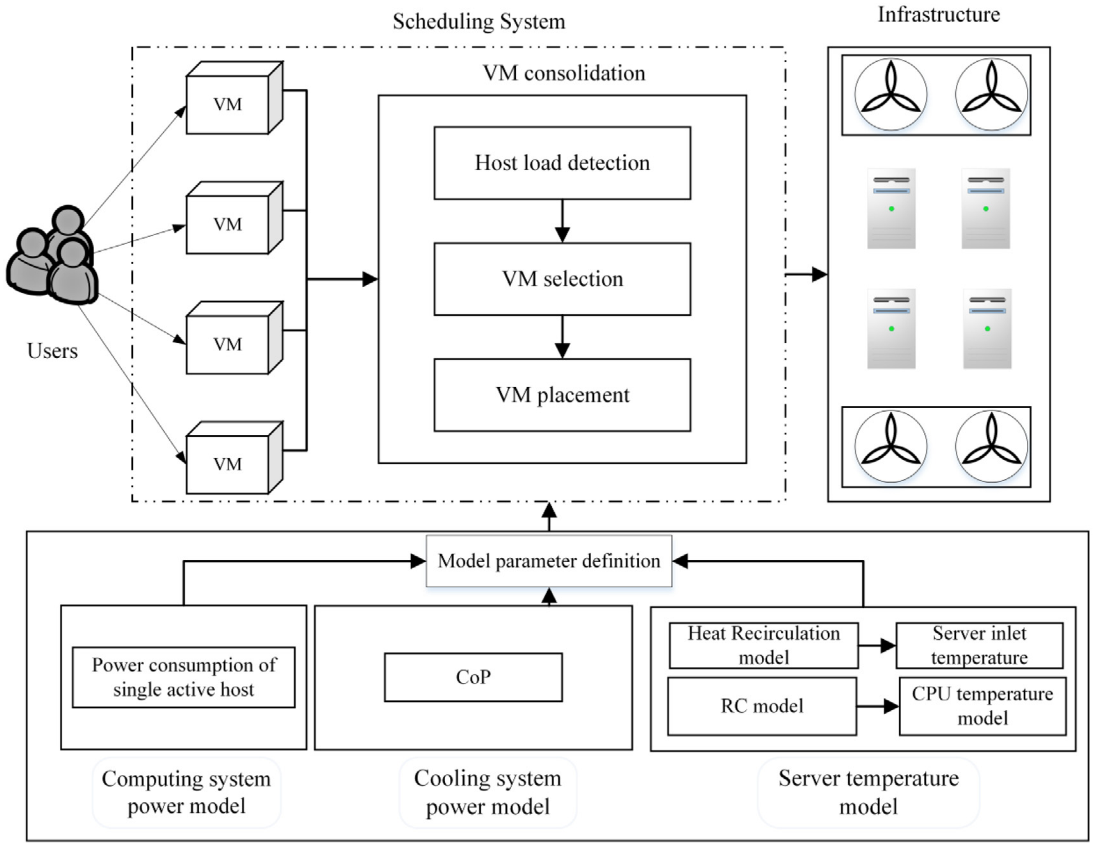


Fig. 2. System model.

3.2.3. Server temperature model

The inlet temperature of the host is mainly affected by the CRAC air supply temperature, its power consumption and the heat recirculation effect. The temperature model established by Tang et al. [10] is used, so the inlet temperature of the host can be directly defined as the following linear function:

$$T_i^{in}(t) = T_{sup} + \sum_{j=1}^{N} d_{i,j} \times P_j(t) \tag{4}$$

In Eq. (4), the $d_{i,j}$ represent the effect of heat recirculation to h_i from h_j, and it is the number of row j in i in the thermal distribution matrix D [10]; The $P_j(t)$ is power of h_j at time t. The T_{sup} is CRAC cold air supply temperature; The N is the number of hosts in the heat recirculation zone. Eq. (4) shows that the inlet temperature of the host is affected by its physical position and heat recirculation effect.

CPU temperature modeling is the most critical indicator of temperature modeling, and the RC model is one of the mature methods for calculating CPU temperature [30]. Its model can be expressed as :

$$T_i(t) = PR + T_i^{in} + \left(T_0 - PR - T_i^{in}\right) \times e^{-\frac{t}{RC}} \tag{5}$$

In Eq. (5), the $T_i(t)$ is the CPU temperature of the h_i at time t. R and C are the host's the thermal resistance and heat capacity, respectively; P is the energy consumption of the active host; T_0 is the CPU initial temperature obtained by Eq. (4). The RC model assumes that the power and inlet temperature of the CPU is stable, and $PR + T_i^{in}$ expresses the stable CPU temperature. It is also a function related to time t. With the increase of time, the CPU will continue to approach its stable temperature. Eq. (5) shows the dynamic temperature change of a single host in the heat recirculation zone.

3.3. Problem description

The total energy consumption of cloud data centers is mainly composed of computing system energy consumption and cooling system energy consumption. The sum of energy consumption of all hosts is computing system energy consumption, which can be expressed as:

$$P_{IT} = \sum_{t=0}^{T} \sum_{p=1}^{N} x_j P_i \tag{6}$$

In Eq. (6), the P_i is the host's computational energy consumption, x_j is a binary variable, when h_j is active from time 0 to time t, its value is 1, otherwise for 0. The N is the number of hosts in the heat recirculation zone and T is total scheduling interval. The timely shutdown of low-load hosts can save unnecessary overhead, so the key is how to adjust the allocation of host workload.

Energy consumption of the cooling system is defined as the ratio of calculated energy consumption to CoP:

$$P_{cooling} = \frac{P_{IT}}{CoP(T_{sup})}, \tag{7}$$

The mentioned Eq. (7) indicates that in order to provide colder airflow. The cooling system needs to consume more energy to remove the heat brought by the host calculation.

Total energy consumption is expressed by Eq. (8):

$$P_{total} = P_{IT} + P_{cooling} \tag{8}$$

The problem is expressed as the workload scheduling problem to minimize the total energy consumption. The total energy

Table 1
Definition of notations.

