Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

Research on virtual machine consolidation strategy based on combined prediction and energy-aware in cloud computing platform

Jinjiang Wang, Hangyu Gu, Junyang Yu*, Yixin Song, Xin He and Yalin Song

Abstract

In the era of information explosion, the energy consumption of cloud data centers is significant. It's critical to reduce the energy consumption of large-scale data centers while guaranteeing quality of service (QoS), especially the energy consumption of video cloud computing platforms. The application of virtual machine (VM) consolidation has been regarded as a promising approach to improve resource utilization and save energy of the data centers. In this paper, an energy efficient and QoS-aware VM consolidation method is proposed to address the issues. A combined prediction model based on grey model and ARIMA is applied to host status detection, and we provide a new scheme that VM placement policy based on resource utilization and varying energy consumption to search most suitable host and VM selection policy called AUMT selecting VM with low average CPU utilization and migration time. Extensive experimental results based on the cloudsim simulator demonstrate that proposed approach enables to achieve the objectives reducing energy consumption, number of migrations, SLAV and ESV by an average of 56.07%, 79.21%, 91.01% and 84.34% compared with the benchmark methods and the AUMT can reduce energy consumption, the number of migrations and ESV by an average of 15.46%, 28.11% and 3.96% compared with the state-of-the-art method.

Keywords: Cloud computing platforms, Virtual machine consolidation, Energy consumption, QoS

Introduction

The vital issue of energy comsuption is always with data centers today. The mainly factors are high operating cost and envirmental impact for cloud provider. According to Koomey's report [1], the total operating cost plays an important role in the annual electricity cost of a large-scale data center account for about 41%. However, according to Shehabi's latest report [2], power consumption about 45% through a new resource management method can reduce.

The virtualization technology of large-scale data center [3] provides an opportunity to dynamically consolidate VMs of the data center. Dynamic VM consolidation technology uses real-time VM migration to package as many VMs as possible on a host and switches the low-utilized hosts to a low power consumption mode to save the energy and improve resource utilization of hosts for the data center [4]. However, considering the variable workloads of applications running on the VM [5, 6], additional migrations lead to hamperring the quality of service (QoS) and results in increasing in some aspects about response time, failure and time-out, [7] as well as it may increase the costs of energy consumption and VM migration, thus dynamic VM consolidation may lower the QoS and even increase energy consumption if the technology applied inappropriately.

Since the fluctuating workloads of VMs running on the hosts in the data center, some working state of hosts will be seem as overloaded and other hosts are underloaded [8]. Dynamic VM consolidation is exactly efficient method that improve resource utilization and keep in a normal working state to perform tasks while

*Correspondence: jyyu@henu.edu.cn

Department of Software, Henan University, Kaifeng, Henan, China

maintaining service of agreement (SLA) for whole hosts in the data center. Dynamic VM consolidation is divided by several steps to effectively reallocate VMs to hosts involves (1)detecting the overloaded host; (2) selecting the migrated VM from the overloaded host; (3)detecting the underloaded host; (4) selecting the targeted host for migration; (5)execute the consolidation [9]. Specifically, when the host with overloaded status, the host try to migrate VMs to suitable hosts to keep its in better working condition meanwhile guaranteeing the QoS for the data center, and taking into consideration the additional energy consumption generated by switching the power state of the host from idle to low power state [10, 11], it is necessary to switch all the underloaded hosts and limit the frequency with the objective of energy saving. Current and future CPU utilization indicators are considered as reliable characteristics of overloaded and underloaded hosts, and CPU utilization has the greatest impact on energy consumption [12, 13], so predicting short-term future CPU utilization based on historical data enables to determine host's state to turn off the underloaded hosts to save energy and to reduce the number of additional VM migrations for the overloaded hosts while guaranteeing quality of service (QoS) to some extent.

In this paper, it mainly focus on the reduction of energy consumption, the number of virtual machine migrations and SLA violations while guaranteeing QoS in the cloud data centers, we devise a VM consolidation framework involves identification of the underloaded and overloaded hosts, an efficient virtual machine placement strategy that building new mapping relationships between the most suitable hosts and migrated virtual machines and then virtual machine selection strategy that selecting virtual machines from the overloaded hosts to migrate.The main contributions of this paper are as follows:

- Formulation of combined prediction model based on grey model and the ARIMA model to predict CPU utilization of all hosts to determine the status about overloaded or underloaded.
- Proposal of VM selection policy called AUMT that selecting VM with minimum cost in combination of both average CPU utilization and migration time.
- Proposal of VM placement heuristics approach called CUECC that determining targeted host with the maximum reward in combination of both real-time CPU utilization and energy consumption changes when the virtual machine placed.
- Extensive experiments were conducted on Cloudsim for evaluating the performance of the proposed approach based on real-world workload traces.

The rest of this paper is organized as follows. Related works are discussed in Section 2. Section 3 provides system framework and host status detection approach based on combined prediction. Section 4 introduces the VM selection strategy AUMT, VM placement strategy. Section 5 evaluates the proposed approach based on the experimental environment, performance metrics, comparison benchmarks and extensive simulation results. Section 6 concludes and describes future work.

Related work

Extensive researches has been foucsed on the energy efficiency of data centers. With the wide popularization of virtualization technology, a good deal of previous works have used VM consolidation as an effective solution energy saving for data center. The methods [13, 14] use real-time migration to pack existing VM into fewer hosts and periodically shut down idle hosts. Generally speaking, the problem of dynamic VM consolidation can be divided into several sub-problems. Previous approaches in this area usually underlined a sub-problem of the general process.

In some methods, VM consolidation is regarded as an optimization problem and solved by known convex optimization solutions. In the heterogeneous cloud centers, Wu et al. [15] in order to reduce the migration cost of VMs and the energy consumption, an improved grouping genetic algorithm based on the score function is proposed, and the final experimental results can meet the experimental objectives. Ashraf et al. [16] propose a novel multi-objective ant colony algorithm for VM consolidation that can meet the objectives of minimum number of VM migrations and active hosts in heterogeneous cloud centers.

Beloglazov et al. [13] considering the variable workloads in cloud data centers, propose host status detection based on historical CPU utilization involves interquartile range (IQR), median absolute deviation (MAD) and local regression (LR). Beloglazovet et al. [12] propose a low static CPU utilization threshold, when host's the CPU utilization below the threshold, all VMs will be migrated to active hosts without overloaded and the host will switch to idle state to save energy.

Farahnakianet [17] predict the CPU utilization of the future host using a linear regression method based on the historical CPU utilization to ensure whether the future CPU utilization of the host is overloaded, and to decide to migrate some VMs of the host and reduce SLA violations; meanwhile, the underloaded host according to the predicted value lower the low threhold will migrate all VMs and the host is switched to the sleep state to reduce overall energy consumption for cloud data center.

Haghshenas and Mohammadi [18] propose a new linear regression method to predict the CPU utilization of hosts based on the historical data and to select the targeted hosts with higher utilization for VM allocated. Suhib et al. [14] based on the historical CPU utilization of the hosts, a Markov prediction model is applied to identify the future hosts' status about overloaded, normal or underloaded, which can avoid addtional migrations. Li et al. [19] devise the method that host overloaded detection based on a robust linear regression, which is applied to improve the accuracy of the results about predicted value, eight error reduction methods are used in the paper, it enables to reduce SLAV to some extent. The new approach about EQ-VMC was proposed in [20], the authors introduce improved discrete difference evolution algorithm to obtain the deployment vector between each VM and all physical machines in the global search space and optimize the vector to find the most suitable host for the migrated VMs.

Laili et al. [21] propose a new iterative budget algorithm, which is adopted to reduce the costs include migration, communication and underloaded for node, then the proposed approach based on budget heuristic strategy and multi-stage selection strategy to deploy the appropriate host during VM migration. Sharma et al. [22], in order to solve the situation of unreliable physical resources deployed during VM consolidation, it mainly focus on a failure-aware VM consolidation mechanism. This mechanism provides real-time monitoring of failures of consolidation occuring and immediately consolidate physical resources in the data center. Jheng et al. [23] devise a gray prediction model to predict the future CPU utilization of the host, but the model does not guarantee the accuracy of the prediction results due to fluctuations in workload. Chehelgerdi-Samani and Safi-Esfahani [24] using the known ARIMA model, a framework called PCVM.ARIMA was developed to detect host overloads during VM consolidation and thus reduce unnecessary migration of VMs.

Xu et al. [25] introduce a lightweight interference-aware VM live migration strategy based on designing a simple multi-resource demand-supply model to cope with the incurred performance interference and cost on both source and destination servers during and after such VM migration. Xu et al. [26] propose a Heterogeneity and interference-aware VM provisioning framework for tenant applications called (Heifer) to create VM instances of the good-performing hardware type by explicitly exploring the hardware heterogeneity and capturing VM interference. Xu et al. [27] devise future research challenges pertinent to the modeling methods and mitigation techniques of VM performance overhead in the IaaS cloud based on the obtained insights into the pros and cons of each existing solution. Liu et al. [28] design control framework by taking advantage of the Lyapunov optimization techniques to make online decisions on request admission control, routing, and virtual machine (VMs) scheduling. Deng et al. [29] devise Reliability-Aware server consolidation strategy called RACE to address when and how to perform energy-efficient server consolidation in a reliability-friendly and profitable way.

Syh et al. [30] use the grey Markov prediction model to identify the host's future state with overloaded or underloaded and to confirm its' effectiveness in reducing the number of VM migrations and energy consumption but without account of the long execution time. Calheiros et al. [31] apply the ARIMA model with 91% prediction accuracy for the variable workloads and evaluate the accuracy of its resource utilization and QoS. The combined prediction model can improve the accuracy between real and predicted value [32]. For this reason, in this study, we use a combined model based on ARIMA and GM(1,1), which can improve accuracy to a certain extent [32].

System framework and host status detection based on combined prediction model

In this section, we introduce the system framework based on the proposed approach and describe the proposed strategy for the key elements, host status detection, of VM consolidation.

