Received: 1 June 2017 | Revised: 11 December 2017 | Accepted: 3 January 2018

DOI: 10.1002/dac.3537

RESEARCH ARTICLE

WILEY

A learning-based approach for virtual machine placement in cloud data centers

Mostafa Ghobaei-Arani[1] | Ali Asghar Rahmanian[2] | Mahboubeh Shamsi[3] | Abdolreza Rasouli-Kenari[3]

[1]Young Researchers and Elite Club, Qom Branch, Islamic Azad University, Qom, Iran

[2]Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands

[3]Faculty of Electrical and Computer Engineering, Qom University of technology, Qom, Iran

Correspondence
Mahboubeh Shamsi, Faculty of Electrical and Computer Engineering, Qom University of Technology, Qom, Iran.
Email: shamsi@qut.ac.ir

Summary

In recent years, the increasing use of cloud services has led to the growth and importance of developing cloud data centers. One of the challenging issues in the cloud environments is high energy consumption in data centers, which has been ignored in the corporate competition for developing cloud data centers. The most important problems of using large cloud data centers are high energy costs and greenhouse gas emission. So, researchers are now struggling to find an effective approach to decreasing energy consumption in cloud data centers. One of the preferred techniques for reducing energy consumption is the virtual machines (VMs) placement. In this paper, we present a VM allocation algorithm to reduce energy consumption and Service Level Agreement Violation (SLAV). The proposed algorithm is based on best-fit decreasing algorithm, which uses learning automata theory, correlation coefficient, and ensemble prediction algorithm to make better decisions in VM allocation. The experimental results indicated improvement regarding energy consumption and SLAV, compared with well-familiar baseline VM allocation algorithms.

KEYWORDS

cloud computing, energy consumption, learning automata, virtual machine placement, virtualization

1 | INTRODUCTION

Today, cloud computing is one of the most challenging research topics in the field of information technology (IT). Because of its importance, the researchers put the cloud computing up to the list of top 10 technologies. The data centers are among the main components of cloud computing, and high energy consumption is one of the most challenging issues related to data centers. In 2006, the cost of energy consumption for data centers had been 405 billion dollars; in its increasing trend according to the predictions, it will double until 2011. Besides the costs, high energy consumption increases temperature and reduces the reliability and lifetime of hardware resources.[1-4] Also, due to high amount of carbon dioxide (CO_2) and greenhouse gas emissions, environmental issues addressed in Johnson and Marker[5] and therefore using power-management techniques considered in Liu et al.[6]

Idle hosts are one of the most important energy wasters into the cloud data centers. These hosts have low utilization, but they use hardware resources as busy hosts. It means the hosts consume a large amount of energy when being at light load or even idle, compared with the time of having maximum use of their resources.[7]

The cloud data centers use virtualization techniques to reduce energy consumption. The virtualization techniques enable multiple virtual machines (VMs) located on a single physical machine or host. So, it is possible to reduce the number of hardware resources and improve the performance of cloud data centers.[8-10] One of the other benefits of virtualization is easy transferring of a VM from a host to another one called "live migration." Live migration transfers running VMs from 1 host to another host without interrupting the operation of VMs.[11] When the number of VMs on a specific host is decreasing, these VMs migrate to another host, and the idle host goes to the sleep or shutdown mode. This technique decreases the number of active hosts called "host consolidation." Host consolidation includes 4 stages: (1) detecting overloaded hosts from which some VMs should be migrated; (2) selecting the VMs that should be migrated from an overloaded host; (3) detecting under-loaded hosts which should migrate all their VMs; and (4) placing VM on destination host for migrating VMs. In this paper, we are focusing on the fourth stage of host consolidation.[12]

It is supposed that M is the number of available physical machines, and capacity of their resources such as memory, CPU, and bandwidth of the network is specified. N is the number of VMs whose needs should be met by memory, CPU, and bandwidth capacity.[13]

A good VM to host mapping is placing VMs on suitable hosts so that minimizing the number of the used physical machines. Also, total needs of resources for the located VMs in the host should not exceed the capacity of the physical machine. In fact, the placement of a VM means finding a suitable physical machine for the VM regarding the minimum number of the hosts.[14-17]

Also, the service providers should be able to provide high-quality services to increase utilization and decrease energy consumption. One of the important necessities for cloud computing environment is to ensure the quality of services that have been defined in the service level agreement (SLA). It is possible to save energy consumption substantially using placement techniques and create a tradeoff among increasing utilization, decreasing energy consumption, and SLA. The VM placement problem is similar to bin packing program which is one of the NP-hard problems; hence, approximation algorithms are typically used to solve the corresponding optimization problem.

In this paper, a new approach is presented based on the combination of an ensemble prediction algorithm[18] and learning automata theory for the dynamic placement of VMs in cloud data centers to reduce energy consumption. The employed policies in the proposed approach select a host with minimum energy usage to decrease both energy consumption and Service Level Agreement Violation (SLAV). The proposed approach considers to the minimum correlation coefficients between the selected VM and the VMs running on the host, future load of the VMs. An ensemble prediction algorithm is used to predict the load of very next future of VMs to make decisions. The proposed approach handles heterogeneous VMs effectively and does not require any information about the applications running on VMs.

The main contributions of this research can be summarized as follows:

- We introduce the proposed framework for VM placement in order to reduce energy consumption in cloud data centers.
- We proposed an algorithm for VM placement in cloud data centers in order to reduce energy and SLAV using a novel algorithm with a mixture of approaches such as learning automata theory, ensemble prediction algorithm, and correlation.
- We consider minimum correlation coefficients between the current VM and VMs of the destination host and energy consumption of destination host after allocation, at the time of placement.

The rest of this paper is organized as follows: in Section 2, the related works are considered. Section 3 contains the detailed description of the proposed approach. The performance of the proposed approach will be evaluated in Section 4. Finally, the conclusion and future works are presented in Section 5.

2 | RELATED WORKS

Various studies performed to solve the problem of VM placement using heuristic algorithms such as first Fit, best fit (BF), constraint programming, random number programming, evolutionary algorithms, etc.

In Wang and Liu,[19] the researchers addressed the problem of live migration by considering to the dot production of the resources vector and available capacity vector of the host, comparing with a threshold. There was an event extractor in the proposed model which it received the information from the resources including the start time of VM, the size of VM, VM migration, and so on. Then, it sent the placement map and planned to the supervisor. The supervisor uses the

size and the number of VMs migration to decide the VM should migrate or not. The aim was preventing sequential and non-beneficial migrations. The approach controlled the consumed energy by threshold limit and used the modified first Fit algorithm for bin packing.

In Jiang et al,[20] the workload integration was addressed using ant colony algorithm. They used an ant colony optimization algorithm to solve the problem of multi-dimensional packing. In the proposed algorithm, each ant received all packs (for example, VMs) and all bins (for example, hosts). Then, it started to choose the host, according to a probable decision-making rule which described the optimizing parameters of works for the ant. It updated the amount of pheromone. The rule guided the ants to choose the most promising items based on the information about current pheromone; the ant would choose the pack with higher probability.

In Vu and Hwang,[21] the best-fit decreasing (BFD) method was used to place the VMs. Their goal was decreasing the network traffic with the energy consumption decrement. They modeled the relationship between VMs and weighing up the communication. The model converted to the graph. Then, the weighted graph converts to a hierarchical model using a recursive algorithm for placing the VM. In the case of low utilization, VM migration considered between physical hosts to keep performance. This approach does not involve the network traffic, and it is not suitable for wide area networks.

Goodarzi et al[22] in California University addressed the placement of VMs problem by building several copies of VMs, dynamic programming, and local search. Their algorithm optimized the trade-off between consumed energy and utilization using these copies. Several copies are usually used to increase reliability. Interesting idea behind the algorithm is, the main VM is responsible for representing the requested service from the service provider. The other VMs should be kept idle until they were needed so that idle VMs cause an increase in availability and utilization and they also decrease the consumed energy. These authors used dynamic programming and linear search to solve the bin packing problem.