Notation	Definition	Notation	Definition
t	Time interval t	$d_{i,j}$	Effect of heat recirculation to h_i from h_j
h_i	The ith host	R	Thermal resistance of host
N	Number of hosts	C	Heat capacity of host
P_i	Power of h_i	U_{max}	Upper utilization threshold of host
$P_i^{dynamic}$	Dynamic power of h_i	T_{red}	Upper temperature threshold of host
P_i^{idle}	Idle power of h_i	$VM_{j,i}(R_{cpu}, R_{mem})$	CPU and memory capacity occupied by VM_j at h_i runtime
P_{IT}	Power of computing system	$h_x(R_{cpu}, R_{mem})$	CPU and memory capacity of $host_x$
$P_{cooling}$	Power of cooling system	s	Safety factor of AMD algorithm
P_{total}	Power of total data center	AMD	Value of average median deviation
V_i	Number of VMs in h_i	$value$	Evaluation standard of PTOACO algorithm
$U(VM_{i,k}(t))$	Efficiency of VM_k running in h_i at time interval t	α	Adjustable parameter of PTOACO algorithm
T_{sup}	Air supply temperature of CRAC	$Power$	Maximum energy consumption expected by the target host in the ant-generated allocation scheme of PTOACO algorithm
$T_i(t)$	CPU temperature of h_i at time interval t	$sumPower$	Total Power of all ants in PTOACO algorithm
T_0	Initial temperature of CPU	$temperature$	Sum of all target host temperatures in the ant-generated allocation scheme in PTOACO algorithm
$T_i^{in}(t)$	Inlet temperature of h_i at time interval t	$sumTemperature$	Total temperature of all ants in PTOACO algorithm
L	Number of host utilization history		

consumption of the cloud data center is minimized when the constraints of CRAC air supply temperature, the host utilization rate, the critical temperature of the host CPU and the host workload (represented by VM) constraints are considered.

$$\begin{aligned} MINNIZE\ P_{total} &= P_{IT} + P_{cooling} \\ &= \sum_{t=0}^{T}\sum_{p=1}^{N}(1+\frac{1}{COP(T_{sup})})P_{IT} \\ Subject\ to\ :&U(h_i) \le U_{max} \\ &T_i(t) < T_{red} \\ &\sum_{j=0}^{m} VM_{j,i}(R_{cpu}, R_{mem}) \le h_x(R_{cpu}, R_{mem}) \\ &x_j \in \{0, 1\} \end{aligned} \tag{9}$$

The objective function in Eq. (9) is committed to minimizing the data center's energy consumption as a whole. Constraints ensure that the host does not cause CPU utilization and temperature to exceed the threshold due to the increase in workload and that the target host selected can meet the resource requirements of the VM. x_j is a binary variable, and its value is 1 when the VM is assigned; Otherwise, it is 0.

The notations involved in this paper are shown in Table 1.

3.4. Algorithm

3.4.1. Algorithm overview

VM consolidation process of PTDS includes host overload detection algorithm AMD, VM selection algorithm MM and VM placement algorithm PTOACO. The relationship is shown in Fig. 3.

The AMD algorithm is used to determine whether host utilization is overloaded to detect overloaded hosts. When detecting the underloaded hosts, this paper uses the method of Beloglazov et al. [17] to iterate all the underloaded hosts. If all the virtual machines in the host can be migrated to other hosts, the host is determined to be an underloaded host. The VM selection phase uses the MM algorithm to find suitable VMs for migration. The VM placement phase uses PTOACO to select the target host for the VM to be migrated, considering the host's expected energy consumption and temperature.

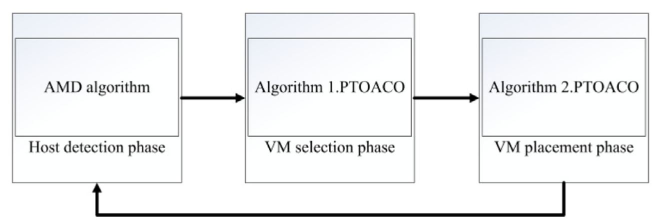


Fig. 3. VM consolidation process.

3.4.2. Algorithm details

The traditional utilization threshold determination method is a fixed threshold. If the host utilization exceeds the threshold, the system must migrate some VMs to reduce the utilization to prevent potential SLA violations. However, a fixed utilization threshold is unsuitable for a dynamic workload environment. The proposed AMD can adjust the value of the utilization threshold according to the strength of the CPU utilization deviation. It is a measure of statistical dispersion, which improves the shortcoming that the square of the difference between the data and the mean in the standard deviation needs to be calculated, resulting in a more significant weight of the value of a large deviation. In order to identify the server with overload utilization, it will check whether the utilization of the host will be greater than the upper threshold U_{max}:

$$\begin{aligned} AMD &= \frac{1}{L}\sum_{i=1}^{L}(|history_i - median(history)|) \\ U_{max} &= 1 - s \times AMD \end{aligned} \tag{10}$$

In Eq. (10), *AMD* is the average value of the difference between each record and the median in the server utilization history record, representing the average dispersion degree of the server's utilization history; $history_i$ represents the ith term of the host utilization history; L is the number of host utilization history; *median* is the median value operation; $s \in \mathbb{R}^+$ represents the safety factor. The evaluation level can be adjusted by changing it. If it is increased, it can avoid violating SLA but reduce server utilization. On the contrary, it can optimize resource utilization but increase the possibility of violating SLA. The complexity of AMD algorithm is $O(V_i \times length_{history})$, where Vi is the number of

VMs allocated to the h_i, and $length_{history}$ is the CPU history length of each VM.

Algorithm 1 shows the process of selecting the VM to be migrated. In this phase, the MM [17] is used. It first traverses the host list, arranges the hosts' VMs according to the utilization descending order (line 3), and then finds the most suitable VM for moving out of the host. In addition, the VMs will remain in the original host until it is removed in the next scheduling interval if there is no space in any of the other servers. The selected VM makes the utilization rate of the host lower than the highest threshold and reduces the utilization rate at least after being removed.

The complexity of MM algorithm is $O(N_{over} \times V_{hover})$, where N_{over} is the number of overloaded hosts, and V_{hover} is the VMs' number of each overloaded host.

Algorithm 1. MM.

```
Input: VMList, hostList
Output: vmsToMigrate
1: FOR hostList DO
2:     VMList ← h.getVmList()
3:     Sort VMList in descending order of utilization
4:     hUtil ← h.getUtil()
5:     bestFitUtil ← MAX
6:     WHILE hUtil > THRESH_UP DO
7:         FOR each vm in VMList DO
8:             IF vm.getUtil() > hUtil − THRESH_UP THEN
9:                 t ← vm.getUtil()-hUtil+THRESH_UP
10:                IF t<bestFitUtil THEN
11:                    bestFitUtil ← t
12:                    bestFitVm ← vm
13:                END IF
14:            ELSE
15:                IF bestFitUtil=MAX   THEN
16:                    bestFitVm ← vm
17:                END IF
18:                BREAK
19:            END IF
20:        END FOR
21:        hUtil ← hUtil-bestFitVm.getUtil()
22:        vmsToMigrate.add(bestFitVm)
23:        vmList.remove(bestFitVm)
24:    END WHILE
25: End For
26: RETURN   vmsToMigrate
```

Before introducing PTOACO, we first describe the process of VM allocation using an ant colony optimization algorithm. According to the rules of the ant colony algorithm, assigning a VM to a host is called an allocation path. In order to find a set of optimal feasible solutions, all ants are divided into two parts. According to the pheromone concentration matrix, one part of the ants choose the path with the highest concentration. The other part of the ant randomly chooses the allocation path to avoid premature convergence to the optimal local solution. Each ant selects the target host for each VM through the rule, obtains a set of feasible solutions, uses the best solution, and then uses the update rule to update the pheromone concentration of each path. Repeat the above operation until the final optimal feasible solution is obtained. PTOACO will consider the resource requirements of VMs, such as CPU, memory, and storage capacity, and select the most suitable host to ensure that the host will not exceed the utilization rate and temperature threshold after migration.

