Future Generation Computer Systems 113 (2020) 329–342

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Memory-aware resource management algorithm for low-energy cloud data centers

Bin Liang [a,b], Xiaoshe Dong [a,*], Yufei Wang [a], Xingjun Zhang [a]

[a] School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, China
[b] School of Computer Science, Xi'an Shiyou University, Xi'an, China

ARTICLE INFO

Article history:
Received 15 February 2020
Received in revised form 26 May 2020
Accepted 12 July 2020
Available online 16 July 2020

Keywords:
Cloud data center
Memory-aware
Cloud task mapping
Virtual machine deployment

ABSTRACT

The continuous advancement of cloud computing technology has driven the vigorous development of cloud data centers. This manifests itself not only in increasing numbers, but also in rapid expansion scale. At the same time, the problems of high energy consumption, high cost and high carbon emissions began to stand out. These factors have become a bottleneck restricting the further development of cloud computing technology. As a result of the application of virtualization technology, mature cloud service providers use virtual machines instead of physical machines to provide users with computing, storage and other services. Therefore, the scheduling algorithm of cloud tasks and virtual machines has been widely studied by academia as a core problem. However, most of the current cloud data center resource management algorithms take CPU utilization as the main consideration, and increase the CPU utilization through virtual machine integration to reduce the energy consumption of cloud data centers. However, these algorithms will cause waste and low utilization of other resources in the cloud data center due to excessive consideration of CPU utilization. This paper systematically analyzes the mapping relationship between cloud tasks, virtual machines and physical machines. At the same time, the performance of cloud tasks, virtual machines, and physical machines is modeled in the cloud data center. By comprehensively considering the CPU and memory characteristics of cloud tasks, the memory priority cloud task mapping rule is established. Based on the rule, the memory-aware resource management algorithm for low-energy cloud data centers (MALE) is proposed. The algorithm maps cloud tasks and deploys virtual machines according to the memory requirements of the cloud tasks, thereby achieving the goal of simultaneously reducing the total fee of cloud users and the energy consumption of cloud data centers. Finally, the algorithm is compared with the other three algorithms. The experimental simulation results show that the effect of the proposed algorithm in this paper is significantly better than the comparison algorithm for the total fee of cloud users and the energy consumption of the cloud data center.

1. Introduction

Large-scale cloud data centers inevitably bring higher energy consumption. For enterprises, high energy consumption means high costs. With the global energy shortage and climate warming, the energy consumption of cloud computing has become one of the bottlenecks restricting the development of cloud computing. In the face of high energy consumption, the academia and the industry have jointly proposed the concept of green cloud computing. The realization of green cloud computing depends not only on technological progress and the improvement of manufacturing levels, but also on the integration and efficient use of existing resources.

The energy saving methods in cloud computing are mainly divided into two methods: static energy saving and dynamic energy saving. Static energy saving mainly deals with computer hardware equipment. This method mainly includes the energy-saving design of the circuit layer and the low-power architecture of the processor. At the same time, an improved algorithm of the GPU is used to propose an energy-efficient black-box scheduling technology for the integrated processor. The core of the scheduler is the energy consumption model of each processor. This model takes into account the runtime characteristics of the load, and each processor has to recalculate its energy consumption model when performing different types of tasks. For computing tasks, faster speeds also mean reduced energy consumption [1]. However, because this design has internally solidified the hardware, it cannot make real-time adjustments based on the specific conditions of the cloud data center. Therefore, it is difficult to achieve the purpose of truly reducing energy consumption.

* Corresponding author.
E-mail address: xsdong@mail.xjtu.edu.cn (X. Dong).

https://doi.org/10.1016/j.future.2020.07.026
0167-739X/© 2020 Elsevier B.V. All rights reserved.

Dynamic energy saving mainly includes dynamic voltage and frequency scaling (DVFS) technology and optimized scheduling algorithm. For DVFS task scheduling on a single processor, the more classic method is EDF algorithm [2]. Later, the DVFS energy saving scheduling algorithm that satisfies the task deadline time constraint is also studied. The difference is that the latter is mainly for real-time task scheduling, and the same batch of tasks have the same arrival time and latest end time [3]. Although DVFS technology can play a good role in reducing the energy consumption of a single physical machine, it will not be able to generate a global optimal solution due to local optimization when processing the overall problem of the cluster. Therefore, the improvement of the scheduling algorithm is a good research approach. This algorithm can describe the problem as a given set of tasks submitted and the corresponding resource requirements by cloud users. How to map these cloud tasks with a limited combination of resources to optimize some given goals. The goals generally include cloud task execution time, user fees, cloud provider profits, and cloud data center energy consumption [4].

In the research on energy consumption of cloud data centers, some literatures have established corresponding energy consumption models. These algorithms obtained the parameters of the model through experimental simulation. First, the virtual machine was initially mapped according to the model, and then the virtual machine in the overloaded physical machine was migrated [5]. Some algorithms use the difference in execution time of cloud tasks on different virtual machines to analyze the time loss caused by changing the execution order of cloud tasks. In the end, the algorithm prioritizes the expensive cloud tasks to reduce the energy consumption of physical machines. The copy backup strategy is also one of the methods to reduce energy consumption. During the virtual machine mapping process, only the virtual machine copy can be accessed, thereby reducing the energy consumption during the mapping process. In the end, the overall energy consumption is reduced [6]. The efficient virtual machine mapping algorithm also plays a role in reducing energy consumption. It can minimize the number of activated physical machines through different filling methods, and shut down idle physical machines to reduce the energy consumption of cloud data centers [7,8]. At the same time, some algorithms migrate virtual machines in cloud data centers based on the utilization of physical machines. They migrate virtual machines deployed on physical machines that are above the upper threshold of utilization and under the lower threshold of utilization to other physical machines. However, this approach will increase the migration loss of virtual machines [9–11]. Although the above algorithm has achieved certain results. But cloud data centers as a complex system need to consider some other factors. First of all, different operating cycles of virtual machines and different resource usage rates will cause different mapping effects. Different mapping effects will cause great differences in energy consumption. Different mapping orders will also produce different mapping effects. Secondly, the fee of cloud users using virtual machines is also an important factor limiting the development of cloud computing. Excessive fee will reduce the efficiency of cloud data center use. Third, due to the fact that most cloud data centers use heterogeneous clusters composed of cheap machines. Each physical machine will show different performance due to different configurations of CPU and memory. The previous algorithms were used to solve the mapping problem of homogeneous clusters, and did not consider the heterogeneity of the cluster. Finally, the attributes of virtual machines and physical machines are generally composed of multi-dimensional factors such as CPU utilization and memory size. Most of the previous algorithms only considered CPU utilization or CPU utilization as the main factor, which would cause a huge waste of other resources of the physical machine.

Based on the above reasons, we first systematically analyze the cloud tasks, virtual machines and physical machines and establish a cloud data center scheduling model in this paper. Secondly, it considers many factors that affect the energy consumption of cloud data centers and establishes a cloud data center energy consumption model. Thirdly, the memory characteristics of virtual machines and the heterogeneous characteristics of cloud data centers are studied. The memory priority cloud task mapping rule are proposed. Then, based on the above rule and cloud data center scheduling model, a memory-aware resource management algorithm for low-energy cloud data centers (MALE) is proposed. The algorithm readjusts the mapping of cloud tasks and the establishing virtual machines to reduce the total fee of cloud users and the energy consumption of cloud data centers. Finally, the algorithm is verified using the established verification platform in this paper.

The main contributions of this paper are as follows:

(1) We have established the cloud data center scheduling model and energy consumption model.

(2) We propose a memory priority cloud task mapping rule.

(3) We propose a memory-aware resource management algorithm for low-energy cloud data centers (MALE).

(4) To verify the effectiveness of the algorithm, we embedded a simulation verification platform in CloudSim, including a virtual machine generator and a physical machine generator to simulate the real environment of a cloud data center. Finally, the effect of HTVM2 algorithm is verified by CloudSim.

The remaining sections of this paper are organized as follows: Section 2 introduces the predecessors' relevant work. Section 3 describes the cloud data center model and rule. Section 4 is the core algorithm (MALE) of this paper. The test results of the algorithm will appear in Section 5. The full text is summarized in Section 6.

2. Related works

As the scope of cloud computing applications and the scale of cloud data centers continue to expand, a large number of studies have adopted efficient scheduling algorithms and green energy-saving energy supplies. The purpose of these algorithms is to reduce the fee of cloud users and the energy consumption of cloud data centers.

[12] proposed an energy management scheme for heterogeneous multi-core systems. Based on the price theory in economics, this solution uses three methods of DVFS, load balancing among different processing units, and task migration to minimize the energy consumption of the system. [13] proposed an energy-aware scheduling algorithm to deal with batch disk requests. This problem is abstracted as a set cover problem, and the greedy solution of the set cover problem is used to solve the problem. [14] divides nodes into two parts based on the same idea. Their storage methods are also basically the same. Common data is stored on the working nodes, and other data is stored on the low-power nodes. [15] eliminates the server idle state energy consumption by quickly switching the server between the running state and the ultra-low power consumption state. Compared with the traditional DVFS method, this method can save more energy consumption and ensure a shorter response time. [16] believes that the chain storage method is helpful to achieve efficient use of energy. Each node stores part of the data and selects some nodes to process the task during the task processing. At the same time, load balancing between nodes can be achieved by task migration. [17] proposed a flexible energy-saving scheduling method for heterogeneous computing systems, which reduced the energy consumption of the system on the basis of meeting user time expectations. When the system load