Kord et al[23] offered an approach for VM placement whose objective was to compromise decrease of energy and SLAV. They used minimum correlation coefficients to locate the VM. They also utilized fuzzy analytic hierarchy process to make a compromise between energy decrement and SLAV. They also offered a centralized management model. Whenever a VM wanted to choose a host, it selected the host with minimum consumed energy and minimum correlation coefficients.

Beloglazov et al[24] used host consolidation to solve the energy problem. They detected overloaded and underloaded hosts using upper and lower utilization thresholds to trade off the energy reduction and the SLAVs decrement. If the host had fewer efficiency than lower utilization threshold, its VMs transferred to another host. Then, the host with zero utilization goes to the shutdown mode. The BFD approach also applied to solve placement issues, and the algorithm is known as modified best-fit decreasing (MBFD). These authors considered various policies to detect more overloaded hosts and review various policies to select different VMs for the integration stages of the VM. Results of their research showed that the local regression minimum migration time policy had the best outcome in terms of energy. They considered other parameters as the multiple of consumed energy and percentage of violation of SLAs to compare the policies. This approach focused on the energy consumption at the time of placement and reduction of SLA relying on the adequacy of resources.

In Ferdaus et al,[25] the researchers proposed a hierarchy algorithm and used bee colony algorithm to decrease energy consumption. The algorithm clustered the VMs according to their CPU usage, and it determined the best place for each VM. It also decreased the communication delay, and it had a minimum live migration of VMs to achieve the best performance.

In Fang et al,[26] placing the VM performed using BF greedy algorithm and a hierarchical clustering in 2 steps. In the first step, VMs with high traffic allocate to the same physical machine. In the next step, the resources cluster and the VMs are allocated to the suitable cluster. They use minimum cuts of the binary tree to optimize the network issues. The goal of the work is reducing the number of active nodes and overall energy consumption.

In Panigrahy et al,[27] energy issues addressed in the cloud architecture with multi-dimensional resources. The authors offered a solution for reducing energy consumption in data centers, using 3 local search algorithms and a genetic algorithm solution. The big problem was unsustainability of the algorithm. To overcome the problem, the authors considered other computing resources such as RAM and disk besides the CPU. They supposed hosts are homogeneous and VMs are heterogeneous resources, and then, they addressed the problem as a multi-dimensional packing issue.

In Wang and Liu,[19] a VM selected which has the lowest value of dot production of resources vector and capacity vector. To achieve the best utilization, a VM which need higher CPU and lower memory should place on a physical machine with lower CPU and higher memory capacity. This approach might have an incorrect selection because it does not consider the length of vectors.

In Feller et al,[28] the author's focus was on the balance of hosts' resource efficiency using ant colony algorithm. Their objective was energy decrement and reducing wasting resources. The authors utilized vector algebra. They modeled the host capacity, VMs requests, and resource efficiency as vectors. They also considered a matrix for VM placement whose elements showed which host hosts which VM. They also used vectors to minimize wasting resources and choose the best mapping among the possible mappings by ant colony algorithm.

In Ghobaei-Arani et al,[29] an efficient approach for improving VM placement using learning automata in cloud computing environment is proposed. Their approach groups both virtual and physical machines, and taking into account the maximum absolute deviation during the VM placement, the energy consumption as well as the SLA deviation using learning automata in cloud data centers.

In Arianyan et al,[30,31] the authors focused on consolidation problem as an efficient resource management solution to reduce energy consumption in cloud data centers. They proposed novel heuristics for 2 main phases of consolidation problem including Window Moving Average (WMA) policy for detection of overloaded hosts that considers all input criteria including CPU, RAM, and network bandwidth in the decision process and reduces the occurrence of VMs' migrations caused by instantaneous load peaks and also a novel multi-criteria VM selection method namely Multi-criteria Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) with Prediction VM Selection (MTPVS) policy that selects the VMs to be migrated from overloaded hosts to both eliminating the hotspots quickly and minimizing the SLAVs due to VM migrations.

In Horri et al,[32] a novel quality of services-aware VMs consolidation approach is proposed that adopts a method based on resource utilization history of VMs. Their proposed approach considers the trade-off between energy consumption and quality of service in the cloud environment.

Varasteh et al[33] presented a survey and taxonomy for host consolidation techniques in cloud data centers. Special attention has been devoted to the parameters and algorithmic approaches used to consolidate VMs onto hosts. Also, they presented a system model and reviewed various approaches to host consolidation presented in the literature and classified them from 5 points of view: time of applying the technique, constraints and requirements considered during the optimization process, and algorithmic method used to find the near-optimal solution of the optimization problem, their objective functions, and evaluation methods.

3 | PROPOSED APPROACH

In this section, we explain our proposed approach in more detail. The proposed algorithm is based on BFD algorithm, which uses learning automata theory, correlation coefficient, and ensemble prediction algorithm to make better decisions in VM allocation. The learning automata theory is appropriate for learning where there is incomplete information regarding the environment and in dynamic, complex, or random environments with a large number of uncertainties such as cloud environment, computer networks, etc. In recent years, learning automata used as a technique for energy management in smart grid and cloud computing.[34,35] The proposed approach focuses on reducing both energy consumption and the SLAV simultaneously. In order to decrease energy consumption, the proposed approach tries to find a host which consumes minimum energy. Also, the proposed approach uses 2 measures for decreasing the SLAV: the first one is the adequacy of resources and, the second one is the minimum correlation coefficients between the current VMs and the host VMs. As our approach is based on learning automata, in the next section, we will introduce learning automata theory. Then, we explain multiple correlation coefficients because our placement policy needs multiple correlation coefficients. Finally, the proposed framework and algorithm are presented.

3.1 | Multiple correlation coefficient

Multiple correlation coefficient is the prediction quality measure of the dependent variable in multiple regression analysis. Its value is a correlation between the predicted values and the real value. Suppose that there are n subjects and p independent variables collected from each subject. The target value of the variable y computes from the set of p independent predictor variables x_1, x_2, ..., x_p. The value of variable y_i, observed from subject i, relates to p independent variables $x_{1...p}$ as shown in Equation 1.

$$y_i = \beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip} + e_i \quad (1)$$

Where y_i is the value of the dependent variable observed from subject i, p is the number of predictors, β_j is j-th coefficient of predictor x_i, $j = 0, ..., p$, and e_i is an error in observed value for subject i.

Suppose that $\mathbf{X}$ is a matrix of $n \times (p + 1)$ whose first column is 1. The matrix includes data from p independent variable collected from n subjects. The variable $\mathbf{y}$ is a vector of $n \times 1$ of the observed real value of dependent variable. These matrixes are shown in Equations 2 and 3.

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{bmatrix} \tag{2}$$

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \tag{3}$$

Vector of the predicted value of dependent variable $\mathbf{y}$ is shown as $\widehat{\boldsymbol{y}}$ which is obtained from Equation 4.

$$\widehat{\boldsymbol{y}} = \mathbf{Xb} \text{ where } \mathbf{b} = \left(\mathbf{XX^T}\right)^{-1}\mathbf{X^T y} \tag{4}$$

Where $\mathbf{X^T}$ is the transposed matrix of $\mathbf{X}$. Predicting quality of multiple correlation coefficient is $R^2_{y,1...,p}$ shown in Equation 5.

$$R^2_{y.1....p} = \left[\frac{\mathrm{COV}(y, \widehat{y})}{\sqrt{\mathrm{var}(y)\,\mathrm{var}(\widehat{y})}}\right] \tag{5}$$

Here, COV shows covariance and Var is variance. It measures the correlation degrees of 2 linear variables, and its value is between zero and one. A zero value indicates no linear relationship between the 2 variables. So, if there is a linear relationship with a positive slope, we will have a value of 1. The above equation could compute as Equation 6.

$$R^2_{y.1....p} = \left[\frac{\sum_1^n (y_i - m_y)^2 \left(\widehat{y_i} - m_{\widehat{y}}\right)^2}{\sum_1^n (y_i - m_y)^2 \sum_1^n \left(\widehat{y_i} - m_{\widehat{y}}\right)^2}\right] \tag{6}$$

In this equation, m_y and $m_{\widehat{y}}$ are the means of y and $\widehat{y}$.

