Taylor & Francis
Taylor & Francis Group

Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/taut20

Uniform distribution elephant herding optimization (UDEHO) based virtual machine consolidation for energy-efficient cloud data centres

G. Kanagaraj & G. Subashini

To cite this article: G. Kanagaraj & G. Subashini (2023) Uniform distribution elephant herding optimization (UDEHO) based virtual machine consolidation for energy-efficient cloud data centres, Automatika, 64:3, 529-539, DOI: 10.1080/00051144.2023.2196116

To link to this article: https://doi.org/10.1080/00051144.2023.2196116

AUTOMATIKA
2023, VOL. 64, NO. 3, 529–539
https://doi.org/10.1080/00051144.2023.2196116

REGULAR PAPER

OPEN ACCESS Check for updates

Uniform distribution elephant herding optimization (UDEHO) based virtual machine consolidation for energy-efficient cloud data centres

G. Kanagaraj[a] and G. Subashini[b]

[a]Department of Computer Science and Engineering, Kumaraguru College of Technology, Coimbatore, India; [b]Department of Robotics and Automation Engineering, PSG College of Technology, Coimbatore, India

ABSTRACT
Information technology (IT) providers should use cloud-based services due to their flexibility, reliability, and scalability to handle the rising requirement for processing capacity. The maintenance of dependable services between cloud providers and their customers in a cloud environment, on the other hand, depends on Quality of Service (QoS) assurance. Virtual machine (VM) consolidation is nondeterministic polynomial time (NP) hard issue, and numerous heuristic techniques have been suggested to solve it. In this work, the suggested VM consolidation technique takes into account both current and future uniform distribution elephant herding optimization (UDEHO) based VM consolidation approaches for resource utilization via host overload detection (utilization prediction based potential overload detection (UP-POD)) and host underload detection (UP-PUD). A UDEHO method efficiently predicts resource use in the future. Depending on the power utilization and the number of migrations, a power-saving value is advised for identifying under-loaded hosts. Furthermore, the CloudSim toolkit is used to construct and test these techniques using the same experimental parameters. Lastly, the findings demonstrate that the suggested methodologies considerably decrease the number of VM migrations by about 0.073%, the energy usage of about 11%, and SLA violations by 6.15% while retaining QoS guarantees when compared to conventional techniques.

ARTICLE HISTORY
Received 20 November 2022
Accepted 23 March 2023

KEYWORDS
Cloud computing; cloud data centres; nondeterministic polynomial time (NP); utilization prediction model; Quality of Service (QoS); dynamic virtual machine (VM) consolidation; and uniform distribution elephant herding optimization (UDEHO)

1. Introduction

Due to the major advantages of the cloud for users and providers in the areas of economics, the environment, and technology, adoption and migration of the cloud are on the rise worldwide. Cloud computing has quickly evolved into one of the contemporary economy's backbones since its inception. Government agencies, academic institutions, and commercial enterprises are all cloud users who have adopted it and reaped its benefits to a significant extent. Cloud computing also allows for the rapid establishment of new companies, the worldwide expansion of enterprises, speeding up scientific investigation, and creating of new applications and models. For customers that desire pay-as-you-go on-demand access to services, cloud providers can provide a variety of cloud services [1,2]. Numerous companies that offer public cloud services, like Amazon, Yahoo, and Microsoft, construct enormous cloud data centres worldwide the globe to provide cloud computing services to their clients [3].

Cloud data centres should preferably assign resources to users in a manner that matches the requisite Quality of Service (QoS) as defined by cloud customers via service level agreement (SLA). An SLA in cloud computing is described as a two-sided contract between a cloud provider and its customers that specifies the content of services offered, the degree of performance, fees, and penalties for failure to supply the services. Any breach of the QoS results in an SLA violation, and as a result, service providers must pay a penalty [4]. As a result, using less energy is the main objective of this investigation in cloud data centres while maintaining QoS standards. Cloud infrastructures have grown increasingly difficult and complex as a result of the fast rise of cloud services and their accompanying technologies. As a result, amongst the highly significant difficulties in current cloud systems is resource management, which has a direct impact on the successful deployment of cloud services. Hence, guaranteeing that the fewest amount of physical machines (PMs) reduce energy expenses in cloud data centres effectively by being operational.

Consolidation of VMs is one amongst the most effective techniques in cloud computing's energy-efficient resource management; this strategy improves resource usage while decreasing energy usage. Consolidation is the live movement of VMs across hosts with minimal performance disruption. Consolidation aims to reduce the number of hosts hosting virtual machines (VMs) while setting idle hosts to power-saving modes [5].

Static and dynamic VM consolidation is the two main kinds. When a job comes, the dimensions and

CONTACT G. Kanagaraj kanagaraj.techguy@gmail.com Kumaraguru College of Technology, Athipalayam Rd, Chinnavedampatti, Coimbatore, Tamil Nadu 641049, India

location of VMs on PMs are predetermined, and the positioning does not alter over time. Therefore, since PMs resources for various types of VMs are established, this type of VM consolidation is typically acceptable for short-running tasks lasting a few hours [6]. The majority of energy savings are based on basic heuristics on past VMs demand trends. Despite the possibility of an increase in the application provider's costs in times of both high and low demand, the resources available may be inadequate [7]. By moving VMs across PMs or live migrating them, dynamic VM consolidation can result in the utilization of fewer PMs without significantly disrupting services. It considers performance since it is dependent on QoS, which is set by SLA among the tenant and the service provider. It improves data centre power effectiveness by shutting off underutilized servers to conserve energy [8]. Dynamic provisioning-based energy usage may be a highly effective way for improving resource use and lowering energy usage.

By moving VMs across PMs or live migrating them without significantly disrupting services, dynamic VM consolidation can result in the utilization of fewer PMs. It considers performance since it is dependent on QoS that is established by an SLA among the tenant and the service provider. This will improve data centre power effectiveness by shutting off underutilized servers to conserve energy. Energy utilization based on dynamic provisioning may be the most efficient method for improving resource use and lowering energy usage. A great way to reduce resource and energy consumption is through dynamic VM consolidation [9]. Many VMs are hosted on the same physical server using hardware virtualization technology, and each VM can run single or several applications. Furthermore, individual tasks may be divided among fewer servers, thanks to hardware virtualization, increasing resource efficiency. VMs may be condensed and packed on fewer PMs employing live VM migration methods, lowering energy usage. VM consolidation is often divided into four parts [10]: selecting VMs, locating under-loaded hosts, finding overloaded hosts, and placing VMs. The difficulty of VM consolidation is mostly addressed in the first and second stages of the work. More specifically, anytime a host is found to be overloaded, a few of the VMs on that server must be carefully picked for migration to other acceptable hosts. Switching a host's power state from idle to low-power and vice-versa wastes extra energy. As a result, switching hosts' states is crucial to save power, but limiting their frequency is more critical.