System framework

Figure 1 pictures the system architecture of this paper, which presents the whole procedure of VM consolidation.

Our implementation consists of m heterogeneous hosts (i.e.,$H =< h_1, h_2, \ldots, h_m >, i \in < 1, m >$) in a cloud data center. Each host is characterized by different resource types such as CPU, memory size, network bandwidth and storage capacity. Additionally, CPU is usually measured in million instructions per second (MIPS). At any given time, many simultaneous users use the services of a cloud data center. The provisioning of n VMs ($i.e., V =< v_1, v_2, \ldots, v_n >, j \in < 1, n >$) is requested by users. Since the workloads fluctuating, the request utilization of running hosts and VMs will vary all time. For this reason, this paper proposed a novel VM consolidation algorithm, which can be implemented regularly to optimize the performance of the cloud data center. Our proposed approach is executed every 5 minutes in the cloud data center to reduce energy consumption and the number of active hosts and the number of migrations. The system architecture contains two types of agents, the global manager (GM) deployed under the master node, and the local managers (LMs) where all hosts are fully

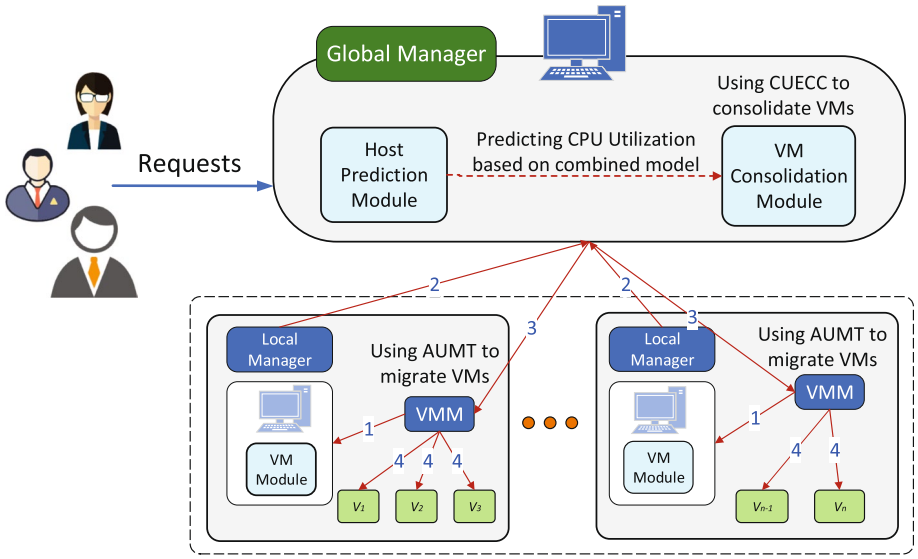


Fig. 1 System architecture. Combined model:= Combined prediction model of GM(1,1) and ARIMA, CUECC:=determining targeted host with the maximum reward in combination of both real-time CPU utilization and energy consumption changes when the virtual machine placed, AUMT:=VM selection strategy that selecting VM with minimum cost in combination of both average CPU utilization and migration time

distributed [30]. The following operations are performed in each iteration;

(1) Each LM will regularly monitor the current resource utilization of all VMs of the host, and each LM uses a combined prediction method to predict the future CPU utilization of the host based on the historical CPU utilization.
(2) GM will collect the running status of the host in the LMs, the CPU utilization and the number of VMs running on the host.
(3) GM will send relevant migration commands to the VM monitor(VMM) to execute the host detection algorithm based on the combined prediction model, and use the proposed algorithm to select and place the VM.
(4) After VM monitor receiving the migration command, migration will start.

Power and energy consumption model

When a set of hosts in active state in the cloud data centers, due to the change of host's working status, the CPU of host will change instantaneously and power varies according to CPU utilization. The power consumption is defined as follows:

$$P(u_i) = P_i^{idle} + (P_i^{max} - P_i^{idle}) * u_i \tag{1}$$

where $P_i^{max}, P_i^{idle}, u_i$ present maximum power of host when experiencing full CPU utilization, minimum power of host with sleep state and host's CPU utilization respectively.

CPU utilization may change over time, so power of the host will be fluctuate with CPU utilization varying, which means that the host's energy consumption is a function of power and CPU utilization. Therefore, E_i^{consum}, the energy consumption generated by the host in active status, is defined as follows:

$$E_i^{consum} = \int_{t_1}^{t_2} P(u_i(t))dt \tag{2}$$

Live migration cost

By using VM live migration technology [33], VMs can be transferred between hosts without being suspended. The average performance degradation is equivalent to 10% of the CPU utilization of the VM during the migration [34]. Therefore, the cost of migration based on research [13] is defined as follows:

$$t_j^{mig} = \frac{v_j^{ram}}{h_i^{bw}} \tag{3}$$

$$v_j^{degra} = 0.1 \times \int_{t_0}^{t_0 + t_j^{mig}} u_j(t)dt \tag{4}$$

where t_j^{mig}, v_j^{ram} present the migration time required, RAM used for v_j, h_i^{bw} is the available bandwidth of the host h_i; u_j, v_j^{degra} present utilization of VM and the performance degradation during VM migration.

Host status detection based on combined prediction

In this section, formulation of combined prediction model based on GM(1,1) and ARIMA in this subsection and apply it in the host status detection approaches.

GM(1,1)

In the gray prediction model family, the most commonly used is the gray model [35]. Grey forecasting is an exponential forecasting model. The accumulated generation operation (AGO) of the original sequence can reduce the noise of the original sequence. The first-order linear differential equation is used to model the data sequence from AGO, which can predict the future trend [30].

Suppose there are n samples in a set of data, and each sample is independent and has no relationship. Assume the host's historical CPU utilization data with n samples (time point) as:

$$X^{(0)} = \{x^{(0)}(1), x^{(0)}(2), \ldots, x^{(0)}(n-1), x^{(0)}(n)\} \quad (5)$$

Construct the AGO. Let $X^{(1)}$ be the transformation sequence of $X^{(0)}$

$$X^{(1)} = \{x^{(1)}(1), x^{(1)}(2), \ldots, x^{(1)}(n)\}, where \quad x^{(1)}(k) = \sum_{i=1}^{k} x^{(0)}(i), k \in < 1, \ldots, n > \quad (6)$$

Consequently, the model of the first-order differential equation GM(1, 1) is:

$$\frac{dx^{(1)}}{dt} + ax^{(1)} = b \quad (7)$$

To obtain the predicted value of the primitive data at time $(k+1)$, the inverse AGO (IAGO) is used to establish the following gray model [30]:

$$\widehat{x}^{(0)}(k+1) = \left[x^{(0)}(1) - \frac{b}{a}\right] e^{-ak}(1 - e^{a}) \quad (8)$$

ARIMA

Box and Jenkins proposed a method that includes a five-step process of identifying, selecting, and evaluating conditional mean models [36], which is based on the autoregressive integrated moving average (ARIMA) model.

This model is expressed as ARIMA(p,d,q). Where p, d and q are non-negative real numbers. ARMA model is suitable for stationary time series data. ARMA involves two critical phases that 1) using AR model to gain the current value based on linear combination of p past observation and a random error together with a constant term and 2) applying MA model (line regression) to obtain current observation of the time series against the random shocks of one or more prior observations. When time series data is non-stationary, the defined by $(1-L)y_t = y_t - y_{t-1}$ difference operation for time series data can obtain stable data. Thus, ARIMA(p,d,q) is ARMA with d different times. ARMA(p,q) and ARIMA(p,d,q) [36] are expressed by Eqs. (9) and (10), respectively.

$$y_t = c + \epsilon_t + \sum_{i=1}^{p} a_i y_{t-i} + \sum_{j=1}^{q} b_j \epsilon_{t-j} \quad (9)$$

$$\left(1 - \sum_{i=1}^{p} a_i L^i\right)(1-L)^d y_t = \left(1 + \sum_{j=1}^{q} b_j L^j\right)\epsilon_t \quad (10)$$

Combined prediction

Combined prediction model performs better in the accuracy compared with ARIMA and GM(1,1) [32], it's beneficial to predict host's CPU utilization according to short-term historical data to determine future working state of host and to avoid addtional migrations and improve QoS for cloud data centers. Thus, the combined prediction model is appied in this paper. The predicted value denoted by $Y_{predict}$ is defined as follows:

$$Y_{predict} = \alpha_1 y_t + \alpha_2 \widehat{x}^{(0)}(k+1) \quad (11)$$

Subjects to the constraints are:

$$0 < \alpha_1 < 1, \qquad 0 < \alpha_2 < 1 \quad (12)$$

$$\alpha_1 + \alpha_2 = 1 \quad (13)$$

Host overload detection

In the cloud data centers, the host with overloaded status exerts an impact on QoS and increases SLA violations. It is important to determine whether future working state of the host is overloaded and to reduce additional migrations and energy consumption caused by the overloaded host. Hence, host overloaded detection Algorithm 1 is

proposed in this subsection. The approach according to the CPU utilization based on real-time and the predicted value calculated by Eq. (11) using historical short-term CPU utilization.

We assume a set of n VMs and a set of m heterogeneous hosts in the data center, the mapping relationships between VMs and hosts denoted by x_{ij} is displayed as Eq. (14), when a new VM is placed on the host, the value about x_{ij} is equal to one otherwise zero.

$$x_{ij} = \begin{cases} 1, \ if \quad v_j \quad placed \quad on \quad h_i \\ 0, \ otherwise \end{cases} \tag{14}$$

It's known that the host's CPU utilization is subject to fluctuating workloads of VMs. And host's CPU utilization varies depending on the number of VMs placed on the host and the varying CPU utilization of VM. So the CPU utilization of the host is calculated as follows:

$$u_i = \frac{\sum_{j=1}^{n} x_{ij} \times v_j^{mips} \times u_j}{h_i^{mips}} \tag{15}$$

where the v_j^{mips} and h_i^{mips} present the maximum CPU capacity of the host and VM in MIPS, respectively and the u_j is CPU utilization of VM running on the host.