In VM placement problem, the correlation degree between VM and other VMs on selected physical machines should be calculated. The matrix $\mathbf{X}$ includes the utilization rate of p VMs running on the selected host. For each VM, the utilization rate history is reviewed, and n utilization rates save as columns of matrix $\mathbf{X}$. The vector $\mathbf{y}$ includes n utilization rates of the selected VM.

3.2 | The learning automata theory

The learning automata are an abstract model that can perform a limited number of actions. The learning automata learn from the experiences taken from the environment and selected its current action. After performing an action, the environment gives feedback to the learning automata, and the automata use these feedbacks to choose the next actions. Figure 1 shows the relationship between learning automata and environment.[36]

In this paper, we use learning automata with variable structures. Learning automata with variable structures is shown by 5 items $\equiv \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{p}, \mathbf{T}, \mathbf{c}\}$.

$\alpha \equiv \{\alpha_1, \alpha_2, \alpha_3 \cdots, \alpha_r\}$ is the set of actions in learning automata.

$\beta \equiv \{\beta_1, \beta_2, \beta_3 \cdots, \beta_m\}$ shows the input set of automata.

$p \equiv \{p_1, p_2, p_3, ... p_n\}$ is the probability vectors for choosing an action.

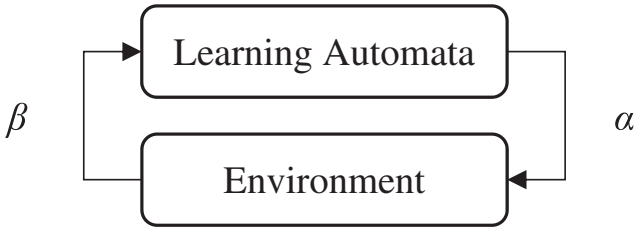

FIGURE 1 The relationship between learning automata and its environment

$T \equiv p(n+1) = T[\alpha(n), \beta(n), p(n)]$ is learning algorithm.

$c \equiv \{c_1, c_2, c_3, \cdots, c_r\}$ is the probability of fining an action.

If the learning automata choose act α_i at step i and receive a proper response from the environment, $p_i(n)$ will increase, and the other probabilities will decrease. By an improper response, $p_i(n)$ will decrease, and the other probabilities will increase. After all changes, the sum of $p_i(n)$ remains constant and equals to one.

i. A proper response will be made by Equations 7 and 8.

$$p_i(n+1) = p_i(n) + \alpha[1 - p_i(n)] \tag{7}$$

$$p_j(n+1) = (1-\beta)p_j(n) \quad \forall j \, j \neq i \tag{8}$$

ii. The improper response will be made by Equations 9 and 10.

$$p_i(n+1) = (1-\beta)p_i(n) \tag{9}$$

$$p_j(n+1) = \frac{\beta}{r-1} + (1-\beta)p_j(n) \forall j \, j \neq i \tag{10}$$

In the above equation, α shows reward and β is a penalty.

3.3 | Proposed framework

We introduce the proposed framework in this section. Figure 2 shows the proposed framework for running the proposed algorithm.

In this framework, first, the customers send their requests to the brokers. On the other hand, the cloud providers also send their offers to the brokers. Brokers are responsible for delivering the customers' requests to the best service providers (which select according to their policies) as a requesting for a VM. It is also possible; the customers send these requests to the providers directly. In fact, brokers work as the mediator between service providers and customers. Once a Request receives, the service provider tries to allocate VM into one of the physical machines using placement manager.

Physical machines send their information to placement manager including the total capacity of their resources, the used capacity of resources, the VMs that are running on them, and utilization of each VM.

The placement manager includes 5 modules: 2 modules for detecting overloaded hosts and under-loaded hosts, an energy manager module, and manager of the violation of SLAs. Besides them, there is a module for selecting VM based on specified policies. The placement manager collects information from the physical machines and sends them to its modules.

The first module detects overloaded host using the received information from placement manager, and it also estimates which host will be overloaded after adding the current VM. Then, the module sends this data to both the manager

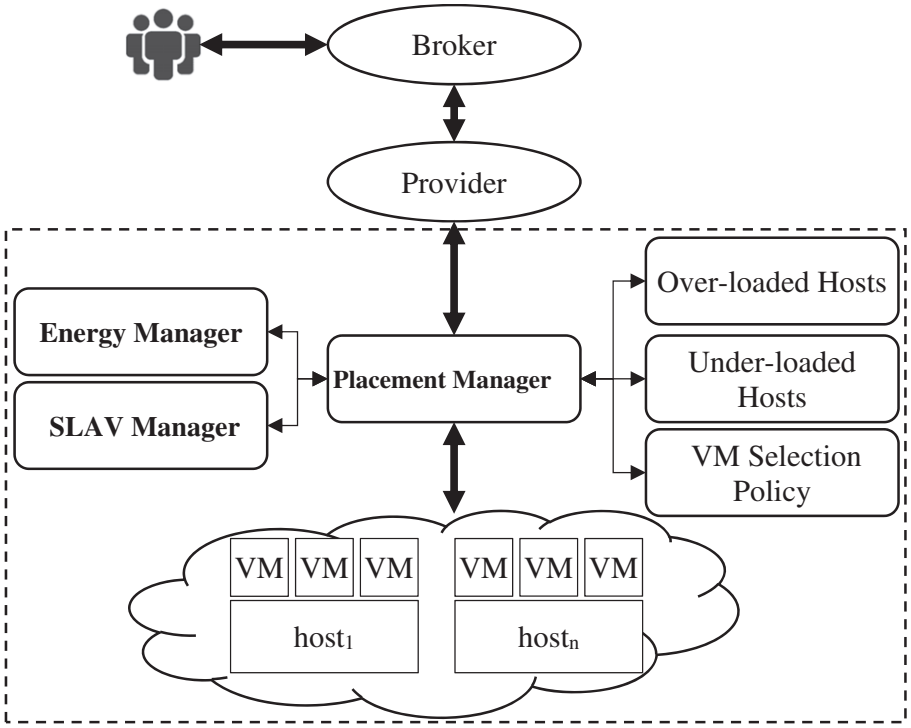

FIGURE 2 The proposed framework

of SLAV and VM selection module. On the other hand, the underloaded detector module identifies hosts with fewer loads and sends back the result to the placement manager.

The VM selection module uses received information, then, introduces a VM running on the overloaded host according to the specified policies by placement manager. The placement manager migrates the selected VM to an under-loaded physical machine.

It is essential to find the best destination physical machine. Energy manager module computes the energy of each physical machine using the information received from placement manager. It estimates the energy of a physical machine after allocating the current VM on it. Energy Manager is responsible for identifying the host which consumes energy less than the other hosts, and it introduces it to placement manager for destination host of migration.

The SLAV manager identifies the hosts do not have enough resources because selecting these hosts will increase SLAV. It also calculates the correlation between the selected VM and the VMs running on the chosen host. It sends the correlation coefficients of resources to the placement manager.

Placement manager collects all data from above modules and specifies the most suitable destination physical machine to allocate the selected VM. The most suitable physical machine is the host who has minimum energy and minimum correlation coefficients. If there is a conflict between minimum energy and minimum correlation coefficients, the manager uses the learning automata to make the best decision. The details of the algorithm are described in the next section.

3.4 | Proposed algorithm

The proposed algorithm is a developed version of Power-aware Best-Fit-Decreasing (PABFD) algorithm[12] in combination with an ensemble prediction algorithm and learning automata theory. Algorithm 1 shows pseudo-code of the proposed algorithm. The algorithm input is 2 lists: a list of hosts and a list of the migrating VMs. First, the entire VMs on the migration list are sorted in descending order based on their CPU usage requests in MIPS. Then, the hosts are divided into several categories as follows:

- The first list includes all hosts without running VMs, which is called empty machines (line 7–8).
- The second list includes under-loaded hosts (lines 9–10).