The technique is employed in this research to forecast short-term future resource use formed on past data from sample hosts. This article suggests a VM consolidation method that considers present and future uniform distribution elephant herding optimization (UDEHO) based VM consolidation method for resource usage via host UP-PUD. As a result, cloud providers can improve energy effectiveness and the SLA performance assurance. The suggested technique decreases energy usage while limiting the number of migrations through various simulations based on real-world workloads. As a result, it improves cloud data centre performance with an increased SLA performance guarantee.

2. Literature review

Takouna et al. [11] presented a robust consolidation strategy to attain energy-performance equilibrium. The suggested method is made up of three techniques: overutilized host identification, VM location, and choice. Additionally, wasteful VM migration is minimized by using an adaptive historical window selection technique. The CloudSim simulator was used to create the concept, and simulations of a genuine Planet Lab workload trace were conducted for several days to assess it. Moreover, it can reduce network energy usage as a consequence of VM relocation.

Farahnakian et al. [12] provided a framework for predicting computer processing unit (CPU) consumption relying on the linear regression (LR) approach. Depending on the history of usage in every host, the suggested technique approximates the short-term future CPU consumption. During the live migration procedure, it is employed to foresee overloaded and underloaded hosts. The host then enters sleep mode to decrease power utilization. The suggested approach can dramatically reduce energy utilization and SLA violation rates, based on test results from over a thousand Planet Lab VMs with actual workload traces. CPU consumption forecast may simply lead to unnecessary migrations, increasing the overhead like VM migration energy costs, performance deterioration due to migration, and additional traffic.

Mastroianni et al. [13] eco Cloud, a self-organizing and flexible solution for VM consolidation on two resources: CPU and Random Access Memory, was created. The method is very simple to implement since decisions about VM allocation and migration are made using probabilistic algorithms using just local data. A fluid-like mathematical model and tests on a real data centre show that the method quickly combines workloads and balances VMs that are CPU- and RAM-bound, enabling efficient use of both resources. For cloud data centres with fluctuating workloads, resource usages are ineffective measurement methodologies.

Hieu et al. [14] presented VMCUP-M to increase cloud data centre energy efficiency. Multiple usages in this respect apply to both resource kinds and the time range used to forecast future consumption. The suggested technique is run during the VM consolidation procedure to base on particular server history, forecast the long-term usage of various resource kinds. Findings indicate that a union with numerous use predictions lowers migrations number and server power usage while meeting SLA. Even though the criteria under consideration are not static, these techniques make VM migration choices based only on current resource use.

Ismaeel et al. [15] conducted a thorough review with a focus on energy conservation, the most modern proactive dynamic provisioning architecture in a data centre. Cloud data centres (CDCs) with diverse settings are the focus of proactive dynamic VM consolidations. A general structure is described, along with numerous steps that result in a comprehensive consolidation process. It is critical to ensure that the level of QoS is preserved in accordance with the SLA while attempting to fully use data centre resources.

Sayadnavard et al. [16] proposed a methodology for predicting future resource use using discrete-time Markov chain (DTMC). Through the DTMC framework in conjunction with the PM reliability model results in more accurate PM categorization depending on their state. Then, using the multi-objective VM placement methodology, which is based on the dominance-based multi-objective artificial bee colony technique, the ideal VMs to PMs mapping is obtained. This mapping can efficiently balance resource waste, energy consumption, and system reliability to meet SLA and QoS requirements. A performance evaluation study using the Clouds tool shows the recommended strategy's effectiveness.

Liu et al. [17] suggested by maintaining VMs that are prone to migration thrashing on the same physical servers rather than relocating them, the dynamic consolidation with minimization of migration thrashing (DCMMT) framework prioritizes VMs with high capacity while drastically reducing the number of migrations necessary to provide SLA. The suggested technique spreads present VM consolidation techniques by requiring that high-capacity VMs not be transferred.

The recommended technique reduces the number of servers used and, to the maximum degree possible, gets rid of migration thrashing. The findings of these studies were encouraging, as data centres may readily benefit from the DCMMT process because it requires little adjustments to be added to an existing resource management system.

Hsieh et al. [18] suggested a VM consolidation method which considers present UP-POD and UP-PUD. A Gray–Markov (GM)-based model correctly predicts resource consumption in the future. The new methodologies were applied to real-world workload traces in clouds and contrasted against current benchmark techniques in the test. Jheng et al. [19] offered the first string in the study field, a workload prediction approach employing the GM forecasting model to distribute VMs.

To begin, the time-dependent workload is employed at the same time every day to anticipate the VM workload inclination to increase or decrease. Then, contrast the expected value to the prior time period on workload consumption, and then decide which VM should be migrated to PM for a balanced workload and reduced power usage. The findings of the simulation indicate that the suggested strategy not only employs fewer data points to effectively forecast workload but also distributes VM resources in a power-efficient manner.

Table 1 shows the comparison of the existing approaches with merits and demerits. Hence by analysing the existing approaches of cloud computing solutions that may not solely minimize operational prices but conjointly cut back the environmental impact. By setting up many virtual machine (VM) instances on a real server using virtualization, cloud providers may address the energy inefficiency, maximizing resource usage and ROI (ROI). Switching off idle nodes will result in a reduction in energy usage since they no longer use any power when idle. Additionally, live migration enables the VMs to be dynamically condensed on the fewest possible physical nodes in accordance with their current resource needs. However, modern service applications frequently encounter very unpredictable workloads that result in dynamic resource use patterns, making efficient resource management in clouds challenging.

Table 1. Comparison of the existing approaches.

Authors	Approaches	Results/Merits	Demerits
Takouna et al. [11]	Three algorithms over-utilized host identification, VM selection, and VM placement make up the suggested methodology.	According to the findings, our method decreased the average number of SLA breaches by 31.8%, 74.8%, and 38%, respectively, and the number of power changes and migrations.	Implementing cloud infrastructure in real time is challenging.
Farahnakian et al. [12]	Algorithm for Predicting CPU Usage Based on Linear Regression	Energy use and SLA violation rates are dramatically reduced.	Between dependent and independent variables is assumed to be linear
Mastroianniet al. [13]	An adaptive and self-organizing method for VM consolidation	Good scalability is achieved	As a data centre's size increases, its efficiency diminishes.
Farahnakian et al. ([26])	Using an ant colony approach, energy-conscious dynamic VM consolidation is achieved in cloud data centres.	It increases how effectively PMs use resources and lowers how much energy they use.	When dealing with a huge amount of data, the ACO has several drawbacks in terms of convergence speed and solution correctness.
Hieu et al. [14]	To increase energy efficiency, use the virtual machine consolidation algorithm with multiple usage prediction (VMCUP-M).	While maintaining the SLA, decreases the number of server migrations and the power consumption of the servers.	It is difficult to assess the suggested algorithm's performance in actual data centres.