The CPU utilization varying real-time since the varying workloads of the VM running on the host. Since time series data is the historical data of CPU utilization recorded every 5 minutes on each host in this paper. To obtain the dynamic upper threshold denoted by tu, it is defined as follows:

$$tu = 1 - s * Mad \tag{16}$$

Where the s is safety parameter and Mad presents the method (Mean absolute deviation) to handle historical data.

```
   Input: host
   Output: true or false
 1 predictionUtilization←Using algorithm3;
 2 UtilizationOfHost=RequestedMIPS/totalMIPS ← Using Eq.(15);
 3 tu← Using Eq.(16);
 4 if length<18 then
 5 |   if UtilizationOfHost>tu then
 6 |   |   return true;
 7 |   end
 8 |   else
 9 |   |   return false;
10 |   end
11 end
12 if length>18 then
13 |   if UtilizationOfHost>tu and predictionUtilization>tu then
14 |   |   return true;
15 |   end
16 |   else
17 |   |   return false;
18 |   end
19 end
```

Algorithm 1: Host overload detection

As the Algorithm 1 describes that host overloaded detection. Firstly, the input of the algorithm is a active host, use Algorithm 3 to get the predicted value based on combined prediction model to predict CPU utilization of the host in line 1, then calculate the current host utilization using Eq. (15) in line 2. Obtaining dynamic upper threshold in line 3. In proposed approach (in lines 4-18), if the utilization of the host is higher than the upper threshold, it's status is regarded as overloaded. Using Eq. (16) to set the upper threshold. In terms of the length of the historical CPU utilization data, the length of 18 performs best. If the length of historical CPU utilization data is less than 18 and the current utilization of host is higher than the threshold, the host is considered overloaded, otherwise it's not. Similarly, when the length of historical CPU utilization data is more than 18, the status of host is regarded as overloaded as current CPU utilization of host and predicted value are higher than the threshold.

Host underload detection

When the LM detects the host with underloaded status based on the combined prediction model, GM sends migration conmmands to VMM to migrate all VMs, reducing the number of active hosts with low CPU utilization and switching the hosts to idle mode are main access to reduce energy consumption, thus it is vital to ensure the most underloaded host from the list about host and the algorithm of host underloaded detection is presented as follow.

```
   Input: hostlist // The list of host
   Output: underUtilizedHost // Selecting the most underloaded host
 1 Initialize u_i^predict, u_i;
 2 Initialize minUtilization=1;
 3 Initialize underUtilizedHost=null;
 4 tu←0.3;
 5 for Host host: hostlist do
 6 |   if length < 18 then
 7 |   |   u_i ← Using Eq.(15);
 8 |   |   if u_i < tu and !areAllVmsMigratingOutOrAnyVmMigratingIn(host) then
 9 |   |   |   minUtilization=u_i;
10 |   |   |   underUtilizedHost=host;
11 |   |   end
12 |   |   return underUtilizedHost;
13 |   end
14 |   else
15 |   |   minUtilization=1;
16 |   |   u_i^predict ←Using algorithm3;
17 |   |   u_i ← Using Eq.(15);
18 |   |   double meanUtilization=(u_i^predict + u_i)/2;
19 |   |   if meanUtilization < minUtilization then
20 |   |   |   minUtilization= meanUtilization;
21 |   |   |   underUtilizedHost=host;
22 |   |   end
23 |   |   return underUtilizedHost;
24 |   end
25 end
```

Algorithm 2: Host underload detection

As the Algorithm 2 shows that underloaded host detection. Firstly, the input and output of the algorithm is the list of host and is the host with most insufficient CPU utilization, respectively. Initiating the CPU utilization of host with predicted and real value using Eqs. (11) and (15)the minimum CPU utilization and

underloaded host in lines 1-3, the low threshold we set is 0.3 in line 4. Selecting the host with minimum CPU utilization in lines 5-34 in this paper. When the length of historical CPU utilization is less than 18, if the host's real CPU utilization is lower than the threshold we set, the host considered to be an underloaded(in lines 6-13); when the length is more than 18 (in lines 14-24), then gain the host of real and predicted CPU utilization in lines 16-17 and gain the mean value about two types of utilization in line 18, if the host's CPU utilization of real and the predicted value are lower than the threshold we set and searching the host with minimum mean CPU utilization in lines 15-23, only those conditions satisfied the host will be regarded as most underloaded.

```
Input: CPU_utilization[] //The historical data
Output: PredictionValue // Gaining the predicted value
1 Data[]←CPU_utilization[];
2 Initialize sum_PredictionValue[];
3 Initialize Average_Value=0, d=1;
4 sum_PredictionValue+=ARIMA(p,d,q);
5 Average_Value=Math.mean(sum_PredictionValue);
6 greyPredictionValue←using Eq.(8);
7 PredictionValue ←using Eq.(11);
8 return PredictionValue;
```

Algorithm 3: Get PredictionValue

As presented in Algorithm 3, the combined prediction model is used to predict the CPU utilization of the host, and finally more accuratly predicted value about CPU utilization is obtained. The input of the algorithm is historical CPU utilization data and perform the first-order linear difference on the data to obtain a relatively stable sequence, then use the ARIMA model to obtain the predicted value, then to reduce the error by an average of a set of predicted values based on ARIMA model and use the grey model to gain the CPU prediction utilization, finally use the Eq. (11) to get the combined predicted value.

The proposed VM placement policy

In this section, we introduce a new VM selection policy with consideration of both historical CPU utilization and migration time, and then propose a new VM placement strategy based on the real-time CPU utilization and variable power consumption when a new VM placed on the targeted host.

VM selection strategy

In the cloud data centers, varying CPU utilization of VMs is one of the key drivers of fluctuating CPU utilization of hosts. The VMM module performs when the host was identified as overloaded, in this article, the VM selection strategy called AUMT based on both average CPU utilization of VMs and the migration time, the migrated VMs with regard to proposed method is to alleviate the cloud data centers of energy consumption, the number of migrations and is to improve QoS.

The average CPU utilization of a VM is an indicator of the working status of VM placed on the host, and VM with low CPU utilization will have little influence on performance degradation when performing migration, denoted by u_j^{ave}, is defined below:

$$u_j^{ave} = \frac{\sum_{k=1}^{length} u_j}{length} \tag{17}$$

where *length* and u_j^{ave} indicate the length of VM's historical CPU utilization and average CPU utilization of VM v_j.

$$cost_j = u_j^{ave} \times t_j^{mig} \tag{18}$$

When VM migration triggerring, the VM with the minimum value calculated by Eq. (18) is preferred to select.

The proposed algorithm for VM selection

```
Input: overloadedHost // Implementing VM selection from the overloaded host
Output: VmToMigrate // Selecting the VM from the host
1  migratableVms = getMigratableVms(host);
2  if migratableVms.isEmpty() then
3  |   return null;
4  end
5  Initialize VM=null, minMetric=Double.MAX_VALUE;
6  for VM in migratableVms do
7  |   Initialize metric ← Using Eq.(18) ;
8  |   if metric < minMetric then
9  |   |   minMetric=metric;
10 |   |   VmToMigrate=VM;
11 |   end
12 end
13 return VmToMigrate;
```

Algorithm 4: VM selectionThe pseudo-code of the AUMT is shown in Algorithm 4, the input and output of this algorithm is an overloaded host and the VM is to migrate, obtaining VMs migrated from the host (in line 1) and finding the VM with minimum value calculated by 18 (in lines 6-12)), eventually, return the VM.

Destination host selection

In the procedure of VM consolidation, it's vital to solve the mapping relationship between hosts and VMs, which means to find the most proper host. When the VMM triggers migrating, this study gives priority to considering whether the host's status with overloaded when the VM placed on the host, it avoids repeated migration. Therefore, the increase in CPU utilization and power consumption of the host, denoted by u_i^{incre} and p_i^{incre}, are defined as follows:

$$u_i^{incre} = \frac{v_j^{mips} * u_j}{h_i^{mips}} \tag{19}$$

and the p_i^{incre} is calculated as follows:

$$p_i^{incre} = p(u_i + u_i^{incre}) - p(u_i) \tag{20}$$

Where $p(u_i)$ presents that power consumption before the VM migrates to the targeted host and $p(u_i + u_i^{incre})$ shows that power consumption generated by the VM migrated to the host.

When the VM migrates to the targeted host, we should ensure the host's capacity of residual CPU utilization to meet the resource request of the VM, the capacity of residual CPU utilization, denoted by Δu_i, is defined as follows:

$$\Delta u_i = tu - (u_i + u_i^{incre}), \Delta util_i > 0 \tag{21}$$

Where the value about tu is obtained by the Eq. (16), only the value about Δu_i is more than zero, the host will be a targeted host.

In order to select the most suitable host and achieve the objectives of reducing energy consumption and the number of migrations, we take into consideration host's real-time CPU utilization and power consumption and try to place all VMs on the host within the upper threshold. Meanwhile, high CPU utilization results in the SLA violations, all VMs allocated will placed on the host in a normal status. The function is calculated using Eq. (22) to alleivate the problem below:

$$Score(h_i) = (1 - u_i) \times \frac{1}{1 + e^{-p_i^{incre}}} \tag{22}$$

The aim of this Eq. (22) is to illustrate that the same power consumption generated by the host when the VM placed on it has advantage in selecting a host with low CPU utilization and the VM placed on the same CPU utilization of host with high energy-efficiency, which enables to increase resource utilization for the cloud data centers.

