




# Energy and quality of service-aware virtual machine consolidation in a cloud data center

Anurina Tarafdar<sup>1</sup> · Mukta Debnath<sup>1</sup> · Sunirmal Khatua<sup>1</sup>  · Rajib K. Das<sup>1</sup>

Published online: 18 February 2020

© Springer Science+Business Media, LLC, part of Springer Nature 2020

## Abstract

The large-scale virtualized Cloud data centers consume huge amount of electrical energy leading to high operational costs and emission of greenhouse gases. Virtual machine (VM) consolidation has been found to be a promising approach to improve resource utilization and reduce energy consumption of the data center. However, aggressive consolidation of VMs tends to increase the number of VM migrations and leads to over-utilization of hosts. This in turn affects the quality of service (QoS) of the applications running in the VMs. Thus, reduction in energy consumption and at the same time ensuring proper QoS to the Cloud users are one of the major challenges among the researchers. In this paper, we have proposed an energy efficient and QoS-aware VM consolidation technique in order to address this problem. We have used Markov chain-based prediction approach to identify the over-utilized and under-utilized hosts in the data center. We have also proposed an efficient VM selection and placement policy based on linear weighted sum approach to migrate the VMs from over-utilized and under-utilized hosts considering both energy and QoS. Extensive simulations using real-world traces and comparison with state-of-art strategies show that our VM consolidation approach substantially reduces energy consumption within a data center while delivering suitable QoS.

**Keywords** Cloud computing · Energy · Quality of service · Virtual machine consolidation · Virtual machine placement · Virtual machine selection

---

✉ Sunirmal Khatua  
skhatuacomp@caluniv.ac.in

Anurina Tarafdar  
anurinataraftdar@gmail.com

Mukta Debnath  
muktapearl.debnath@gmail.com

Rajib K. Das  
rajib.k.das@ieee.org

<sup>1</sup> Department of Computer Science and Engineering, University of Calcutta, Kolkata, India

## 1 Introduction

Cloud computing enables dynamic and on-demand provisioning of infrastructure, platform and software as services in a pay-per-use manner. These are termed as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS), respectively [1]. Instead of bearing the huge cost needed to build and maintain their own infrastructure, many large and small enterprises outsource their computational requirements to the Cloud.

Cloud service providers (CSPs), namely Google, Amazon, etc., have deployed large-scale virtualized data centers across the world in order to deliver services to their customers. These data centers containing numerous physical machines (hosts or servers) are responsible for consumption of enormous amount of electrical energy and emission of greenhouse gases. It has been reported that in 2016, the global data centers consumed around 416.2 terawatt hours of electricity which accounted for about 3% of the global electricity supply and 2% of the total greenhouse gas emissions [2]. Analysts predict that the data centers' energy requirement would increase about 15 times by 2030, resulting in around 8% of the global electricity demand [3]. This massive amount of energy consumption and emission of greenhouse gases by the data centers have a negative impact on the environment and lead to global warming. Also, owing to the rising energy costs, the CSPs are facing reduction in their profit margin. Reports show that energy consumption of the data centers accounts for around 12% of the monthly operational costs for the CSPs [4]. Thus, to develop a greener environment and increase the Return on Investment (ROI) of the providers, promotion of energy efficiency in a virtualized data center is extremely important.

The numerous servers in a data center largely contribute to the total energy consumption of the data center [5]. Each server contains several virtual machines (VMs) which execute the tasks of the Cloud users. Task execution by the VMs increases the CPU and memory utilization of the servers which in turn leads to increase in energy consumption. However, it must be noted that the huge power consumption of the servers not only occurs due to the execution of large number of Cloud services, but also due to the inefficient usage of the computing resources. Research shows that generally servers operate at 10–50% of their maximum CPU utilization [6] and idle servers consume about 70% of their peak power consumption [7]. To improve resource utilization and obtain increased ROI, the CSPs use VM consolidation techniques. The VM instances are dynamically consolidated to a minimum number of servers using live migration, keeping their current resource requirements into consideration. Idle servers are switched to sleep mode to eliminate idle power consumption [8]. It has been reported in [4] that the use of VM consolidation has led to only 4% rise in data center energy consumption from 2010 to 2014, compared to about 24% increase from 2005 to 2010.

VM consolidation in Cloud computing, however, has several challenges. Owing to the variable workload of the applications running on the VMs, their aggressive consolidation might lead to some VMs not getting their demanded resources during peak load. This in turn would hamper the quality of service (QoS) of the applications by leading to increase in response time, failure or time-out. Additionally, the performance overheads [9] associated with excess migration of VMs, increase the

completion time of the tasks executing in the VM instances and hence affect the QoS. An important objective of the CSPs is satisfaction of QoS defined via service level agreements (SLAs). At the same time, energy consumption of the data center has to be reduced to decrease operational costs and pave the way for sustainable and eco-friendly cloud computing.

Although many research works have been conducted to reduce energy consumption in a Cloud data center by VM consolidation, most of them have not considered the performance overheads associated with VM migrations [8, 10–13]. Again, many existing studies have considered homogeneous data centers and have tried to consolidate the VMs so as to reduce the number of active hosts [11, 14]. However, the hosts in a data center are mostly heterogeneous having different resource capacities and different nature of power consumption. In such a context, simply minimizing the number of active hosts is not equivalent to minimizing energy consumption. Moreover, in Cloud environment, it is important not only to reduce energy consumption but at the same time lessen SLA violations. Thus, a trade-off is essential.

Keeping the above aspects into consideration, in this paper, we propose an energy efficient and QoS-aware dynamic VM consolidation approach that can be used in a heterogeneous virtualized data center by a CSP providing IaaS. The VM consolidation approach involves identification of the over-utilized and under-utilized hosts in the data center, proper selection of VMs from these hosts and thereafter effective placement of the selected VMs to other hosts in the data center. The main contributions of this paper are as follows:

1. Formulation of a prediction-based over-utilized and under-utilized host detection technique where we predict the CPU utilization of each of the VMs allocated to a host using the Markov chain model and thus obtain the predicted CPU utilization values of the host. Based on these values, the decision of considering the host to be over-utilized or under-utilized and subsequent VM migrations are taken.
2. Proposal of energy and QoS-aware VM selection and VM placement heuristics that enable load balancing, improve resource utilization and lessen the number of VM migrations within the data center.
3. Formulation of a VM consolidation approach by integrating the proposed prediction-based over-utilized and under-utilized host detection technique, VM selection and VM placement policies to achieve proper trade-off between energy consumption and QoS.
4. Exploration of the efficacy of our proposed VM consolidation approach by performing extensive simulations using the CloudSim 3.0.3 simulation platform [15] and the PlanetLab workload data [16].

The rest of the paper is organized in the following manner: In Sect. 2, we present some related works on VM consolidation in Cloud computing. In Sect. 3, we describe the power and energy model for a host in the data center, several overheads of VM migration and metrics for evaluation of QoS. Our VM consolidation model is also discussed in this section. sections 4, 5 and 6 describe our proposed over-utilized and under-utilized host detection, VM selection and placement and VM consolidation strategies, respectively. We evaluate our proposed techniques and analyze the

simulation results in Sect. 7. We conclude our work giving it a future direction in Sect. 8.

## 2 Related work

Reduction in energy consumption is one of the main challenges for data center infrastructures. The important power consumers of a data center are servers, network equipment and cooling systems [17]. To reduce the energy consumption of the network equipment, energy-efficient routing considering minimum network devices are used [18]. To reduce the energy consumption of the cooling systems, new innovative techniques like evaporative cooling and free cooling are being used. When the servers in the data center get heated up, cooler air from the environment is let in. As in many places of the world the outside air remains cooler than that inside the data center for most of the time in a year, free cooling is becoming popular because of its cost-effective and energy-saving nature [19, 20]. Research shows that among servers, network equipment and coolers, the servers are the main power consumers contributing to around 40–50% of the total energy consumption of the data centers [17, 21]. Thus, in this paper, we focus on VM consolidation which has become a popular method for reducing energy consumption and improving resource utilization of the servers in a data center. In the last few years, several studies have been conducted on energy-aware VM consolidation. We briefly discuss some of the approaches for VM consolidation in the following:

*Greedy approaches* VM consolidation is considered as a general version of the bin-packing problem, and greedy heuristics like First Fit (FF), Best Fit (BF), First Fit Decreasing (FFD), Best Fit Decreasing (BFD) are used for it. Beloglazov et al. [8] have proposed Modified Best Fit Decreasing (MBFD) algorithm to place a VM to the host that generates least increase in power consumption. Also three VM selection policies, namely the minimization of migrations (MM) policy, the highest potential growth (HPG) policy and the random choice (RC) policy were proposed to select VMs for migration from over-loaded and under-loaded hosts. The MM policy reduces the number of VM migrations, thereby improving QoS. Verma et al. [22] have presented pMapper, a power and migration cost-aware VM placement algorithm. Here, the physical machines are classified into receivers and donors, and VMs are allocated from the donors to the receivers. pMapper tries to reduce migration costs by keeping the VMs in the receivers unmoved. Although this reduces migration overheads, it hinders power savings. Li et al. [11] have proposed a framework named EnaCloud that extends the FF and BF algorithm to reduce energy by reducing number of active hosts. Experiments are conducted on a set of homogenous servers.