Those hosts in the above lists will not be considered as the destination of the migrating VMs in the first step. The third list includes the hosts have enough resources to allocate the migrating VM, and they would not be overloaded after allocation (Lines 11–24).

Algorithm 1 Pseudo-code of the proposed algorithm.

```
Input: Host list, VM list.
Output: Allocation.
1. Initialize: Empty host list, Underutilized host list.
2. Sort VMs in decreasing MIPS ().
3. For each VM v in VM's list, do.
4.     Min power = Max
5.     Allocated host = Null
6.     For each host in host list do
7.         If host is empty then
8.             Append host to empty host queue
9.         Else If host is underutilized then
10.            Append host to underutilized host queue
11.        Else If host has enough resources and host is not overutilized after allocation of VM v then
12.            Power = Get power after allocation (host, VM)
13.            Predicted_Load = Ensemble_Prediction_module(VM, host)
14.            Load_Deviation= calculate load deviation for VM v according to Equation 11
15.            If power < min power then
16.                If Predicted_Load < Safety_Threshold and
17.                   Predicted_Deviation + Predicted_Load < 100% then
18.                    Allocated host = Host
19.                    Min power = Power
20.                Else If Predicted_Load < Safety_Threshold then
21.                    Candid host list add (host)
22.                End If
23.            End If
24.        End If
24.    If allocated host = null then
25         Allocated host = get host by automaton (candid host list, VM list, host list)
26.    If allocated host = null then
27.        Allocated host = get host by MBFD (underutilized host list, VM list, host list)
28.    If allocated host==null then
29.        Allocated host=get host by MBFD (empty host list, VM list, host list)
30. Return Allocated host.
```

More precisely, if a host has enough space for the migrating VM and power increase of the host after VM allocation is minimum and the future load of the host after VM allocation according to the ensemble model prediction algorithm (Algorithm 2) is lower than the *Safetly_Threshold*, which is set to 90% load of CPU, and the sum of predicted load and load deviation is under full utilization mode then the migrating VM can be allocated on the host without any concern (Lines 11–19). Otherwise, if all mentioned conditions unless the last condition in Lines 16 and 17 are provided, the host may be chosen as the destination for the migrating VM, but it would not be the best choice. Hence, the host is appended to the list of candidate hosts (See line 21).

After the process of evaluating allocation of the migrating VM on individual available hosts, no suitable host for the VM has not chosen yet if allocated host is *null*. Thus, we first try to allocate the migrating VM on one of the candidate hosts using learning automata host selection (Algorithm 3) in lines 24 to 25. If no suitable allocated host is not found among candidate hosts, we try to allocate the migrating VM on one of hosts in underutilized host list by *PABFD* algorithm (See Lines 26–27).[12] Finally, in the last try, if no suitable destination for migrating VM is not found among candidate and underutilized hosts, the migrating VM is allocated on one of hosts in empty host lists by the *PABFD* algorithm (See Lines 28–29).[12]

Equation 11 calculates the load deviation for a specific host as follows:

$$Load\ Deviation = \frac{1}{k}\sum_{j=1}^{k}\frac{\sqrt[2]{(Vl_i(t-j+1)-Vl_i(t-j))^2}}{Vl_i(t-j)} \tag{11}$$

Where K is the time window length and $Vl_i(t)$ is the CPU load of VM i at time interval t.

Algorithm 2 shows the pseudo-code of the applied ensemble CPU load prediction algorithm. The inputs of this algorithm are the migrating VM and the host to be allocated, and output of the algorithm is the predicted load of the next time interval for the host after allocation of the migrating VM. In the first step, a time series for CPU load history of the VM I is created (Line 5). Then, MAPE error rate of recent CPU load predictions of the VM for individual constituent prediction models is calculated (Lines 6–9). According to the calculated error rate, the best constituent model at the current time interval is chosen, which is the one with minimum error rate (MER[*Constituent Prediction Model Name*]). Based on the chosen model, CPU load of the VM for the next time interval is predicted, and thus sum of the very future load of the host is summation predicted load for all its VMs (See lines 11–20). Finally, the predicted load is returned as the output of the function.

Algorithm 2 Pseudo-code algorithm for Ensemble Prediction Module.

Input: *migrating VM*, *host*.
Output: Predicted load.
1. Predicted_Load = 0.
2. Vm_List=getVmList(*host*).
3. Vm_List.Add(migrating VM).
4. **Foreach** *VM* V_i *in Vm_List* **do**.
5. ts_i = Create_TimeSeries(Resource_Usage_History(V_i).
6. MER[Moving Average] = Calculate recent error rate for V_i according to **Equation 12**
7. MER[Exponential Smoothing] = Calculate recent error rate for V_i according to **Equation 12**
8. MER[Linear Regression] = Calculate recent error rate for V_i according to **Equation 12**
9. MER[Double Exponential Smoothing] = Calculate recent error rate for V_i according to **Equation 12**
10. Best_Prediction_Model = Select model with minimum MAPE error rate according to MER values
11. **Switch** (Best_Prediction_Model)
12. **Case** Moving Average:
13. Predicted_Load = Predicted_Load + **Predict**(Vm,Moving Average)
14. **Case** Exponential Smoothing:
15. Predicted_Load = Predicted_Load + **Predict**(Vm,Exponential Smoothing)
16. **Case** Linear Regression:
17. Predicted_Load = Predicted_Load + **Predict**(Vm,Linear Regression)
18. **Case** Double Exponential Smoothing:
19. Predicted_Load = Predicted_Load + **Predict**(Vm,Double Exponential Smoothing)
20. **End Switch**
21. **Return** Predicted_Load

$$Recent\ Error\ Rate = \frac{1}{k}\sum_{j=1}^{k}\frac{\sqrt[2]{\left(Vl'_i(t-j)-Vl_i(t-j)\right)^2}}{Vl_i(t-j)} \tag{12}$$

Where k is the time window length and $Vl_i(t)$ and $Vl'_i(t)$ are the actual and predicted CPU load of VM i at time interval t.

Algorithm 3 shows the pseudo-code of the host selection process by the learning automaton. Each learning automata has a number of actions. In our algorithm, the number of actions is equal to the number of physical machines. The automata are allowed to select only 1 action at any time. In other words, our automaton selects only 1 host from the list of candidate at any time. The learning automaton has a vector of hosts selection probabilities. A host with higher probability has more chance to select as the destination host. This probability vector is an array called P[i]. The probability vector initializes with uniform distribution equals to $\frac{1}{\text{number of the hosts}}$. The value of P[i] is the probability of selecting the *i*-th host. The automaton selects 1 host from the candidate list. The host should have enough resources for VM, and it should not overload after allocating resources to VM (line 5). Then, the algorithm estimates the energy consumption of chosen host by acquiring the list of VMs running on it (lines 6–8). The amount of energy is compared with the energy of the other hosts after accepting the current VM. It also calculates the correlation between the current VM and all VMs running on other hosts. If the chosen host has the minimum value of energy and correlation, the automaton is rewarded and the probability of selecting *i*-th host increases (lines 9–12). If only one of the criteria, the energy value or correlation coefficient, has the minimum value, the algorithm calculates the EnC (Energy and Correlation) value using α (the reward factor) and β (the penalty factor) (lines 13–15). If calculated EnC has the minimum value among all hosts, the automaton is also rewarded and the probability of selecting *i*-th host increases (lines 16–18). Otherwise, the automaton is fined and the probability of selecting *i*-th host decreases (lines 18–20). These steps repeat for all hosts and, finally, the host with higher probability selects as the destination host (line 22).

In the process of reward and fine, Equations 7, 8, 9, and 10 are used. The parameters α and β are equal, because our algorithm is random learning automata with a linear relationship between reward and penalty. We suppose α and β are equal to preserve the algorithm sustainability.