Due to the positive feedback characteristics of the traditional ant colony algorithm, the pheromone concentration in the algorithm's initial state is the same. The ants almost choose to update the pheromone concentration randomly. When the pheromone is updated, the ants will leave more pheromones in the path with higher pheromone concentration. This process will increase the difference caused by the initial state, resulting in local optimization, but the existence of a globally optimal solution is ignored. In this paper, the evaluation criteria of the solution and the updated method of pheromone concentration are improved to reduce the influence of optimal local problems on the results of the ant colony algorithm.

The pseudo-code of PTOACO is shown in Algorithm 2. PTOACO reduces the cost of computing systems and avoids potential thermal risks. Firstly, the algorithm sets the pheromone as the expected energy consumption after the host allocates the VM and obtains the initial pheromone concentration by calculation (lines 1 to 3). Then start iteratorNum iteration, and the former first criticalNum ants are selected to assign the VM to the host according to the maximum pheromone concentration in the VM-host mapping. The other ant randomly sets the allocation scheme (lines 6 to 10) to avoid premature algorithm convergence to the optimal local solution. Cyclically traversing all ants to calculate their respective, is the evaluation standard of the ant allocation scheme, and it is calculated by Eq. (11):

$$value = \alpha * \frac{Power}{sumPower} + (1 - \alpha) \times \frac{temperature}{sumTemperature} \tag{11}$$

In Eq. (11), the parameter α is not less than 0 and not more than 1, which is used to adjust the bias of the solution. When α increases, the influence of energy consumption on the selection of ants increases, and the influence of temperature decreases. On the contrary, the influence of temperature increases, and the influence of energy consumption decreases. Power is the maximum energy consumption expected by the target host in the ant-generated allocation scheme, and sumPower is the total Power of all ants. temperature is the sum of all target host temperatures in the ant-generated allocation scheme, and sumTemperature is the total temperature of all ants.

In order to simulate the pheromone volatilization process of ants in nature, the pheromones of all distribution paths are attenuated to the original p% at each iteration (line 12). After calculating the value of all ants, rank all ants in ascending order (line 13). maxQ is the maximum proportion of pheromone concentration that can be increased. The top 20% of ants are selected to update the pheromone concentration. According to the position of each ant in the ranking, the weight of the increased pheromone concentration is set. For example, the pheromone concentration of the distribution path corresponding to the kth ant increases to the original (maxQ-difference ×(k-1))%. The first ant increases the pheromone concentration to the original maxQ%. The increased proportion of the pheromone concentration of the distribution path corresponding to the 20% ants is close to 0% (line 15). In this way, the risk of ignoring the optimal solution due to insufficient initial advantages can be avoided.

The complexity of PTOACO algorithm is $O(iteratorNum \times antNum \times hostNum \times vmNum)$, where *iteratorNum* is number of iterations and the *antNum* is number of ants, and these two parameters can be freely adjusted; *hostNum* is number of potential target hosts, and *vmNum* is the number of VMs to be migrated.

Algorithm 2. PTOACO.

```
Input: vmsToMigrate, hostList
Output: VM-host mapping
1:FOR each VM-host mapping DO
2:   Sets the initial pheromone for the VM-host mapping by calculating the expected energy consumption after allocating VMs to the host.
3:END FOR
4:FOR Number of traversals iteratorNum DO
5:   FOR each ant k DO
6:      IF k<=criticalNum THEN
7:         Each ant sets the VM-host allocation scheme according to the maximum pheromone in the VM-host mapping
8:      ELSE
9:         Each ant sets the VM-host allocation scheme randomly
10:     END IF
11:  END FOR
12:  Pheromones of all distribution paths are attenuated to the original p% at each iteration
13:  Calculate the value and ascending order of each ant according to equation (10)
14:  FOR top 20% of ants k DO
15:     The pheromone concentration of the distribution path corresponding to the kth ant increases to the original (maxQ-difference× (k-1))%, where maxQ is the largest proportion of the pheromone concentration increase.
16:  END FOR
17:END FOR
18:RETURN VM-host mapping with maximum pheromone concentration
```

4. Experiments and results

4.1. Experimental setup

To prove the effectiveness of the proposed PTDS scheme, we created a large number of simulation experiments. We assume the cloud data center contains ten zones, and each zone has ten racks. The racks are arranged in 5 × 2 rows, and each rack has ten servers. Assuming that the heat recirculation effect exists within each zone and the host temperature is affected by other hosts. The experiment adopts the heat distribution matrix used by Tang et al. [10]. The entire data center contains 1000 hosts, each of which is an IBM x3550 M3 machine with eight processors and 4 GB RAM. There are two different types of hosts with different processors, namely Intel Xeon X5670 (6 judges, 2.93 GHz, 12MB L3 Cache) processor and Intel Xeon X5675 (6 judges, 3.07 GHz, 12MB L3 Cache) processor. The experiment sets four types of single-core VMs specifications that correspond to Amazon EC2 [37]. Because the simulation experiment uses the real-time workload traces obtained by PlanetLab from the CoMon project [12,38], the amount of RAM is divided by the number of cores for each VM type, as shown in Table 2.

Table 2
VM types.

VM size	Core	Processing speed (MIPS)	Ram (MB)	Bandwidth (Mbits/s)
Extra Large	1	2500	870	100
Large	1	2000	1740	100
Micro	1	1000	1740	100
Nano	1	500	613	100

Table 3
Experimental parameters.