is heavy, adjust the processing speed of each node to increase the task processing speed. When the system load is light, the processing speed of the nodes is reduced as much as possible under the condition that meets the user time expectations to reduce energy consumption. [18] pointed out the difference in the minimum execution time of different cloud tasks on virtual machines. Use this difference to change the order of cloud task execution, thereby reducing virtual machine execution time and energy consumption in cloud data centers. [19] proposed an online optimization method that balances the performance and energy consumption of a single data center. This method dynamically turns on or off several servers based on historical and current traffic conditions. Therefore, while reducing data center energy consumption and switching costs, it can ensure that the queuing delay of task loads is minimized. [20] considered the energy efficiency management of homogeneous resources in Internet hosting. Its main challenge is to determine the resource requirements so that each application allocates resources in the most efficient manner at its current request load level. This can be negotiated based on the available budget and the service needs of the current user. [21] developed a system for monitoring to detect hot spots and remap virtual machines if necessary. To select the virtual machines to migrate, it sorts them using the ratio of volume to capacity. This is a metric based on CPU, network and memory load. [22] studies the energy-saving scheduling of virtual resources in private clouds. This paper proposes a layout-based virtual machine energy-saving scheduling method that uses the minimum load priority mechanism for virtual machine allocation. But this method is not applicable to the case of heterogeneous nodes. [23] constructed an application-independent model to estimate the energy consumption and performance of virtual machine migration. This model is based on the learning of historical data. [24] introduced an improved Max–Min algorithm, which uses slower resources to perform tasks that require longer completion times. The advantages of this algorithm are reduced waiting time, improved resource utilization, and reduced task completion time. [25] introduced multi-standard decision-making techniques and replaced meta-heuristic algorithms in order to avoid problems such as inefficient resource sharing, low utilization, waste of resources, and loss of revenue caused by static allocation strategies and pricing models. [26] proposed a greedy algorithm based on quadratic exponential smoothing for virtual machine allocation and migration. It allocates CPU-intensive virtual machines and memory-intensive virtual machines to the same physical machine, so as to maximize the use of physical machine resources and reduce energy consumption. The energy-saving method in [27] takes advantage of data center hardware heterogeneity. It allows energy-optimized components and services to be provided to dynamically adapt hardware components to meet different workload and performance requirements. [28] proposed a virtual machine mapping method based on the characteristics of virtual machines and physical machines. This method improves the utilization of physical machine and the energy consumption of cloud data center by preferentially filling appropriate virtual machines. The algorithm proposed in [29] sets the lower limit of physical machine utilization. Migrate virtual machines on physical machines below the lower utilization limit, and randomly migrate the migrated virtual machines. [11] used PSO algorithm to schedule virtual machines to be migrated on the physical machine with higher load. At the same time, the dynamic integration of the virtual machine is performed on the physical machine with a lower load to reduce energy consumption. However, methods based on simulation optimization usually require multiple iterations to find the final solution, and the convergence speed of the iteration is related to practical applications. Therefore, it is not suitable for applications that require high real-time performance. [30] introduced a cloud computing resource rescheduling strategy for workflow tasks. First, the related background of resource rescheduling for distributed systems is introduced. Then, It models the workflow tasks and cloud data centers. It includes monitoring module, response module, learning module and memory module. Finally, simulation experiments verify the feasibility and effectiveness of the algorithm. [31] proposed a novel scheduling algorithm designed to meet the high performance and fast requirements of heterogeneous processor scheduling time, called the heterogeneous earliest completion time (HEFT) algorithm. The HEFT algorithm selects the task with the highest upward ranking value at each step and assigns the selected task to the virtual machine, minimizing its earliest completion time using an insertion-based method. In [32], the heterogeneous earliest completion time (E-HEFT) was improved and an algorithm was proposed to load balance across virtual machines under user-specified financial constraints. At the same time, in order to meet the dynamic change of heterogeneous earliest completion time (HEFT) that cannot effectively assign tasks, it tries to minimize the given workflow application.

[33] considers different ways to purchase different types of new energy to power the data center. The problem is modeled as minimizing the total energy cost of the data center while meeting the constraints of different carbon emission levels. [34] proposed a solution to estimate data center power capacity based on new energy production and requested load. This solution gives a reference for the supply of electrical energy from different sources, so as to achieve a balance between supply and demand. [35] optimized the use of spatial differences in carbon emission rates to reduce the actual carbon emissions of data centers, but this study did not take into account the dynamic differences in carbon emission rates in the dimension of time. Therefore, there is room for further exploration for deep energy saving and emission reduction of cross-domain data centers. [36] proposed a green mobile cloud computing network architecture, where user mobile terminals can communicate with base stations equipped with small server clusters. It uses new energy produced by solar panels to supply power, and the insufficient part directly uses the power grid. [37] proposed a method based on label reduction to reduce the energy consumption of the first-level instruction cache of the on-chip multiprocessor. It can allocate the entire instruction page in memory to a group of cores, thereby minimizing label reduction conflicts on each core. [38] considered using the latest server with multiple sleep modes. Sleep modes with smaller transition delays typically consume more power while sleeping. It expresses this problem as an integer linear programming problem with millions of decision variables over the entire time period. [39] designed a cloud computing resource elasticity providing mechanism for workflow tasks, including resource expansion strategy and resource integration strategy. Then the corresponding task elastic fault-tolerant scheduling strategy and task copy elastic fault-tolerant scheduling strategy are designed. [40] first verified the conflict between minimizing electricity costs and minimizing carbon emissions based on empirical research on several representative geographically distributed cloud services. They then jointly considered electricity bills, service level agreement requirements and emission reduction budgets. [41] mainly studied the green scheduling problem of cloud data centers in an economical manner through energy transactions with the power grid. Data centers can be powered by their own wind or solar energy or energy purchased from renewable energy power plants. Renewable energy can be used to directly power the data center, or sold back to the grid to provide part of the data center's high energy consumption. [42] analyzed the difference between the minimum completion time and the secondary small completion time of the cloud task on the same physical machine. Finally,

the cloud task with the largest difference is executed. This can reduce the running time of virtual machines and ultimately reduce energy consumption in cloud data centers. [43] provided an analysis framework for characterizing and optimizing power performance trade-offs in cloud platforms. It maximizes cloud provider operating profit and minimizes power consumption. To achieve these goals, they use Lyapunov optimization technology to design and analyze the optimal control framework to make online decisions on request admission control, routing, and virtual machine scheduling. [44] proposed an energy-efficient scheduling algorithm for virtual machines based on deadline constraints. It believes that there is an optimal frequency for a physical machine to handle certain virtual machines. The physical machine with the best performance and power ratio will be allocated to the virtual machine.

In summary, most of the cloud resource management algorithms in the above research are based on the CPU utilization as the research object. By increasing the CPU utilization, the energy consumption of cloud data centers is reduced. However, there are various configurations of physical machines in actual cloud data centers. Each physical machine contains various resources such as CPU and memory. At the same time, different operating cycles of virtual machines and different resource usage rates will result in different mapping effects. Different mapping effects will cause great differences in energy consumption. Different mapping orders will also produce different mapping effects. The fee of cloud users using virtual machines is also an important factor limiting the development of cloud computing. Excessive fee will reduce the efficiency of using cloud data centers. In this paper, we systematically analyze cloud tasks, virtual machines, and physical machines to establish a cloud data center scheduling model. The memory priority cloud task mapping rule is proposed. Then, based on the above rule and cloud data center model, the memory-aware resource management algorithm for low-energy cloud data centers (MALE) is proposed. The algorithm readjusts the mapping of cloud tasks and the order in which virtual machines are created, thereby reducing the total fee of cloud users and the energy consumption of cloud data centers. Finally, the algorithm is verified using the verification platform established in this paper.

3. Cloud data center model and rule

3.1. Cloud data center scheduling model

In this section, we first introduce the scheduling model of the cloud data center. The cloud data center studied in this article includes a host cluster, $\mathrm{PM} = \{\mathrm{pm}_1, \mathrm{pm}_2, \ldots, \mathrm{pm}_n\}$, where n represents the number of physical machines in this cloud data center. Each physical machine can be described as $\mathrm{pm}_i = \{c_i, m_i, s_i, p_i, t_i\}$, where c_i, m_i, s_i, p_i and t_i respectively represent the number of CPU cores, memory size (GB), CPU core processing speed (MIPS), CPU core usage price (\$/h), and physical machine running time (s). Each physical machine can hold a certain number of virtual machines, $\mathrm{VM}_i = \{\mathrm{vm}_1, \mathrm{vm}_2, \ldots, \mathrm{vm}_m\}$, where m represents the total number of virtual machines contained on the physical machine. Each virtual machine also contains the number of CPU cores, memory size (GB), CPU core processing speed (MIPS), CPU core usage price (\$/h), the physical number of the running this virtual machine and virtual machine running time (s) can be expressed as $\mathrm{vm}_{ij} = \{c_{ij}, m_{ij}, s_{ij}, p_{ij}, n_{ij}, t_{ij}\}$. In order to ensure the performance of each virtual machine deployed on the physical machine, the total CPU core and memory of the virtual machine contained in each physical machine cannot exceed the capacity of the physical machine. At the same time, the cloud tasks submitted by the user form a cloud task set, $\mathrm{CT} = \{\mathrm{ct}_1, \mathrm{ct}_2, \ldots, \mathrm{ct}_t\}$, where t represents the number of cloud tasks in this set. Through the scheduling algorithm, the cloud tasks in the set are mapped to the corresponding virtual machines for execution in sequence. Each cloud task can be described as $\mathrm{ct}_{ijk} = \{l_{ijk}, m_{ijk}, n_{ijk}\}$, where l_{ijk}, m_{ijk} and n_{ijk} represent the length of the cloud task (MI), memory size (GB) and the number of the virtual machine executing the cloud task.

In order to maximize resource utilization, modern data centers use virtualization technology to virtualize hardware resources into virtual machines in the data center. It provide cloud users with computing and storage services in units of virtual machines. When a user submits a series of cloud tasks to a cloud data center, in order to meet user needs or reasonably utilize resources, the scheduling system will establish a virtual machine according to the cloud task queue and map the cloud task to the virtual machine. During the execution of the task, virtual machines are migrated due to performance improvements and system requirements, until the completion of all cloud tasks to complete the destruction of the virtual machine. The mapping relationship established between cloud tasks, virtual machines and physical machines is shown in Fig. 1.