Algorithm 3 Pseudo-code algorithm for host selection by the learning automata.

```
Input: Candid hosts list, VM.
Output: Allocated host;
1.  Foreach host i in candid hosts list do.
2.    If host i has enough resources for VM then
3.          Allocated host = Select host i by probability of random P[i]
4.          VM list of allocated host = Allocated host. Get VM list ()
5.          Power = Estimate power (allocated host, VM)
6.          Mcc = Get VM correlation (VM, allocated host, VM list of allocated host);
7.          Min power = Get min power (candid hosts list, VM)
8.          Min correlation = Get min MCC (candid hosts list, VM)
9.          If power <= min power and Mcc <= min correlation then
10.               Update probabilities according to Equation 7 to 8
11.         Else If (power >=  min power and Mcc <= Min correlation)
12.          or (power <= min power and Mcc >= min correlation) then
13.               EnC = alpha * power + beta * Mcc
14.               min EnC = get Min EnC (candid hosts list, VM)
15.               If EnC <= Min EnC then
16.                    Update probabilities according to Equation 7 to 8
17.               Else
18.                    Update probabilities according to Equation 9 to 10
19.               End If
20.         End If
21.   End If
22. Allocated host=Select host by Max P[i] in candid hosts list
23.Return Allocated host
```

4 | PERFORMANCE EVALUATION

In this section, we use the cloudsim[37] to simulate the proposed approach. We consider 4 different VM types for the simulation, as shown in Table 1. Also, the type of hosts and the number of VMs on various days are shown in Tables 2 and 3, respectively.

Also, we consider 4 scenarios to evaluate the efficiency of the proposed algorithm. The scenarios along with their descriptions are shown in Table 4. The comparing algorithm for VM allocation section are PABFD,[12] UMC,[32] and TPSA.[31] We call them as baseline algorithms for VM allocation section.

4.1 | Evaluation of the proposed algorithm

In this section, we compare the proposed VM allocation algorithm with other important baseline algorithms regarding energy consumption, SLAV, number of migrations, and number of shutdowns.

We executed each experiment 10 times, and the average of the executions are provided in this section for individual experiments. In order to evaluate the superiority of the proposed algorithm, we compare it with well-known VM allocation algorithms such as PABFD,[12] UMC,[32] and TPSA.[31] To do so, we selected 4 measurements in our plots, including the percentage of SLAV, energy consumption, number of shutdowns, and the number of migrations. Figures 3--6 show the results of the experiments over different days in the dataset.

TABLE 1 Type of VMs

Type	Size	BW	RAM	PE	MIPS
1	2.5 GB	100 Mbit/s	870	1	2500
2	2.5 GB	100 Mbit/s	1740	1	2000
3	2.5 GB	100 Mbit/s	1740	1	1000
4	2.5 GB	100 Mbit/s	613	1	500

TABLE 2 Type of hosts

Type	Storage	BW	RAM	PE	MIPS
1	1 GB	1 Gbit/s	4096	2	1860
2	1 GB	1 Gbit/s	4096	2	2660

TABLE 3 Number of VMs on various days

Date	20110303	20110306	20110309	20110322	20110325	20110403	20110409	20110411	20110412	20110420
Number of VMs	1052	898	1061	1516	1078	1463	1358	1233	1054	1033

TABLE 4 Type of scenarios

Scenario name	Description
LR/MMT/SM	LR policy for detecting overloaded hosts; MMT policy for VM selection; SM policy for detecting under-loaded hosts.[12]
LR/MC/SM	Use LR policy for detecting overloaded host; MC policy for VM selection; SM policy for detecting underloaded hosts.[9]
WMA/MRR/SM	Use window moving average (WMA) policy for detecting overloaded hosts[30]; maximum requested resources (MRR) policy for VM selection[30]; SM policy for detecting underloaded hosts.[9]
WMA/MDM/VDT	Use WMA policy for detecting overloaded hosts[30]; minimum down-time migration (MDM) policy for VM selection[30]; VM-based dynamic threshold (VDT).[32]

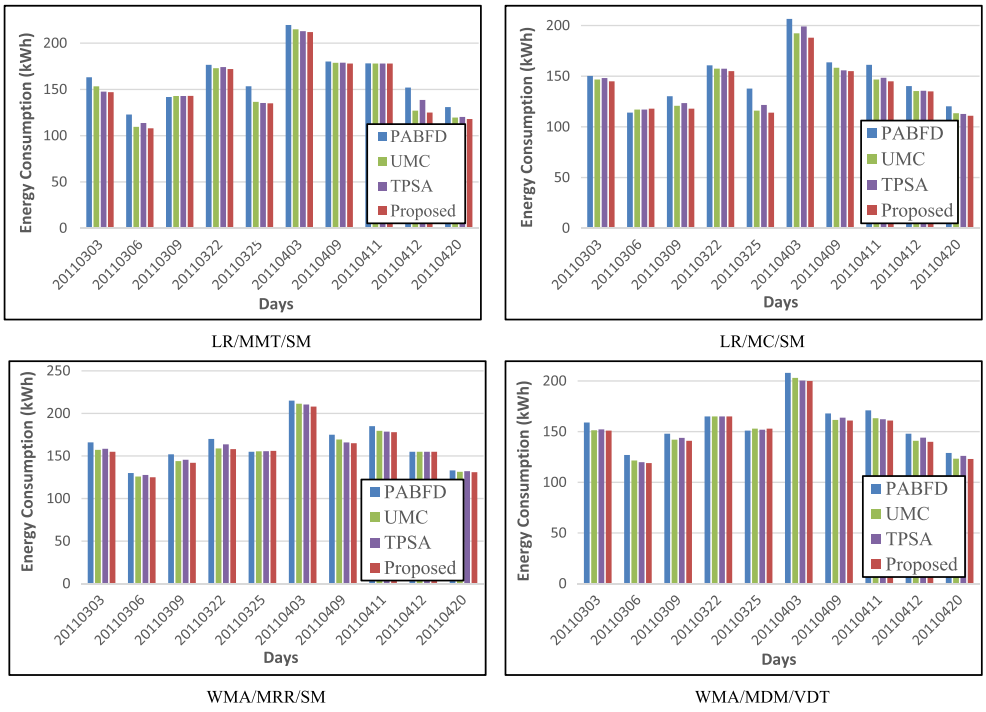

FIGURE 3 The energy consumption

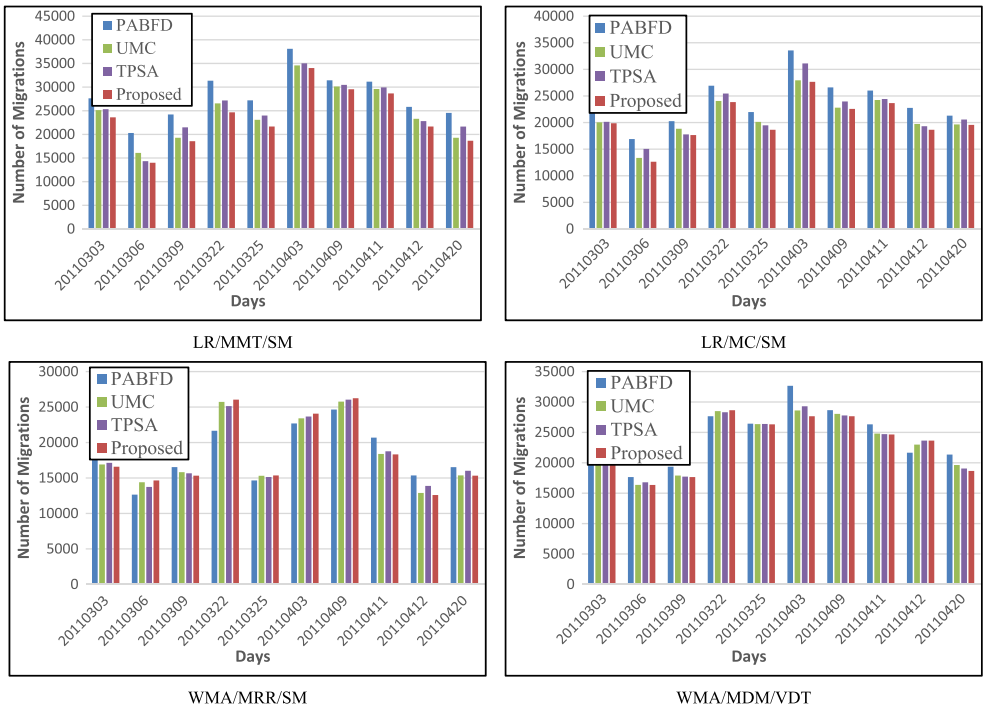

FIGURE 4 The number of VM migrations

Regarding Figure 3, the consumed energy of the proposed VM allocation algorithm decreased compared with the baseline algorithms in all 10 days. Try to allocate VMs on hosts with higher CPU load with considering energy increase after allocation, CPU load after allocation, and the type of host are among the most important reasons for this improvement.