In [23], Murtazaev et al. have presented Sercon algorithm where an all or none migration strategy is followed. All the VMs in an active host are tried to be placed on other hosts. If possible, a new placement scheme with a less number of active hosts is obtained. In [24], QoS-aware energy efficient VM consolidation techniques have been proposed. The authors have presented a prediction-based migration

technique for the VMs based on the predicted CPU utilizations. Also a deadline-aware VM migration technique was proposed where by setting appropriate deadlines for the VMs, a trade-off between energy consumption and QoS is achieved. Zakarya et al. [4] have proposed energy-performance-cost (EPC)-aware VM migration heuristic that reduces energy consumption, improves performance of the VMs and reduces cost of the Cloud service consumers. A VM is migrated from source host to target host only if it improves energy efficiency and performance.

*Evolutionary approaches* Metaheuristics like ant colony optimization (ACO), genetic algorithm (GA) and particle swarm optimization (PSO) are used for solving the VM consolidation problem. Gao et al. [25] have adopted ACO to consolidate VMs in a virtualized data center to reduce energy consumption and improve response time of the applications. Farahnakian et al. [26] have proposed ant colony system-based VM consolidation (ACS-VMC) for better energy savings. Zhang et al. [27] have presented an effective evolutionary approach for VM allocation in reservation-based Cloud considering energy and service request acceptance ratio. In [28], the authors use an improved grouping genetic algorithm (IGGA) for VM consolidation to improve power savings and reduce migration costs. The migration cost of a VM is defined as a product of its memory, page dirty ratio and application sensitivity. A particle swarm optimization-based VM placement approach is presented in [13] to minimize energy consumption and maintain QoS while provisioning data intensive services in a heterogeneous data center. QoS is measured in terms of response time, throughput, availability and reliability of services. However, the migration of VMs and the costs associated with it are not taken into account. Liali et al. [29] propose an iterative budget algorithm for VM consolidation considering energy consumption, VM migration cost and communication overhead between the VMs. On the critical side, many of the above mentioned evolutionary techniques involve a lot of iterations, which are time consuming in nature.

*Integer programming* In [12], the authors have provided an integer programming formulation of the VM consolidation problem and have proposed a heuristic to provide services using reduced number of physical machines. An optimal cloud resource provisioning algorithm is presented by formulating a stochastic programming model in [30]. It helps to reduce the total cost of on-demand and reservation-based resource provisioning and proves to be advantageous for Cloud service consumers. Ghribi et al. [14] have formulated a linear integer program to consolidate VMs to a minimum number of physical machines and reduce energy consumption. However, these exact solutions cannot be obtained in feasible time when the number of VMs and hosts in the data center are large.

*DVFS-based approaches* Dynamic voltage and frequency scaling (DVFS) technology helps to reduce processor energy consumption by making it operate on different frequencies with different voltage supplies. Ding et al. [10] have proposed an energy efficient scheduling algorithm for the VMs where an optimal frequency is set for the hosts using DVFS, and the VMs are allocated considering the performance-power ratio of the hosts. Here, the performance overheads of VM migrations are

not taken into account. Alnowiser et al. [31] have presented the Enhanced Weighted Round Robin (EWRR) algorithm for energy efficient task scheduling in the Cloud. Here, DVFS is used to specify the minimum VM frequency for each task based on the task's complexity and deadline. A DVFS-aware VM consolidation technique to reduce energy consumption of a data center is presented in [32].

It is evident from the above discussion that a sizable amount of research work has been conducted on energy-aware VM consolidation in a Cloud data center. Our work is distinguishable from the above literature in the following aspects: (1) We propose an energy efficient and QoS-aware VM consolidation approach considering three aspects- over-utilized and under-utilized host detection, VM selection and VM placement that can be used by IaaS providers to dynamically consolidate VMs in a Cloud data center; (2) Identification of over-utilized and under-utilized hosts is done by analyzing the CPU utilization history of each of the VMs allocated to the hosts and predicting their CPU utilization in future using Markov chain model; (3) The migration costs of VMs are taken into account during VM selection and VM placement; (4) Heterogeneity of physical machines as well as VMs is considered; (5) The VM consolidation decisions are fast and do not require any highly time consuming computation.

### 3 System model

In this section, we describe the power and energy model used for hosts in a data center and the overheads related to migration of VMs. Also, we briefly state the metrics used to measure the SLA violations and evaluate the QoS. Finally, we discuss our VM consolidation model.

#### 3.1 Power and energy model

The energy consumed by the large number of servers within a data center has a major contribution toward the total energy consumption of the data center [5]. Compared to other system resources of a host in the data center, the CPU has been found to consume the maximum energy [8, 28]. Studies have also shown that there lies a linear relationship between the power consumption of a host and its CPU utilization [7, 25, 28]. The use of processor utilization-based power model in determining the power consumption of a host is highly dominant in the literature [8, 24, 33, 34]. Hence, we adopt this power model to obtain the power consumed by a host. It has been specified below:

$$P(u) = P_{\text{idle}} + (P_{\text{max}} - P_{\text{idle}}) \cdot u, \quad (1)$$

where  $P_{\text{idle}}$  and  $P_{\text{max}}$  represent the power consumption of an idle and a fully utilized host, respectively; and  $u$  is the current CPU utilization. Owing to the variation of workload, the CPU utilization of a host changes with time and hence can be represented as a function of time,  $u(t)$ . Thus, the total energy consumption,  $E_h$  of a host can be calculated using the energy model [8] given as follows:

$$E_h = \int_t P(u(t))dt. \quad (2)$$

### 3.2 Overhead of virtual machine migration

Live migration of VMs helps to dynamically consolidate them to a minimum number of hosts. However, it affects the performance of applications running in the migrating VMs and has several other overheads [9]. Due to migration, a performance degradation of about 10% of the CPU utilization has been observed in applications running in that VM [16]. Time taken for a VM migration depends on the memory used by the VM and the available network bandwidth. The performance degradation and delay of a VM  $vm_i$  due to a migration [4, 16] has been stated below:

$$P_{\text{degr}_i} = \frac{1}{10} \int_{t_0}^{t_0+T_{\text{mig}_i}} u_i(t)dt, \quad (3)$$

$$T_{\text{mig}_i} = \frac{\text{Mem}_i}{\text{BW}_i}, \quad (4)$$

where  $T_{\text{mig}_i}$  is the time taken for migration of  $vm_i$ ;  $\text{Mem}_i$  is the memory used by  $vm_i$ ;  $\text{BW}_i$  is the available network bandwidth;  $P_{\text{degr}_i}$  represents the performance degradation experienced by  $vm_i$ ;  $t_0$  indicates the time when the migration of  $vm_i$  starts; and  $u_i(t)$  is the CPU utilization of  $vm_i$ .

### 3.3 Quality of service (QoS) and service level agreement (SLA) violation

One of the major objectives of the CSPs is to ensure proper QoS to their customers, defined using SLAs. Excessive consolidation of VMs can lead to over-utilization of hosts, deteriorate the performance of the VMs and take increased time in completion of the jobs running in the VMs. This in turn leads to SLA violations and affects the QoS. In [16], the authors have defined some metrics for evaluating the level of SLA violations in an IaaS platform. They are: SLA violation time per active host (SLATAH) as defined in Eq. (5), performance degradation due to migrations (PDM) defined in Eq. (6) and the combined SLA Violation (SLAV) metric stated in Eq. (7).

$$\text{SLATAH} = \frac{1}{N} \sum_{j=1}^N \frac{T_{\text{over}_j}}{T_{\text{active}_j}}, \quad (5)$$

where  $N$  is the total number of hosts in a data center;  $T_{\text{over}_j}$  is the time duration during which the host  $j$  has experienced 100% CPU utilization; and  $T_{\text{active}_j}$  is the total time for which the host  $j$  has been active (serving VMs).

$$\text{PDM} = \frac{1}{M} \sum_{i=1}^M \frac{C_{\text{degr}_i}}{C_{\text{req}_i}}, \quad (6)$$

where  $M$  is the total number of VMs;  $C_{degr_i}$  is the estimated performance degradation of the VM  $vm_i$  due to migrations; and  $C_{req_i}$  is the total CPU capacity requested by  $vm_i$  in its lifetime. During all the migrations of  $vm_i$ ,  $C_{degr_i}$  has been estimated to be 10% of its CPU utilization and has been calculated using Eq. (3).