3.2. Cloud data center energy consumption model

It is generally believed that the energy consumption of cloud data centers is mainly composed of energy consumption of physical machines and energy consumption of cooling. Because the energy consumption of physical machines accounts for the largest proportion of the total energy consumption, and other energy consumption such as cooling often has a fixed linear proportional relationship with the energy consumption of physical machines. Therefore, the energy consumption of physical machines can reflect the energy consumption of cloud data centers. How to reduce the energy consumption of physical machines has become the main goal of this paper. This article assumes that the relationship between the total energy consumption of cloud data center and the energy consumption of a physical machines is as follows:

$$E^{Total} = \mathrm{S} \times E^{PM} \tag{1}$$

S represents a constant. In general, the number of virtual machines purchased by cloud users is far less than the number of cloud tasks. Therefore, the number of physical machines provided by cloud providers for establishing virtual machines is relatively small. Assuming that the number of physical machines is N, the energy consumption of the physical machines is:

$$E^{PM} = \sum_{\mathrm{i}=1}^{N} E^{PM_\mathrm{i}} \tag{2}$$

This article only considers the power consumption of the CPU and memory that have a large impact on power consumption. According to the conclusions in [45–47], the power of a physical machine is linearly or positively related to the CPU utilization of the physical machine. The higher CPU utilization means the greater power. At the same time, [48] indicates that the power of a physical machine is equivalent to about 70% of the full load in the idle state. Generally, the utilization of physical machine tends to a fixed value of a cloud data center, and $\mathrm{CPU}_{\mathrm{PMi}}$ is used to indicate the CPU utilization of the numbered i physical machine when it is fully loaded. The power consumption of the memory is also related to the CPU utilization. When the CPU utilization is increased, the power consumption of the memory will also increase. Use $\mathrm{MEM}_{\mathrm{PMi}}$ to represent the power consumption of the memory when the numbered i physical machine is full. In this paper, the constant K_1 is used to represent the relationship

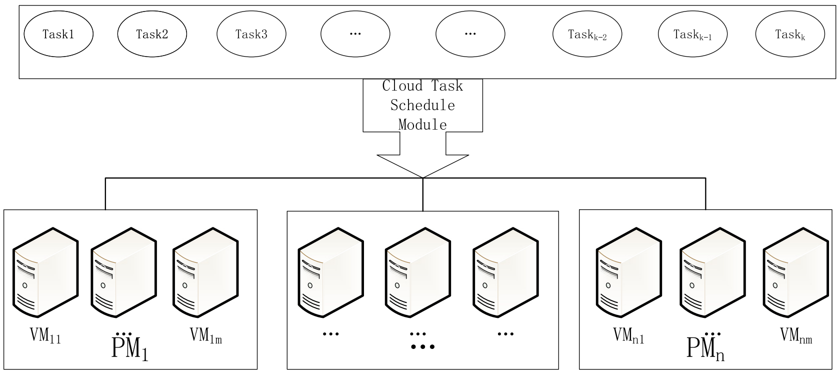

Fig. 1. Mapping relationship between cloud tasks, virtual machines and physical machines.

between the power consumption of physical machine and CPU utilization. The constant K_2 is used to represent the relationship between the power consumption of physical machine and memory. Therefore, the energy consumption and power consumption of the numbered i physical machine are as follows:

$$E^{PM_i} = T_i \times P^{PM_i} \tag{3}$$

$$P^{PM_i} = K_1 \times CPU_{PM_i} + K_2 \times MEM_{PM_i} \tag{4}$$

According to the above analysis, the energy consumption of the cloud data center can be expressed as follows:

$$E^{Total} = S \times N \times \sum_{i=1}^{N} \left(T_i \times \left(K_1 \times CPU_{PM_i} + K_2 \times CPU_{PM_i} \right) \right) \tag{5}$$

S, N, K_1, K_2, CPU_{PMi} and MEM_{PMi} are all constant or fixed value. Therefore, the energy consumption of a cloud data center is mainly determined by the total running time or average running time of the physical machines. The main way to reduce the energy consumption of cloud data centers is to change the mapping of cloud tasks and the deployment of virtual machines under a certain number of cloud tasks and physical machines.

3.3. Memory priority cloud task mapping rule

The scheduling algorithm of cloud data center mainly solves two problems. On the one hand, the first is the establishment and deployment of virtual machines. Different virtual machines have different CPU and memory parameters, and they also show different performance. Usually large and mature cloud service providers have standard virtual machine types for users to choose. The number ratio of CPU cores and memory provided by Amazon is generally 1: 2, that is, one CPU core is equipped with 2 GB memory. With different required virtual machine performance, the number of CPU cores will gradually increase, but the ratio is usually fixed. The second is to complete the mapping of the cloud task on the virtual machine. The scheduling system will schedule the appropriate virtual machine to perform the task according to the different attributes and requirements of the cloud task. The mapping of a cloud task requires that the virtual machine can simultaneously meet the parameters of the cloud task such as CPU and memory. Cloud users usually select the virtual machine with the shortest running time for mapping when the budget is sufficient, that is, a virtual machine with higher CPU performance. Although this improves the running time of some cloud tasks, it will eventually result in a longer running time due to lower utilization of other resources such as memory, and also increase the energy consumption of the physical machine. On the other hand, as the complexity of cloud tasks continues to increase, the fee of cloud users will continue to increase. When users submit tasks, cloud users will also make rational choices on the services provided by cloud computing providers. Only a reasonably priced and highly efficient cloud service can guarantee to stand out in the market competition. Therefore, cloud service providers must not only consider the energy consumption of cloud data centers, but also optimize the fee of cloud users. The goal of optimization in this article is to reduce the total fee of cloud users and the energy consumption of cloud data centers through the memory priority approach when the number of cloud tasks and physical machines is constant. However, the situations that need to be considered are often more complicated in the scheduling of cloud computing. First, all the physical machines are heterogeneous in cloud data centers. You must comprehensively consider the differences in physical machines in order to really increase utilization when deploying virtual machines. Second, because the number of CPU cores and memory size are usually configured proportionally, memory utilization and task completion time must be fully considered when mapping cloud tasks. Based on the above considerations, when researching the mapping of cloud tasks, this paper proposes a memory priority cloud task mapping rule. It analyzes homogeneous cloud data centers, and then extends the rule to heterogeneous cloud data centers.

Rule 1. Memory priority cloud task mapping rule. When performing cloud task mapping, in order to improve memory utilization, we must select or establish a virtual machine with the most suitable memory capacity and the least impact on the physical machine runtime in a homogeneous cloud data center. First, select a virtual machine from the established virtual machines with suitable memory and do not extend the running time of the physical machine. Secondly, a new virtual machine with suitable memory is selected for deployment in the free resources of the physical machine. Finally, select a virtual machine from the established virtual machines with suitable memory and will extend the shortest running time of the physical machine. Each time the mapping of the cloud task is completed, the virtual machine and physical machine parameters are updated until the mapping of all cloud tasks is completed.

We first verify the effectiveness of this rule in a homogeneous cloud data center. Assume that the cloud data center provides two physical machines $PM = \{pm_1, pm_2\}$. Where $pm_1 = \{8, 16, 100, 0.35, 0\}$ and $pm_2 = \{8, 16, 100, 0.35, 0\}$ respectively represent the number of CPU cores, memory size (GB), CPU core processing speed (MIPS), CPU core usage price (\$/h), and physical machine running time (s). The parameters of the two physical machines are the same, they are homogeneous cloud data centers. The cloud task submitted by the user is $CT = \{ct_1, ct_2, ct_3, ct_4, ct_5\}$. Where $ct_1 = \{300\,000, 4, -1\}$, $ct_2 = \{400\,000, 8, -1\}$, $ct_3 = \{200\,000, 2, -1\}$, $ct_4 = \{500\,000, 4, -1\}$ and $ct_5 = \{500\,000, 8, -1\}$ respectively represent the length of the cloud task (MI), memory size (GB) and the number of the virtual machine executing the cloud task. In the initial state, the running time of the physical machine is 0, and the number of the virtual machine executing the cloud task is preset to -1. The specific situation is shown in Fig. 2. Each small cell represents 1 GB memory in the figure. Because the ratio of the virtual machine CPU core number and memory size is fixed at 1: 2. So for convenience, the number of CPU cores is not listed here. The initial situation is shown in the first area. First, no virtual machines have been created in mapping ct_1. Therefore, a virtual machine vm_{11} was created on pm_1 in order. In order to meet the memory requirements of ct_1 and effectively use the memory of the physical machine, the memory of vm_{11} is set to 4 GB. Because the ratio of the number of virtual machine CPU cores and memory sizes is generally 1: 2, two CPU cores are required. At the same time, according to the task length of ct_1, the running time of the task is 1500 s, the final $vm_{11} = \{2, 4, 100, 0.35, 1, 1500\}$, $pm_1 = \{8, 16, 100, 0.35, 1500\}$. The virtual machine mapping results are shown in the second area. Secondly, it requires 8 GB memory in mapping ct_2. At this time, there is no virtual machine with a memory greater than or equal to 8 GB, so the virtual machine vm_{12} is continuously created on pm_1 in order. In order to meet the memory requirements of ct_2 and effectively use the memory of the physical machine, the memory of vm_{12} is set to 8 GB, the CPU is 4 cores, and $vm_{12} = \{4, 8, 100, 0.35, 1, 1000\}$. Because the running time of vm_{12} is less than pm_1, the parameters of pm_1 are unchanged. The virtual machine mapping results are shown in the third area. The situation of ct_3 mapping is similar to ct_2. A virtual machine of vm_{13} will be created on pm_1, $vm_{13} = \{1, 2, 100, 0.35, 1, 2000\}$. Because the running time of vm_{13} is greater than pm_1, the parameters of pm_1 are changed to $pm_1 = \{8, 16, 100, 0.35, 2000\}$. The virtual machine mapping results are shown in the fourth area. Once again, when mapping ct_4, we find vm_{11} can meet the memory requirements of ct_4. However, if ct_4 is mapped to vm_{11}, the running time of vm_{11} will be extended to 2500 s, which exceeding the current running time of pm_1 by 2000 s. As a result, the running time of pm_1 is deferred and energy consumption is increased. Therefore, a new virtual machine needs to be created for the mapping of ct_4. Because the remaining memory of pm_1 cannot meet the requirements of the virtual machine, the deployment of the virtual machine is completed on pm_2 with the parameter $vm_{21} = \{2, 4, 100, 0.35, 2, 2500\}$ and $pm_2 = \{8, 16, 100, 0.35, 2500\}$. The virtual machine mapping results are shown in the fifth area. Finally, if vm_{12} is selected when performing ct_5 mapping, the running time of the physical machine will be extended. Therefore, the scheduling system tries to find a free physical machine space to deploy a new virtual machine. Because the remaining memory of pm_1 cannot meet the requirements of the virtual machine, the deployment of the virtual machine is completed on pm_2 with the parameter $vm_{22} = \{4, 8, 100, 0.35, 2, 1250\}$. Because the running time of vm_{22} is less than pm_2, the parameters of pm_2 are unchanged. The virtual machine mapping results are shown in the sixth area. This completes the mapping of all cloud tasks. The entire mapping process and the final parameters of the virtual machines and physical machines are shown in Table 1.