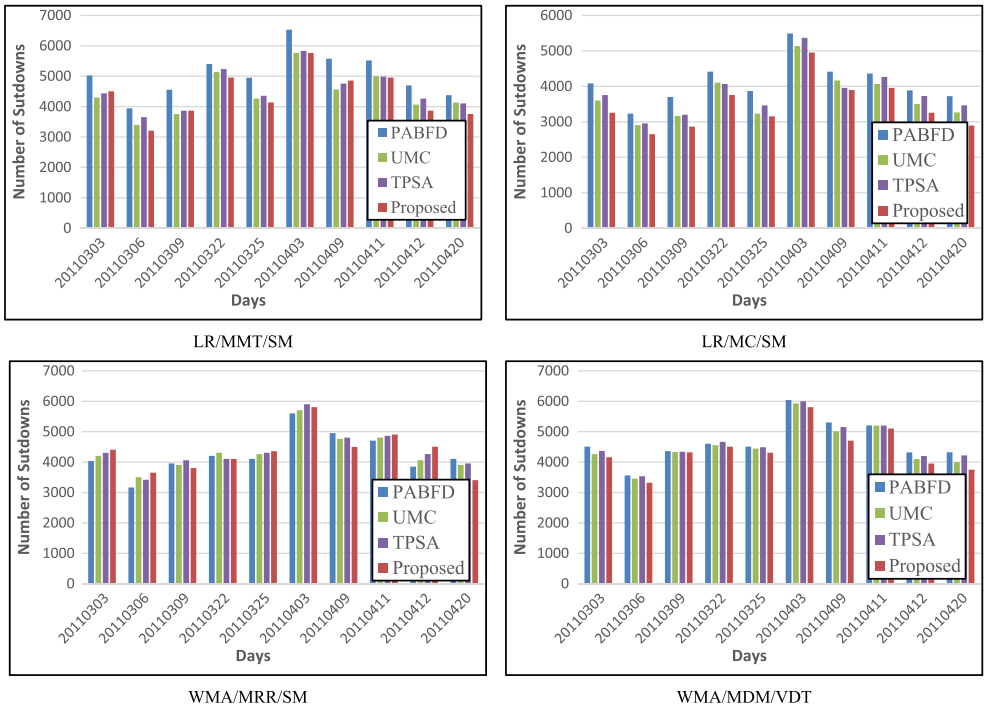

FIGURE 5 The number of shutdowns

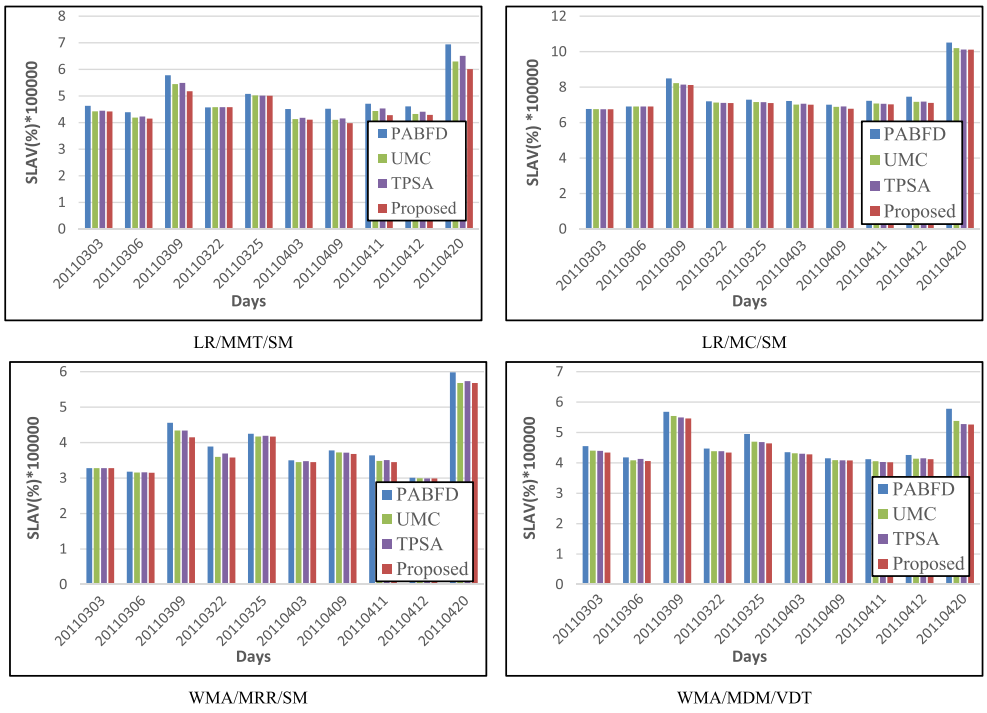

FIGURE 6 The average SLAV

As shown in Figure 4, the number of migrations in most of scenarios for the proposed VM allocation algorithm is decreased. Actually, considering future CPU load of the being allocated host affects on the number of VM migrations. In case of VM allocation according to current VM and host load, there is no consideration of the probable future for the host. By contrast, we do predict the future load of individual VMs with an accurate ensemble prediction model and thus decide whether this allocation would make this being allocated host overloaded or not. By this way, we are able

to protect more hosts from overload situation, and that is why less number of overloaded hosts and so less number of migrating VMs from overloaded hosts (See Figure 4).

There is worth mentioning that algorithms of other phases (ie, host underload detection algorithm, host overload detection algorithm, VM selection algorithm) do influence the performance of the whole resource manager. Hence, we can see that the number of VM migration for the proposed algorithm in scenario WMA/MRR/SM is not better than other approaches. The mentioned fact is the reason for the results of this scenario.

Figure 5 illustrates the number of host shutdowns for different VM allocation algorithms over different scenarios. While the proposed algorithm considers host load type in determining the new location of the migrating VMs, we can see a reduction in the number of shutdowns. Take for instance, a host with low load has the lowest priority for allocation at the same time hosts with high load in the future (according to our look ahead predictions) do not have much priority for allocation. Because allocation algorithm does not let allocation on a host which is very probable to be overloaded, it indirectly decreases the number of migrations because a subset of VMs from overloaded hosts should have to be migrated out to other active hosts or we have to turn on a recently turned off host again. Further, SLAV rate of the proposed algorithm outperforms other baseline algorithms because SLAV rate is calculated according to a number of migrations and the amount of times that hosts experience overload situation due to lack of enough resources (See Figure 6).

4.2 | Total comparison

Figure 7 shows the average of energy consumption of different allocation algorithms over different scenarios. As seen, the proposed algorithm obtains minimum energy consumption.