The reason for formulating the SLATAH metric as stated by the authors [16] is that if a host is over-loaded or has 100% CPU utilization, then the performance of the applications running in the VMs allocated to that host, is limited by the host capacity. Thus, the VMs in that host would not be able to acquire their desired performance level during the peak workloads. Again, the overall performance degradation of VMs due to migrations can be measured using the PDM metric. The SLATAH metric value increases with the over-utilization of hosts and the PDM metric value increases with the rise in number of VM migrations. Finally, a combined metric SLA violation (SLAV) was proposed to measure the performance degradation and hampering of QoS caused by both host overloading as well as VM migrations. It has been stated below:

$$SLAV = SLATAH \cdot PDM, \quad (7)$$

In this paper, we use the SLATAH, PDM and SLAV metrics to measure the level of SLA violations and represent the QoS.

As energy can be reduced at the expense of an increased level of SLA violations, we consider the combined Energy and SLA violations (ESV) [16] metric as defined in the following:

$$ESV = E \cdot SLAV, \quad (8)$$

where  $E$  denotes the total energy consumed by all the hosts in the virtualized data center.

### 3.4 VM consolidation model

The overall broker-based VM consolidation architecture is depicted in Fig. 1. Once the users request for their VMs, the *Initial VM Placement Module* decides on which physical machines (PMs) of the data center those VMs must be placed. It uses the resource availability information of the PMs obtained from the Monitoring Module, for taking these decisions. The Monitoring Module retrieves the resource availability information of the PMs through the Host Monitoring Agent (HMA) installed in each PM of the data center. The Initial VM Placement Module decides to place a VM on a PM only if it satisfies the resource requirements of the VM. As initially the CPU utilization pattern of VMs are not known, the placement decisions are taken using the Best Fit heuristic in order to reduce the number of active PMs. This in turn reduces the energy consumption of the data center. The Cloud data center consists of a VM provisioner which provisions the VMs on the hosts based on the placement decisions of the Initial VM Placement Module.

The Guest Monitoring Agent (GMA) installed in each VM and the Host Monitoring Agent (HMA) installed in each PM continuously collect the resource utilization values of the VM and the PM, respectively and pass the data to the *Monitoring Module*. Based on these data, using suitable technique (discussed in Sect. 4), this module



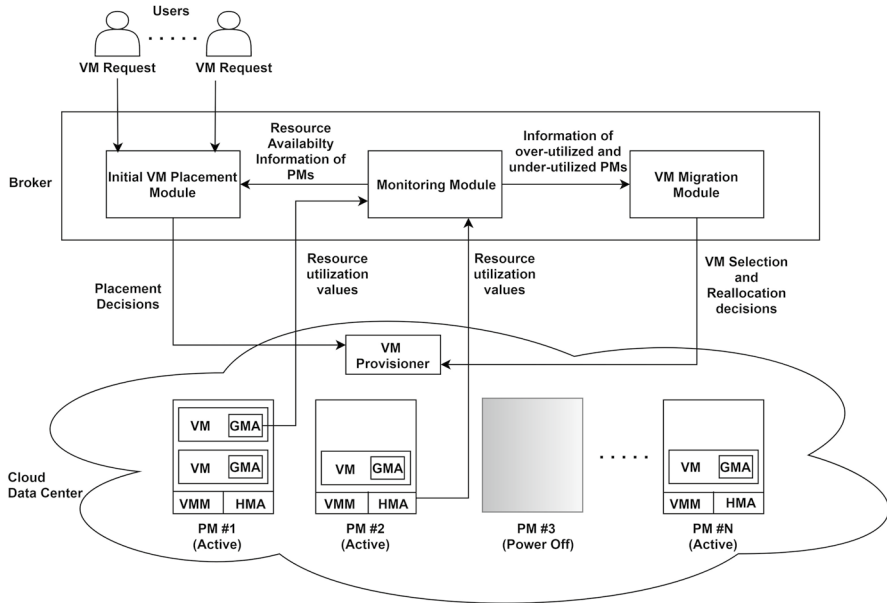


Fig. 1 VM consolidation model

periodically identifies the over-utilized and under-utilized hosts in the data center. Thereafter it instructs the *VM Migration Module* to take decisions for reallocation of VMs from these hosts using suitable VM Selection and VM Placement policies (discussed in Sect. 5). Based on these decisions, the VM Provisioner performs the live migration of the selected VMs in the data center.

In this paper, we focus on the Monitoring Module and the VM Migration Module to perform energy efficient and QoS-aware dynamic VM consolidation in a Cloud data center (discussed in Sects. 4, 5 and 6).

#### 4 Identification of over-utilized and under-utilized hosts in the data center

In a VM consolidation approach, it is important to properly identify the over-utilized and under-utilized hosts in the data center so that VMs can be selected for migration from them. In [8, 16, 35], threshold-based policies have been used to decide when a host is over-utilized or under-utilized. An upper and a lower CPU utilization threshold is set for the hosts in the data center. Whenever the current CPU utilization of a host violates the upper or the lower threshold, it is considered to be over-utilized or under-utilized, respectively. However, it is known that the CPU utilization of a host changes over time due to the dynamic nature of the workload of the VMs allocated to it [16]. So there lies the possibility of a host which has been detected to be over-utilized at a particular time, having much less CPU utilization in near future. Also, an under-utilized host may not remain the same in the next few instances of time.

Based on this observation, we propose a technique to predict the CPU utilization of a host in future. As the CPU utilization of a host depends on the workloads of the VMs allocated to it, it is appropriate to analyze the CPU utilization history of each of the allocated VMs, predict their future CPU utilization and thereby obtain the predicted CPU utilization of the host.

#### 4.1 Prediction of utilization based on Markov chain model

Markov chain is a mathematical model used widely for prediction [36, 37]. The main components of a Markov chain are as follows: a set of states,  $S = \{s_1, s_2, \dots, s_r\}$ ; Transition Probability  $p_{i,j}$  that represents the probability of transition from state  $s_i$  to state  $s_j$ ; and a State Transition Matrix comprising of the transition probabilities between different states. We use Algorithm 1, to determine the set of predicted CPU utilization values for a host.

---

#### Algorithm 1: predictUtilization

---

**Input:**  $host_{vmList}$ ,  $win$ ,  $host_{mips}$   
**Output:**  $Pred$

- 1  $n \leftarrow$  the number of states
- 2  $Pred \leftarrow \emptyset$  // Predicted CPU utilization values of the host
- 3  $x \leftarrow |host_{vmList}|$  // Number of VMs currently allocated to the host
- 4 Create a 2D matrix  $vmPredictedUtils$  of size  $x \times win$
- 5 **for**  $j := 1$  **to**  $x$  **do**
- 6  $stateSequence \leftarrow \emptyset$
- 7  $vm_{hist} \leftarrow$  CPU utilization history of  $vm_j$
- 8 **foreach**  $v_u$  in  $vm_{hist}$  **do**
- 9  $stateSequence \leftarrow stateSequence \cup \{f(v_u, n)\}$
- 10 **end**
- 11  $vm_{cur} \leftarrow$  current CPU utilization of  $vm_j$
- 12  $c \leftarrow \lfloor vm_{cur} \cdot (n - 1) \rfloor$  // Therefore  $st_c$  is the current state for  $vm_j$
- 13 Create Transition Matrix  $TM$  for  $vm_j$ , using Eq. (9) and Eq. (10)
- 14  $r \leftarrow \{0, 0, \dots, 0\}$  //  $r$  is a row vector
- 15  $r[c] \leftarrow 1$
- 16 **for**  $k := 1$  **to**  $win$  **do**
- 17  $r \leftarrow r \times TM$
- 18  $m \leftarrow i$  //  $r[i]$  is maximum value in  $r$ ,  $st_m$  is the predicted state for  $vm_j$
- 19  $v_u^m \leftarrow \frac{m}{n-1}$  //  $v_u^m$  is the CPU util value of  $vm_j$  corresponding to state  $st_m$
- 20  $vmPredictedUtils[j][k] \leftarrow v_u^m$
- 21 **end**
- 22 **end**
- 23 **for**  $k := 1$  **to**  $win$  **do**
- 24  $sum \leftarrow 0$
- 25 **for**  $j := 1$  **to**  $x$  **do**
- 26  $sum \leftarrow sum + (vmPredictedUtils[j][k] \times vm_{mips_j})$
- 27 **end**
- 28  $u_{host_k} \leftarrow \frac{sum}{host_{mips}}$  //  $u_{host_k}$  is the  $k^{th}$  predicted CPU utilization of host
- 29  $Pred \leftarrow Pred \cup \{u_{host_k}\}$
- 30 **end**
- 31 **return**  $Pred$

---

The input to Algorithm 1 are: the set of VMs currently allocated to the host ( $host_{vmList}$ ); the prediction window size ( $win$ ); and the maximum CPU capacity of the host represented in Million Instructions Per Second ( $host_{mips}$ ). The set of predicted CPU utilization values of the host  $Pred$ , is returned by this Algorithm, where  $|Pred| = win$ .