Item		Value
Data center	Number of zones	10
	Number of racks in zone	10
	Number of hosts in rack	10
CRAC	Air supply temperature	25 °C
Server	Heat capacity	340 J/K
	Thermal resistance	0.34 K/W
	Initial CPU temperature	318 K
	Threshold of CPU temperature	70 °C
	Upper utilization threshold	80%
Simulation	Simulation time	24 h
	Simulation platform	CloudSim V.4.0

To evaluate the performance of dynamic migration, simulation experiments are performed using the parameters in Table 3. The work of Tang et al. [10] inspires the layout of the data center room. We assume each zone is affected by the heat recirculation effect, and Tang et at. also provides a heat distribution matrix to simulate the heat recirculation effect in the zone. According to ASHRAE [39], the air supply temperature of CRAC is 25 °C. In the host CPU temperature Eq. (5), the heat capacity and thermal resistance are 340 J/K and 0.34 K/W, respectively, and the initial CPU temperature is 318K [40].

The simulation experiment uses the real-time workload traces obtained by PlanetLab from the CoMon project [12,38]. The data is recorded at an interval of 5 min, as shown in Table 4, where the Mean represents the mean value of data and the SD represents the standard deviation. All algorithms are written in Java programming language, running on the Core i5-8500 CPU, 3.00 GHz, 16 GB RAM machine.

4.2. Metrics

The experiment evaluates the efficiency of the proposed scheme based on four standard metrics (energy consumption, hotspots, SLA violation, active hosts).

Energy: This metric represents energy consumption of each scheme in kWh.

SLA violation: This metric indicates the performance overhead caused by dynamic consolidation of VMs [38]. Meeting

Table 4
Workload data characteristics.

Workload datasets	Date	Number of VMs	Mean (%)	SD (%)
CoMon project workload trace	03/03/2011	1052	12.31	17.09
	06/03/2011	898	11.44	16.83
	09/03/2011	1061	10.70	15.57
	22/03/2011	1516	9.26	12.78
	25/03/2011	1078	10.56	14.14
	03/04/2011	1463	12.39	16.55
	09/04/2011	1358	11.12	15.09
	11/04/2011	1233	11.56	15.07
	12/04/2011	1054	11.54	15.15
	20/04/2011	1033	10.43	15.21

QoS requirements is significant for Cloud computing environments. Since the minimum throughput, maximum response time and bandwidth delivered by the system can vary depending on the application running. It is necessary to define a workload-independent metric $\mathrm{SLA_{violation}}$. When the utilization rate of the host is close to 100%, the VM performance level on the host will be limited by host capacity, which can be described as SLA violation time $\mathrm{SLA_{TAH}}$ for each active host. In addition, the consolidation of VMs will lead to performance degradation, which can be described as Performance Degradation due to Migration (PDM).

$$SLA_{TAH} = \frac{1}{N}\sum_{i=1}^{N}\frac{T_{si}}{T_{ai}} \qquad PDM = \frac{1}{M}\sum_{j=1}^{M}\frac{C_{d_j}}{C_{r_j}} \tag{12}$$

$$SLA_{violation} = SLA_{TAH} \times PDM \tag{13}$$

In Eq. (12), the N is the number of hosts; T_{si} is the total time during which the h_i has experienced the utilization of 100% leading to an SLA violation; T_{ai} is the total of the h_i being in the active state; M is the number of VMs; C_{d_j} is the estimate of the performance degradation of the j-th VM caused by migrations; C_{r_j} is the total CPU capacity (MIPS) requested by the j-th VM during its lifetime. The overall SLA violation $SLA_{violation}$ for cloud infrastructure can be obtained by combining SLA_{TAH} and PDM, as shown in Eq. (13).

Hotspots: This metric represents the number of hosts that have exceeded the temperature threshold.

Active hosts: This metric represents the number of active hosts during the experiment.

4.3. Baseline schemes

This paper verified the effectiveness of PTDS by comparing it with benchmark scheduling schemes.

Random-Random: In the VM placement and selection phase, a completely random strategy is used to randomly select the host placed without considering any constraints.

GRANITE-MMT: The VM placement phase uses a VM placement and migration algorithm GRANITE proposed by Li Xiang et al. [28] to minimize the total energy consumption. In order to achieve better cooling efficiency, the host will be greedily selected in the placement phase. The Minimum migration time (MMT) algorithm is used in the VM selection phase.

ETAS-MMT: The VM placement phase uses a dynamic consolidation VM algorithm proposed by Ilager et al. [30]. This algorithm uses a GRASP meta-heuristic online scheduling algorithm to reduce energy consumption while preventing hotspots. MMT is used in the VM selection phase.

PABFD-MM: The VM placement phase uses the Power-aware Modified Best Fit Decreasing algorithm (PABFD) proposed by Beloglazov et al. [17] which dynamically consolidates VMs. The basic idea is to set the upper and lower utilization threshold for the host to maintain the utilization of all VMs in the host. The VM selection phase uses a migration minimization algorithm (MM).

RACC-MDT: Ding et al. [24] proposed a VM consolidation framework based on resource utilization and heterogeneous host PPR. Residual Available Computing Capacity (RACC) technology is used to detect overloaded hosts in the VM placement stage. The Minimum Data Transfer algorithm (MDT) based on dynamic programming is used in the VM selection phase. This framework can effectively solve the trade-off between host computing overhead and performance.

TASA-MMT: The VM placement phase uses the thermal-aware scheduling algorithm (TASA) proposed by Wang et al. [27]. TASA allocate workloads based on their task-temperature profiles and allocate suitable resources for job execution. MMT is used in the VM selection phase.

HAWDA-MMT: The VM placement phase uses the thermal hotspot adaptive workload deployment algorithm HAWDA [34]. HAWDA uses a worst-case prediction model to predict the temperature and deploys workload on the server. MMT is used in the VM selection phase.

This paper configures the same simulation parameters for each scheduling scheme to compare, including the hardware conditions, workload and simulation environment of the data center. For the parameters of PTOACO, the parameters α, the number of iterations, the number of ants, criticalNum, the proportion of pheromone attenuation and the maximum proportion of pheromone increase are set to 0.5, 30, 30, 10, 80% and 150%, respectively. The experiment proves that the setting can obtain great results.