The VM Placement strategy

The GM collects the running status of host in the data center, when the LM monitors the status of host is underloaded or overloaded the GM sends commands to each VMM to perform migration according to the live and predicted CPU utilization of host, the most underloaded host is to migrates all VMs to some hosts then to switch the host to idle state to save energy, the overloaded host performs the migration module to migrate some VMs to suitable host to avoid SLA violations. How to address the problem mapping relationships between VMs and hosts is also regarded as multi-dimensional bin-packing and NP-hard problem [37], we propose a heuristic approach to tackle the problem that using the score calculated by Eq. (22) with maximum value symbols that the host is suitable for placement. The Algorithm 5 is embedd with the approach of host's status detection and targeted host selection as shown below.

```
   Input: HostList, vmsToMigrate // HostList: the list of host; vmsToMigrate: some VMs prepare
          to migrate
   Output: MigrationMap // Gaining the mapping relationship between host and VM
1  Sort vmsToMigrate in descending order of CPU Utilization;
2  for vm in vmsToMigrate do
3      Initialize minPerformance=Minimum_Value, allocatedHost=null;
4      for host in HostList do
5          if host is overloadedHost then
6              continue;
7          end
8          if host is underloadedHost then
9              continue;
10         end
11         initialize delta ← using Eq.(21);
12         if Δu_i>0 then
13             if AvailableMips >RequestedTotalMips then
14                 if host.isSuitableForVm(vm) then
15                     powerdiff ← Using Eq.(20);
16                     performance ← Using Eq.(22);
17                     if performance > minperformance then
18                         minperformance =performance;
19                         allocatedHost=host;
20                     end
21                 end
22             end
23         end
24     end
25     MigrationMap←map(vm,allocatedHost);
26 end
27 return MigrationMap;
```

Algorithm 5: VM Placement

As the Algorithm 5 shows that VM placement based on the proposed approach. Firstly, after determining the list of VMs to be allocated, sorts the VMs in CPU utilization descending in line 1. About the list of host, use the combined prediction model to predict the CPU utilization of the hosts until the remaining hosts of status without overloaded and underloaded (in lines 5-10). If the value about Δu_i is more than zero, the host has sufficient resource capacity to place the VM, otherwise it cannot be placed. As confirming $\Delta u_i > 0$, and then meet the conditions that there are available CPU utilization in MIPS for host when the VM places it. In the end, apply the Eq. (22) for each host and select the host with a maximum score as the targeted host and execute migration.

Time complexity analysis: We assume that the number of N VMs migrated and a set of M hosts selected, after sorting the utilization of VMs, the time complexity is $O(NlogN)$ at this time. When VMs are placed on the host, the host's selection time is complicated, which is $O(M)$, and the time complexity of Algorithm 5 is $O(NlogN + MN)$, meanwhile the time complexity is $O(n^2)$ when M is equal to N in the worst case.

Experimental evaluation

In this section, we describe our relevant experimental setup, comparison benchmarks and performance metrics, which are introduced to evaluate the performance of the proposed algorithm in this paper.

Table 1 Power Consumption of the selected servers at different load levels(in Watts)

Host Type	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
G4	86	89.4	92.6	96	99.5	102	106	108	112	114	117
G5	93.7	97	101	105	110	116	121	125	129	133	135

Experiment setup

This paper uses cloudsim [38] as a simulation platform to test our proposed approaches. In the cloud platform, we simulated 800 heterogeneous hosts. The state of the hosts involves two types: 400 HP ProLiant ML110 G4 server (Intel Xeon 3040, 2 cores* 1.86 GHz) and 400 HP ProLiant ML110 G5 servers (Intel Xeon 3075, 2 cores* 2.26 GHz) and Table 1 shows the relationship between energy consumption and CPU utilization of different servers. In the experiment, we use four types of Amazon EC2 VMs as shown in Table 2. In order to validate the performance of our proposed algorithm in real cloud data centers,we used 10 workloads provided by the planetlab project [39]. PlanetLab is a computer cluster project distributed all over the world. The project collects CPU utilization data VMs from servers in more than 500 locations around the world. The project monitors the CPU utilization of host every five minutes, and the measurement period is one day. We chose a 10-day workloads tracking from March 2011 to January 2011 and Table 3 shows specific data.

Comparison benchmarks

To validate the performance of the proposed algorithm, the power-aware heuristic algorithm (PABFD) proposed in paper [40] is compared. we choose benchmark consolidation algorithms with five host's state detection algorithms are composed of static threshold (THR), interquartile range (IQR), local regression (LR), mean absolute deviation (MAD) and LR robust (LRR) and two vitual machine migration selection algorithms are composed of minimum migration time(MMT) and maximum correlation (MC) are embedded with PABFD to compare the proposed approach. Meanwhile, α_1, α_2 are equal to 0.47 and 0.53, the safety parameter for IQR, LR, LRR and MAD are set to 1.2 and for THR is set to 0.8. All comparative experiments are compared using cloudsim under the state of 10 workloads.

Table 2 Configurations for Amazon EC2 VMs

VM type	CPU[MIPS]	RAM[GB]	Number for cores
High-CPU medium instance	2500	0.85	1
Extra-large instance	2000	1.7	1
Small instance	1000	1.7	1
Micro instance	500	0.613	1

Performance metrics

In a cloud environment, a user submits a request to create a VM to the data center and signs a service level agreement with the data center. According to [13], the service level agreement is defined by the capabilities that the host and the previously recommended software measurement environment must meet the service quality requirements. *SLATAH* indicates that the percentage of the active host where the utilization is 100% is defined as:

$$SLATAH = \frac{1}{M}\sum_{i=1}^{M}\frac{T_{over_i}}{T_{active_i}} \tag{23}$$

where M, T_{over_i} and T_{active_i} present the number of hosts in active status, the time experiencing 100% CPU utilization of the host and the running time of host in active state (serving VMs) respectively.

When VM live migration technology triggers, the performance of VMs migrated will be affected. The performance degradation due to VM migration, denoted by *PDM*, is defined as follw:

$$PDM = \frac{1}{N}\sum_{j=1}^{N}\frac{C_{degr_j}}{C_{req_j}} \tag{24}$$

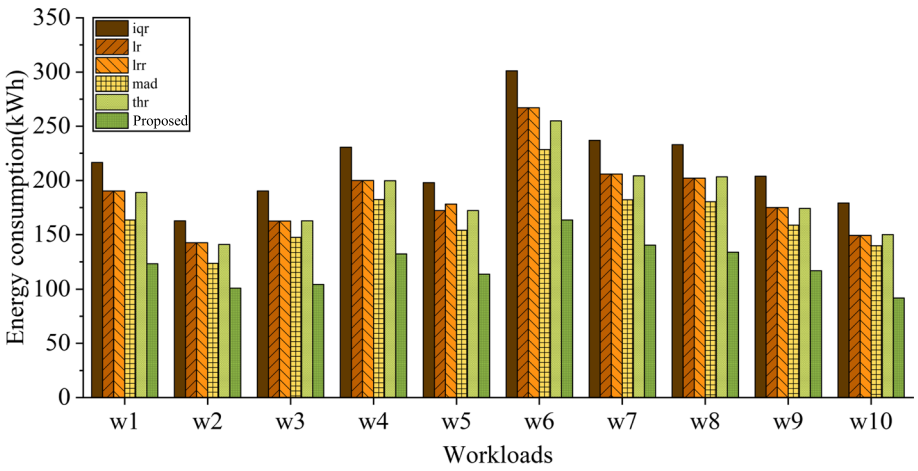
where N, C_{degr_j} and C_{req_j} present the number of VMs, the performance degradation caused by migration and the

Table 3 Planetlab trace data

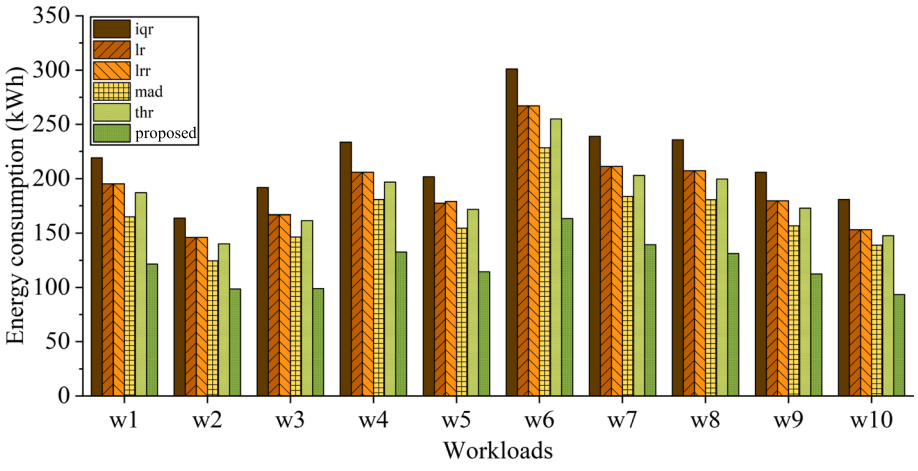
Workloads	Date	Number of servers	Number of VMs	Mean	St.dev
w1	2011/03/03	800	1052	12.31%	17.09%
w2	2011/03/06	800	898	11.4%	16.83%
w3	2011/03/09	800	1061	10.70%	15.57%
w4	2011/03/22	800	1516	9.26%	12.78%
w5	2011/03/25	800	1078	10.56%	14.14%
w6	2011/04/03	800	1463	12.39%	16.55%
w7	2011/04/09	800	1358	11.12%	15.09%
w8	2011/04/11	800	1233	11.56%	15.07%
w9	2011/04/12	800	1054	11.54%	15.15%
w10	2011/04/20	800	1033	10.43%	15.21%

Table 4 Simulation results of PABFD and proposed algorithm

Strategy	Energy consumption(kWh)	Migrations	SLAV($\times 10^{-4}$)	ESV(kWh $\times 10^{-2}$)
iqr_mmt	215.21	29575	10.49	22.36
lr_mmt	186.66	26991	9.35	17.42
lrr_mmt	187.24	26924	9.36	17.49
mad_mmt	166.06	30710	10.84	17.88
thr_mmt	185.15	25609	8.87	16.29
Proposed	**122.13**	**6071.1**	**0.85**	**1.02**
iqr_mc	217.25	30493	10.28	22.14
lr_mc	191	28743	9.12	17.35
lrr_mc	191.14	28781	9.23	17.59
mad_mc	165.96	30785	11.06	18.30
thr_mc	183.52	25177	8.92	16.22
Proposed	**120.61**	**5663**	**0.89**	**1.05**