The CPU utilization of each VM in the  $host_{vmList}$  is predicted in lines 5 to 22. At first, the utilization values in  $vm_{hist}$  is used to obtain a sequence of states ( $stateSequence$ ). We consider there are  $n$  possible states for a VM, namely  $st_0, st_1, \dots, st_{n-1}$ . The function  $f(v_u, n)$  used to obtain the state corresponding to the CPU utilization  $v_u$  of a VM is stated as follows:  $f(v_u, n) = st_{\lfloor v_u \cdot (n-1) \rfloor}$ . Again, the CPU utilization of a VM in state  $st_j$  is calculated as  $\frac{j}{n-1}$ . The current CPU utilization ( $vm_{cur}$ ) of the VM is converted to the current state ( $st_c$ ).

Once the sequence of states ( $stateSequence$ ) for a VM is determined by mapping the utilization values present in  $vm_{hist}$  to the corresponding states, the transition matrix  $TM$  for the VM is created using Eq. (9) and Eq. (10).

$$TM = \begin{pmatrix} P_{0,0} & P_{0,1} & \dots & P_{0,(n-1)} \\ P_{1,0} & P_{1,1} & \dots & P_{1,(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ P_{(n-1),0} & P_{(n-1),1} & \dots & P_{(n-1),(n-1)} \end{pmatrix}, \tag{9}$$

where  $TM$  is a  $n \times n$  matrix and  $n$  represents the number of possible states for a VM. To calculate the value present in a cell of the matrix  $TM$ , we use Eq. (10) which is specified below:

$$P_{i,j} = \frac{t_{i,j}}{t_i}, \tag{10}$$

where  $t_i$  indicates the total number of transitions from the state  $i$  in the  $stateSequence$  of a VM; and  $t_{i,j}$  represents the total number of transitions from the state  $s_i$  to state  $s_j$  in the  $stateSequence$  of that VM.

Thereafter using the Markov chain model, we obtain the set of predicted utilization values for a VM. In Algorithm 1,  $r$  is a row vector of size  $n$ . Initially,  $r[0], r[1], \dots, r[n - 1]$  are all set to zero except for  $r[c]$  (corresponding to the current state,  $st_c$ ), which is set to one. After calculating  $r$ , in each iteration, the state ( $st_m$ ) which maximizes the values in  $r$  is predicted. The state ( $st_m$ ) is converted to the corresponding utilization ( $v_u^m$ ) and added to the matrix  $vmPredictedUtils$ .

After the CPU utilization of each of the VMs in  $host_{vmList}$  are predicted and stored in the  $vmPredictedUtils$  matrix, the possible future CPU utilization values of the host is calculated in lines 23 to 30 of this Algorithm. Here,  $u_{host_k}$  indicates the  $k$ th predicted CPU utilization of the host and  $vm_{mips_j}$  represents the CPU performance of the  $j$ th VM ( $vm_j$ ) allocated to the host in Million Instructions per Second (MIPS).

## 4.2 Prediction-based approach for over-utilized and under-utilized host detection

Based on the predicted CPU utilization values of the hosts obtained using Algorithm 1, we determine the set of over-utilized and under-utilized hosts in the data center, as described in Algorithms 2 and 3, respectively.

---

### Algorithm 2: Prediction Based Over-utilized Host Detection (PBOHD)

---

**Input:**  $hostList, Thr_{up}$   
**Output:**  $over_{list}$

```

1  $win \leftarrow$  prediction window size
2  $over_{list} \leftarrow \emptyset$ 
3 foreach host  $h$  in  $hostList$  do
4    $host_{vmList} \leftarrow$  set of VMs currently allocated to host  $h$ 
5    $host_{mips} \leftarrow$  MIPS of host  $h$ 
6    $Pred \leftarrow predictUtilization(host_{vmList}, win, host_{mips})$  // invokes Algorithm 1
7    $Pred_{over} \leftarrow \{u \mid u \in Pred \wedge u > Thr_{up}\}$ 
8   if  $|Pred_{over}| > \frac{win}{2}$  then
9      $over_{list} \leftarrow over_{list} \cup \{h\}$ 
10  end
11 end
12 return  $over_{list}$ 

```

---



---

### Algorithm 3: Prediction Based Under-utilized Host Detection (PBUHD)

---

**Input:**  $hostList, Thr_{low}$   
**Output:**  $under_{list}$

```

1  $win \leftarrow$  prediction window size
2  $under_{list} \leftarrow \emptyset$ 
3 foreach host  $h$  in  $hostList$  do
4    $host_{vmList} \leftarrow$  set of VMs currently allocated to host  $h$ 
5    $host_{mips} \leftarrow$  MIPS of host  $h$ 
6    $Pred \leftarrow predictUtilization(host_{vmList}, win, host_{mips})$  // invokes Algorithm 1
7    $Pred_{under} \leftarrow \{u \mid u \in Pred \wedge u < Thr_{low}\}$ 
8   if  $|Pred_{under}| > \frac{win}{2}$  then
9      $under_{list} \leftarrow under_{list} \cup \{h\}$ 
10  end
11 end
12 return  $under_{list}$ 

```

---

Algorithm 2 takes the set of hosts in the data center ( $hostList$ ) and the upper CPU utilization threshold for the hosts ( $Thr_{up}$ ) as input. It returns the set of over-utilized hosts in the data center ( $over_{list}$ ) as output. Similarly, input to Algorithm 3 is the set of hosts in the data center ( $hostList$ ) and the lower CPU utilization threshold for the hosts ( $Thr_{low}$ ). Output of this Algorithm is the set of under-utilized hosts in the data center ( $under_{list}$ ). As described in these Algorithms, we consider a host to be

over-utilized or under-utilized, not if it's current CPU utilization violates the upper or the lower CPU utilization threshold, respectively. Rather, a host is regarded to be over-utilized or under-utilized only if it is mostly found to remain so in near future. Each of the Algorithms 2 and 3 (PBOHD and PBUHD) invokes Algorithm 1 which predicts the CPU utilization of a host using the Markov chain model as discussed in Sect. 4.1. Here,  $win$  indicates the prediction window size. Line 8 of Algorithm 2 states that if a host is predicted to remain over-utilized for more than half the prediction window size, it is added to the  $over_{list}$ . Similarly if a host is predicted to remain under-utilized for more than half the prediction window size, it is added to the  $under_{list}$  (line 8 of Algorithm 3). The Algorithms 1, 2 and 3 are used by the Monitoring module (refer Fig. 1) to identify the over-utilized and under-utilized hosts in the data center.

If  $p$  is the total number of hosts in the data center and  $q$  is the maximum number of VMs allocated to a host, then time complexity of Algorithm 1 is  $O(q)$  and that of Algorithms 2 and 3 is  $O(pq)$ .

## 5 Virtual machine selection and it's placement on proper host

### 5.1 Increase in energy consumption due to allocation of a VM

As stated in Sect. 3.1, the power consumption of a host has a linear relationship with it's CPU utilization. It is known that the CPU utilization of a host depends on the workload of the VMs placed on it. Let a host in the data center has  $k$  VMs allocated to it. Then it's CPU utilization is calculated using Eq. (11) stated as follows:

$$u_k = \frac{\sum_{i=1}^k (vm_{mips_i} \cdot vm_{util_i})}{host_{mips}}, \quad (11)$$

where  $u_k$  is the CPU utilization of the host having  $k$  VMs allocated to it;  $host_{mips}$  is the maximum CPU capacity of that host represented in MIPS;  $vm_{mips_i}$  and  $vm_{util_i}$  represent the MIPS and the CPU utilization of  $i$ th VM on the host, respectively. The power consumption of this host  $P(u_k)$ , can be calculated using the Eq. (1) stated in Sect. 3.1.

Now, if a new VM is placed on this host, then it's CPU utilization and power consumption can be represented as  $u_{k+1}$  and  $P(u_{k+1})$  respectively. The increase in power consumption of the host ( $P_{incr}$ ) due to the allocation of the  $(k + 1)$ th VM on it, is calculated using Eq. (12) stated below:

$$\begin{aligned} P_{incr} &= P(u_{k+1}) - P(u_k) \\ &= \left[ \frac{P_{max} - P_{idle}}{host_{mips}} \right] \cdot (vm_{mips_{k+1}} \cdot vm_{util_{k+1}}) \end{aligned} \quad (12)$$

where  $vm_{mips_{k+1}}$  and  $vm_{util_{k+1}}$  are the MIPS and the CPU utilization of the newly allocated  $(k + 1)$ th VM;  $P_{idle}$  and  $P_{max}$  are the power consumption of the host when it is

idle and when it is fully utilized, respectively. We define the term power factor ( $P_f$ ) of a host as follows:

$$P_f = \left[ \frac{P_{\max} - P_{\text{idle}}}{\text{host}_{\text{mips}}} \right] \quad (13)$$

Thus, it is evident from Eq. (12) that if a VM can be allocated to a host with considerably lesser power factor ( $P_f$ ), then the increase in power and energy consumption of the data center can be reduced.