As mentioned in the previous description, the internal physical machines will not all have the same configuration when deploying modern cloud data centers. This is almost impossible to achieve in reality. At the same time, this goes against the original intention of cloud computing. However, this rule guarantees that the sum of all virtual machine memory and other resources deployed on the same physical machine will not exceed the capacity of the physical machine when performing cloud task mapping. Therefore, this rule should still apply to heterogeneous cloud data centers. Then we will verify the heterogeneous cloud data center.

Assume that the cloud data center provides two physical machines $PM = \{pm_1, pm_2\}$. Where $pm_1 = \{8, 16, 100, 0.35, 0\}$ and $pm_2 = \{2, 4, 200, 1.00, 0\}$ respectively represent the number of CPU cores, memory size (GB), CPU core processing speed (MIPS), CPU core usage price (\$/h), and physical machine running time (s). The cloud task submitted by the user is $CT = \{ct_1, ct_2, ct_3, ct_4, ct_5\}$. Where $ct_1 = \{300\,000, 4, -1\}$, $ct_2 = \{400\,000, 8, -1\}$, $ct_3 = \{200\,000, 2, -1\}$, $ct_4 = \{500\,000, 4, -1\}$ and $ct_5 = \{500\,000, 8, -1\}$ respectively represent the length of the cloud task (MI), memory size (GB) and the number of the virtual machine executing the cloud task. In the initial state, the running time of the physical machine is 0, and the number of the virtual machine executing the cloud task is preset to -1. The specific situation is shown in Fig. 3. Each small cell represents 1 GB memory in the figure. Because the ratio of the virtual machine CPU core number and memory size is fixed at 1: 2. So for convenience, the number of CPU cores is not listed here. The initial situation is shown in the first area. First, no virtual machines have been created in mapping ct_1. Therefore, a virtual machine vm_{11} was created on pm_1 in order. In order to meet the memory requirements of ct_1 and effectively use the memory of the physical machine, the memory of vm_{11} is set to 4 GB. Because the ratio of the number of virtual machine CPU cores and memory sizes is generally 1: 2, two CPU cores are required. At the same time, according to the task length of ct_1, the running time of the task is 1500 s, the final $vm_{11} = \{2, 4, 100, 0.35, 1, 1500\}$, $pm_1 = \{8, 16, 100, 0.35, 1500\}$. The virtual machine mapping results are shown in the second area. Secondly, it requires 8 GB memory in mapping ct_2. At this time, there is no virtual machine with a memory greater than or equal to 8 GB, so the virtual machine vm_{12} is continuously created on pm_1 in order. In order to meet the memory requirements of ct_2 and effectively use the memory of the physical machine, the memory of vm_{12} is set to 8 GB, the CPU is 4 cores, and $vm_{12} = \{4, 8, 100, 0.35, 1, 1000\}$. Because the running time of vm_{12} is less than pm_1, the parameters of pm_1 are unchanged. The virtual machine mapping results are shown in the third area. The situation of ct_3 mapping is similar to ct_2. A virtual machine of vm_{13} will be created on pm_1, $vm_{13} = \{1, 2, 100, 0.35, 1, 2000\}$. Because the running time of vm_{13} is greater than pm_1, the parameters of pm_1 are changed to $pm_1 = \{8, 16, 100, 0.35, 2000\}$. The virtual machine mapping results are shown in the fourth area. Once again, when mapping ct_4, we find vm_{11} can meet the memory requirements of ct_4. However, if ct_4 is mapped to vm_{11}, the running time of vm_{11} will be extended to 2500 s, which exceeding the current running time of pm_1 by 2000 s. As a result, the running time of pm_1 is deferred and energy consumption is increased. Therefore, a new virtual machine needs to be created for the mapping of ct_4. Because the remaining memory of pm_1 cannot meet the requirements of the virtual machine, the deployment of the virtual machine is completed on pm_2 with the parameter $vm_{21} = \{2, 4, 100, 0.35, 2, 2500\}$ and $pm_2 = \{8, 16, 100, 0.35, 2500\}$. The virtual machine mapping results are shown in the fifth area. Finally, if vm_{12} is selected when performing ct_5 mapping, the running time of the physical machine will be extended. Therefore, the scheduling system tries to find a free physical machine space to deploy a new virtual machine. However, it is found that the remaining space of pm_1 and pm_2 cannot meet the memory requirements of ct_5. The ct_5 can only select virtual machines that extend the running time of the physical machine for mapping. At this time, we can choose the physical machine with the shortest running time for mapping. In this example, only vm_{12} can meet the memory requirements. So after mapping ct_5 to vm_{12}, $vm_{12} = \{4, 8, 100, 0.35, 1, 2250\}$ and $pm_1 = \{8, 16, 100, 0.35, 2250\}$. The virtual machine mapping results are shown in the sixth area. This completes the mapping of all cloud tasks. The entire mapping process and the final parameters of the virtual machines and physical machines are shown in Table 2.

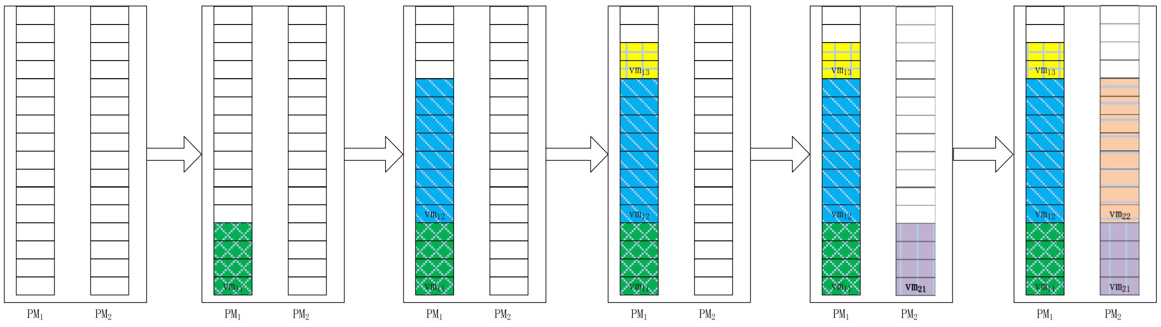

Fig. 2. Virtual machines deployment in homogeneous cloud data center example.

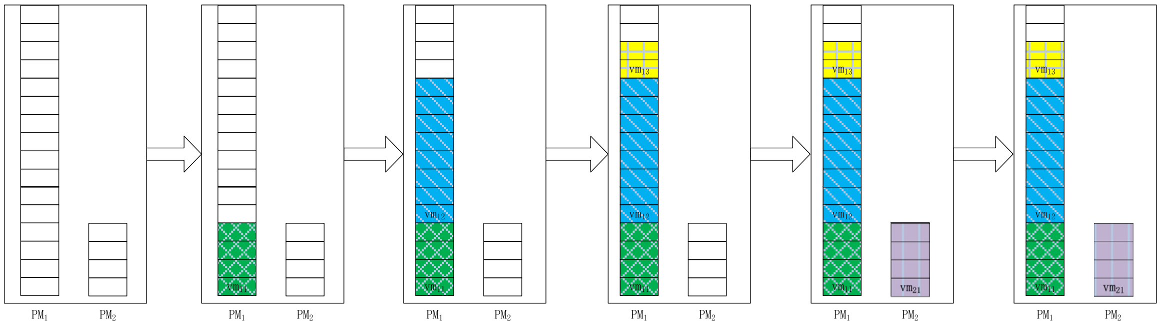

Fig. 3. Virtual machines deployment in heterogeneous cloud data center example.

Table 1
Parameters of virtual machines and physical machines in homogeneous cloud data center example.

	pm_1	pm_2	vm_{11}	vm_{12}	vm_{13}	vm_{21}	vm_{22}
Initial situation	{8,16,100 ,0.35,0}	{8,16,100 ,0.35,0}					
Mapping ct_1	{8,16,100 ,0.35,1500}	{8,16,100 ,0.35,0}	{2,4,100,0. 35,1,1500}				
Mapping ct_2	{8,16,100 ,0.35,1500}	{8,16,100 ,0.35,0}	{2,4,100,0. 35,1,1500}	{4,8,100,0. 35,1,1000}			
Mapping ct_3	{8,16,100 ,0.35,2000}	{8,16,100 ,0.35,0}	{2,4,100,0. 35,1,1500}	{4,8,100,0. 35,1,1000}	{1,2,100,0. 35,1,2000}		
Mapping ct_4	{8,16,100 ,0.35,2000}	{8,16,100 ,0.35,2500}	{2,4,100,0. 35,1,1500}	{4,8,100,0. 35,1,1000}	{1,2,100,0. 35,1,2000}	{2,4,100,0. 35,2,2500}	
Mapping ct_5	{8,16,100 ,0.35,2000}	{8,16,100 ,0.35,2500}	{2,4,100,0. 35,1,1500}	{4,8,100,0. 35,1,1000}	{1,2,100,0. 35,1,2000}	{2,4,100,0. 35,2,2500}	{4,8,100,0. 35,2,1250}

Table 2
Parameters of virtual machines and physical machines in heterogeneous cloud data center example.

	pm_1	pm_2	vm_{11}	vm_{12}	vm_{13}	vm_{21}
Initial situation	{8,16,100 ,0.35,0}	{2,4,200 ,1.00,0}				
Mapping ct_1	{8,16,100 ,0.35,1500}	{2,4,200 ,1.00,0}	{2,4,100,0.35 ,1,1500}			
Mapping ct_2	{8,16,100 ,0.35,1500}	{2,4,200 ,1.00,0}	{2,4,100,0.35 ,1,1500}	{4,8,100,0.35 ,1,1000}		
Mapping ct_3	{8,16,100 ,0.35,2000}	{2,4,200 ,1.00,0}	{2,4,100,0.35 ,1,1500}	{4,8,100,0.35 ,1,1000}	{1,2,100,0.35 ,1,2000}	
Mapping ct_4	{8,16,100 ,0.35,2000}	{2,4,200 ,1.00,1250}	{2,4,100,0.35 ,1,1500}	{4,8,100,0.35 ,1,1000}	{1,2,100,0.35 ,1,2000}	{2,4,200,1.00 ,2,1250}
Mapping ct_5	{8,16,100 ,0.35,2250}	{2,4,200 ,1.00,1250}	{2,4,100,0.35 ,1,1500}	{4,8,100,0.35 ,1,2250}	{4,8,100,0.35 ,1,2000}	{2,4,200,1.00 ,2,1250}

Combined with the example analysis of memory priority cloud task mapping rule, memory utilization can be guaranteed by readjusting the mapping of cloud tasks and the order in which virtual machines are created. Homogeneous cloud data centers are just a special case of heterogeneous cloud data centers. Therefore, the mapping rule can also be used to eventually improve the mapping efficiency. The core algorithm MALE of this article will be introduced in the next section.