Therefore, if VMs are aggressively consolidated, applications will perform worse as demand rises, resulting in higher resource utilization. The energy-performance trade-off is another issue that cloud service companies must handle. This research focuses on resource management techniques that may be used by a provider in a virtualization CDC and that are both energy and performance efficient.

3. Proposed methodology

Here, an effective usage prediction technique based on the UDEHO model is used to anticipate short-term future CPU consumption based on data collected from the considered hosts. An efficient VM consolidation strategy is shown that maximizes VM placement for the greatest projected benefit reducing the number of state hosts in an active state. The advantage is derived from two major factors: The quantity and frequency of SLA breaches during VM migrations. Furthermore, integrating data on current and near-future CPU consumption provides a reliable system for classifying hosts that are overloaded and underloaded. This problem may be handled by detecting host overload UP-POD and host underload UP-PUD. When overloaded hosts are found and when underloaded hosts are found, whole VMs with a potential increase in CPU consumption are relocated from these hosts to maintain QoS, and entire VMs from these hosts are relocated to save energy usage. As a result, cloud providers may enhance the energy effectiveness and performance guarantee of SLA. The recommended utilization-prediction-aware VM consolidation method for cloud data centres is divided into several pieces in this section. Sections 3.1 and 3.2 go into the specifics of the cloud data centre's system architecture and give a UDEHO prediction model. Furthermore, and most importantly, Section 3.3 presents the resource usage prediction methods (UP-POD and UP-PUD) and power saving value depending on expected CPU use. Figure 1 depicts the suggested system design.

3.1. System architecture

In a cloud data centre, the suggested technique comprises m heterogeneous hosts (that is, $H = \langle h_1, h_2..., h_m\rangle$). Various resource types, like CPU, size of memory, network bandwidth, and storage capacity, distinguish every host. Furthermore, CPU performance is typically evaluated in MIPS. A cloud data centre's services are used by numerous users at the same time. Users request the provisioning of n VMs (that is, $V = \langle v_1, v_2..., v_n\rangle$). The best fit decreasing (BFD) approach is used to first allocate VMs to hosts, that is, among most extensively used heuristic techniques for bin-packing problems.

The BFD method eliminates all unutilized space in destination hosts. The system chooses a host whose existing resources are nearby to the quantity of resources sought by VM. This describes why the BFD method executes the first allocation of VMs so well. Nevertheless, because of dynamic workloads with frequent fluctuation, operating hosts' and virtual

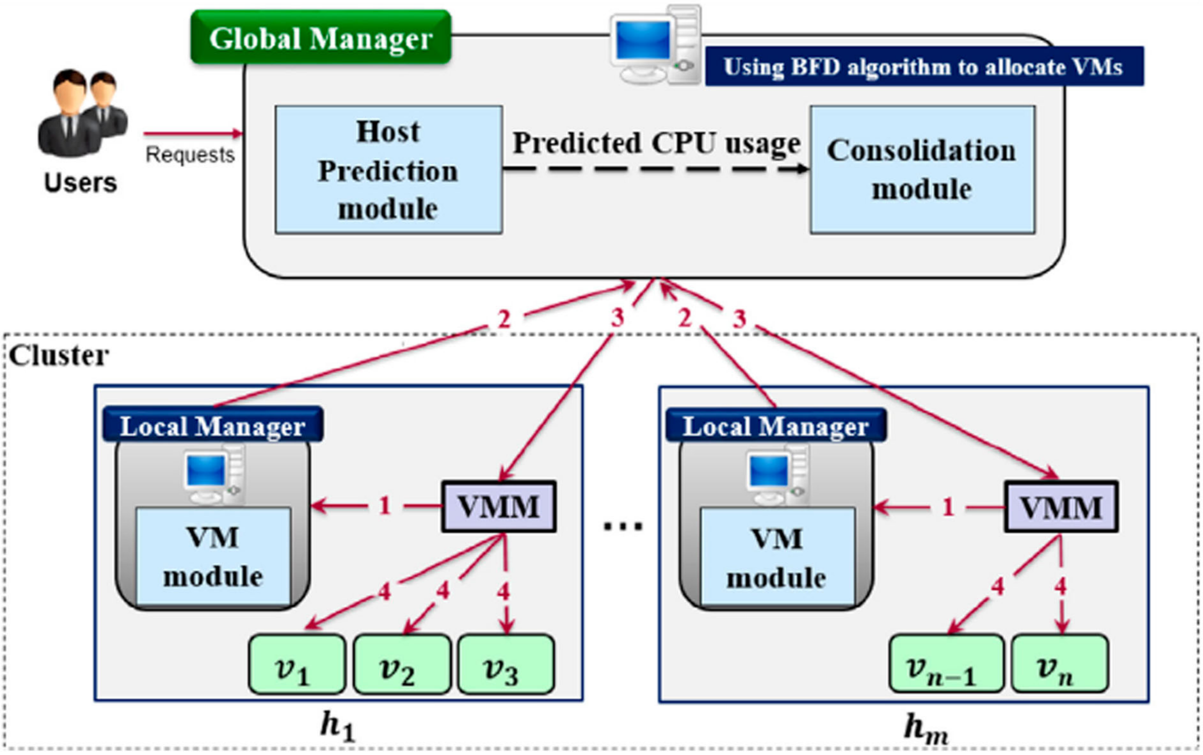

Figure 1. Suggested system architecture.

machines' requested usages evolve over time: thus, the initial allocation method is improved with a VM consolidation method which is implemented on a regular basis to enhance cloud data centres' performance. To reduce energy costs and the number of active state hosts, the recommended solution is implemented every 5 min in a cloud data centre. System architecture is made up of two sorts of agents: (1) a global manager installed on a master node and (2) completely dispersed local managers (LMs) scattered across entire hosts. At every cycle, two agents do the subsequent actions:

(1) Every LM monitors present resource use of entire VMs on a host on a regular basis. Every LM correctly forecasts a host's future CPU use based on past data in a log file by using the UDEHO prediction algorithm.
(2) For the purpose of understanding the overall position of hosts, the GM solicits status information from LMs.
(3) To implement the UP-POD and UP-PUD techniques of the recommended methodology, GM sends migration commands to the VM monitor. Based on the consolidation methods, the instructions specify that VMs have to be transferred to which destination hosts.
(4) The VMMs migrate VMs after getting instructions from GM.