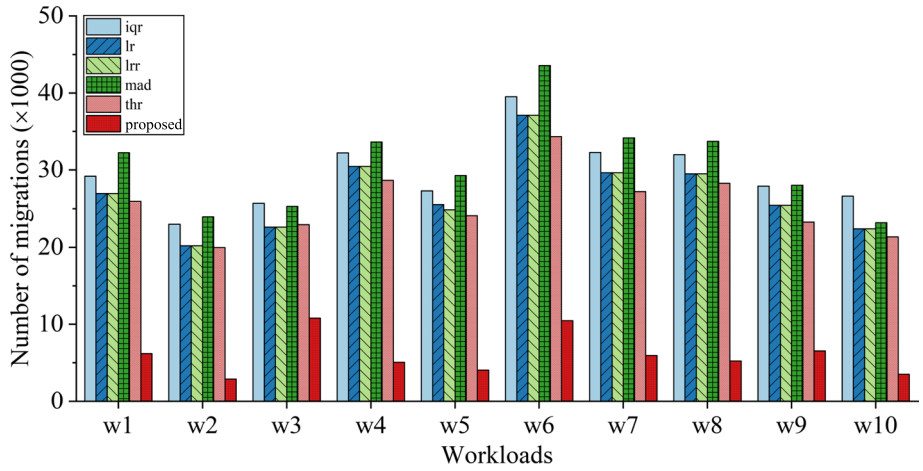


(a) MMT VM selection method

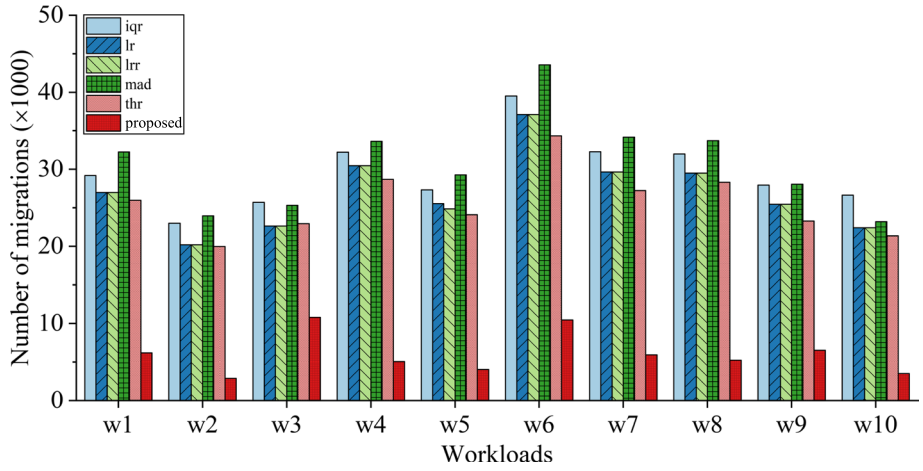


(b) MC VM selection method

Fig. 2 Energy consumption of the cloud data center based on the PABFD and the proposed approach are implemented with different VM selection method



(a) MMT VM selection method



(b) MC VM selection method

Fig. 3 Number of migrations of the cloud data center based on the PABFD and the proposed approach are implemented with different VM selection method

total CPU capacity requested by v_j during its life cycle respectively.

A combined metric is introduced to violate the service level agreement (SLA) to measure the performance degradation and service quality obstacles caused by overloaded host and VM migration.

$$SLAV = SLATAH \times PDM \tag{25}$$

We use metrics *SLATAH* and *PDM* to measure the degree of SLAV and express the QoS.

Since energy consumption can be reduced at the cost of increased violations of SLA, the performance metric according to ESV combined with energy consumption and SLA violations (SLAV) are discussed [41], which is defined as follows:

$$ESV = E \times SLAV \tag{26}$$

where E presents the energy consumption produced by all hosts in the data center.

Simulation results and analysis

In this section, we use the performance metrics to evaluate the proposed algorithm compared with the bechmarks method.

The following Table 4 displays the comparsion of energy consumption, the number of migrations, SLAV

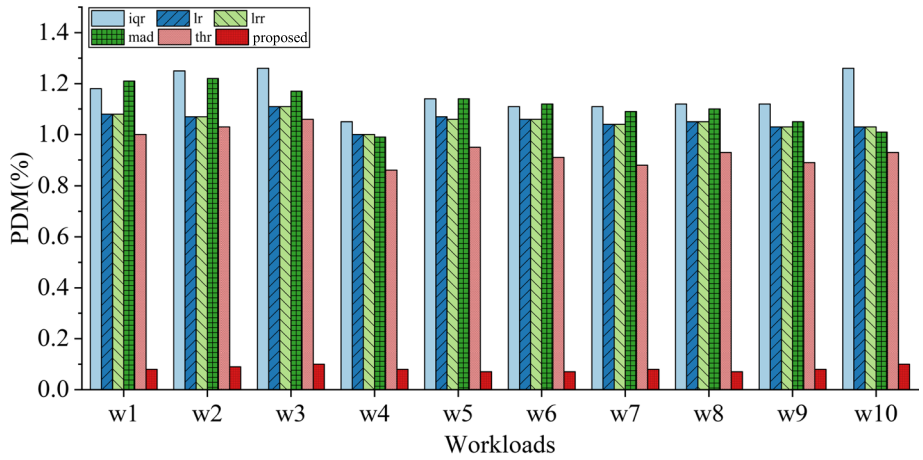
and ESV, and shows specific simulation results by an average of 10 workloads.

Comparison of energy consumption

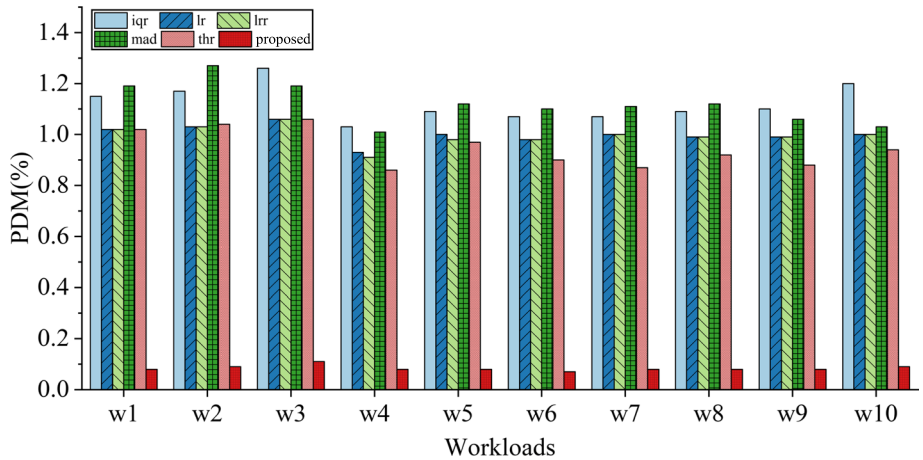
The simulation results as presented by Fig. 2 depict that energy consumption generated by all hosts in the data center. The proposed algorithm can achieve up to an average of 34.62% and 35.97% energy-saving improvement compared with baseline policy PABFD when the VM selection algorithm is MMT and MC as shown in Fig. 2a and b. The proposed approach is to increase resource utilization for running hosts and reduce the number of hosts with low resources utilization based on the host's status detecting algorithms to switch these hosts to idle status to save energy in the cloud data centers. Meanwhile, the introduction of Eq. (22) enables to balance live energy consumption and CPU utilization of host and finally it is effectiveness for energy saving.

Comparsion of number of migrations

The simulation results, as presented by Fig. 3, indicate the performance metric with regard to number of migrations in the data center by using proposed algorithm. The proposed algorithm can reduce the number of migrations by an average of 78.19% and 80.23% compared with the PABFD algorithm when the VM selection algorithm is MMT and MC as shown in Fig. 3a and b. The proposed approach is to predict the host's future resource utilization based on a combined prediction model to determine the host's status. When the host is regarded as overloaded or underloaded by the predicted value, the host will reduce the number of VMs running on this host to avoid addtional migrations or



(a) MMT VM selection method



(b) MC VM selection method

Fig. 4 PDM of the cloud data center based on the PABFD and the proposed approach are implemented with different VM selection method

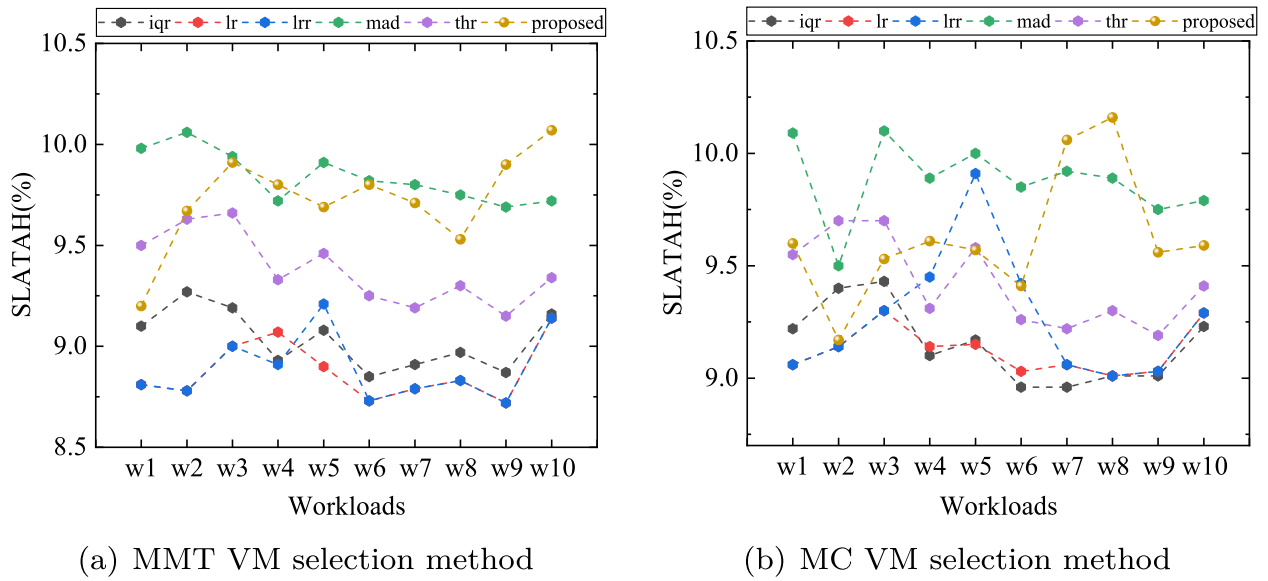


Fig. 5 SLATAH of the cloud data center based on the PABFD and the proposed approach are implemented with different VM selection method

reduce the number of hosts with low resource utilization to save energy for the cloud data centers.