4. Male algorithm

From the previous analysis, it can be concluded that no matter what kind of cloud data center is, it is necessary to reduce the energy consumption of the cloud data center by reducing the running time of the physical machine under the premise of meeting the memory utilization. Based on this idea, we propose a memory-aware resource management algorithm for low-energy cloud data centers (MALE). The algorithm uses memory priority cloud task mapping rule and reduces the average execution time of physical machines to reduce the total energy consumption of cloud data centers.

According to the previous conclusion, the MALE algorithm as shown in Algorithm 1, mainly includes the following seven steps:
Step 1: Analyze cloud tasks submitted by cloud users and form a cloud task mapping queue.
Step 2: Select the cloud task from the cloud task queue in order to map, and traverse the established virtual machine queue according to the requirements of the cloud task. Calculate the virtual machines whose memory meets the requirements of the cloud task to determine whether there are virtual machines that will not extend the running time of the physical machine after the cloud task is mapped. If it exists, go to Step 3, otherwise Step 4.
Step 3: Sort the data in ascending order based on the absolute value of the difference between the running time of the virtual machine and the physical machine. After that, select the first virtual machine for mapping, and then turn to Step 7.
Step 4: Traverse the physical machine according to the requirements of the cloud task. Determine whether there is space to create a new virtual machine that meets the cloud task. If it exists, go to Step 5, otherwise Step 6.
Step 5: Create a new virtual machine on the physical machine that meets the cloud task. After that, the cloud task is mapped to the virtual machine, and the virtual machine is added to the established virtual machine queue, and then goes to Step 7.
Step 6: Traverse the established virtual machine queue according to the requirements of the cloud task again. Calculate the virtual machines whose memory meets the requirements of the cloud task, and sort them in descending order according to the absolute value of the difference between the running time of the virtual machine and the physical machine. Select the first virtual machine for mapping.
Step 7: Remove the mapped cloud task from the queue. Determine whether there is an unmapped cloud task. If it exists, repeat Step 2. Otherwise, end the entire scheduling algorithm.

The algorithm first selects the virtual machine with suitable memory from the established virtual machines and does not extend the running time of the physical machine to be mapped (line 1–line 2). Second, the algorithm selects a new virtual machine with suitable memory for deployment in the free resources of the physical machine (line 3–line 4). Finally, the algorithm selects the virtual machine with suitable memory from the established virtual machines and the shortest running time of the physical machine to be mapped (line 5–line 7). Finally, complete the mapping of all cloud tasks.

The line 2 of the Algorithm 1 selects a virtual machine with suitable memory from the established virtual machines and does not extend the running time of the physical machine to perform mapping, as shown in Algorithm 2. The algorithm traverses the established virtual machines in turn. If a virtual machine with suitable memory is found and does not extend the running time of the physical machine, the cloud task is mapped.

The line 4 of Algorithm 1 creates a new virtual machine with suitable memory in the free resources of the physical machine for deployment, as shown in Algorithm 3. The algorithm traverses the physical machine in turn. If the physical machine memory is found to be sufficient, a new virtual machine is mapped for cloud tasks.

The line 6 of Algorithm 1 selects the virtual machine with suitable memory from the established virtual machines and will extend the shortest running time of the physical machine where it is located to perform the mapping as shown in algorithm 4. The algorithm traverses the virtual machines that have been created. If two mapping matrices are used at the same time, the virtual machine with the shortest running time of the extended physical machine is found to perform mapping.

Table 3
Experimental parameters.

Parameters	Value
Length of the cloud task	100 000–1 000 000 (MI)
Memory size of the cloud task	{2 GB, 4 GB, 6 GB, 8 GB}
CPU cores of the physical machine	{2, 4, 8, 16}
Memory size of the physical machine	{4 GB, 8 GB, 16 GB, 32 GB}
CPU core processing speed of the physical machine	100–200 (MIPS)
CPU core usage price of the physical machine	0.35–1.00 ($/h)

It can be seen from the above steps that the MALE algorithm will preferentially select the established virtual machine without extending the running time of the physical machine. This can reduce the execution time of physical machines, thereby further reducing the energy consumption of cloud data centers. At the same time, because it can be applied to homogeneous and heterogeneous cloud data centers. In the next section we will further verify its efficiency through experiments.

5. Test results

In order to verify the effect of the MALE algorithm in this section, the algorithm is implemented using CloudSim. We have embedded a simulation verification platform in CloudSim. At the same time, we have implemented the Min-Min algorithm [49], Sufferage algorithm [50] and E-HEFT algorithm [32] in this simulation verification platform. Later, a comparison module was added to the platform, which can compare the energy consumption of cloud data centers between different mapping algorithms, and the output result is the ratio between the energy consumption of the two algorithms. At the same time, reference the methods in [11,51,52] to generate the corresponding cloud task parameters and physical machine parameters for simulation. As shown in Table 3, we refer to the virtual machine examples provided by Amazon when creating virtual machines. The ratio of the number of CPU cores and the size of the memory is generally 1: 2. The MALE algorithm is verified by a simulation verification platform. In this section, the total fee of the cloud users, the average running time of physical machines, and the energy consumption ratio of the cloud data center will be verified from the number of physical machines and the number of cloud tasks. The total fee of cloud users subscribers is as follows:

$$P^{Total} = \sum_{j=1}^{M} \left(c_j \times s_j \times t_j\right) / 3600 \tag{6}$$

Among them, P^{Total} represents the total fee of cloud users. M is the number of created virtual machines. The c_j, s_j and t_j represent the number of CPU cores, CPU core usage price ($/h) and the virtual machine running time (s).

The average running time of a physical machines is as follows:

$$T^{Average} = \sum_{i=1}^{N} t_i / N \tag{7}$$

Among them, $T^{Average}$ represents the average running time of the activated physical machines in the cloud data center. The N is the number of activated physical machines in the cloud data center. The t_i represents the running time of the numbered i physical machine.

Algorithm 1: (MALE)

```
Input: Cloud Task collection: CT={ct_1, ct_2, ..., ct_t},
       PM collection: PM={pm_1, pm_2, ···, pm_n},
       Cloud Task mapping: A=(a_ij)_t*m,
       VM mapping: E=(e_jk)_m*n
Output: VM collection: VM={vm_1, vm_2, ···, vm_m},
        Cloud Task collection: CT={ct_1, ct_2, ..., ct_t},
        PM collection: PM={pm_1, pm_2, ···, pm_n},
        Cloud Task mapping: A=(a_ij)_t*m,
        VM mapping: E=(e_jk)_m*n
1  while ct_i of CT not null do
2    ct_i.n=VMSelectWithoutDelay(VM, PM, ct_i, A=(a_ij)_m*n, E=(e_jk)_m*n)
3    if (ct_i.n==-1)
4      ct_i.n=VMSetup(VM, PM, ct_i, A=(a_ij)_m*n, E=(e_jk)_m*n)
5      if (ct_i.n==-1)
6        ct_i.n=VMSelectWithDelay(VM, PM, ct_i, A=(a_ij)_m*n, E=(e_jk)_m*n)
7      end if
8    end if
9  end while
```

Algorithm 2: (VMSelectWithoutDelay)

```
Input: Cloud Task i:ct_i,
       VM collection: VM={vm_1, vm_2, ···, vm_m},
       PM collection: PM={pm_1, pm_2, ···, pm_n},
       Cloud Task mapping: A=(a_ij)_t*m,
       VM mapping: E=(e_jk)_m*n
Output: VM Number of Cloud Task i: ct_i.n
1  while vm_j of VM not null do
2    if (vm_j.m==ct_i.m)
3      pm_k base on E=(e_jk)_m*n
4      if (pm_k.t>( vm_j.t+ ct_i.l/(pm_k.s* pm_k.m/2)))
5        Update A=(a_ij)_t*m
6        Update E=(e_jk)_m*n
7        Update vm_j
8        ct_i.n=j
9      end if
10   end if
11 end while
```

Energy consumption ratio of cloud data center is as follows:

$$S^E = \sum_{i=1}^{N_i} (t_i \times p_i) \Big/ \sum_{j=1}^{N_j} (t_j \times p_j) \tag{8}$$

Among them, S^E represents the energy consumption ratio of the cloud data center of the two scheduling algorithms. The t_i, p_i and N_i represent the number of activated physical machines, the running time of the numbered i physical machine and power of the first algorithm in the cloud data center. The t_j, p_j and N_j represent the number of activated physical machines, the running time of the numbered j physical machine and power of the second algorithm in the cloud data center.