## 5.2 Consideration of QoS while migration of a VM

Apart from decreasing the energy consumption of the data center, we aim to propose a VM consolidation technique that would also ensure improved QoS. As discussed in Sect. 3.2, there is some performance degradation of the VMs during their migrations. Duration of a VM migration can be obtained using Eq. (4). In Eq. (4), we assume the available network bandwidth  $BW_i$  used for migration of VM  $vm_i$  is calculated as follows:

$$BW_i = \min(BW_{\text{srcHost}_i}, BW_{\text{destHost}_i}), \quad (14)$$

where  $BW_{\text{srcHost}_i}$  is the available network bandwidth of the source host from which VM  $vm_i$  is migrating; and  $BW_{\text{destHost}_i}$  is the available network bandwidth of the host to which VM  $vm_i$  is being placed after migration.

Longer duration of migration leads to increased performance degradation of the VMs that can be measured using Eq. (3). This in turn increases SLA violations, thereby affecting QoS. Thus, to decrease this performance degradation and achieve good QoS, we aim to reduce the migration duration of the VMs as well as the number of VM migrations in the data center.

## 5.3 VM selection policy

When a host in the data center is identified to be over-utilized or under-utilized, VMs have to be selected for migration from them. In Eq. (12), it is evident that selection of VMs with less value of ( $vm_{\text{mips}_i} \cdot vm_{\text{util}_i}$ ) will lead to less increase in energy consumption. Again, to reduce the duration of migration (Eq. (4)) and improve QoS, VMs with smaller memory (RAM) should be selected for migration. Thus, when a host is found to be over-utilized, we determine the energy–QoS ( $EQV$ ) cost of each of the VMs allocated to it using Eq. (15) stated as follows:

$$EQV_i = w_1 \cdot \left( \frac{vm_{\text{mips}_i} \cdot vm_{\text{util}_i}}{vm_{\text{mips}_{\max}} \cdot vm_{\text{util}_{\max}}} \right) + w_2 \cdot \left( \frac{vm_{\text{RAM}_i}}{vm_{\text{RAM}_{\max}}} \right), \quad (15)$$

where  $EQV_i$  indicates the energy–QoS cost of the VM  $vm_i$  allocated to the over-utilized host;  $vm_{\text{mips}_i}$  and  $vm_{\text{util}_i}$  is the MIPS and the CPU utilization of  $vm_i$  respectively;

and  $vm_{RAM_i}$  represents the RAM of  $vm_i$ . While calculating the energy-QoS cost of a VM, we first normalize each component and then apply the linear weighted sum approach. Here,  $w_1$  and  $w_2$  are the weight parameters such that  $w_1 + w_2 = 1$ ;  $0 < w_1 < 1$ ;  $0 < w_2 < 1$ ; and  $w_1, w_2 \in \mathbb{R}$ . Migration of a VM with smaller  $EQV$  cost will be more energy-efficient and lead to lesser performance degradation, as compared to a VM with greater  $EQV$  cost.

---

**Algorithm 4:** Energy and QoS aware VM Selection (EQVS)
 

---

**Input:**  $over_{list}, Thr_{up}$   
**Output:**  $migrationList_{ov}$

```

1  $migrationList_{ov} \leftarrow \emptyset$ 
2 foreach host  $h$  in  $over_{list}$  do
3    $u_{cur_h} \leftarrow$  current CPU utilization of host  $h$ 
4    $host_{vmList} \leftarrow$  set of VMs allocated to host  $h$ 
5   calculate the  $EQV$  cost of every VM in  $host_{vmList}$  using Eq. (15)
6   sort the VMs in  $host_{vmList}$  in ascending order of their  $EQV$  cost
7   foreach  $vm_i$  in  $host_{vmList}$  do
8     if  $u_{cur_h} > Thr_{up}$  then
9        $migrationList_{ov} \leftarrow migrationList_{ov} \cup \{vm_i\}$ 
10       $u_{cur_h} \leftarrow u_{cur_h} - \left( \frac{vm_{mips_i} \cdot vm_{util_i}}{host_{mips}} \right)$ 
11    end
12    else
13      break
14    end
15  end
16 end
17 return  $migrationList_{ov}$ 

```

---

Our proposed VM selection policy based on the calculated  $EQV$  cost of the VMs is presented in Algorithm 4. We name this technique as Energy and QoS-aware VM selection policy (EQVS). In Algorithm 4,  $over_{list}$  represents the set of over-utilized hosts in the data center;  $Thr_{up}$  is the upper CPU utilization threshold for the hosts;  $migrationList_{ov}$  is the set of VMs selected for migration from over-utilized hosts;  $vm_{mips_i}$  and  $vm_{util_i}$  is the MIPS and the CPU utilization of  $vm_i$  respectively; and  $host_{mips}$  is the MIPS of the host. Time complexity of EQVS is  $O(pq \log q)$  where  $p$  is the number of over-utilized hosts in the data center and  $q$  is the maximum number of VMs allocated to a host.

We have performed experiments by taking different values of weight parameters  $w_1$  and  $w_2$  while calculating the  $EQV$  cost of a VM. We have observed that a higher value of  $w_1$  in cost calculation tends to select VMs with lower CPU requirement for migration, giving less preference to the memory requirement of the VM. This generally causes lesser energy consumption but poorer QoS (high PDM and SLAV values). On the contrary, when higher value of  $w_2$  is used, a VM with smaller memory (RAM) is preferably selected for migration. This improves QoS (low PDM and SLAV values) but increases energy consumption. Empirically we find the values  $w_1 = 0.3$  and  $w_2 = 0.7$  give the best energy and QoS trade-off.

When a host in the data center is found to be under-utilized, all the VMs allocated to it are selected for migration.

#### 5.4 Energy and QoS-aware VM placement

During VM consolidation, a proper VM placement technique is needed. As VM placement is a NP-hard problem [22, 38, 39], we propose a heuristic for efficient allocation of VMs on the hosts keeping the aspects discussed in Sects. 5.1 and 5.2 into consideration. The proposed heuristic is named as energy and QoS-aware VM placement (EQVP) and has been presented in Algorithm 5. This Algorithm takes the following parameters as input: the set of hosts in the data center (hostList); the set of VMs to be allocated to the hosts (vmList); excludedHostList indicating the set of hosts in the data center which should not be considered for VM placement as described in Sect. 6; and the CPU utilization thresholds  $Thr_{up}$  and  $Thr_{low}$  for the hosts. It returns the array  $A$  representing the allocation of VMs to the hosts.

---

#### Algorithm 5: Energy and QoS aware VM Placement (EQVP)

---

**Input:** hostList, vmList, excludedHostList,  $Thr_{up}$ ,  $Thr_{low}$

**Output:** Allocation array  $A$  indicating mapping between hosts and VMs

```

1 acceptableHostList  $\leftarrow$  hostList - excludedHostList
2 sort vmList in descending order of their ( $vm_{mips} \cdot vm_{util}$ ) value
3 foreach  $vm_i$  in vmList do
4    $A[vm_i] \leftarrow$  null //  $A[vm_i]$  indicates the host on which  $vm_i$  will be placed
5    $H_n \leftarrow \{h \mid h \in \text{acceptableHostList} \wedge Thr_{low} \leq u_{cur_h} \leq Thr_{up}\}$ 
6   if  $|H_n| > 0$  then
7     Calculate the  $EQH$  cost of each host in  $H_n$  using Eq. (16)
8     sort  $H_n$  in ascending order of their  $EQH$  cost
9      $A[vm_i] \leftarrow \text{getSuitableHost}(H_n, vm_i)$  // Algorithm 6
10  end
11  if  $A[vm_i] = \text{null}$  then
12     $H_u \leftarrow \{h \mid h \in \text{acceptableHostList} \wedge u_{cur_h} < Thr_{low}\}$ 
13    if  $|H_u| > 0$  then
14      sort  $H_u$  in descending order of their  $u_{cur_h}$  value
15       $A[vm_i] \leftarrow \text{getSuitableHost}(H_u, vm_i)$  // Algorithm 6
16    end
17  end
18  if  $A[vm_i] = \text{null}$  then
19     $H_s \leftarrow \{h \mid h \in \text{acceptableHostList} \wedge h \text{ is in sleep mode}\}$ 
20    if  $|H_s| > 0$  then
21      Calculate the  $EQH$  cost of each host in  $H_s$  using Eq. (17)
22      sort  $H_s$  in ascending order of their  $EQH$  cost
23       $A[vm_i] \leftarrow \text{getSuitableHost}(H_s, vm_i)$  // Algorithm 6
24    end
25  end
26 end
27 return  $A$ 