3.2. UDEHO algorithm

The EHO method [20,21] is a heuristic intelligence approach that relies on elephant migratory tendencies that are utilized for CPU prediction in the VM. The elephant herd possesses the following two features based on observation and research of the elephants. The first distinguishing feature is the presence of several clans in an elephant herd, for the future prediction of CPU in the VM, each group has its own patriarch, and its members adhere to his or her instructions. Another distinguishing feature of the herd is the absence of one adult male elephant. Whenever the elephants grow up, the young elephants will live apart from the elephants in order to anticipate the CPU in the VM in the future. UDEHO's goal is divided into two parts: clan update and separating. It is defined by Equation (1)

$$x_{n,i,j} = x_{i,j} + r * \alpha * (x_{b,i} - x_{i,j}) \tag{1}$$

wherein $x_{i,j}$ and $x_{n,i,j}$ are according to Elephant J's previous and current status in clan i; $\alpha \in [0,1]$ is a scaling factor; and $x_{b,i}$ is location with the greatest fitness data in clan i. r is a normal-distributed arbitrary value in the range [0, 1]. The updating procedure of most individuals is represented by Equation (1), however, the matriarch in every clan has not been changed for future prediction of CPU in the VM. As a result, the matriarch's updating process is depicted in Equations (2)–(3)

$$x_{n,i,j} = \beta * x_{c,i} \tag{2}$$

$$x_{c,i} = \frac{1}{n_i} \times \sum_{j=1}^{n_i} x_{i,j} \tag{3}$$

The uniform distribution function is used to obtain wherein β is the scale factor between [0, 1]. Equation (4) gives a general formula for uniform distribution's probability density function (PDF)

$$f(x) = \frac{1}{B - A} \tag{4}$$

Here A is the VM's forecast range for CPUs and $(B- A)$ the standard uniform distribution is the scaling parameter is the situation wherein $A = 0$ and $B = 1$. Centre location in clan $i$is $x_{c,i}$, which may be computed using Equation (3). For future CPU prediction in the VM, the elephant number in clan i is n_i. A connection exists between the knowledge about every member of the clan and the updating of the matriarch position (VM position) in Equation (2). From the second characteristic of the elephant herd, the separation operator may be taken out for use in the VM's following CPU prediction. Equation (5) describes the separation process

$$x_{w,i} = x_{min} + r * (x_{max} - x_{min}) \tag{5}$$

wherein $x_{w,i}$ is the location in clan i with the lowest fitness value; x_{max} and x_{min} are upper and lower bounds of elephant's location, correspondingly; r is an arbitrary integer with a normal distribution between [0, 1]. Individuals in EHO refresh themselves with data from other members of the clan in order to find an improved forecast value of CPU in the VM. As illustrated in Algorithm 1, the separation of people is substituted by the arbitrary production of new individuals, which excludes testing of new individuals for future CPU prediction in the VM. The notion of updating people from the original EHO was kept in the UDEHO method.

Algorithm 1. UDEHO Algorithm

```
Initialization. Set population and parameters
Fitness evaluation by the CPU requirement in the VM
While $t < T_{max}$ do
   for $i = 1 : n_c$do
      for $j = 1 to n_j$ (number of elephants in one clan) do
         modify $x_{i,j}$ and produce $x_{n,i,j}$ as per Equation (1)
         if $x_{i,j} = x_{b,i}$ then
            modify $x_{i,j}$ & produce $x_{n,i,j}$ as per equation (2-3)
         end if
      end for
   for $i = 1 : n_c$ do
      substitute worst elephant prediction of CPU in VM in clan i via
       Equation (5)
   end for
      Assess individuals (prediction of CPU in the VM) as per their
       new position
   end while
Result
```

Firstly, establish a key point. SepPoint is responsible for determining the probability Pr_a of executing the separation strategy. Equations (6)–(7) explain how to calculate SepPointand Pr_a, correspondingly.

$$\text{SepPoint} = n_c(i), i = \left\lceil \frac{(s_{n_c} + 1)}{2} \right\rceil \tag{6}$$

$$Pr_a = \text{SepPoint} * \frac{1}{n_c(k)} \tag{7}$$

wherein n_c denotes the number of clans, s_{n_c} the size of n_c, the index k for CPU prediction in the VM is determined by the current number of generations, and $n_c(k)$ how many clans there are today. Equation demonstrates (7), the more clans there are, the less likely it is that the separation process will be performed. And, after the number of clans influences SepPoint, the separation procedure is carried out. Next, I added an individual assessment link and set a low probability Pr. When a new person (VM) outperforms the present one in terms of evaluation, the departure is carried out. When a new person is not as excellent as the existing one, but rand $< Pr$, the separation process is carried out. The separation procedure is not carried out if this condition is not met. R and is a arbitrary value between [0, 1] with a normal distribution.

3.3. Resource utilization prediction algorithm

Resource usage is via host overload detection UP-POD and host underload detection UP-PUD.

3.3.1. Utilization prediction-based on overload detection

Every host, whether overcrowded or not, should be detected in every dynamic VM consolidation operation. Algorithm 2 presents the suggested usage prediction based potential overload detection (UP-POD), which is prompted by the latest work. Algorithm 2 takes as input a collection of active hosts H_{active}. It is identified which hosts in H_{active} are overloaded. Following that, the overloaded hosts are uploaded to Hover as output to carry out the migration choice. The following is a step-by-step explanation of Algorithm 2. Line 1 gives the GM $Ut_h(t)$, that is described as $Lo_h(t)$ (host h load at time t) divided by $Ca_h(t)$. In line 2, a time-series-based prediction model may be used to compute the short-term CPU utilisation (i.e. $Ut_h(t + 1)$) by compiling earlier information about a host's CPU usage that has been recorded in a log file. The UDEHO model is used to forecast $Ut_h(t + 1)$, which is shown in Section 3.1. Moreover, the past data on CPU use recorded at 5-min intervals in every host is used as time series data input. After preprocessing, the choice is determined on line 3 using the dynamic upper threshold approach. Here, a host is deemed overloaded if its usage exceeds the upper threshold. The dynamic upper threshold is determined using the median absolute deviation (MAD) method, and the parameter s is set at 2.5 in accordance with previous work. A measure of statistical dispersion is the MAD. It performs better with distributions devoid of a mean or variance, such as the Cauchy distribution, making it a more reliable estimator of scale than the sample variance or standard deviation. Compared to the standard deviation, the MAD is a robust statistic that can withstand outliers in a data collection. Outliers can have a significant impact on the standard deviation since it is based on squared deviations from the mean, which generally gives greater weight to big deviations. A limited number of outliers' distances are irrelevant to the MAD's calculations in terms of their size. In lines 5–9, the time-series model requires enough previous data to anticipate the $Ut_h(t + 1)$ correctly. If there is insufficient historical CPU use data on every host, a choice is made using $Ut_h(t)$. Tests using historical data with durations of 12, 16, 20, 24, and 28 were run during the simulation. Depending on the outcomes, the suggested methods outperform historical data with 24 lengths. Therefore, when historical data length $da_h(t)$ (host h historical data at time t) is smaller than 24, the decision is made by $Ut_h(t)$. If the hosts' $Ut_h(t)$ values are more than th_u, they will be regarded overloaded and added to Hover (upper threshold value of CPU utilization). In contrast, if $da_h(t)$ is more than 24, the host is deemed overloaded and is added to Hover when current and expected short-term CPU utilization values $> th_u$ $(Ut_h(t) > th_u$ and $Ut_h(t + 1) > th_u) >$. The scenario indicates that the host is a prospective candidate for carrying out when it is overloaded now and in the near future, the migration option. Algorithm 2 as a result looks at not only current situation and also scenario in the near future. Algorithm 2 can avoid redundant migrations, reducing overall number of migrations and executing an acceptable migration choice; furthermore, the SLA violation rate may be predicted in advance.