Figure 8 illustrates the average number of migrations of the VM allocation algorithms over different scenarios. As seen, the proposed algorithm has a minimum number of migrations. In the prior section, we explained the reasons

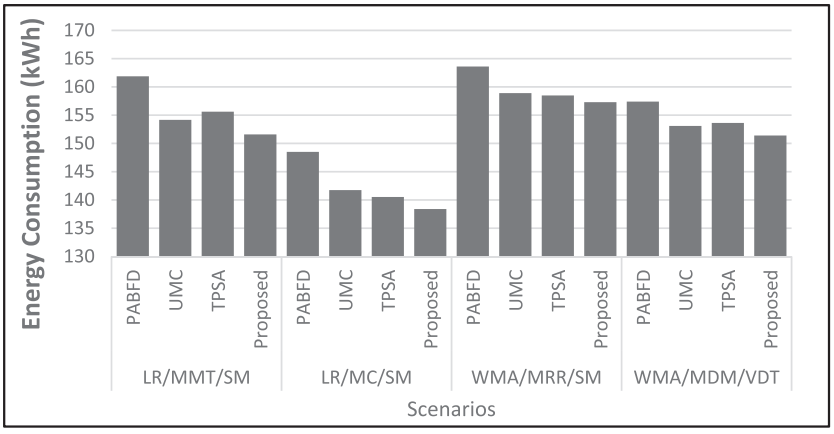

FIGURE 7 Comparing energy consumed in various scenarios

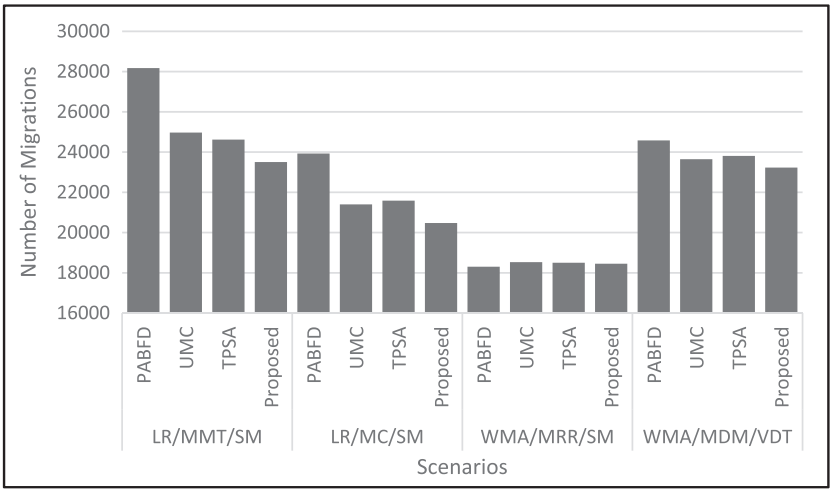

FIGURE 8 Comparing the number of VM migration in various scenarios

for this improvement. Figure 9 also shows a marked improvement in the number of host shutdowns compared with other baseline algorithms.

While a number of migrations and the amount of time hosts experience overloaded situation are 2 reasons of SLAV, according to Figures 8 and 9, we can conclude the SLAV rate of the proposed algorithm is more efficient compared with

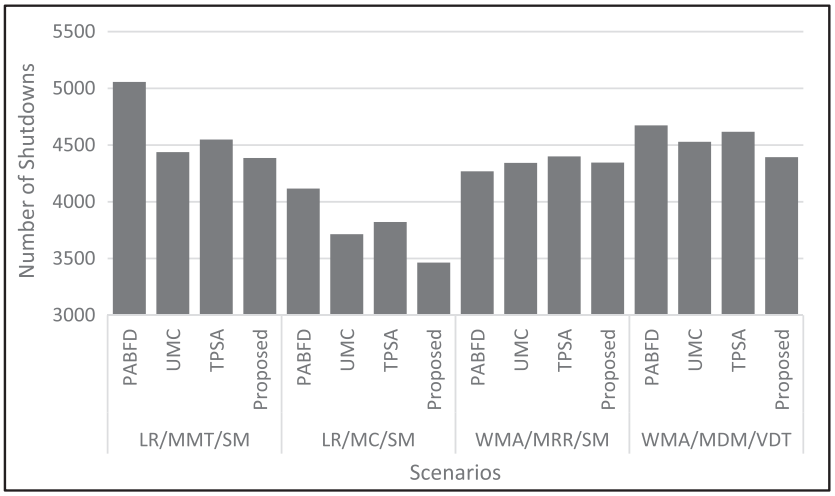

FIGURE 9 Comparing the number of shutdown hosts in various scenarios

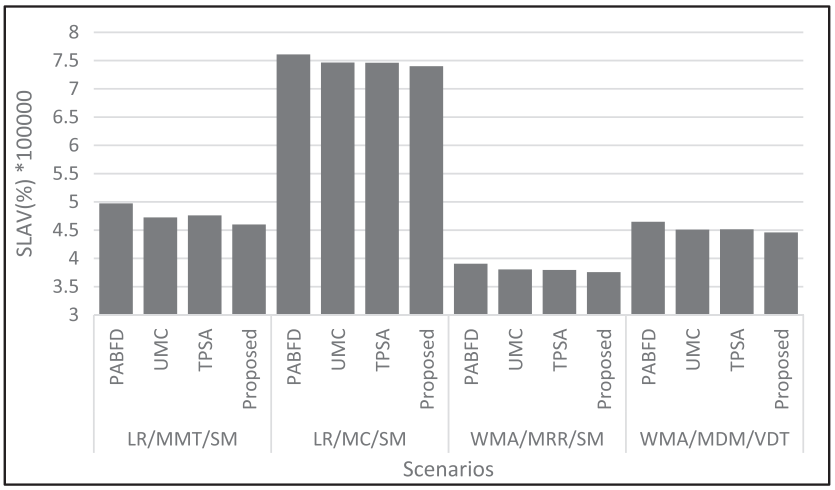

FIGURE 10 Comparing violation of SLA in various scenarios

TABLE 5 Statistical comparison of the proposed VM allocation algorithm with other baseline algorithms

Baseline algorithm	Metrics	Energy consumption	Migration	Shutdown	SLAV
PABFD	*Mean*	10.27	3454.5	652.7	0.207
	SD	9.106	1181.105062	148.1126073	0.139208078
	t-value	3.566	9.249040184	13.93546887	4.702252101
	p-value	**0.003**	**3.41287E − 06**	**1.06717E − 07**	**0.000558312**
UMC	*Mean*	2.599	627.5	250.3	0.043543544
	SD	2.7	541.055604	100.8288429	0.032318004
	t-value	3.044	3.667514424	7.850115857	4.260683251
	p-value	**0.006**	**0.002587386**	**1.28686E − 05**	**0.001054513**
TPSA	*Mean*	2.596	619.8	357.6	0.041954202
	SD	4.048	363.8765969	141.2414796	0.04932613
	t-value	2.028	5.386385688	8.00636254	2.689666447
	p-value	**0.036**	**0.000220396**	**1.09973E − 05**	**0.012405034**

other 3 algorithms. Figure 10 also depicts the SLAV rate of different VM allocation algorithms and prove the mentioned deduction.

Table 5 shows the statistical results of paired t-test that determine the significance level of proposed algorithm compared with other algorithms regarding energy consumption and SLAV. The table provides mean and standard deviation differences between the proposed and baseline algorithms, t-value, and *P*-value, respectively. To do a statistical evaluation, a paired t-test with a significance level of $p < 0.05$ is done to evaluate if the differences were the statistical difference. The table shows that there is a meaningful difference between proposed algorithm and other algorithms while *P*-value in all cases is lower than 0.05. Given certainty, more than 0.95 proves that the proposed VM allocation algorithm has a significant improvement compared with other baseline algorithms. Thus, the null hypothesis is rejected, which proves the fact that differences are significant.

5 | CONCLUSION AND FUTURE WORKS

Concerning the daily growth in the number of data centers and their energy consumption, it is essential to developing the good energy management approaches. These approaches should use the cloud resources more efficient. In this paper, we proposed an algorithm to reduce consumed energy in data centers. The proposed algorithm focuses on both energy reduction and SLAV decrement. The employed policies select a host with minimum energy usage to decrease both energy consumption and SLAV. The proposed algorithm considers to the minimum correlation coefficients between the selected VM and the VMs running on the host, future load of the VMs. An ensemble prediction algorithm is used to predict the load of very next future of VMs to make decisions. Results show that, by using this algorithm, both energy and violation of SLA are reduced. It also uses the learning automata to solve the compromise between energy reduction and SLAV. For future works, the neural network and the learning automata could be combined to achieve better efficiency, especially in the learning phase.