```

---



**Algorithm 6:** getSuitableHost

---

**Input:** listOfHosts,  $vm_i$   
**Output:** The suitable host  $h$  for VM  $vm_i$

```

1 foreach host  $h$  in listOfHosts do
2   | if  $h$  has sufficient resources for  $vm_i$  then
3     |   | return  $h$ 
4     |   end
5 end
6 return null

```

---

In Algorithm 5, at first we try to place each VM  $vm_i$  in vmList on one of the hosts ( $H_n$ ) in the data center whose current CPU utilization lies within the thresholds. In order to obtain a suitable host for placement of VM  $vm_i$ , we calculate the energy-QoS ( $EQH$ ) cost of each host  $h$  in  $H_n$  using Eq. (16) stated as follows:

$$EQH_h = \alpha_1 \cdot \left( \frac{P_{f_h}}{P_{f_{\max}}} \right) + \alpha_2 \cdot \left[ 1 - \min \left( \frac{BW_h}{BW_{srcHost_{vm_i}}}, 1 \right) \right] + \alpha_3 \cdot u_{cur_h}, \quad (16)$$

where  $P_{f_h}$  is the power factor of host  $h$  obtained using Eq. (13);  $P_{f_{\max}}$  is the maximum  $P_f$  value among all the hosts in  $H_n$  used for normalization;  $BW_h$  is the available network bandwidth of host  $h$ ;  $BW_{srcHost_{vm_i}}$  is the available network bandwidth of the source host of VM  $vm_i$ ; and  $u_{cur_h}$  is the current CPU utilization of host  $h$ . Here  $\alpha_1$ ,  $\alpha_2$  and  $\alpha_3$  are weight parameters such that  $\alpha_1 + \alpha_2 + \alpha_3 = 1$  and  $\alpha_1, \alpha_2, \alpha_3 \in \mathbb{R}$ .

A host having lesser  $P_{f_h}$  value is more energy-efficient as explained in Sect. 5.1. On the contrary, a host having lesser  $BW_h$  would increase the VM migration duration thereby affecting the QoS. Again, placement of  $vm_i$  on a host in  $H_n$  having lesser  $u_{cur_h}$  would improve resource utilization and reduce the chance of VM migrations from that host due to over-utilization in future. Thus it is clear that the first component of  $EQH_h$  considers the energy efficiency of the host, whereas the QoS aspect is considered by the second and the third components. Hence accordingly, we set the weight parameters  $\alpha_1 = 0.5$  and  $\alpha_2 = \alpha_3 = 0.25$ . The host in  $H_n$  having minimum  $EQH$  cost is most suitable for placement of VM  $vm_i$ , in terms of energy efficiency and QoS.

If none of the hosts in  $H_n$  have sufficient resources for placement of  $vm_i$ , we try to place it on one of the hosts in  $H_u$  as stated in lines 11 to 17 of Algorithm 5. The host in  $H_u$  having highest CPU utilization is first considered for VM placement so that its CPU utilization no longer remains lesser than  $Thr_{low}$ , reducing chances of VM migrations due to under-utilization from it. If the other hosts in  $H_u$  are found to remain under-utilized in future, VMs can be migrated from them and they can be switched to sleep mode to reduce energy consumption of the data center.

If  $vm_i$  could not be allocated to any host in  $H_n$  and  $H_u$ , then a suitable host is activated from sleep mode and the VM is allocated to it as stated in lines 18 to 25 of Algorithm 5. Here,  $H_s$  represents the set of hosts that are in sleep mode. To obtain a suitable host for placement of  $vm_i$ , we calculate the energy-QoS ( $EQH$ ) cost of each host  $h$  in  $H_s$  using the following equation:

$$EQH_h = \beta_1 \cdot \left( \frac{P_{s_h}}{P_{s_{\max}}} \right) + \beta_2 \cdot \left[ 1 - \min \left( \frac{BW_h}{BW_{srcHost_{vm_i}}}, 1 \right) \right], \quad (17)$$

where  $P_{s_h}$  is the increase in power consumption of host  $h$  due to the placement of  $vm_i$ ; and  $P_{s_{\max}}$  is the maximum  $P_{s_h}$  value among all hosts in  $H_s$ .  $P_{s_h}$  has been defined below:

$$P_{s_h} = P_{idle_h} + P_{f_h} \cdot (vm_{util_i} \cdot vm_{mips_i}), \quad (18)$$

where  $P_{f_h}$  is the power factor of host  $h$  calculated using Eq. (13);  $vm_{util_i}$  and  $vm_{mips_i}$  is the CPU utilization and MIPS of  $vm_i$  respectively.

In Eq. (17), the first component of  $EQH_h$  considers the energy efficiency of the host and the second component considers the QoS aspect. Hence we take the weight parameters  $\beta_1 = \beta_2 = 0.5$ . A host in  $H_s$  having smaller  $EQH$  cost is more suitable for placement of  $vm_i$  in comparison to a host having greater  $EQH$  cost. If  $p$  is the number of hosts in the data center and  $q$  is the number of VMs to be allocated on the hosts, then the time complexity of Algorithm 5 is  $O(pq \log p)$ . Algorithms 4 and 5 are used by the VM Migration Module (refer Fig. 1) for VM selection and reallocation.

## 6 Virtual machine consolidation

We have integrated the Algorithms- PBOHD, PBUHD, EQVS and EQVP described in Sects. 4 and 5 to develop the Energy and QoS-aware VM Consolidation (EQC) technique presented in Algorithm 7. EQC periodically consolidates the VMs in a data center to improve energy efficiency and QoS.

---

### Algorithm 7: Energy and QoS aware VM Consolidation (EQC)

---

**Input:** hostList,  $Thr_{up}$ ,  $Thr_{low}$

- 1  $over_{list} \leftarrow$  PBOHD(hostList,  $Thr_{up}$ )
- 2  $migrationList_{ov} \leftarrow$  EQVS( $over_{list}$ ,  $Thr_{up}$ )
- 3  $A_{ov} \leftarrow$  EQVP(hostList,  $migrationList_{ov}$ ,  $over_{list}$ ,  $Thr_{up}$ ,  $Thr_{low}$ )
- 4 place each VM  $vm_i$  to corresponding host  $A[vm_i]$  in Allocation array  $A_{ov}$
- 5  $under_{list} \leftarrow$  PBUHD(hostList,  $Thr_{low}$ )
- 6 sort  $under_{list}$  in ascending order of their  $u_{cur_h}$  value
- 7  $host_s \leftarrow$  hosts in sleep mode
- 8 **foreach** host  $h$  in  $under_{list}$  **do**
- 9      $Ex \leftarrow over_{list} \cup host_s \cup h$
- 10     $migrationList_{un} \leftarrow$  all VMs allocated to host  $h$
- 11     $A_{un} \leftarrow$  EQVP(hostList,  $migrationList_{un}$ ,  $Ex$ ,  $Thr_{up}$ ,  $Thr_{low}$ )
- 12    **if** a suitable host for each VM in  $h$  could be found using EQVP **then**
- 13     | place each VM  $vm_i$  to corresponding host  $A[vm_i]$  in Allocation array  $A_{un}$
- 14     |  $host_s \leftarrow host_s \cup h$
- 15    **end**
- 16    **else**
- 17     | do not migrate any VM from host  $h$
- 18    **end**
- 19 **end**

---

The input to Algorithm 7 are:  $hostList$  representing the set of hosts in the data center; and the CPU utilization thresholds  $Thr_{up}$  and  $Thr_{low}$  for the hosts. At first the over-utilized hosts ( $over_{list}$ ) in the data center are identified using the PBOHD Algorithm and VMs are selected for migration from them using the EQVS policy. Then the selected VMs are placed on suitable hosts using the EQVP technique. It is logical that VMs selected from over-utilized hosts must not be placed on other over-utilized hosts in the data center. Thus, in this case  $excludedHostList = over_{list}$ . Thereafter, the under-utilized hosts in the data center are identified using the PBUHD Algorithm and the VMs allocated to them are selected for migration. Here,  $u_{cur_h}$  is the current CPU utilization of host  $h$ ,  $Ex$  denotes the excludedHostList and  $host_s$  is the set of hosts in the data center currently switched to sleep mode. If it is possible to allocate all the VMs from an under-utilized host to other hosts using EQVP, it is switched to sleep mode thus reducing the total energy consumption of the data center.

## 7 Performance evaluation

### 7.1 Experimental setup

As it is difficult to perform large-scale experiments repeatedly on real Cloud setup, we have used the CloudSim 3.0.3 [15] simulation platform to evaluate our proposed heuristics and compare it with other state-of-art approaches. CloudSim is a modern and well-accepted simulation framework for the Cloud computing environment that supports the creation of a large-scale virtualized data center, power modeling and VM provisioning techniques. Our simulations have been conducted on a computer having Intel(R) Core(TM) i5-7200U CPU @ 2.50 GHz processor, 4 GB RAM and 64-bit Windows 10 Operating System.