3.3.2. Utilization prediction-based underload detection

Following the identification of overloaded hosts, the underload detection method is initiated. To lower the number of active state host and hence decrease energy utilization, finding underused hosts and switching them over to low-power methods are crucial. Algorithm 3 presents the suggested UP-PUD. Algorithm 3 takes as input a collection of active hosts H_{active}. Which hosts in H_{active} are candidates of underloaded hosts are identified for each host, and these hosts are then added to H_{under} (set of candidates of underloaded hosts) as output. The following is a step-by-step explanation of Algorithm 3. Algorithm 3's process and idea are identical to those of Algorithm 2. The difference in lines 5–9, the host is added to H_{under} if $Ut_h(t) \leq th_l$ (lower threshold value of CPU usage). Furthermore,

Algorithm 2: UP-POD Algorithm

```
INPUT: $H_{active}$
OUTPUT: $H_{over}$
$Ut_h(t) = Lo_h(t)/Ca_h(t)$; /* Required MIPS(t)/Total MIPS(t) */
forecast $Ut_h(t+1)$; /* by UDEHO prediction */
initialize $th_u$; /* using MAD = 1−s*Mad */
For every $h \in H_{active}$ do
    if $Da_h(t) < 24$ then
        if $Ut_h(t) > th_u$ then
            return true;
        End
        Else
            return false;
        End
        End
        if $Ut_h(t) > th_u$ and $Ut_h(t+1) > th_u$ then
            return true;
        End
    Else
            return false
    End
End
```

Algorithm 3: UP-PUD algorithm

```
INPUT: $H_{active}$
OUTPUT: $H_{over}$
$Ut_h(t) = Lo_h(t)/Ca_h(t)$; /* Required MIPS(t)/Total MIPS(t) * /
forecast $Ut_h(t+1)$; /* by UDEHO prediction * /
initialize $th_l = 30\%$ ;
For every $h \in H_{active}$ do
    if $Da_h(t) < 24$ then
        if $Ut_h(t) < th_l$ then
            return true;
        End
        Else
            return false;
    End
    end
    if $Ut_h(t) \leq th_l$ and $Ut_h(t+1) \leq th_l$ then
            return true;
    End
    Else
        return false
    End
End
```

in 10–13 lines, $da_h(t)$ when the host is regarded to be underloaded and added to the list if it is at least $24H_{\text{under}}$ if the short-term CPU use that is now and anticipated $\leq th_l$ $(Ut_h(t) > th_l \text{and} Ut_h(t+1) > th_l)$.

Once Algorithm 3 selects and adds candidates of under-loaded hosts to H_{under}, the suggested $S_h$value is used to choose the ultimate under-loaded host from H_{under}. According to Fu et al. [22], the host's power usage is nearly proportionate to its CPU use. As a result, utilizing Equation (8), the power usage of every host may be estimated

$$P(\mu) = 0.7 * P_{\max} + 0.3 * P_{\max} * \mu \tag{8}$$

where in the notation $P_{\max}$ signifies the host's maximum power consumption value when fully loaded. The notation μrepresents the host's CPU use, which varies. As a result, under the VM consolidation strategy, the host's CPU usage is primarily evaluated for employment. The latest research [23] computes a power-efficient value for active state hosts. The value is used to choose an under-utilized host. By enhancing the power-efficient value, a power-saving value (S_h) is given in Equation (9) based on the prediction model. This metric is used to identify under-loaded hosts more accurately. Equation (9) describes it

$$S_h = \frac{P_h + \hat{P}_h}{M_h} \tag{9}$$

In Equation (9), P_h denotes the hth host's power usage in the cloud data centre, $\hat{P}_h$ indicates the hth host's power usage calculated using $Ut_h(t+1)$, and M_h denotes VMs number operating on the hth host. Lastly, as the under-loaded host, the host with the highest S_h value can be selected. Clearly, because S_h only considers the host's current power usage, power utilization at time $t+1$, and VM migrations' number, it will discover an under-loaded host more effectively.

4. Experimental setup

Workload categories, the simulation environment, and performance measures are used in this part to compare the efficiency of the suggested methodology versus current methods.

4.1. Workload data

Simulation employs workloads from the same 10-day period for efficient comparison with current work. The VMs' CPU usage correlates to their workloads and statistical analyses. Planet Lab data provided as part of the common project is used to conduct the research, which are based on real-world workloads that are publicly accessible: monitoring apparatus from Planet Lab. Workload information was gathered on 10 different days in March and April 2011 and comprises CPU use of a VM recorded at 5-min intervals. Every VM has 288 data on CPU consumption, which are fed into dynamic VM consolidation. Furthermore, information is compiled from over 1000 VMs hosted on servers at over 500 different locations throughout the world. Workload really represents an IaaS cloud environment, such as Amazon EC2, where individual users create and operate virtual machines.

4.2. Details of simulation environment

The study applies the Clouds 3.0.3 toolbox to objectively analyse the performance of the proposed short-term-based VM consolidation technique time series prediction [24,25]. A data centre with 800 disparate PMs was used in the simulation.

The HP ProLiant ML110 G4 servers have 1860 MIPS per core, whereas the HP ProLiant ML110 G5 servers have 2660 MIPS per core, making up half of the PMs in each workload. Every PM is designed to have two cores, four gigabytes of memory, and one gigabit per second of

Table 2. VM Information.

Type of VM	CPU (in MIPS)	RAM (in GB)
High CPU medium instance	2500	0.85
Extra-large instance	2000	3.75
Small instance	1000	1.7
Micro instance	500	0.613

network connectivity. Table 2 shows the CPU (in MIPS) rating and RAM quantity of 4 VM instances utilized in CloudSim that correspond to Amazon EC2.

4.3. Performance metrics

The suggested solution has four goals: (1) decrease power consumption, (2) lower SLA violation rates, (3) decrease the number of active state hosts, and (4) decrease migrations number. Consequently, the measures listed below are used to evaluate the effectiveness of the suggested technique and current methodologies.