ORCID

Mostafa Ghobaei-Arani http://orcid.org/0000-0003-2639-0900
Ali Asghar Rahmanian http://orcid.org/0000-0001-9249-1633

REFERENCES

1. Fereydooni A, Shamsi M, Arani MG. EDLT: an extended DLT to enhance load balancing in cloud computing. *International Journal of Computer Applications*. 2014;108(7):6-11.
2. Rahmanian AA, Dastghaibyfard GH, Tahayori H. Penalty-aware and cost-efficient resource management in cloud data centers. *International Journal of Communication Systems*. 2017;30(8):
3. Ghobaei-Arani M, Jabbehdari S, Pourmina MA. An autonomic resource provisioning approach for service-based cloud applications: a hybrid approach. In: *Future Generation Computer Systems*; 2017.
4. Ghobaei-Arani M, Jabbehdari S, Pourmina MA. An autonomic approach for resource provisioning of cloud services. *Cluster Computing*. 2016;19(3):1017-1036.
5. Johnson, P. and Marker, T., 2009. Data centre energy efficiency product profile. Pitt & Sherry, report to equipment energy efficiency committee (E3) of The Australian Government Department of the Environment, Water, Heritage and the Arts (DEWHA).
6. Liu, L., Wang, H., Liu, X., Jin, X., He, W.B., Wang, Q.B. and Chen, Y., 2009, June. GreenCloud: a new architecture for green data center. In *Proceedings of the 6th international conference industry session on Autonomic computing and communications industry session* (pp. 29-38). ACM.
7. Beloglazov, A. and Buyya, R., 2010, May. Energy efficient resource management in virtualized cloud data centers. In *Proceedings of the 2010 10th IEEE/ACM international conference on cluster, cloud and grid computing* (pp. 826-831). IEEE Computer Society.
8. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I. and Warfield, A., 2003, October. Xen and the art of virtualization. In *ACM SIGOPS operating systems review* (Vol. 37, no. 5, pp. 164-177). ACM.
9. Rahmanian AA, Horri A, Dastghaibyfard G. Toward a hierarchical and architecture based virtual machine allocation in cloud data centers. *Int J Commun Syst*. 2018;31(4). https://doi.org/10.1002/dac.3490
10. Ghobaei-Arani M, Rahmanian AA, Aslanpour MS, Dashti SE. 2017. CSA-WSC: cuckoo search algorithm for web service composition in cloud environments. *Soft Computing*, (pp. 1-26). https://doi.org/10.1007/s00500-017-2783-4

11. Clark, C., Fraser, K., Hand, S., Hansen, J.G., Jul, E., Limpach, C., Pratt, I. and Warfield, A., 2005, May. Live migration of virtual machines. In *Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation-Volume 2* (pp. 273-286). USENIX association.
12. 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 and Computation: Practice and Experience*. 2012;24(13):1397-1420.
13. Gupta RK, Pateriya RK. Survey on virtual machine placement techniques in cloud computing environment. *International Journal on Cloud Computing: Services and Architecture (IJCCSA)*. 2014;4(4):1-7.
14. Pathan N, Shetty B. Virtual machine placement in cloud. *International Journal of Computer Science and Information Technologies*. 2014;1833-1835.
15. Ghiasi H, Arani MG. Smart virtual machine placement using learning automata to reduce power consumption in cloud data centers. *SmartCR*. 2015;5(6):553-562.
16. Aslanpour MS, Dashti SE, Ghobaei-Arani M, Rahmanian AA. 2017. Resource provisioning for cloud applications: a 3-D, provident and flexible approach. *The Journal of Supercomputing*, (pp. 1-32). 10.1007/s11227-017-2156-x
17. Aslanpour MS, Ghobaei-Arani M, Toosi AN. Auto-scaling web applications in clouds: a cost-aware approach. *Journal of Network and Computer Applications*. 2017;95:26-41.
18. Rahmanian AA, Ghobaei-Arani M, Tofighy S. A learning automata-based ensemble resource usage prediction algorithm for cloud computing environment. *Future Generation Computer Systems*. 2018;79:54-71.
19. Wang, X. and Liu, Z., 2012, January. An energy-aware VMs placement algorithm in cloud computing environment. In *Intelligent System Design and Engineering Application (ISDEA), 2012 Second International Conference on* (pp. 627-630). IEEE.
20. Jiang D, Huang P, Lin P, Jiang J. Energy efficient VM placement heuristic algorithms comparison for cloud with multidimensional resources. In: *International Conference on Information Computing and Applications*. Berlin Heidelberg: Springer; 2012, September: 413-420.
21. Vu HT, Hwang S. A traffic and power-aware algorithm for virtual machine placement in cloud data center. *International Journal of Grid & Distributed Computing*. 2014;7(1):350-355.
22. Goudarzi H, Pedram M. Energy-efficient virtual machine replication and placement in a cloud computing system. In: IEEE, ed. *In Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*; 2012, June:750-757.
23. Kord N, Haghighi H. An energy-efficient approach for virtual machine placement in cloud-based data centers. In: IEEE, ed. *In Information and Knowledge Technology (IKT), 2013 5th Conference on*; 2013, May:44-49.
24. Beloglazov A, Buyya R. Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints. *IEEE Transactions on Parallel and Distributed Systems*. 2013;24(7):1366-1379.
25. Ferdaus MH, Murshed MM, Calheiros RN, Buyya R. Virtual machine consolidation in cloud data centers using ACO metaheuristic. In: *Euro-Par*; 2014, August:306-317.
26. Fang W, Liang X, Li S, Chiaraviglio L, Xiong N. VMPlanner: optimizing virtual machine placement and traffic flow routing to reduce network power costs in cloud data centers. *Computer Networks*. 2013;57(1):179-196.
27. Panigrahy, R., Talwar, K., Uyeda, L. and Wieder, U., 2011. Heuristics for vector bin packing. *research. microsoft. com*.
28. Feller E, Rilling L, Morin C. Energy-aware ant colony based workload placement in clouds. In: IEEE Computer Society, ed. *Proceedings of the 2011 IEEE/ACM 12th International Conference on Grid Computing*; 2011, September:26-33.
29. Ghobaei-Arani M, Shamsi M, Rahmanian AA. An efficient approach for improving virtual machine placement in cloud computing environment. *Journal of Experimental & Theoretical Artificial Intelligence, pp*. 2017;1-23.
30. Arianyan E, Taheri H, Sharifian S. Novel heuristics for consolidation of virtual machines in cloud data centers using multi-criteria resource management solutions. *The Journal of Supercomputing*. 2016;72(2):688-717.
31. Arianyan E, Taheri H, Sharifian S. Novel energy and SLA efficient resource management heuristics for consolidation of virtual machines in cloud data centers. *Computers & Electrical Engineering*. 2015;47:222-240.
32. Horri A, Mozafari MS, Dastghaibyfard G. Novel resource allocation algorithms to performance and energy efficiency in cloud computing. *The Journal of Supercomputing*. 2014;69(3):1445-1461.
33. Varasteh A, Goudarzi M. Server consolidation techniques in virtualized data centers: A survey. *IEEE Systems Journal*. 2015;
34. Misra S, Krishna PV, Saritha V, Obaidat MS. Learning automata as a utility for power management in smart grids. *IEEE Communications Magazine*. 2013;51(1):98-104.
35. Ranjbari M, Torkestani JA. A learning automata-based algorithm for energy and SLA efficient consolidation of virtual machines in cloud data centers. *Journal of Parallel and Distributed Computing*. 2017;
36. Fallah M, Arani MG. ASTAW: auto-scaling threshold-based approach for web application in cloud computing environment. *International Journal of u-and e-Service, Science and Technology*. 2015;8(3):221-230.

37. Calheiros RN, Ranjan R, Beloglazov A, De Rose CA, Buyya R. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and experience*. 2011;41(1):23-50.

How to cite this article: Ghobaei-Arani M, Rahmanian AA, Shamsi M, Rasouli-Kenari A. A learning-based approach for virtual machine placement in cloud data centers. *Int J Commun Syst*. 2018;31:e3537. https://doi.org/10.1002/dac.3537