4.4. Experimental results and analysis

Firstly, each scheme's average energy consumption and hotspots are compared, and the results are shown in Fig. 4. It is observed that PTDS has a remarkable effect on limiting energy expenditure, which is 50.65% lower than that of Random-Random with the worst effect, and 25.97% lower than the average of all benchmark scheduling schemes. The Random-Random scheduling scheme does not consider constraints, which means that all the hosts in the sleep state have the same allocation opportunities. Many hosts opened randomly, leading to an increase in energy consumption. The randomness of Random-Random makes it entirely ignore the thermal risk of the computer room in the scheduling process, resulting in 9671 hotspots. GRANITE-MMT, ETAS-MMT, PABFD-MM and RACC-MDT tend to choose the host with minor energy consumption after placement as the target host of the VM. On this basis, GRANITE-MMT will choose the VM to migrate from the host with the temperature of the top 10%, but not all hosts will have the utilization overload. Excessive migration leads to the result that it is inferior to PABFD-MM. And the GRANITE-MMT adopts a thermal-aware scheduling strategy to avoid the increase of hotspots. ETAS-MMT uses GRASP technology to improve the greedy selection process of target hosts and reduce the algorithm's time complexity. However, there is no significant improvement in terms of reducing energy costs. Besides, ETAS-MMT stipulates the temperature constraint in selecting the target host but does not bias the host with low thermal risk in generating solutions, which increases the possibility of hotspots. PABFD-MM's MM algorithm ensures the selected VMs which are just making the host out of the utilization overload state after migrating from the host, so it does not cause excessive migration. And PABFD-MM has an aggressive VM consolidation strategy. Its upper threshold is set to 80% in this paper, thus ensuring that the host will migrate out VMs before the hotspot is generated. RACC-MDT tends to restore overloaded hosts to the normal state

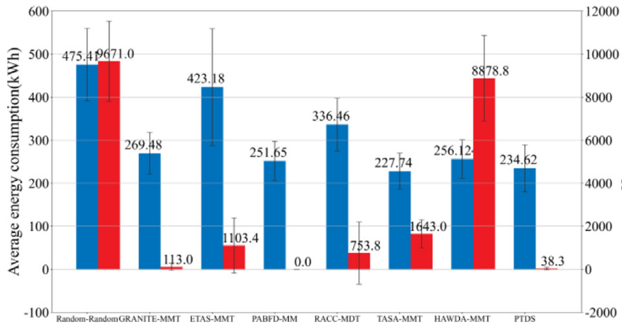


Fig. 4. Average energy consumption.

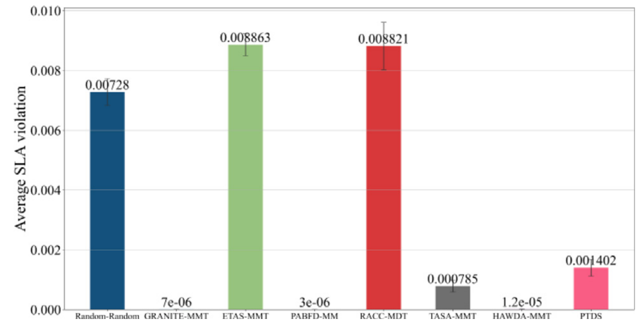


Fig. 5. Average SLA violation.

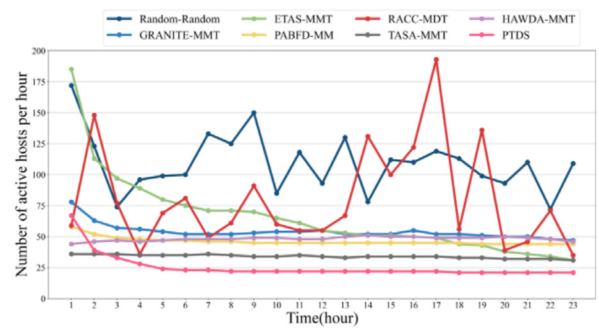


Fig. 6. Number of active hosts per hour.

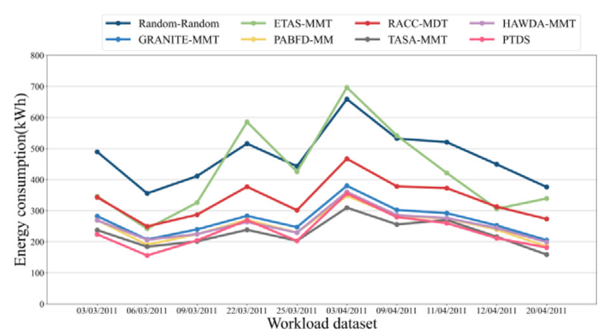


Fig. 7. Energy consumption under different workload datasets.

with the least amount of transmission in selecting hosts to be migrated. However, it increases the number of VMs migrated, increasing the number of open hosts and increasing computational overhead. Moreover, it does not consider the influence of hotspots and cannot avoid the generation of hotspots. The TASA-MMT tends to allocate the VM with the highest task-temperature profile to the "coolest" host. The VM with the highest task-temperature profile means that the VM needs the most CPU capacity, and the "coolest" host means the host's CPU capacity is ample. This characteristic reduces the number of migrated VMs from TASA, which consumes slightly less total energy than PTDS. But the number of hotspots is much higher than PTDS. And TASA-MMT tends to run VMs on the coldest host but does not consider resource constraints, which cannot result in a lower final temperature. The HAWDA-MMT allocates VM on the host with the slightest increase in predicted temperature. The temperature of the host is mainly related to CPU utilization, which results in HAWDA avoiding excessive computational energy consumption. But it only considers CPU capacity, quickly concentrating workloads on a few hosts that are not conducive to preventing hotspots. Compared with the above scheduling scheme, the MM algorithm of PTDS reduces unnecessary VM migration. AMD sets the CPU utilization threshold for the host, and PTOACO ensures that the target host will not exceed the utilization threshold after allocating VMs to achieve good average total energy consumption results. In addition, PTOACO excludes potential temperature-overloaded hosts when selecting target hosts and adjusts the final solution based on the predicted future temperature of the host. The occurrence of hotspots will have many effects: (1) Overheating temperature may lead to server failure. (2) To prevent the occurrence of hotspots, the air supply temperature of the CRAC needs to be lowered, which will lead to an increase in the cooling system's energy consumption, according to Eq. (2).

Fig. 5 shows the average SLA violation, with Random-Random, ETAS-MMT and RACC-MDT having higher SLA violation rates. Although PTDS is slightly worse on SLA violation than GRANITE-MMT, PABFD-MM TASA-MMT and HAWDA-MMT, PTDS optimizes both total energy consumption and hotspots, minimizing SLA violation with better performance.

Taking the 06/03/2011 dataset of the CoMon project as an example, Fig. 6 compares the number of active hosts per hour for each scheduling method. The experimental duration is 24 h. After removing the beginning and end, the data of the middle 23 time nodes are counted. It can be seen intuitively that Random-Random uses the most average number of active hosts compared to other schemes. The active hosts of GRANITE-MMT, ETAS-MMT, and PABFD-MM remain slightly above fifty because they always choose the hosts with the smallest increase in total power growth and allocate the newly arrived workloads to a non-idle host rather than activate a new host. The traversal characteristic of HAWDA-MMT to find hosts with the lowest expected temperature also causes it not to start many idle hosts. The MDT algorithm of RACC-MDT describes the VM selection problem as a 0–1 knapsack problem, which tends to restore the overloaded host to the normal state with the least amount of data transmission. However, it increases the number of VMs to be migrated, leading to an increase in the number of active hosts and the SLA violation caused by VM migration. TASA-MMT cools overheated computing nodes by shutting down overheated hosts. PTDS shuts down underloaded hosts as timely as PABFD-MM, and PTOACO selects the most suitable hosts for VMs from running hosts without activating new hosts to keep the number of active hosts at a low level.