4.3.1. SLA violations

In order to maintain the QoS promise in an IaaS between cloud service providers and consumers, Equation (10) is used to determine cloud service quality, resulting in a good SLA. Two metrics, SLAVO (SLA violations due to over-utilization) and SLA violations due to migration (SLAVM) are used to quantify SLA violations.

$$\text{SLAV} = \text{SLAVO} * \text{SLAVM} \tag{10}$$

where SLAVO denotes the average ratio during the time that the host uses 100% of its CPU, as shown in Equation (11)

$$\text{SLAVO} = \frac{1}{M}\sum_{i=1}^{M}\frac{T_{s_i}}{T_{a_i}} \tag{11}$$

Wherein M is the host number and T_{s_i} is the total time host i that suffered 100% CPU use, resulting in an SLA violation. The symbol T_{a_i} indicates period that host i is active. As illustrated in Equation (12), SLAVM indicates the total performance decrease caused by VM migrations

$$\text{SLAVM} = \frac{1}{N}\sum_{j=1}^{N}\frac{C_{d_j}}{C_{r_j}} \tag{12}$$

wherein N indicates the number of VMs, C_{d_j} indicates performance degradation due to migrating VM, and C_{r_j} represents total CPU usage demanded by VM j throughout its lifespan.

4.3.2. Energy consumption

According to several research, CPU resources consume increased power than memory, network interfaces, or disc storage. The energy use is calculated using real-world data from the SPEC power benchmark results. Significantly, whenever underutilized servers adopt low-power mode, their energy usage drops dramatically. As a result, limiting the number of active hosts is essential.

4.3.3. Number of VM migrations

As a result of higher CPU use on the source host, increased network bandwidth, application unavailability during VM migrations, and total migration time, live VM migration entails significant expenses and performance deterioration. Limiting VM migrations is essential since doing otherwise would almost surely lead to SLA violations.

4.3.4. Energy and SLA violations (ESV)

The suggested VM consolidation approach's major purpose is to decrease both energy costs and SLA breaches at the same time. Since there is still a trade-off between energy consumption and performance, Equation (13) illustrates a combined indicator called energy and SLA violations that may be used to properly analyse the trade-off

$$\text{ESV} = \text{E} \times \text{SLAV} \tag{13}$$

4.4. Comparison benchmarks

The suggested technique is contrasted with the techniques given for detecting overloaded hosts as follows for effective verification. The CloudSim simulator displays these methods.

(1) Static threshold (THR): the hot threshold is set to 90%. Hosts are deemed overloaded if their current CPU usage exceeds 90%.
(2) The MAD and interquartile range (IQR) are two adaptive criteria. The algorithm works in the same way as the THR. The latest research presents a thorough estimate of MAD and IQR.
(3) Dynamic threshold termed as the local regression (LR) technique: hosts that are overloaded are determined by calculating local regression changes over time.

Figure 2 depicts the performance of the SLAVO measure in contrast to THR, IQR, MAD, GM, and the suggested UDEHO method under 10 workloads. Conventional techniques do not outperform the suggested system in terms of performance. The suggested approach has a lower SLAVO value of 6.15%, while other systems like THR, IQR, MAD, and GM have higher SLAVO values for the 10th workload of 7.5%, 7.2%, 7.23%, and 6.75%, respectively (see Table 3).

Figure 3 depicts a performance comparison using the SLAVM measure. When compared to THR, IQR, MAD, and GM, the performance improves more since the technique caused a significant decrease in VM

Table 3. Performance Metrics Comparison of 10 Workloads under Methods.

	SLAVO (%)					SLAVM (%)				
Workloads	THR	IQR	MAD	GM	UDEHO	THR	IQR	MAD	GM	UDEHO
W1	7.12	6.84	7.05	6.20	5.80	0.092	0.094	0.087	0.0654	0.0620
W2	7.20	6.90	7.12	6.40	5.85	0.098	0.105	0.088	0.0630	0.0600
W3	7.40	7.00	7.16	6.90	6.30	0.109	0.108	0.075	0.0721	0.0700
W4	7.00	6.90	7.09	6.30	5.92	0.080	0.086	0.072	0.0680	0.0620
W5	7.00	6.75	7.16	6.50	6.12	0.094	0.090	0.082	0.0780	0.0720
W6	6.10	6.80	7.06	6.10	5.75	0.087	0.085	0.082	0.0740	0.0700
W7	6.90	6.70	6.85	6.00	5.63	0.098	0.086	0.072	0.0650	0.0610
W8	7.20	6.90	6.95	6.02	5.72	0.096	0.084	0.070	0.0620	0.0540
W9	6.60	6.10	6.36	6.50	5.91	0.097	0.096	0.083	0.0780	0.0690
W10	7.50	7.20	7.23	6.75	6.15	0.101	0.099	0.098	0.0850	0.0730

	SLAV (%)					Energy consumption (%)				
Workloads	THR	IQR	MAD	GM	UDEHO	THR	IQR	MAD	GM	UDEHO
W1	0.6550	0.64296	0.61335	0.40548	0.35960	188	182	175	135	115
W2	0.7056	0.7245	0.62656	0.40320	0.35100	140	138	134	110	105
W3	0.8066	0.7560	0.53700	0.49749	0.44100	158	153	149	129	116
W4	0.5600	0.5934	0.51048	0.42840	0.36704	194	190	184	152	135
W5	0.6580	0.6075	0.58712	0.50700	0.44064	165	162	155	124	109
W6	0.5307	0.5780	0.57892	0.45140	0.40250	240	234	230	190	162
W7	0.6762	0.5762	0.49320	0.39000	0.34343	198	194	190	143	116
W8	0.6912	0.5796	0.48650	0.37324	0.30888	194	190	184	156	132
W9	0.6402	0.5856	0.52788	0.50700	0.40779	199	195	192	128	109
W10	0.7575	0.7128	0.70854	0.57375	0.44895	140	135	131	118	103

	ESV METRIC (%)				
Workloads	THR	IQR	MAD	GM	UDEHO
W1	12.314752	11.701872	10.733625	5.473980	4.135400
W2	9.878400	9.998100	8.395904	4.435200	3.685500
W3	12.74428	11.56680	8.001300	6.417621	5.115600
W4	10.86400	11.27460	9.392832	6.511680	4.955040
W5	10.85700	9.841500	9.100360	6.286800	4.802976
W6	12.73680	13.52520	13.31516	8.576600	6.520500
W7	13.38876	11.17828	9.370800	5.577000	3.983788
W8	13.40928	11.01240	8.951600	5.822544	4.077216
W9	12.73998	11.41920	10.135296	6.489600	4.444911
W10	10.60500	9.62280	9.2818740	6.770250	4.624185