5.1. Different number of physical machines

The general data center contains tens of thousands of physical machines. For example, Amazon's AWS service has 11 regional clouds in the world. Each region has multiple groups of data centers. Each group has at least one data center. Each data center

```
Algorithm 3: (VMSetup)
Input: Cloud Task i:$ct_i$,
       VM collection: VM={$vm_1, vm_2, \cdots, vm_m$},
       PM collection: PM={$pm_1, pm_2, \cdots, pm_n$},
       Cloud Task mapping: A=$(a_{ij})_{t*m}$,
       VM mapping: E=$(e_{jk})_{m*n}$
Output: VM Number of Cloud Task i: $ct_i$.n
1  while $pm_k$ of PM not null do
2    while $vm_j$ of VM not null do
3      if ($vm_j$.n==k)
4        $pm_k$.used+=$vm_j$.m
5      end if
6    end while
7    if ($pm_k$- $pm_k$.used> $ct_i$.m)
8      Update A=$(a_{ij})_{t*m}$
9      Update E=$(e_{jk})_{m*n}$
10     Update VM
11     Update PM
12     Update $ct_i$.n=j
13   end if
14 end while
```

```
Algorithm 4: (VMSelectWithDelay)
Input: Cloud Task i:$ct_i$,
       VM collection: VM={$vm_1, vm_2, \cdots, vm_m$},
       PM collection: PM={$pm_1, pm_2, \cdots, pm_n$},
       Cloud Task mapping: A=$(a_{ij})_{t*m}$,
       VM mapping: E=$(e_{jk})_{m*n}$
Output: VM Number of Cloud Task i: $ct_i$.n
1  Mindelay=0
2  NumberMindelay=0
3  while $vm_j$ of VM not null do
4    if ($vm_j$.m==$ct_i$.m)
5      $pm_k$ base on E=$(e_{jk})_{m*n}$
6      if (($pm_k$.t-( $vm_j$.t+ $ct_i$.l/($pm_k$.s* $pm_k$.m/2)))> Mindelay)
7        NumberMindelay=j
8      end if
9    end if
10 end while
11 Update A=$(a_{ij})_{t*m}$
12 Update E=$(e_{jk})_{m*n}$
13 Update $pm_k$
14 Update $vm_j$
15 $ct_i$.n=NumberMindelay
```

has 50 000 to 80 000 servers. However, in order to complete a set of tasks, the cloud data center usually only allocates some physical machines for the deployment of virtual machines. In this experiment, it is assumed that the ten number of physical

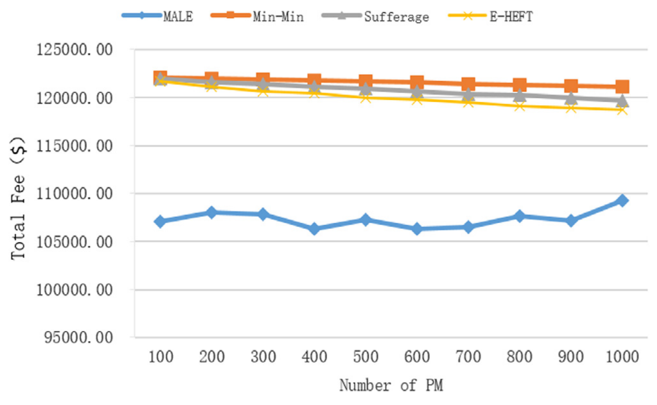

Fig. 4(A). Total fee of cloud users on different number of physical machines.

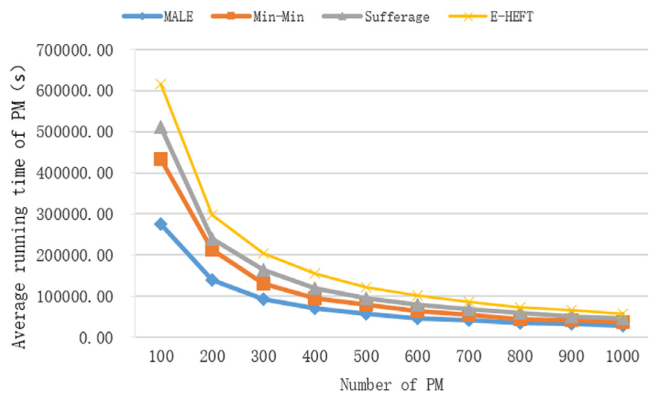

Fig. 4(B). Average running time of PM on different number of physical machines.

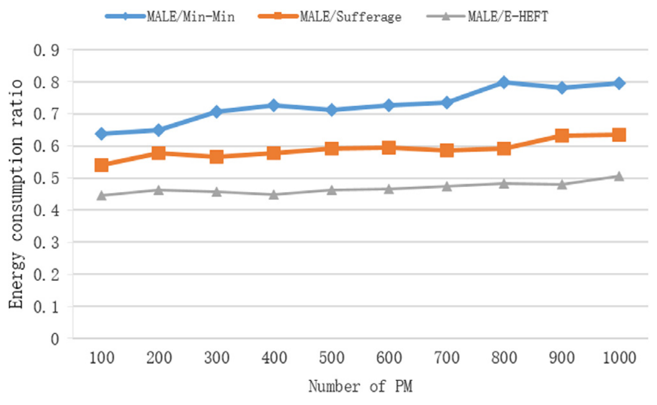

Fig. 4(C). Energy consumption ratio on different number of physical machines.

machines are average selected from 100 to 1000. In order to fully verify the effect of the algorithm on memory utilization, the number of cloud tasks is 160 000, and other parameters are shown in Table 2. This experiment verifies that the total fee of cloud users, the average running time of physical machines, and the energy consumption ratio of cloud data centers on different number of physical machines.

(1) Homogeneous cloud data center

It can be seen from Fig. 4(A) that the total fee of cloud users for the four algorithms does not change much with the number of physical machines, but the total fee of cloud users for MALE algorithm is different from the Min-Min algorithm, Sufferage algorithm and E-HEFT algorithm. The average reductions were 12%, 11% and 11%. This is mainly because the MALE algorithm maps cloud tasks with as few CPU cores as possible while ensuring memory requirements. This improves the utilization efficiency of the physical machine and reduces unnecessary memory waste. As a result, more virtual machines can be created for mapping, which ultimately reduces the total fee of cloud users. It can be seen from Fig. 4(B) that when the number of physical machines is small, the average running time of physical machines of MALE is less than Min-Min algorithm, Sufferage algorithm and E-HEFT algorithm. However, with the increase of the number of physical machines, the average running time of the four algorithms gradually decreases and the gap gradually decreases. This is mainly because the larger the number of physical machines, the more virtual machines will be created. Cloud tasks are more likely to be mapped to the appropriate virtual machine, which further reduces the waste of physical machine memory. As can be seen from Fig. 4(C), the energy consumption ratio of the cloud data center for the MALE algorithm and the other three algorithms does not change much with the number of physical machines. However, compared to Min-Min algorithm, Sufferage algorithm and E-HEFT algorithm, the cloud data center energy consumption of MALE algorithm has been reduced by 27%, 41% and 53% on average. This is mainly because the total energy consumption of a cloud data center is determined by the sum of the energy consumption of all physical machines. In the case of equal numbers of physical machines, it is determined by the average running time of the physical machines. The change in the average running time of the physical machines with the number of physical machines has been given in Fig. 4(B). The reason is not repeated.

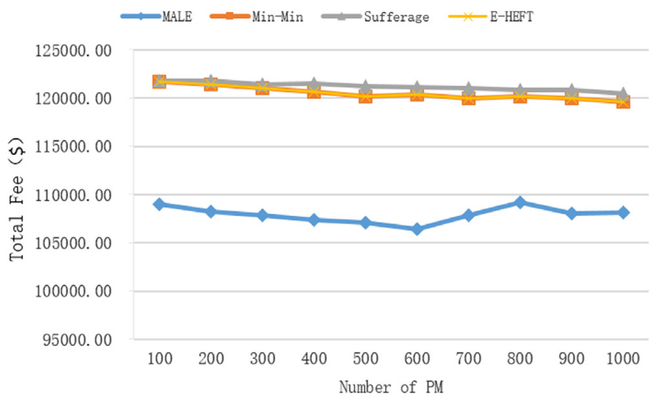

Fig. 5(A). Total fee of cloud users on different number of physical machines.

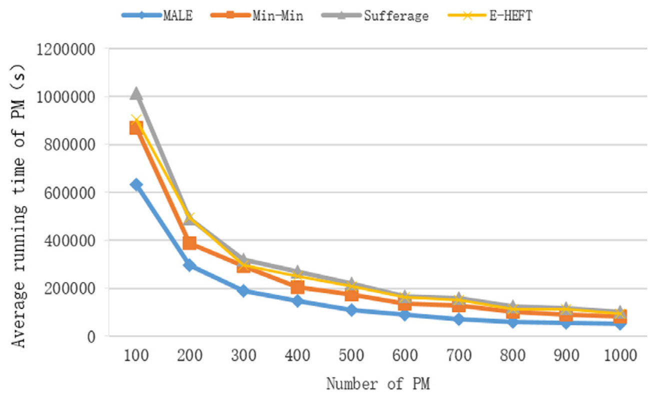

Fig. 5(B). Average running time of PM on different number of physical machines.

(2) Heterogeneous cloud data center

It can be seen from Fig. 5 that compared with the Min-Min algorithm, Sufferage algorithm and E-HEFT algorithm, the total

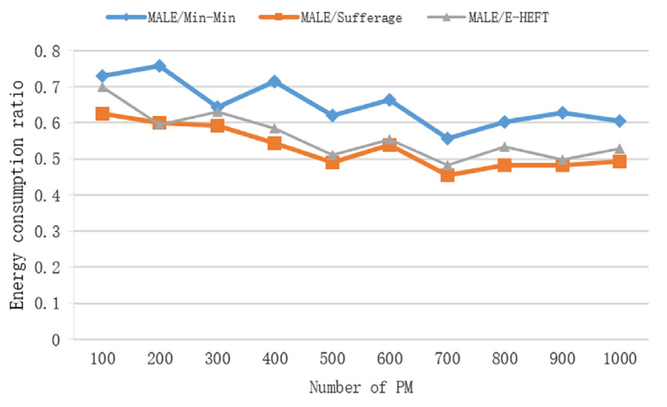

Fig. 5(C). Energy consumption ratio on different number of physical machines.

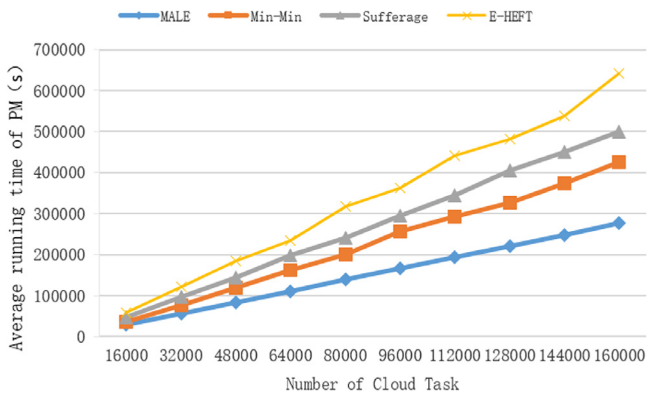

Fig. 6(B). Average running time of PM on different number of cloud tasks.

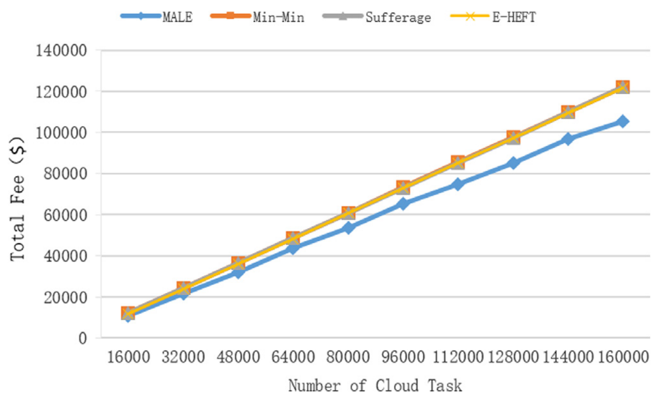

Fig. 6(A). Total fee of cloud users on different number of cloud tasks.