Figs. 7 to 9 show the comparison of results under different workloads. The experimental results show that the proposed scheme PTDS has apparent advantages in minimizing energy consumption, reducing the number of hotspots and ensuring SLA violation, which further proves the effectiveness of PTDS in improving energy efficiency and reducing data center hotspots.

In order to investigate the impacts of the different numbers of hosts and VMs, the ratio of the hosts and VMs number is configured to 1:1, 1:1.25, 1:1.5 and 1:1.75, respectively, and the

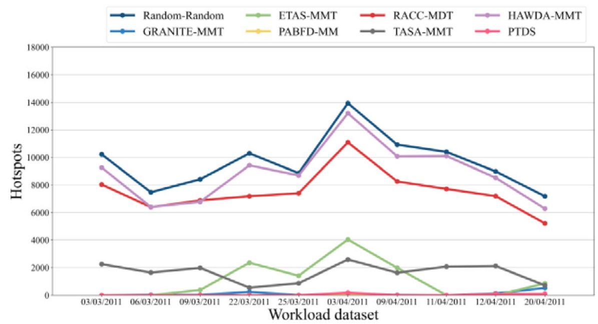


Fig. 8. Hotspots under different workload datasets.

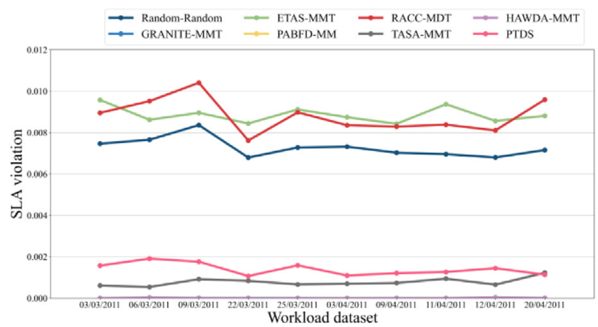


Fig. 9. SLA violation under different workload datasets.

total number of hosts is 1000. The experiment uses random workloads.

As shown in Fig. 10, PTDS can still maintain efficient performance in dealing with random workload datasets. When the number of VMs increases, the total energy consumption of all schemes increases. Among them, the performance of GRANITE-MMT, ETAS-MMT, RACC-MDT, TASA-MMT and HAWDA decreases significantly. Because when the workload is larger than the processing capacity of the host, it will increase the probability of utilizing an overloaded host and hotspots. Once the utilization load of the host exceeds the threshold, these schemes will migrate the VMs in time, thus increasing the energy cost and SLA violation caused by migration. The performance of PABFD-MM is much worse than the result of running the Planetlab workload data. Because the PABFD tends to allocate VMs to the hosts with the least power consumption improvement, but the higher ratio of hosts and VMs number results in high load per host. The PABFD can easily lead to excessive temperature. The selection of the target host by PTDS can improve energy utilization without excessive energy consumption and SLA violation.

As shown in Fig. 11, to analyze the influence of parameters in the PTDS scheme on the experimental results, this paper takes the CoMon project 03/03/2011 dataset as an example to analyze its sensitivity. The experimental results show that the α's size has a significant impact on the total energy consumption and the number of hotspots. Larger α will lead to a decrease in total energy consumption and an increase in the number of hotspots. Therefore, PTDS can control the influence of energy consumption and temperature on the solution by adjusting α.

In summary, the PTDS scheme proposed in this paper, whether dealing with real workloads or random workloads, can put the probability of total energy consumption, hotspots and SLA violation at a low level and has good robustness in dealing with workloads with different characteristics.

5. Conclusion and future works

Cloud data centers generate massive energy consumption on a global scope. It needs to reduce the overall energy consumption of cloud data centers under the premise of satisfying QoS and avoiding thermal risk and provides cloud service providers with the right to adjust the priority relationship between reducing energy consumption and reducing thermal risk. In order to achieve this goal, this paper proposes a VM dynamic scheduling scheme PTDS by studying the heat recirculation effect of the computer room. The VM consolidation process includes host detection, VM selection, and VM placement. In the host detection phase, the AMD algorithm is proposed to determine whether the utilization is overloaded by calculating the mean difference between the host utilization history and the median. The MM algorithm selects the VM to be migrated to the overloaded host in the VM selection phase. In the VM placement phase, a heuristic algorithm PTOACO based on an improved ant colony algorithm is proposed, which not only modifies the update rules of the ant colony algorithm to avoid prematurely falling into local optimum but also modifies the evaluation criteria of the solution, and adjusts the bias between energy consumption and temperature by presetting parameters in the algorithm.

Many experiments are conducted on the real-world workload dataset and random workload obtained by the PlanetLab system. The experiments show that PTDS is superior to the existing schemes in many metrics, which can avoid hotspots and effectively reduce energy consumption and has good robustness. This paper explores the thermal management and energy saving of data centers, analyzes its effectiveness, and lays the foundation for further research on a more effective scheduling scheme.

On this basis, we plan to expand the model in the future further. For example, we can consider the impact of heterogeneous servers relocated according to their thermal profiles and the regional inlet temperature. Other resources, such as the network, can be considered the basis for VM scheduling under temperature constraints. Meanwhile, we can add dynamic control cooling supply technology. According to the heat distribution of the cloud data center, we can dynamically adjust the air supply temperature and wind speed of CRAC to reduce the waste of cooling system resources.

CRediT authorship contribution statement

Rui Chen: Methodology, Software, Writing – original draft. **Bo Liu:** Methodology, Writing – review & editing. **WeiWei Lin:** Methodology, Writing – review & editing, Funding acquisition. **JianPeng Lin:** Review & editing. **HuiWen Cheng:** Review & editing. **KeQin Li:** Review & editing.