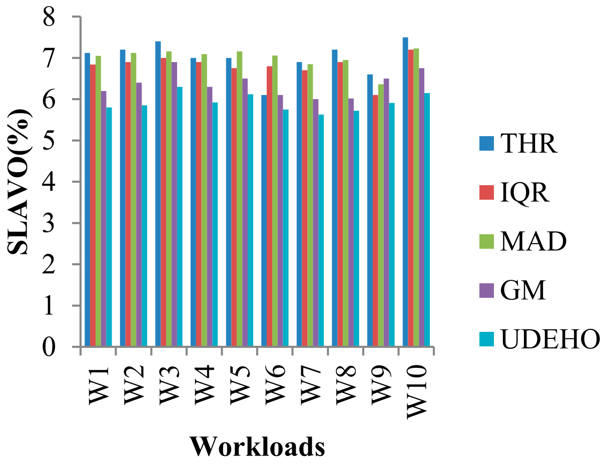

Figure 2. SLAVO measure comparison for 10 workloads.

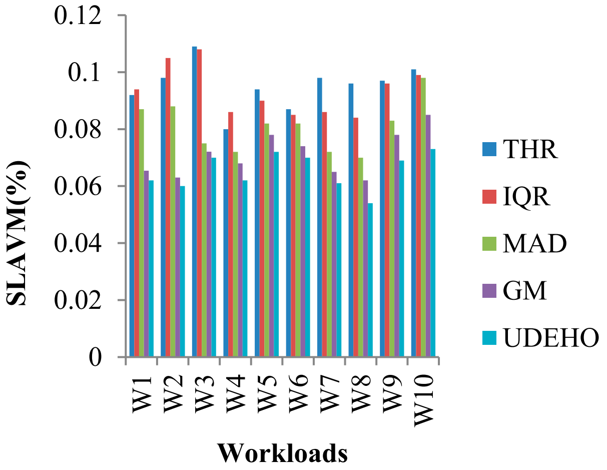

Figure 3. SLAVM measure comparison for 10 workloads.

migration number. The suggested approach has a lower SLAVM value of 0.073%, while other systems like THR, IQR, MAD, and GM have higher SLAVM values for the 10th workload of 0.101%, 0.099%, 0.098%, and 0.085% (see Table 3).

Figure 4 depicts a performance comparison using the SLAV measure (SLAV, x0.00001). Because the suggested technique performs on the SLAV measure, it decreases the SLA violation rate by an average of 40.73%, 37.01%, 36.63%, and 21.75% when compared to THR, IQR, MAD, and GM, correspondingly (SEE Table 3). Nevertheless, multiplying the SLAVO and SLAVM metrics yields the SLAV measure, and while the suggested performance in the SLAVM measure is rather excellent, the performance in the SLAVO measure has a slight increase that is insignificant in the suggested

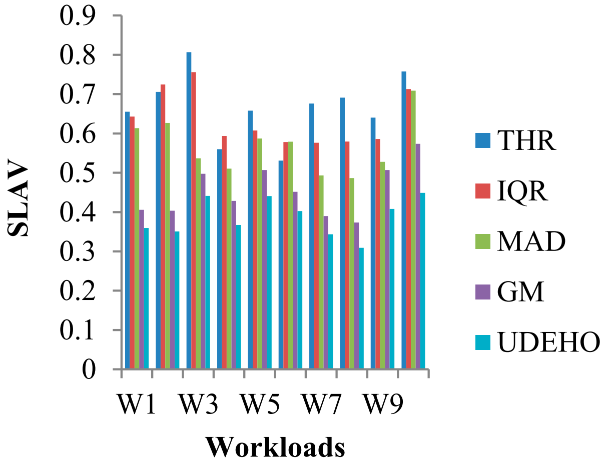

Figure 4. SLAV measure comparison for 10 workloads.

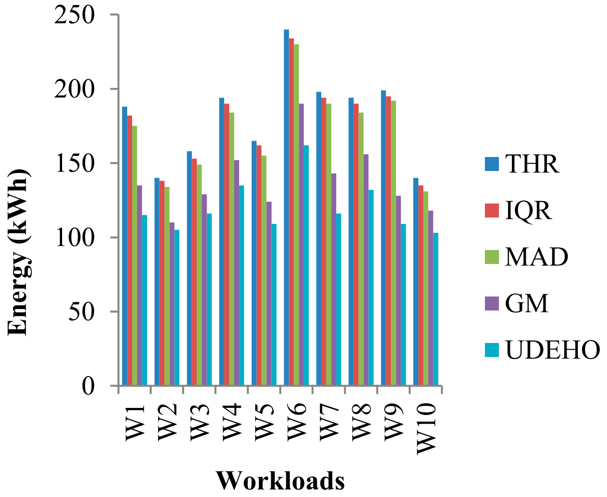

Figure 5. Energy consumption comparison for 10 workloads.

methodology. This is easily noticeable when assessing performance using the SLAV measure.

Figure 5 depicts a performance comparison based on the energy usage parameter. When contrasted to THR, IQR, MAD, and GM, the suggested method minimizes energy usage by an average of 26.4285%, 23.7037%, 21.3740%, and 12.7118% (see Table 3). Those hosts can be chosen more accurately by under loaded hosts that can be found using UP-PUD and power-saving values. Complete VMs on these hosts may be relocated to other suitable hosts after the discovery of the underutilized hosts, and the host may then be put to sleep. As a result, by converting idle hosts to low-power states throughout the consolidation procedure, energy may be conserved.

Figure 6 depicts a performance comparison based on the ESV measure (ESV, x0.001). When contrasted to THR, IQR, MAD, and GM, the suggested technique decreases energy usage by an average of 56.3961%, 51.9455%, 50.1804%, and 31.6984%, correspondingly (see Table 3). As energy usage reduction and violation rate of SLAV, the suggested technique results in such significant improvements. In reality, these significant findings suggest that the technique incorporates an effective trade-off between power cost and QoS assurance.

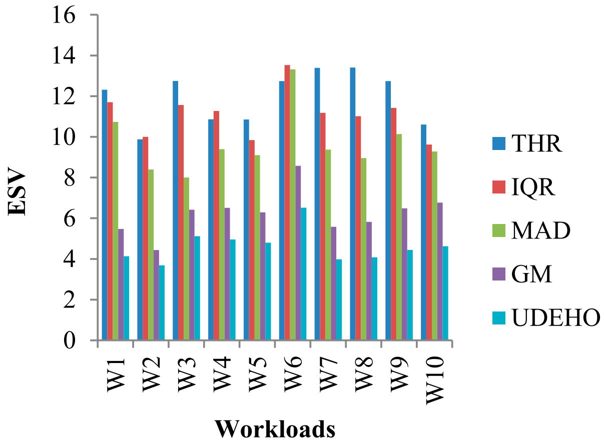

Figure 6. ESV measure comparison for 10 workloads.