Comparison of PDM

The simulation results as exhibited by Fig. 4 illustrate that the performance degradation of VM migration in the data centers. The proposed algorithm can reduce the performance metric about PDM by an average of 92.84%, 91.86% respectively compared with the PABFD algorithm when the VM selection algorithm is MMT and MC as shown in Fig. 4a and b. The results are so promising because the host status detecting algorithms are beneficial for reducing additional migrations and ensuring the targeted host has sufficient resource capacity for placing a new VM before placement procedures triggering.

Comparison of SLATAH

The simulation results as shown by Fig. 5 present that the performance metric about SLATAH in the data center. The proposed algorithm can increase the performance metric with regard to SLATAH by an average of 6% and 2.70% compared with the benchmark method when the VM selection algorithm is MMT and MC as shown in Fig. 5a and b. The increased proportion of performance metrics about SLATAH contrasted with the considerable decrease in energy consumption and PDM, so it is negligible. The reason for the slight increase in the SLATAH is that the proposed approach aims to place all VMs running on the host to increase resources utilization for the data centers to reduce the number of idle hosts to realize energy saving and to avoid additional migrations results in the performance degradation of VM. Not only the resource utilization has increased but the number of VM running on the host also has increased for the data centers, so two factors have raised the amount of SLATAH slightly.

Comparison of SLAV

The simulation results as displayed by Fig. 6 demonstrate that SLA violation calculated by Eq. (25) in the cloud data centers. The proposed algorithm can reduce SLA violation by an average of 91.23%, 90.79% respectively compared with benchmark method when the VM selection algorithm is MMT and MC as shown in Fig. 6a and b. The performance metric about SLAV is a combined value about metrics PDM and SLATAH, which symbols the SLA violations for the data centers. The resutls of proposed approach about SLAV are extremely smaller and draws a conclusion that the peoposed scheme has superiority in reducing SLA violations and improving QoS compared with bechmark method in the cloud data centers.

Comparison of ESV

The simulation results as presented by Fig. 7 illustrate that ESV is calculated by Eq. (26) in the cloud data centers. The proposed algorithm enables to reduce the performance metric about ESV by an average of 94.36% and 94.21% compared with benchmark method the PABFD algorithm when the VM selection algorithm is MMT and MC as shown in Fig. 7a and b. This result is significant at the 94.28% level on average. The performance metric is introduced to evaluate the overall

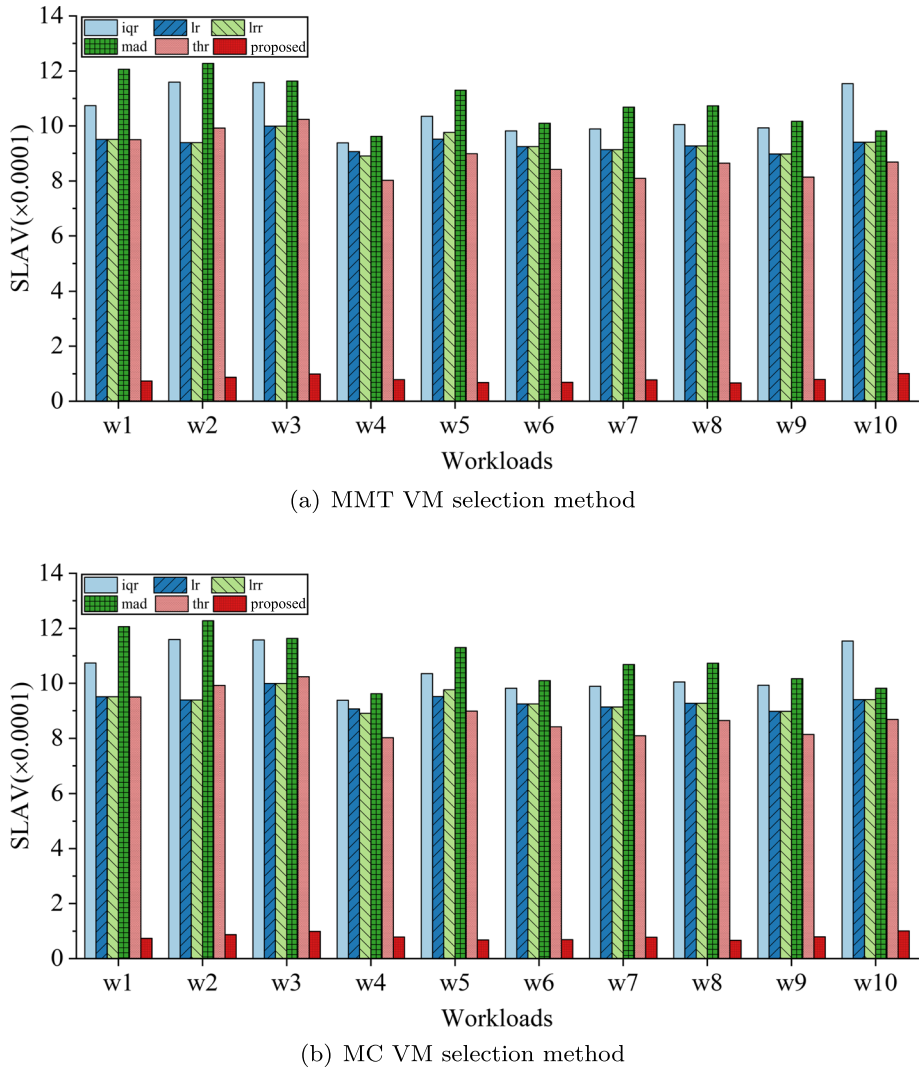


Fig. 6 SLAV of the cloud data center based on the PABFD and the proposed approach are implemented with different VM selection method

conditions about SLA violation and energy consumption. The value about ESV is lower, which symbols that the proposed approach has better perfomrmance in improvement of energy saving and quality of service in the cloud data centers.

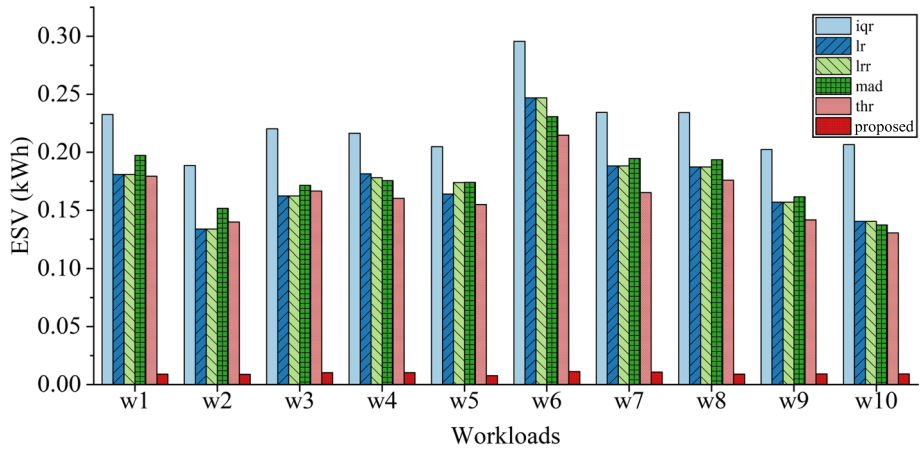
Simulation results based on different VM selection strategy

To validate the proposed VM selection strategy, the performance of AUMT is evaluated by the performance metrics and compared with the state-of-the-art method EQV [7] by using host status method IQR embedd with PABFD.

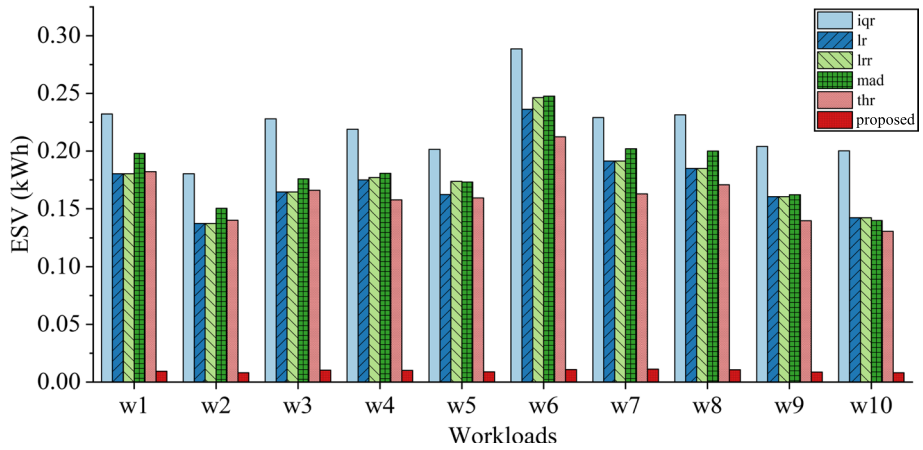
The simulation results as presented by Fig. 8 depict energy consumption generated by all hosts using different VM selection approach in the data center. The AUMT can reduce energy consumption compared with MMT, MC and EQV by an average of 17.33%, 18.24% and 15.46% respectively. The result demostrates that the AUMT has better performance in reducing energy consumption.