Declaration of competing interest

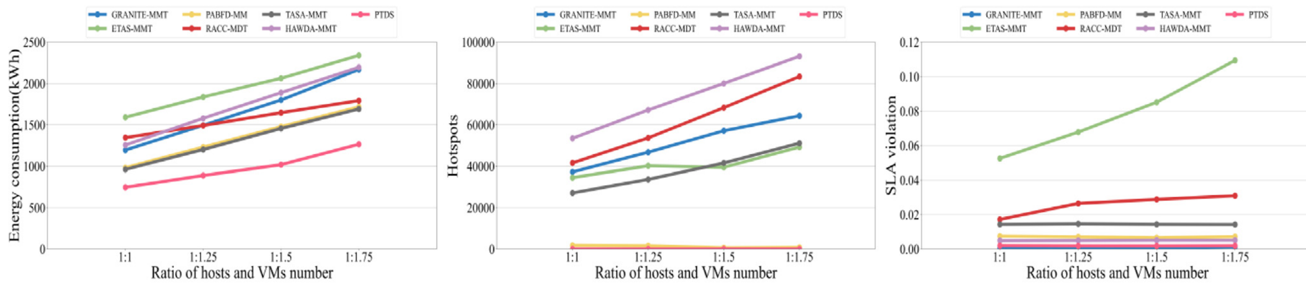
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

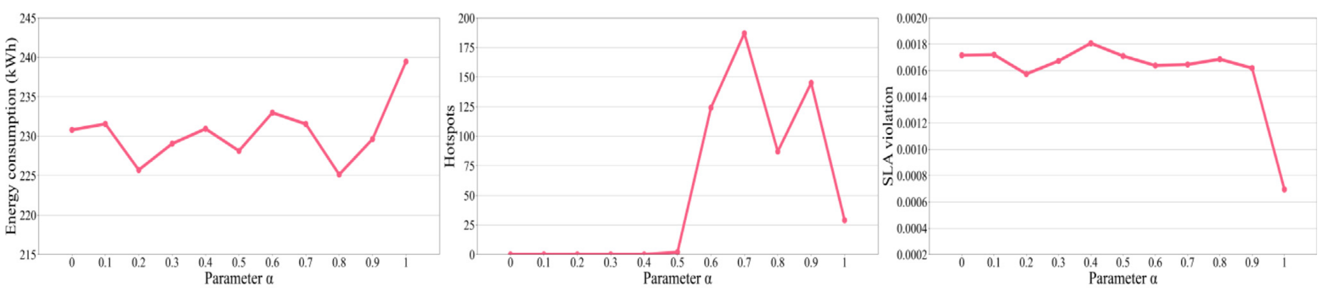
Acknowledgments

This work is supported by Key-Area Research and Development Program of Guangdong Province, China(2021B0101420002), National Natural Science Foundation of China (62072187), Guangdong Major Project of Basic and Applied Basic Research, China (2019B030302002), the Major Key Project of PCL(PCL2021A09), Guangdong Marine Economic Development Special Fund Project, China (GDNRC[2022]17) and Guangzhou Development Zone Science and Technology Project, China (2021GH10, 2020GH10).



(a) (b) (c)

Fig. 10. Performance comparison of schemes under random workloads (setting hosts number=1000 and varying ratios of hosts and VMs number are1:1, 1:1.25, 1:1.5, 1:1.75).



(a) (b) (c)

Fig. 11. Sensitivity analysis of parameter α (a) Energy consumption; (b) Hotspots; (c) SLA violation.

References

[1] A. Shehabi, S.J. Smith, E. Masanet, et al., Data center growth in the United States: decoupling the demand for services from electricity use, Environ. Res. Lett. 13 (12) (2018) 124030.
[2] H.M. Daraghmeh, C.C. Wang, A review of current status of free cooling in datacenters, Appl. Therm. Eng. 114 (2017) 1224–1239.
[3] K. He, Z. Li, D. Deng, et al., Energy-efficient framework for virtual machine consolidation in cloud data centers, China Commun. 14 (10) (2017) 192–201.
[4] M. Arif, T. Mahmood, Cloud computing and its environmental effects, Int. J. Grid and Distrib. Comput. 8 (1) (2015) 279–286.
[5] Q. Zhang, Z. Meng, X. Hong, et al., A survey on data center coolingsystems: Technology, power consumption modeling and control strategy optimization, J. Syst. Archit. 119 (2021) 102253.
[6] S.Y. Hsieh, C.S. Liu, R. Buyya, et al., Utilization-prediction-aware virtual-machine consolidation approach for energy-efficient cloud data centers, J. Parallel Distrib. Comput. 139 (2020) 99–109.
[7] K. Karmakar, R.K. Das, S. Khatua, An ACO-based multi-objective optimization for cooperating VM placement in cloud data center, J. Supercomput. 78 (3) (2022) 3093–3121.
[8] P. Xiao, Z. Ni, D. Liu, et al., A power and thermal-aware virtual machine management framework based on machine learning, Cluster Comput. 24 (3) (2021) 2231–2248.
[9] H. Lu, Z. Zhang, Numerical and experimental investigations on the thermal performance of a data center, Appl. Therm. Eng. 180 (2020) 115759.
[10] Q. Tang, S.K.S. Gupta, G. Varsamopoulos, Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber–physical approach, IEEE Trans. Parallel Distrib. Syst. 19 (11) (2008) 1458–1472.
[11] R.N. Calheiros, R. Ranjan, A. Beloglazov, et al., CloudSim: a toolkit formodeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, Softw. - Pract. Exp. 41 (1) (2011) 23–50.
[12] K.S. Park, V.S. Pai, CoMon: a mostly-scalable monitoring system for PlanetLab, Oper. Syst. Rev. 40 (1) (2006) 65–74.
[13] M.H. Shirvani, A.M. Rahmani, A. Sahafi, A survey study on virtual machine migration and server consolidation techniques indvfs-enabled cloud datacenter: taxonomy and challenges, J. King Saud Univ.-Comput. Inf. Sci. 32 (3) (2020) 267–286.
[14] Y. Chen, A. Das, W. Qin, et al., Managing server energy and operationalcosts in hosting centers, in: Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, 2005, pp. 303–314.
[15] Í. Goiri, W. Katsak, K. Le, et al., Parasol and greenswitch: Managing datacenters powered by renewable energy, ACM SIGPLAN Not. 48 (4) (2013) 51–64.
[16] J.P.B. Mapetu, L. Kong, Z. Chen, A dynamic VM consolidation approach based on load balancing using Pearson correlation in cloud computing, J. Supercomput. 77 (6) (2021) 5840–5881.
[17] A. Beloglazov, J. Abawajy, R. Buyya, Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing, Future Gener. Comput. Syst. 28 (5) (2012) 755–768.
[18] Luo Gang-Yi, Qian Zhu-Zhong, Lu Sang-Lu, A network-aware VM re-scheduling algorithm, Chinese J. Comput. 38 (5) (2015) 932–943.
[19] Zhao Xiao-Gang, Hu Qi-Ping, Ding Ling, et al., Energy saving scheduling strategy based on model prediction control for data centers, J. Softw. 28 (2) (2017) 429–442.
[20] M. Gaggero, L. Caviglione, Predictive control for energy-aware consolidation in cloud datacenters, IEEE Trans. Control Syst. Technol. 24 (2) (2015) 461–474.
[21] N.J. Kansal, I. Chana, Energy-aware virtual machine migration for cloud computing-a firefly optimization approach, J. Grid Comput. 14 (2) (2016) 327–345.
[22] H. Li, G. Zhu, C. Cui, et al., Energy-efficient migration and consolidation algorithm of virtual machines in data centers for cloud computing, Computing 98 (3) (2016) 303–317.
[23] L. Li, J. Dong, D. Zuo, et al., SLA-aware and energy-efficient VM consolidation in cloud data centers using host state 3rd-order Markov chain model, Chin. J. Electron. 29 (6) (2020) 1207–1217.
[24] W. Ding, F. Luo, L. Han, et al., Adaptive virtual machine consolidation framework based on performance-to-power ratio in cloud data centers, Future Gener. Comput. Syst. 111 (2020) 254–270.
[25] J. Shuja, A. Gani, S. Shamshirb, et al., Sustainable cloud data centers: a survey of enabling techniques and technologies, Renew. Sustain. Energy Rev. 62 (2016) 195–214.
[26] H. Sun, P. Stolf, J.M. Pierson, Spatio-temporal thermal-aware scheduling for homogeneous high-performance computing datacenters, Future Gener. Comput. Syst. 71 (2017) 157–170.
[27] L. Wang, S.U. Khan, J. Dayal, Thermal aware workload placement with task-temperature profiles in a data center, J. Supercomput. 61 (3) (2012) 780–803.
[28] X. Li, P. Garraghan, X. Jiang, et al., Holistic virtual machine scheduling in cloud datacenters towards minimizing total energy, IEEE Trans. Parallel Distrib. Syst. 29 (6) (2017) 1317–1331.
[29] E.K. Lee, H. Viswanathan, D. Pompili, Proactive thermal-aware resource management in virtualized HPC cloud datacenters, IEEE Trans. Cloud Comput. 5 (2) (2015) 234–248.