5. Conclusion and future WORK

The dynamic VM consolidation issue is handled in this research by anticipating CPU consumption using the UDEHO model. To optimize the effectiveness of cloud data centres, starting allocation strategy has to be supplemented by a VM consolidation process which may be applied on a regular basis. The primary contribution of the work is the inclusion of a distribution function for random number generation in the EHO method, it enhances how well the VM consolidation method performs. The UP-POD and UP-PUD protocols for host underload and overload detection. When overloaded hosts are found, to maintain QoS, full VMs that may see an increase in CPU use are removed from these hosts; when underloaded hosts are recognized, entire VMs from these hosts are relocated to save energy usage. The proposed technique decreases energy usage while limiting depending on real-world workloads and varying number of migrations at different simulations setup. For instance, when contrasted to THR, IQR, MAD, and GM, the suggested method minimizes energy usage by an average of 26.4285%, 23.7037%, 21.3740%, and 12.7118%. As a result, it enhances cloud data centre performance by improving SLA performance indicators like SLAVO, SLAVM, SLAV, energy usage, and ESV guarantee. The present system has been enhanced to include a Web application that may consolidate and deconsolidate VMs in order to balance CPU load use across PMs in accordance to the number of PMs in use.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Data availability statements

Since no datasets were created or analysed for this topic, data sharing is not relevant.

References

[1] Buyya R, Srirama SN, Casale G, et al. A manifesto for future generation cloud computing: research directions for the next decade. ACM Comput Surv (CSUR). 2018;51(5):1–38.

[2] Gill SS, Tuli S, Xu M, et al. Transformative effects of IoT, blockchain and artificial intelligence on cloud computing: evolution, vision, trends and open challenges. Internet Things. 2019;8:1–30.

[3] Ashraf A, Hartikainen M, Hassan U, et al. Introduction to cloud computing technologies. In: I Porres, T Mikkonen, A Ashraf, editor. Developing cloud software: algorithms, applications, and tools, Turku Centre for Computer Science (TUCS) general publication number 60, bo, Finland. 2013, p. 1–41.

[4] Khoshkholghi MA, Derahman MN, Abdullah A, et al. Energy-efficient algorithms for dynamic virtual machine consolidation in cloud data centers. IEEE Access. 2017;5:10709–10722.

[5] Ghosh R, Komma SPR, Simmhan Y. Adaptive energy-aware scheduling of dynamic event analytics across edge and cloud resources. 2018 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID); May 2018. p. 72–82.

[6] Shim YC. Performance evaluation of static VM consolidation algorithms for cloud-based data centers considering inter-VM performance interference. Int J Appl Eng Res. 2016;11(24):11794–11802.

[7] Zheng Q, Li R, Li X, et al. Virtual machine consolidated placement based on multi-objective biogeography-based optimization. Future Gener Comput Syst. 2016; 54:95–122.

[8] Abdelsamea A, Hemayed EE, Eldeeb H, et al. Virtual machine consolidation challenges: a review. Int J Innov Appl Stud. 2014;8(4):1504–1516.

[9] Chang K, Park S, Kong H, et al. 'Optimizing energy consumption for a performance-aware cloud data center in the public sector. Sustain Comput Informat Syst. 2018;20:34–45.

[10] Beloglazov A, Buyya R. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. Concurrency Comput: Pract Exp. 2012;24(13):1397–1420.

[11] Takouna I, Alzaghoul E, Meinel C. Robust virtual machine consolidation for efficient energy and performance in virtualized data centers. 2014 IEEE International Conference on Internet of Things (iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom), Taipei, Taiwan; 2014. p. 470–477.

[12] Farahnakian F, Liljeberg P, Plosila J. LiRCUP: linear regression based CPU usage prediction algorithm for live migration of virtual machines in data centers. 2013 39th Euromicro Conference on Software Engineering and Advanced Applications, Santander, Spain; 2013. p. 357–364.

[13] Mastroianni C, Meo M, Papuzzo G. Probabilistic consolidation of virtual machines in self-organizing cloud data centers. IEEE Trans Cloud Comput. 2013;1(2): 215–228.

[14] Hieu NT, Di Francesco M, Ylä-Jääski A. Virtual machine consolidation with multiple usage prediction for energy-efficient cloud data centers. IEEE Trans Services Comput. 2017;13(1):186–199.

[15] Ismaeel S, Karim R, Miri A. Proactive dynamic virtual-machine consolidation for energy conservation in cloud data centres. J Cloud Comput. 2018;7(1):1–28.

[16] Sayadnavard MH, Haghighat AT, Rahmani AM. A multi-objective approach for energy-efficient and reliable dynamic VM consolidation in cloud data centers. Eng Sci Technol, Int J. 2021;26:1–13.

[17] Liu X, Wu J, Sha G, et al. Virtual machine consolidation with minimization of migration thrashing for cloud data centers. Math Probl Eng. 2020;2020(7848232): 1–13.

[18] Hsieh SY, Liu CS, Buyya R, et al. Utilization-prediction-aware virtual machine consolidation approach for energy-efficient cloud data centers. J Parallel Distrib Comput. 2020;139:99–109.

[19] Jheng J, Tseng F, Chao H, et al. A novel VM workload prediction using grey forecasting model in cloud data center. International Conference on Information Networking, Phuket; 2014. p. 40–45.

[20] Wang GG, Deb S, Gao XZ, et al. A new metaheuristic optimisation algorithm motivated by elephant herding behaviour. Int J Bio-Inspired Comput. 2016;8(6):394–409.

[21] Li J, Lei H, Alavi AH, et al. Elephant herding optimization: variants, hybrids, and applications. Mathematics. 2020;8(9):1–25.

[22] Fu X, Zhou C. Virtual machine selection and placement for dynamic consolidation in cloud computing environment. Front Comput Sci. 2015;9(2):322–330.

[23] Han G, Que W, Jia G, et al. An efficient virtual machine consolidation scheme for multimedia cloud computing. Sensors. 2016;16(2):1–17.

[24] Rani E, Kaur H. Study on fundamental usage of CloudSim simulator and algorithms of resource allocation in cloud computing. 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Delhi, India; 2017. p. 1–7.

[25] Xavier R, Moens H, Volckaert B, et al. Design and evaluation of elastic media resource allocation algorithms using CloudSim extensions. 2015 11th International Conference on Network and Service Management (CNSM), Barcelona, Spain; 2015. p. 318–326.

[26] Farahnakian F, Ashraf A, Liljeberg P, etal. Energy-aware dynamic VM consolidation in cloud data centers using ant colony system. In IEEE 7th International Conference on Cloud Computing. 2014. p. 104–111