The simulation results as presented by Fig. 9 depict number of migrations by all hosts using different VM selection approach in the data center. The AUMT can reduce number of migrations compared with MMT, MC and EQV by an average of 23.95%, 26.24% and 28.11% respectively. The results show that the AUMT has advantage in reducing addtional migrations.

The simulation results as presented by Fig. 10 show the performance metrics about SALV and ESV using



(a) MMT VM selection method



(b) MC VM selection method

Fig. 7 ESV of the cloud data center based on the PABFD and the proposed approach are implemented with different VM selection method

different VM selection method in the data center. The AUMT enables to reduce SLAV compared with MMT and MC by an average of 37.55% and 36.31% but to increase SLAV compared with EQV by an average of 11.08% as shown in the Fig. 10a. However, the AUMT can reduce ESV compared with MMT, MC and EQV by an average of 47.99%, 47.47% and 3.96% respectively as shown in the Fig. 10b. The results draw a conclusion that the AUMT has a better performance in ESV combined with SLAV and energy consumption contrasted with EQV.

Conslusion and future work

To achieve the goal of reducing energy consumption while guaranteeing the QoS for the cloud data center. In this paper, host's status detection by predicting the CPU utilization based on combined prediction model to ensure the host's future states about overlaoded and underloaded to reduce addtional migrations and reduce the amount of host with low CPU utilization to switch its to idle state to save energy, and VM monitor is activated to trigger migration then to use proposed policy, the fumula about the score calculated by the Eq. (22) based on the live CPU utilization of host and energy consumption generated by a VM migrated to the targeted host, to search most proper host to host VMs. The simulation results based on real plantlab workloads demostrate that the proposed approach compared with benchmark method can acheive the objectives of reducing energy consumption, number of VM migrations, SLA violations and ESV by an average of 35.30%, 79.21%, 91.01% and 94.29% respectively and the AUMT can reduce the data centers of energy consumption, number of migrations and ESV by an average

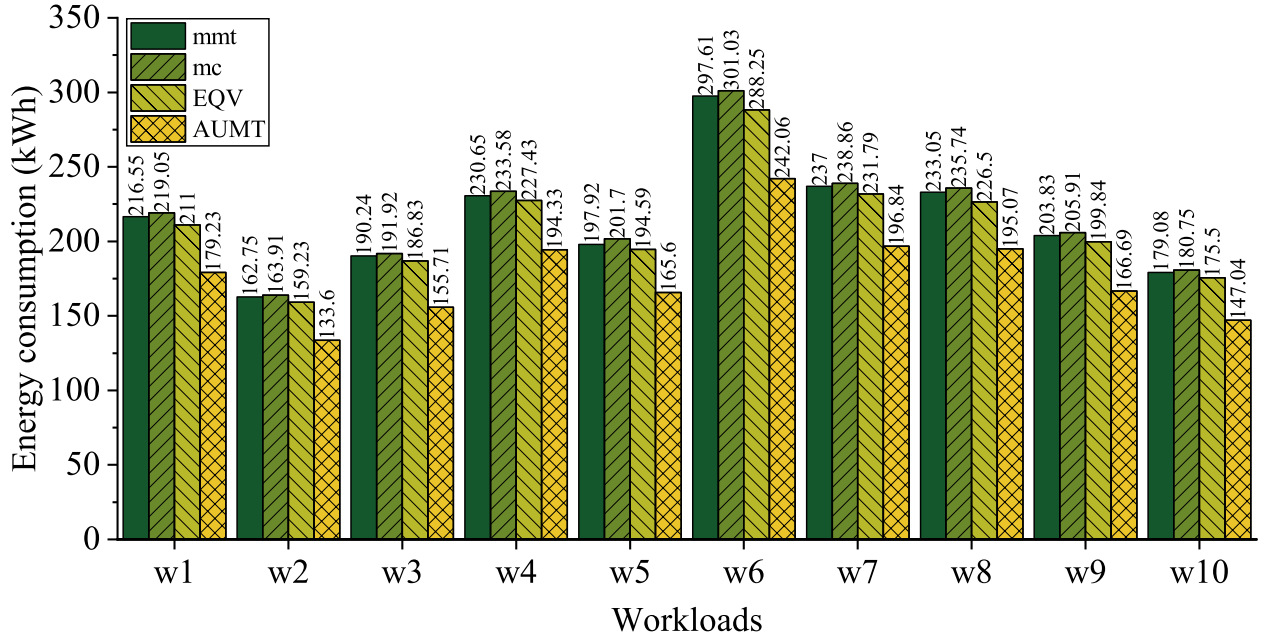


Fig. 8 Energy consumption of the cloud data center based on the PABFD is implemented with different VM selection method

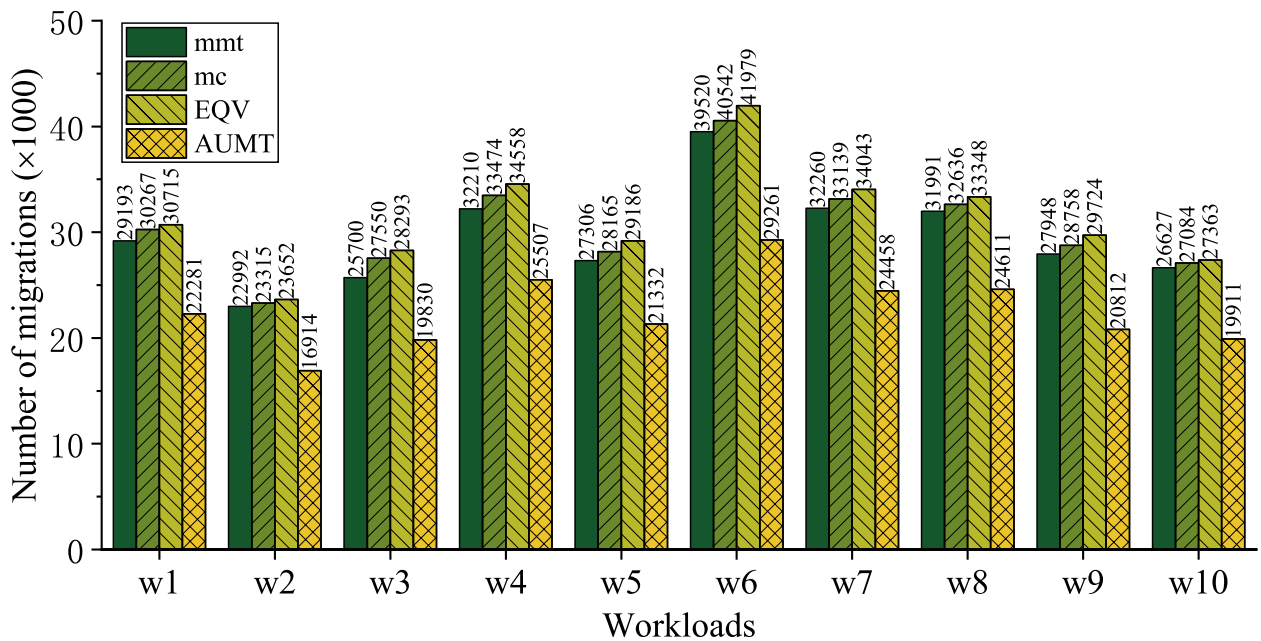


Fig. 9 Number of migrations of the cloud data center based on the PABFD is implemented with different VM selection method

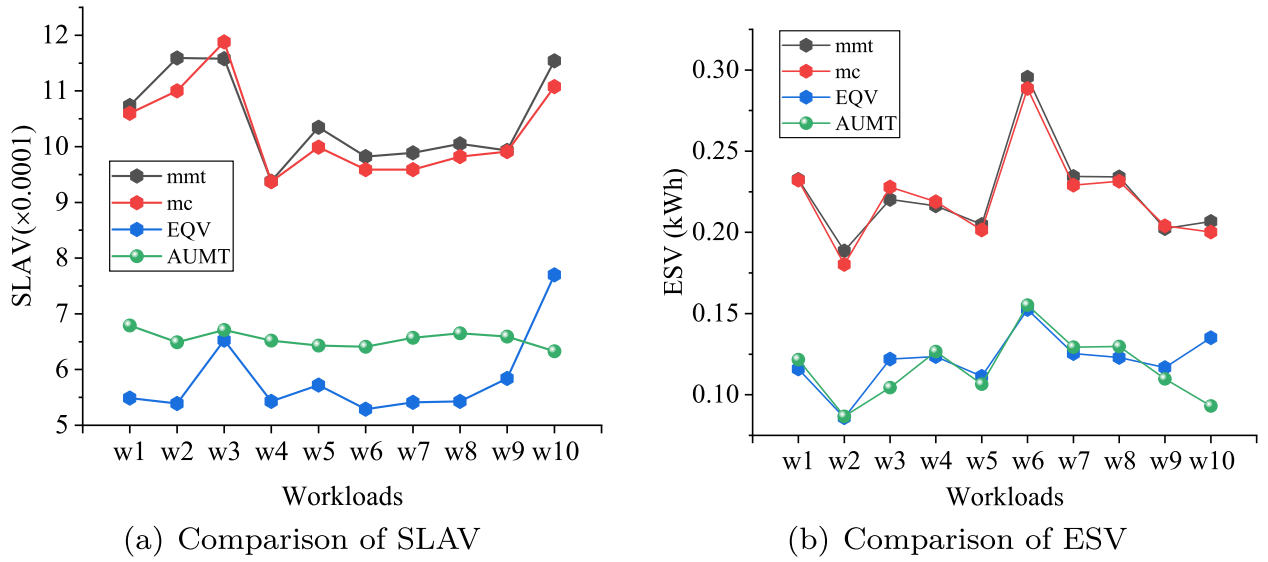


Fig. 10 Performance metrics about SLAV and ESV of the cloud data center based on the PABFD is implemented with different VM selection method

of 15.16%, 28.11% and 3.96% compared with the cutting-edge method EQV. In sum, extensive experimental results validates the performance and effectiveness of the approach.