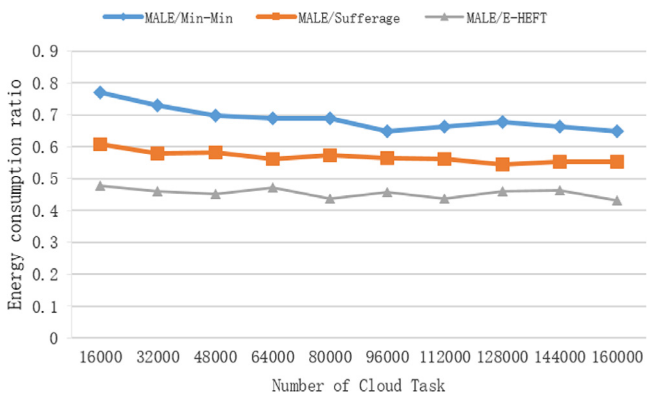

Fig. 6(C). Energy consumption ratio on different number of cloud tasks.

fee of cloud users of MALE algorithm has been reduced by an average of 10%, 11% and 10% in heterogeneous cloud data centers. Compared with Min-Min algorithm, Sufferage algorithm and E-HEFT algorithm, cloud data center energy consumption of MALE algorithm has been reduced by 35%, 47% and 44% on average. The main reason is that the algorithm used memory priority cloud task mapping rule at the beginning of design to fully consider the heterogeneity of physical machines, and performed reasonable processing during cloud task deployment. This not only improves the utilization efficiency of the physical machines, but also reduces unnecessary memory waste. It can be seen that a special case of a homogeneous cloud data center as a heterogeneous cloud data center, which can be handled more conveniently.

5.2. Different number of cloud tasks

For the same reason, the number of physical machines is taken as 100, and the number of cloud tasks is selected on average between 16 000 and 160 000. This experiment verifies that the total fee of cloud users, the average running time of physical machines, and the energy consumption ratio of cloud data centers on different number of cloud tasks.

(1) Homogeneous cloud data center

It can be seen from Fig. 6(A) that the total fee of cloud users for the four algorithms increases sharply with the number of cloud tasks, because the increase in the number of cloud tasks will inevitably increase the total fee of cloud users. However, the total fee of cloud users for MALE algorithm is different from the Min-Min algorithm, Sufferage algorithm and E-HEFT algorithm. The average reductions were 11%, 12% and 11%. This is mainly because the MALE algorithm maps cloud tasks with as few CPU cores as possible while ensuring memory requirements. This improves the utilization efficiency of the physical machine and reduces unnecessary memory waste. As a result, more virtual machines can be created for mapping, which ultimately reduces the total fee of cloud users. It can be seen from Fig. 6(B) that the average running time of the physical machines of the four algorithms also increases dramatically with the number of cloud tasks. The reason is also that the increase in the number of cloud tasks will inevitably increase the average running time of the physical machines. It can be seen from Fig. 6(C) that the trend of the energy consumption ratio of the cloud data center for the MALE algorithm and the other three algorithms with the number of cloud tasks is similar to that shown in Fig. 4(C). However, compared to the Min-Min algorithm, Sufferage algorithm and E-HEFT algorithm, the energy consumption of MALE algorithm has been reduced by 31%, 43% and 55%, respectively. This is mainly because with the increase in the number of cloud tasks, cloud tasks with various needs can more effectively use the space of physical machines. The reason is not repeated.

(2) Heterogeneous cloud data center

As can be seen from Fig. 7, compared to the Min-Min algorithm, Sufferage algorithm and E-HEFT algorithm, the total fee of cloud users for MALE algorithm has been reduced by 12%, 11% and 11%, respectively. Compared with the Min-Min algorithm, Sufferage algorithm and E-HEFT algorithm, energy consumption of MALE algorithm has been reduced by 11%, 30% and 38% on average. The main reason is that the algorithm uses memory priority cloud task mapping rule at the beginning of design to fully consider the heterogeneity of the physical machine. When the cloud task is deployed, the heterogeneity of the physical machine can be fully utilized to be more effective Space for physical machines.

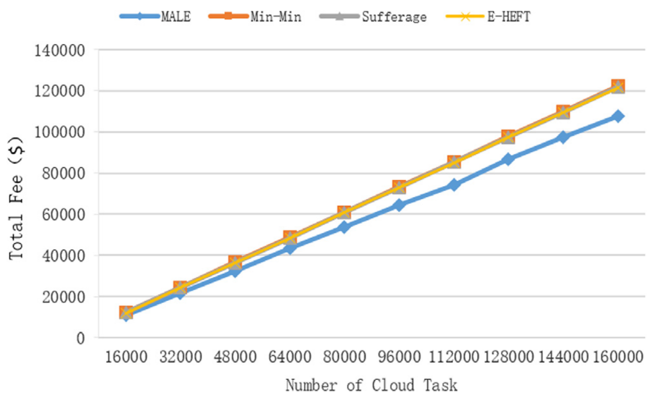

Fig. 7(A). Total fee of cloud users on different number of cloud tasks.

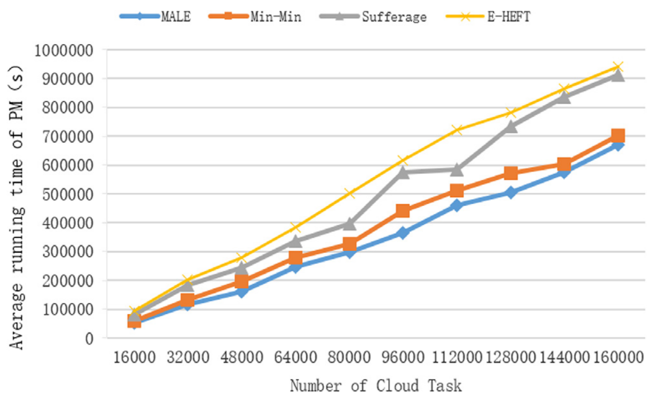

Fig. 7(B). Average running time of PM on different number of cloud tasks.

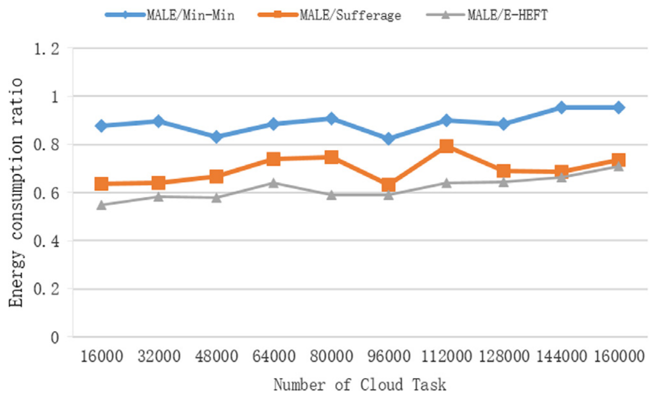

Fig. 7(C). Energy consumption ratio on different number of cloud tasks.

6. Conclusions and future work

In this article, we comprehensively consider the CPU and memory characteristics of cloud tasks and establish a memory priority cloud task mapping rule. According to the memory requirement of cloud task and memory priority cloud task mapping rule, the mapping order of cloud tasks and the establishment order of virtual machines were readjusted. We then propose a memory-aware resource management algorithm for low-energy cloud data centers (MALE). When using this algorithm for cloud task mapping, we try to select virtual machines with fewer CPU cores for mapping on the premise of meeting memory requirements. So as to reduce the total fee of cloud users and the energy consumption of cloud data center. Finally, we implemented the algorithm with CloudSim. Through comparison, we can see that MALE algorithm can effectively reduce the average running time of physical machines compared with existing algorithms, thereby reducing the total fee of cloud users and the energy consumption of cloud data centers.

There are two main directions for future work. First, we further research bandwidth and other resources combining cloud data center energy consumption issues on cloud data center. Classify different cloud tasks and virtual machines to map to the corresponding virtual machines. Second, we combine game theory with dynamic pricing for virtual machines under the condition of guaranteeing the cost of cloud users in cloud data centers. In order to the profit of the cloud provider is maximized.

CRediT authorship contribution statement

Bin Liang: Conceptualization, Methodology, Software, Validation, Writing - original draft. **Xiaoshe Dong:** Formal analysis, Writing - review & editing. **Yufei Wang:** Investigation, Data curation. **Xingjun Zhang:** Visualization, Supervision.

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.

Acknowledgments

This work was supported by the National Key Research and Development Program [No. 2016YFB0200902] and the National Natural Science Foundation of China [No. 61572394].

References

[1] R. Barik, N. Farooqui, B.T. Lewis, C. Hu, T. Shpeisman, A black-box approach to energy-aware scheduling on integrated CPU–GPU systems, in: 2016 IEEE/ACM International Symposium on Code Generation and Optimization (CGO), 2016, pp. 70–81.
[2] G. Quan, X.S. Hu, G. Quan, X.S. Hu, Minimum energy fixed-priority scheduling for variable voltage processors, in: R. Lauwereins, J. Madsen (Eds.), Design, Automation, and Test in Europe: The Most Influential Papers of 10 Years Date, Springer, Netherlands, Dordrecht, 2008, pp. 313–323.
[3] Y.-P. You, C. Lee, J.-K. Lee, W.-K. Shih, Real-time task scheduling for dynamically variable voltage processors, 2001.
[4] S.H.H. Madni, M.S.A. Latiff, Y. Coulibaly, S.i.M. Abdulhamid, Resource scheduling for infrastructure as a service (IaaS) in cloud computing: Challenges and opportunities, J. Netw. Comput. Appl. 68 (2016) 173–200.
[5] X. Li, P. Garraghan, X. Jiang, Z. Wu, J. Xu, Holistic virtual machine scheduling in cloud datacenters towards minimizing total energy, IEEE Trans. Parallel Distrib. Syst. 29 (2018) 1317–1331.
[6] H. Goudarzi, M. Pedram, Energy-efficient virtual machine replication and placement in a cloud computing system, in: 2012 IEEE Fifth International Conference on Cloud Computing, 2012, pp. 750–757.
[7] A. Khosravi, S.K. Garg, R. Buyya, Energy and Carbon-Efficient Placement of Virtual Machines in Distributed Cloud Data Centers, in, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 317–328.
[8] J. Dong, X. Jin, H. Wang, Y. Li, P. Zhang, S. Cheng, Energy-saving virtual machine placement in cloud data centers, in: 2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, 2013, pp. 618–624.
[9] V. Perumal, S. Subbiah, Power-conservative server consolidation based resource management in cloud, Int. J. Netw. Manage. 24 (2014).
[10] T. Ferreto, M. Netto, R. Calheiros, C. De Rose, Server consolidation with migration control for virtualized data centers, Future Gener. Comput. Syst. 27 (2011) 1027–1034.
[11] 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 (2012) 755–768.
[12] T. Muthukaruppan, A. Pathania, T. Mitra, Price theory based power management for heterogeneous multi-cores, 2014.
[13] D. Harnik, D. Naor, I. Segall, Low power mode in cloud storage systems, in: 2009 IEEE International Symposium on Parallel & Distributed Processing, 2009, pp. 1–8.