[30] S. Ilager, K. Ramamohanarao, R. Buyya, ETAS: Energy and thermal-aware dynamic virtual machine consolidation in cloud data center with proactive hotspot mitigation, Concurr. Comput.: Pract. Exper. 31 (17) (2019) e5221.
[31] A. Akbari, A. Khonsari, S.M. Ghoreyshi, Thermal-aware virtual machine allocation for heterogeneous cloud data centers, Energies 13 (11) (2020) 2880.
[32] H. Feng, Y. Deng, J. Li, A global-energy-aware virtual machine placement strategy for cloud data centers, J. Syst. Archit. 116 (2021) 102048.
[33] A. Aghasi, K. Jamshidi, A. Bohlooli, A thermal-aware energy-efficient virtual machine placement algorithm based on fuzzy controlled binary gravitational search algorithm (FC-BGSA), Cluster Comput. (2022) 1–19.
[34] M.H. Jamal, M.T. Chaudhry, U. Tahir, et al., Hotspot-aware workload scheduling and server placement for heterogeneous cloud data centers, Energies 15 (7) (2022) 2541.
[35] S.A. Ludwig, A. Moallem, Swarm intelligence approaches for grid load balancing, J. Grid Comput. 9 (3) (2011) 279–301.
[36] J.D. Moore, J.S. Chase, P. Ranganathan, et al., Making scheduling cool:Temperature-aware workload placement in data centers, in: USENIX Annual Technical Conference, General Track, 2005, pp. 61–75.
[37] P. Humane, Simulation of cloud infrastructure using CloudSim simulator: A practical approach for researchers, in: International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), Controls, Energyand Materials, IEEE, 2015, pp. 207–211.
[38] A. Beloglazov, R. Buyya, Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers, Concurr. Comput.: Pract. Exper. 24 (13) (2012) 1397–1420.
[39] ASHRAE, American society of heating, refrigerating and air-conditioning engineers, 2018, http://tc0909.ashraetcs.org/.
[40] S. Zhang, K.S. Chatha, Approximation algorithm for the temperature-aware scheduling problem, in: 2007 IEEE/ACM International Conference on Computer-Aided Design, IEEE, 2007, pp. 281–288.

Rui Chen is a master student in Computer Science and Technology at South China Normal University. He received his bachelor degree in Computer Science and Technology from Shandong University of Science and Technology in 2020. His research interests include cloud computing.

Bo Liu is currently a professor in the School of Computer Science at South China Normal University. His research interests include cloud storage technology, cloud computing and big data technology.

Weiwei Lin received his B.S. and M.S. degrees from Nanchang University in 2001 and 2004, respectively, and the Ph.D. degree in Computer Application from South China University of Technology in 2007. He has been serving as visiting scholar at Clemson University from 2016 to 2017. Currently, he is a professor in the School of Computer Science and Engineering, South China University of Technology. His research interests include distributed systems, cloud computing, big data computing and AI application technologies. He has published more than 100 papers in refereed journals and conference proceedings. He has been the reviewers for many international journals, including TC, TCYB, TSC, TCC, Information Sciences, Future Generation Computer Systems, etc. He is a senior member of CCF and a member of the IEEE.

JianPeng Lin received the M.S. degree in computer science at Guangdong University of Technology in 2019. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. His research interests include cloud computing, deep reinforcement learning and multi-agent system.

HuiWen Cheng is a master student in Computer Science and Technology at South China Normal University. Her research interests include distributed computing and cloud computing.

Dr. Keqin Li is a SUNY Distinguished Professor of computer science in the State University of New York. He is also a Distinguished Professor of Chinese National Recruitment Program of Global Experts (1000 Plan) at Hunan University, China. He was an Intellectual Ventures endowed visiting chair professor at the National Laboratory for Information Science and Technology, Tsinghua University, Beijing, China, during 2011–2014. His current research interests include parallel computing and high-performance computing, distributed computing, energy-efficient computing and communication, heterogeneous computing systems, cloud computing, big data computing, CPU–GPU hybrid and cooperative computing, multicore computing, storage and file systems, wireless communication networks, sensor networks, peer-to-peer file sharing systems, mobile computing, service computing, Internet of things and cyber–physical systems. He has published over 520 journal articles, book chapters, and refereed conference papers, and has received several best paper awards. He is currently or has served on the editorial boards of IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Computers, IEEE Transactions on Cloud Computing, IEEE Transactions on Services Computing, and IEEE Transactions on Sustainable Computing. He is an IEEE Fellow.