We have simulated a heterogeneous data center consisting of 800 hosts. There are 4 types of hosts in the data center, namely Hitachi HA8000/SS10 (DK1) (dual core CPU with 3067 MIPS per core, 8GB RAM); IBM System x3200 M2 (dual core CPU with 3000 MIPS per core, 4GB RAM); HP ProLiant ML110 G5 (dual core CPU with 2660 MIPS per core, 4GB RAM); and HP ProLiant ML110 G4 (dual core CPU with 1860 MIPS per core, 4GB RAM). In our experiment, we use the results of the SPECpower benchmark [40] that provides data on power consumption by the servers. The nature of power consumption of the aforementioned servers for different CPU Utilization(%) variations [41] is given in Table 1. The power consumption of a server for 100% CPU utilization and 0% CPU utilization represent its  $P_{max}$  and  $P_{idle}$  respectively. There are 200 hosts of each of the 4 types in the simulated data center. Each Hitachi HA8000/SS10 (DK1), IBM System x3200 M2, HP ProLiant ML110 G5 and HP ProLiant ML110 G4 host is modeled to have 1000 Mbits/s, 800 Mbits/s, 400 Mbits/s and 600 Mbits/s network bandwidth, respectively.

Similar to [16], we consider 4 types of VMs in the data center: the first type having 2500 MIPS single core CPU and 0.85 GB RAM; the second type

**Table 1** Power consumption (in Watts) of selected servers for different CPU utilization

Server	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
S1	24.3	30.4	33.7	36.6	39.6	42.5	45.6	51.8	55.7	60.8	63.2
S2	75.2	78.2	84.1	89.6	94.9	100	105	109	112	115	117
S3	93.7	97	101	105	110	116	121	125	129	133	135
S4	86	89.4	92.6	96	99.5	102	106	108	112	114	117

Here, S1, S2, S3 and S4 represent servers Hitachi HA8000/SS10 (DK1), IBM System x3200 M2, HP ProLiant ML110 G5 and HP ProLiant ML110 G4, respectively

comprises of 2000 MIPS single core CPU and 1.7 GB RAM; the third type has a 1000 MIPS CPU core and RAM of 1.7 GB; and the fourth type consists of a 500 MIPS CPU core and 613 MB RAM. The characteristics of the VMs correspond to Amazon EC2 instance types [42].

We have used the PlanetLab workloads provided by the CloudSim 3.0.3 toolkit for conducting our simulations. These workload data have been generated by considering the CPU utilization variations of more than 1000 VMs from servers located in various places of the world [16]. In each PlanetLab workload file, the CPU utilization values for a duration of 1 day taken at an interval of 5 min is given.

The properties of the PlanetLab workload traces are given in Table 2. Initially, the VMs are allocated to the hosts considering the resource requirements as specified by the VM types, assuming 100% CPU utilization. Thereafter, the VMs utilize less resources according to the data provided in the workload trace which create opportunities for dynamic consolidation of the VMs. We have conducted simulations for 24 h on each PlanetLab workload trace to evaluate our proposed algorithms.

## 7.2 Performance metrics

To analyze the performance of our proposed VM consolidation technique and compare them with the state-of-art approaches, we have used some metrics. One of them

**Table 2** Characteristics of PlanetLab workload traces

Date	Number of VMs	Mean CPU utilization of VMs (%)
3 March 2011	1052	12.31
6 March 2011	898	11.44
9 March 2011	1061	10.70
22 March 2011	1516	9.26
25 March 2011	1078	10.56
3 April 2011	1463	12.39
9 April 2011	1358	11.12
11 April 2011	1233	11.56
12 April 2011	1054	11.54
20 April 2011	1033	10.43

is the total energy consumed by the hosts in the virtualized data center where the energy consumption of each host is calculated using the power and energy model defined in Sect. 3.1. Another metric that we consider is the total number of VM migrations in the data center. To measure the level of SLA violations and gain an insight into the QoS provided by the VM consolidation approach, we use the SLAV (SLA violation), PDM (performance degradation due to migration) and SLATAH (SLA violation time per active host) metrics presented in Sect. 3.3. Finally, the trade-off between energy and QoS is determined using the ESV (energy and SLA violations) metric defined in Eq. (8).

### 7.3 Some state-of-art approaches

Some of the over-utilized and under-utilized host detection, VM placement and VM selection strategies present in the literature, are discussed in the following:

1. **Threshold-based over-utilized and under-utilized host detection strategy:** Upper and lower CPU utilization thresholds are set for the hosts in the data center. If the current CPU utilization of a host exceeds the upper threshold, it is considered to be over-utilized. Similarly, if the current CPU utilization is less than the lower threshold, the host is detected as under-utilized [8, 16, 35].
2. **First Fit (FF) VM Placement Policy:** Given a set of hosts, the FF policy assigns a VM to the first host that satisfies its resource requirements [4, 27].
3. **Modified Best Fit Decreasing (MBFD) VM Placement Policy:** In MBFD, the VMs are first sorted in decreasing order of their current CPU utilizations. Then each VM is allocated to the host that causes least increase in power consumption due to this allocation [8].
4. **Minimization of Migration (MM) VM Selection Policy:** MM policy [8] selects minimum number of VMs for migration from a host when it is found to be over-utilized. If a host is under-utilized, all its VMs are migrated.
5. **Minimum Migration Time (MMT) VM Selection Policy:** When a host is found to be over-utilized, the MMT policy migrates the VM that requires minimum time for migration in comparison to other VMs allocated to that host [16].

Many researchers [4, 24, 27, 43] have adopted the aforementioned strategies as baseline methods for evaluation of their work. In this paper, for better evaluation of our proposed over-utilized and under-utilized host detection (PBOHD and PBUHD), VM placement (EQVP) and VM selection (EQVS) policies, we execute them in combination with the state-of-art approaches discussed above. This enables us to comprehend the efficiency of our Algorithms as suitable strategies for each of the three steps of VM consolidation, namely over-utilized and under-utilized host detection, proper selection of VMs from those hosts and effective placement of the selected VMs on other hosts in the data center. Moreover, the efficacy of our VM consolidation approach presented in Algorithm 7 can be determined through this rigorous comparison.

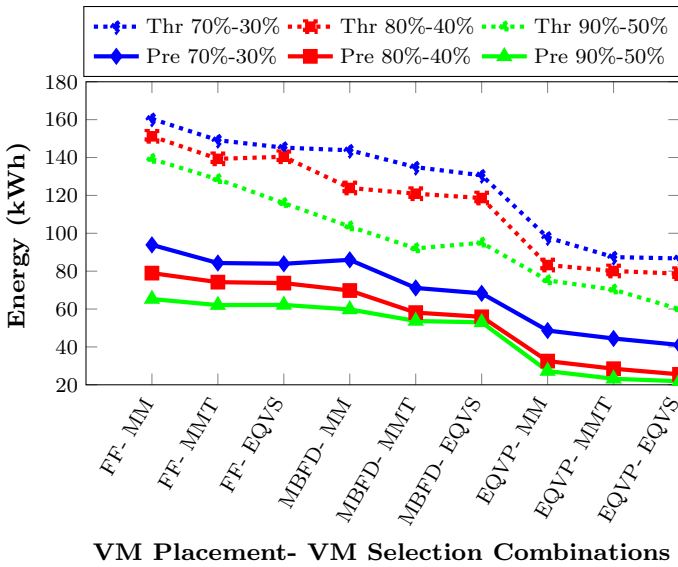
## 7.4 Simulation results and analysis

We have taken three upper and lower CPU utilization threshold combinations for the hosts in the virtualized data center. In each case we have implemented the threshold-based over-utilized and under-utilized host detection policy (discussed in Sect. 7.3) as well as our proposed prediction-based over-utilized and under-utilized host detection strategy (PBOHD and PBUHD) in combination with different VM placement–VM selection techniques. Each combination of an over-utilized and under-utilized host detection, VM selection, and VM placement technique is considered as a VM consolidation approach. The results are presented in Figs. 2, 3, 4, 5, 6 and 7. The results clearly show that our proposed VM consolidation approach EQC, comprising of prediction-based over-utilized and under-utilized host detection, EQVP VM placement policy and EQVS VM selection policy outperforms the other VM consolidation approaches by reducing energy consumption, number of VM migrations and SLA violations. The efficacy of EQC in achieving proper trade-off between energy and QoS is evident from the low ESV metric value generated by it.