[14] Z. Xiao, W. Song, Q. Chen, Dynamic resource allocation using virtual machines for cloud computing environment, IEEE Trans. Parallel Distrib. Syst. 24 (2013) 1107–1117.

[15] D. Meisner, B. Gold, T. Wenisch, PowerNap: Eliminating server idle power, 2009.

[16] W. Lang, J. Patel, J. Naughton, On energy management, load balancing and replication, SIGMOD Rec. 38 (2009) 35–42.

[17] K. Chen, J. Powers, S. Guo, F. Tian, CRESP: Towards optimal resource provisioning for mapreduce computing in public clouds, IEEE Trans. Parallel Distrib. Syst. 25 (2014) 1403–1412.

[18] B. Liang, X. Dong, X. Zhang, A heuristic virtual machine scheduling algorithm in cloud data center, in: 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2019, pp. 180–184.

[19] M. Lin, A. Wierman, L.L.H. Andrew, E. Thereska, Dynamic right-sizing for power-proportional data centers, IEEE/ACM Trans. Netw. 21 (2013) 1378–1391.

[20] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, N. Gautam, Managing server energy and operational costs in hosting centers, SIGMETRICS Perform. Eval. Rev. 33 (2005) 303–314.

[21] T. Wood, P. Shenoy, A. Venkataramani, M. Yousif, Sandpiper: Black-box and gray-box resource management for virtual machines, Comput. Netw. 53 (2009) 2923–2938.

[22] G.v. Laszewski, L. Wang, A.J. Younge, X. He, Power-aware scheduling of virtual machines in DVFS-enabled clusters, in: 2009 IEEE International Conference on Cluster Computing and Workshops, 2009, pp. 1–10.

[23] H. Liu, H. Jin, C.-Z. Xu, X. Liao, Performance and energy modeling for live migration of virtual machines, Cluster Comput. 16 (2013) 249–264.

[24] E.S.T. El-kenawy, A.I. El-Desoky, M.F. Al-rahamawy, Extended max-min scheduling using petri net and load balancing, Int. J. Soft Comput. Eng. (IJSCE) 2 (2012) 198–203.

[25] D. Ardagna, B. Panicucci, M. Trubian, L. Zhang, Energy-aware autonomic resource allocation in multitier virtualized environments, IEEE Trans. Serv. Comput. 5 (2012) 2–19.

[26] J. Wang, C. Huang, K. He, X. Wang, X. Chen, K. Qin, An energy-aware resource allocation heuristics for VM scheduling in cloud, in: 2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing, 2013, pp. 587–594.

[27] K. Chard, K. Bubendorfer, High performance resource allocation strategies for computational economies, IEEE Trans. Parallel Distrib. Syst. 24 (2013) 72–84.

[28] B. Liang, X. Dong, X. Zhang, A power-aware scheduling algorithm in multi-tenant IaaS clouds, in: 2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP), 2019, pp. 250–253.

[29] V. Perumal, S. Subbiah, Power-conservative server consolidation based resource management in cloud, Netw. 24 (2014) 415–432.

[30] G. Yao, Y. Ding, L. Ren, K. Hao, L. Chen, An immune system-inspired rescheduling algorithm for workflow in cloud systems, Knowl.-Based Syst. 99 (2016) 39–50.

[31] H. Topcuoglu, S. Hariri, W. Min-You, Performance-effective and low-complexity task scheduling for heterogeneous computing, IEEE Trans. Parallel Distrib. Syst. 13 (2002) 260–274.

[32] Y. Samadi, M. Zbakh, C. Tadonki, E-HEFT: Enhancement heterogeneous earliest finish time algorithm for task scheduling based on load balancing in cloud computing, in: HPCS 2018 (The 2018 International Conference on High Performance Computing & Simulation), Orléans, France, 2018, pp. 601–609.

[33] X. Liao, H. Jin, H. Liu, Towards a green cluster through dynamic remapping of virtual machines, Future Gener. Comput. Syst. 28 (2012) 469–477.

[34] H. Dou, Y. Qi, An online electricity cost budgeting algorithm for maximizing green energy usage across data centers, Front. Comput. Sci. 11 (2017) 661–674.

[35] L. Andrew, M. Lin, A. Wierman, Optimality, fairness, and robustness in speed scaling designs, 2010.

[36] X. Sun, N. Ansari, Green cloudlet network: A distributed green mobile cloud network, IEEE Netw. 31 (2017) 64–70.

[37] L. Wang, F. Zhang, J.A. Aroca, A.V. Vasilakos, K. Zheng, C. Hou, D. Li, Z. Liu, GreenDCN: A general framework for achieving energy efficiency in data center networks, IEEE J. Sel. Areas Commun. 32 (2014) 4–15.

[38] C. Gu, Z. Li, H. Huang, X. Jia, Energy efficient scheduling of servers with multi-sleep modes for cloud data center, IEEE Trans. Cloud Comput. (2018) 1.

[39] Y. Ding, G. Yao, K. Hao, Fault-tolerant elastic scheduling algorithm for workflow in cloud systems, Inform. Sci. 393 (2017) 47–65.

[40] Z. Zhou, F. Liu, R. Zou, J. Liu, H. Xu, H. Jin, Carbon-aware online control of geo-distributed cloud services, IEEE Trans. Parallel Distrib. Syst. 27 (2016) 2506–2519.

[41] C. Gu, L. Fan, W. Wu, H. Huang, X. Jia, Greening cloud data centers in an economical way by energy trading with power grid, Future Gener. Comput. Syst. 78 (2018) 89–101.

[42] B. Liang, X. Dong, Y. Wang, X. Zhang, A low-power task scheduling algorithm for heterogeneous cloud computing, J. Supercomput. (2020).

[43] Z. Zhou, F. Liu, H. Jin, B. Li, B. Li, H. Jiang, On arbitrating the power-performance tradeoff in SaaS clouds, in: 2013 Proceedings IEEE INFOCOM, 2013, pp. 872–880.

[44] Y. Ding, X. Qin, L. Liu, T. Wang, Energy efficient scheduling of virtual machines in cloud with deadline constraint, Future Gener. Comput. Syst. 50 (2015) 62–74.

[45] Q. Zheng, R. Li, X. Li, J. Wu, A multi-objective biogeography-based optimization for virtual machine placement, in: 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2015, pp. 687–696.

[46] B. Krishnan, H. Amur, A. Gavrilovska, K. Schwan, VM power metering: feasibility and challenges, SIGMETRICS Perform. Eval. Rev. 38 (2011) 56–60.

[47] Y.C. Lee, A.Y. Zomaya, Energy efficient utilization of resources in cloud computing systems, J. Supercomput. 60 (2012) 268–280.

[48] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, I. Brandic, Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility, 2009.

[49] A.V. Karthick, E. Ramaraj, R.G. Subramanian, An efficient multi queue job scheduling for cloud computing, in: 2014 World Congress on Computing and Communication Technologies, 2014, pp. 164–166.

[50] C.C. Huang, C.L. Huang, Development of cloud computing based scheduling system using optimized layout method for manufacturing quality, in: 2012 International Symposium on Computer, Consumer and Control, 2012, pp. 444–447.

[51] S. Garg, P. Konugurthi, R. Buyya, A linear programming driven genetic algorithm for meta-scheduling on utility grids, 2009.

[52] T.S. Somasundaram, K. Govindarajan, CLOUDRB: A framework for scheduling and managing high-performance computing (HPC) applications in science cloud, Future Gener. Comput. Syst. 34 (2014) 47–65.

Bin Liang is a Ph.D. student in the School of Electronic and Information Engineering of Xi'an Jiaotong University. He received his BE degree from Xi'an University of Technology and MS degree from Xi'an Jiaotong University. His research interests focus on cloud computing, energy-aware scheduling, cloud datacenter scheduling and blockchain model.

Xiaoshe Dong received the BE and ME degrees in computer science from Xi'an Jiaotong University, China, in 1985 and 1990, respectively, and the Ph.D. degree in computer science from Keio University, Japan, in 1999. He was a lecturer from 1987–1994 and an associate professor from 1999–2003 in the Department of Computer Science & Technology of Xi'an Jiaotong University, where he has been a full professor with the Department of Computer Science and Technology since 2003. His interests include computer architecture, high performance computing and cloud computing.

Yufei Wang is a Ph.D. student in the School of Electronic and Information Engineering of Xi'an Jiaotong University. He received his BE degree from Xi'an University of Technology and MS degree from Xi'an Jiaotong University. His research interests focus on the reliability of solid-state drives, high performance computing, cloud computing and nonvolatile memory technology.

Xingjun Zhang received his Ph.D degree in Computer Architecture from Xi'an Jiaotong University, China, in 2003. From 1999 to 2005, he was Lecturer, Associate Professor in the Department of Computer Science & Technology of Xi'an Jiaotong University. From Feb. 2006 to Jan. 2009, he was Research Fellow in the Department of Electronic Engineering of Aston University, United Kingdom. He was Associate Professor during 2009–2013 in the Department of Computer Science & Engineering of Xi'an Jiaotong University, where he has been the Full Professor from 2014. His interests include high performance computer architecture, high performance computing, big data storage system and computer networks.