In the future, we will test our approaches on real cloud platforms (such as video cloud computing platforms and openstack) to verify the performance of the proposed strategy in terms of energy consumption, the number of migrations, SLA violations and guaranteeing the QoS of the cloud data center [42].

Acknowledgements
The authors would like to thank the anonymous reviewers for their insightful comments and suggestions on improving this paper.

Authors' Contributions
All authors contributed to the study's conception and design. Material preparation, data collection and analysis were performed by Jinjiang Wang, Junyang Yu, Yixin Song and Hangyu Gu. The first draft of the manuscript was written by Jinjiang Wang and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding
This work was supported in part by Science and Technology R &D Project of Henan Province (Grant No. 212102210078) and the Key Science and Technology Project of Henan Province (Grant No. 201300210400).

Availability of data and materials
The datasets used during the current study are available from the corresponding author on reasonable request.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 11 May 2022 Accepted: 9 August 2022
Published online: 24 September 2022

References

1. Koomey JG (2007) Estimating total power consumption by servers in the us and the world
2. Shehabi A, Smith SJ, Sartor DA, Brown RE, Herrlin M, Koomey JG, Masanet ER, Horner N, Azevedo IL, Lintner W (2016) United states data center energy usage report
3. Barham P, Dragovic B, Fraser K, Hand S, Harris T, Ho A, Neugebauer R, Pratt I, Warfield A (2003) https://doi.org/10.1145/1165389.945462Xen and the art of virtualization. SIGOPS Oper Syst Rev 37(5):164–177
4. Leelipushpam PGJ, Sharmila J (2013) Live VM migration techniques in cloud environment - a survey. In: 2013 IEEE Conference on Information \& Communication Technologies. IEEE, p 408–413
5. Sobel W, Subramanyam S, Sucharitakul A, Nguyen J, Wong H, Klepchukov A, Patil S, Fox A, Patterson D (2008) Cloudstone: Multi-platform, multi-language benchmark and measurement tools for web 2.0. Work Cloud Comput Appl 8:228
6. Pahlevan A, Qu X, Zapater M, Atienza D (2017) Integrating heuristic and machine-learning methods for efficient virtual machine allocation in data centers. IEEE Trans Comput Aided Des Integr Circ Syst 37(8):1667–1680. IEEE
7. Tarafdar A, Debnath M, Khatua S et al (2020) Energy and quality of service-aware virtual machine consolidation in a cloud data center. J Supercomput 76:9095–9126. https://doi.org/10.1007/s11227-020-03203-3
8. Monil MAH, Rahman RM (2015) Implementation of modified overload detection technique with VM selection strategies based on heuristics and migration control. In: 2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS). IEEE, p 223–227
9. Cao Z, Dong S (2014) Energy-aware framework for virtual machine consolidation in cloud computing. In: 2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing. IEEE, p 1890–1895
10. Greenberg A, Hamilton J, Maltz DA, Patel P (2008) The cost of a cloud. ACM SIGCOMM Comput Commun Rev 39(1):68–73
11. Takouna I, Alzaghoul E, Meinel C (2014) Robust virtual machine consolidation for efficient energy and performance in virtualized data centers. In: 2014 IEEE International Conference on Internet of Things (iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom). IEEE, p 470–477
12. Beloglazov A, Buyya R (2012) Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints. IEEE Trans Parallel Distrib Syst 24(7):1366–1379

13. Beloglazov A, Buyya R (2012) Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. Concurr Comput Pract Experience 24(13):1397–1420
14. Melhem SB, Agarwal A, Goel N, Zaman M (2018) Markov prediction model for host load detection and vm placement in live migration. IEEE Access 6:7190–7205. http://dx.doi.org/10.1109/ACCESS.2017.2785280. https://doi.org/10.1109/ACCESS.2017.2785280
15. Wu Q, Ishikawa F, Zhu Q, Xia Y (2016) Energy and migration cost-aware dynamic virtual machine consolidation in heterogeneous cloud datacenters. IEEE Trans Serv Comput 12(4):550–563. IEEE
16. Ashraf A, Porres I (2018) Multi-objective dynamic virtual machine consolidation in the cloud using ant colony system. Int J Parallel Emergent Distrib Syst 33(1):103–120. Taylor \& Francis
17. Farahnakian F, Liljeberg P, Plosila J (2013) LiRCUP: Linear regression based CPU usage prediction algorithm for live migration of virtual machines in data centers. In: 2013 39th Euromicro conference on software engineering and advanced applications. IEEE, p 357–364
18. Haghshenas K, Mohammadi S (2020) Prediction-based underutilized and destination host selection approaches for energy-efficient dynamic VM consolidation in data centers. J Supercomput 76(12):10240–10257. Springer
19. Li L, Dong J, Zuo D, Wu J (2019) SLA-aware and energy-efficient VM consolidation in cloud data centers using robust linear regression prediction model. IEEE Access 7:9490–9500. IEEE
20. Li Z, Yu X, Yu L, Guo S, Chang V (2020) Energy-efficient and quality-aware vm consolidation method. Futur Gener Comput Syst 102:789–809
21. Laili Y, Tao F, Wang F, Zhang L, Lin T (2018) An iterative budget algorithm for dynamic virtual machine consolidation under cloud computing environment. IEEE Trans Serv Comput 14(1):30–43
22. Sharma Y, Si W, Sun D, Javadi B (2019) Failure-aware energy-efficient vm consolidation in cloud computing systems. Futur Gener Comput Syst 94:620–633
23. Jheng J-J, Tseng F-H, Chao H-C, Chou L-D (2014) A novel VM workload prediction using Grey Forecasting model in cloud data center. In: The International Conference on Information Networking 2014 (ICOIN2014). IEEE, p 40–45
24. Chehelgerdi-Samani M, Safi-Esfahani F (2021) PCVM. ARIMA: predictive consolidation of virtual machines applying ARIMA method. J Supercomput 77(3):2172–2206. Springer
25. Xu F, Liu F, Liu L, Jin H, Li B, Li B (2014) iAware: Making Live Migration of Virtual Machines Interference-Aware in the Cloud. In: IEEE Trans Comput, vol. 63, no. 12. pp 3012–3025 https://doi.org/10.1109/TC.2013.185
26. Xu F, Liu F, Jin H (2016) Heterogeneity and Interference-Aware Virtual Machine Provisioning for Predictable Performance in the Cloud. In: IEEE Transactions on Computers, vol. 65, no. 8. pp 2470–2483. https://doi.org/10.1109/TC.2015.2481403
27. Xu F, Liu F, Jin H, Vasilakos AV (2014) Managing Performance Overhead of Virtual Machines in Cloud Computing: A Survey, State of the Art, and Future Directions. In: Proceedings of the IEEE, vol. 102, no. 1. pp 11–31. https://doi.org/10.1109/JPROC.2013.2287711
28. Liu F, Zhou Z, Jin H, Li B, Li B, Jiang H (2014) On Arbitrating the Power-Performance Tradeoff in SaaS Clouds. In: IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 10. pp 2648–2658. https://doi.org/10.1109/TPDS.2013.208
29. Deng W, Liu F, Jin H et al (2014) Reliability-aware server consolidation for balancing energy-lifetime tradeoff in virtualized cloud datacenters[J]. Int J Commun Syst. 27(4):623–642
30. Syh A, Csl A, Rb B, Ayz C (2020) Utilization-prediction-aware virtual machine consolidation approach for energy-efficient cloud data centers - sciencedirect. J Parallel Distrib Comput 139:99–109
31. Calheiros RN, Masoumi E, Ranjan R, Buyya R (2015) Workload prediction using arima model and its impact on cloud applications' qos. IEEE Trans Cloud Comput 3(4):449–458. https://dx.doi.org/10.1109/TCC.2014.2350475
32. Shasha W, An C, Jing S, Shuo L (2009) Application of the combination prediction model in forecasting the gdp of china(in chinese). J Shandong Univ (Nat Sci) 44(2):4
33. Nathuji R, Schwan K (2007) Virtualpower: coordinated power management in virtualized enterprise systems. ACM SIGOPS Oper Syst Rev 41(6):265–278
34. Voorsluys W, Broberg J, Venugopal S, Buyya R (2009) Cost of virtual machine live migration in clouds: A performance evaluation. In: IEEE international conference on cloud computing. Springer, p 254–265
35. Julong D (1989) Introduction to grey system theory. J Grey Syst 1(1):1–24
36. Adhikari R, Agrawal RK, An introductory study on time series modeling and forecasting. arXiv preprint arXiv:1302.6613
37. Beloglazov A, Buyya R (2013) Managing Overloaded Hosts for Dynamic Consolidation of Virtual Machines in Cloud Data Centers under Quality of Service Constraints. IEEE Trans Parallel Distrib Syst. 24(7):1366–1379
38. Calheiros RN, Ranjan R, Beloglazov A, De Rose CA, Buyya R (2011) Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Softw Pract Experience 41(1):23–50
39. Park K, Pai VS (2006) https://doi.org/10.1145/1113361.1113374Comon: A mostly-scalable monitoring system for planetlab. SIGOPS Oper.Syst Rev 40(1):65–74
40. Ferdaus MH, Murshed M, Calheiros RN, Buyya R (2017) Multi-objective, decentralized dynamic virtual machine consolidation using aco metaheuristic in computing clouds. arXiv preprint arXiv:1706.06646
41. Murtazaev A, Oh S (2011) Sercon: Server consolidation algorithm using live migration of virtual machines for green computing. IETE Tech Rev 28(3):212–231
42. Xiong FU, Chen Z (2015) Virtual machine selection and placement for dynamic consolidation in cloud computing environment. Front Comput Sci China (Engl) 2:9

Publisher's Note