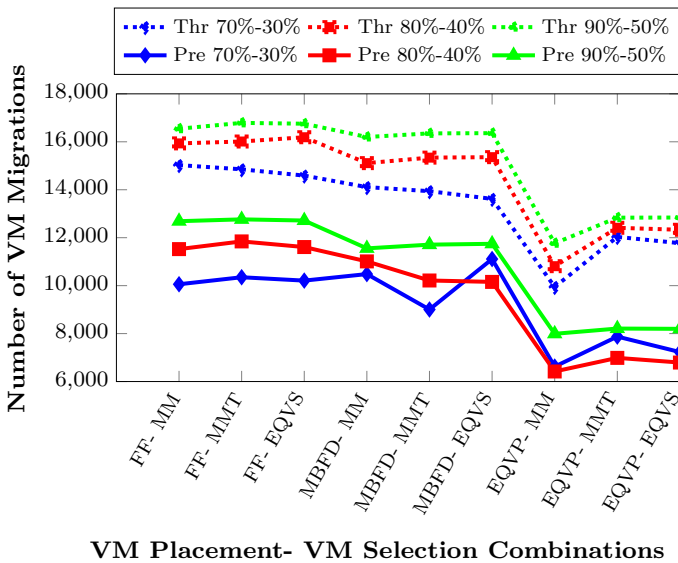
While conducting the prediction-based approach presented in Sect. 4, we have taken  $n$  as 100 and the prediction window size  $win$  as 6. In the CloudSim 3.0.3 platform, simulation is performed at an interval of 5 min. So a prediction window of size 6 is equivalent to a prediction of 30 min. Also, for prediction we need some history data. So for an initial offset phase of 1 h we do not perform any prediction. Thereafter, the CPU utilization values of a host are predicted at every time stamp and the decision of considering the host to be over-utilized or under-utilized is taken, as discussed in Sect. 4. Now, we briefly discuss the impact of over-utilized and under-utilized host detection, VM placement and VM selection strategies on a VM consolidation approach.

*Performance impact of over-utilized and under-utilized host detection strategy on VM consolidation* The results presented in Figs. 2, 3, 4, 5, 6 and 7 clearly show that in a VM consolidation approach, the use of the proposed prediction-based technique (PBOHD and PBUHD) for over-utilized and under-utilized host detection proves to be much more efficient than the threshold-based strategy present in the literature. In every threshold and in every VM placement- VM selection combination, the values of energy consumption, number of VM migrations, SLAV, PDM and ESV metrics obtained when prediction is used are much less than that of their threshold-based counterpart. Also, even though the value of the SLATAH metric obtained during our proposed prediction-based approach is sometimes greater than that of Thr, ultimately the SLAV metric, which is a product of SLATAH and PDM, is lesser in the prediction-based policy. This is because our proposed approach substantially reduces the number of VM migrations thereby lessening the value of the PDM metric. This in turn reduces the value of the SLAV metric.

We have considered three different CPU utilization threshold combinations with 40% interval between the upper and lower CPU utilization thresholds as specified in [8]. Figures 2 and 6 show that higher utilization thresholds reduces energy consumption but increases SLA violations. We can attribute this result to the fact that higher thresholds provide opportunities for aggressive VM consolidation which can



**Fig. 2** Energy consumption of the data center when threshold (Thr)-based and prediction (Pre)-based over-utilized and under-utilized host detection strategies (PBOHD and PBUHD) are implemented with different VM placement-VM selection combinations



**Fig. 3** Number of VM migrations in the data center when threshold (Thr)-based and prediction (Pre)-based over-utilized and under-utilized host detection strategies (PBOHD and PBUHD) are implemented with different VM placement-VM selection combinations

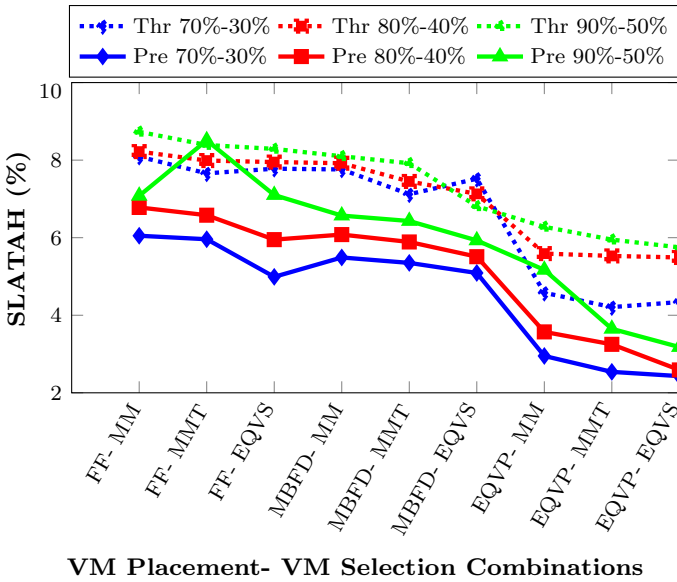


Fig. 4 The SLATAH metric values when threshold (Thr)-based and prediction (Pre)-based over-utilized and under-utilized host detection strategies (PBOHD and PBUHD) are implemented with different VM placement–VM selection combinations

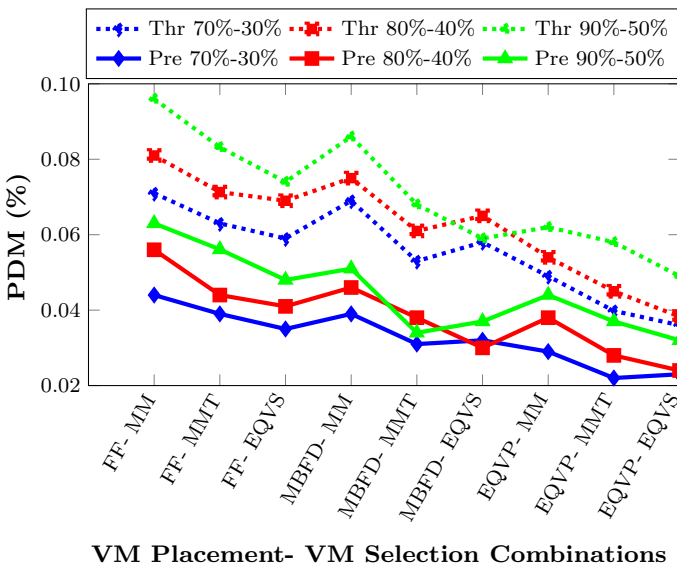
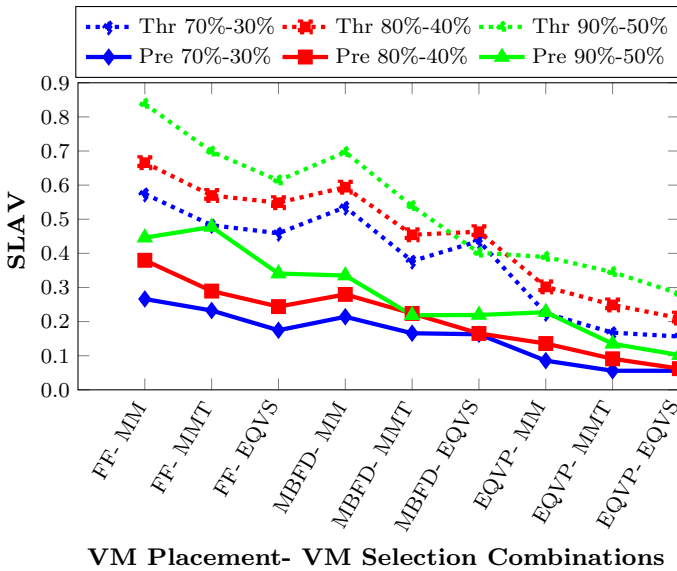
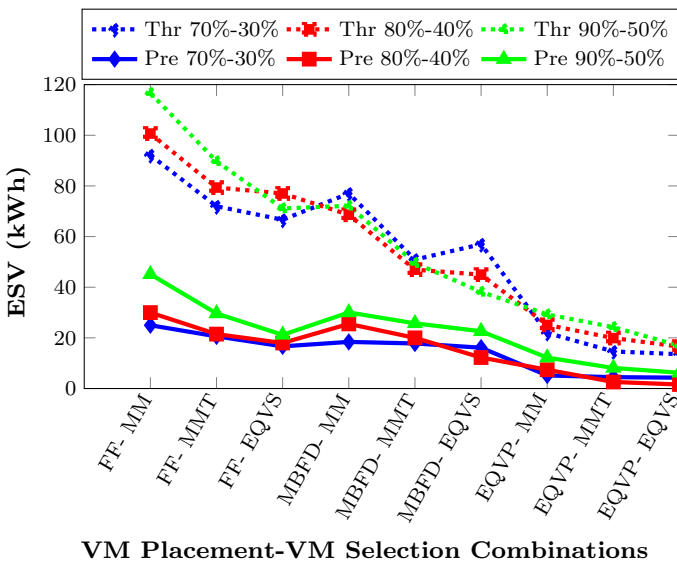


Fig. 5 The PDM metric values when threshold (Thr)-based and prediction (Pre)-based over-utilized and under-utilized host detection strategies (PBOHD and PBUHD) are implemented with different VM placement–VM selection combinations





**Fig. 6** The SLAV metric values when threshold (Thr)-based and prediction (Pre)-based over-utilized and under-utilized host detection strategies (PBOHD and PBUHD) are implemented with different VM placement-VM selection combinations

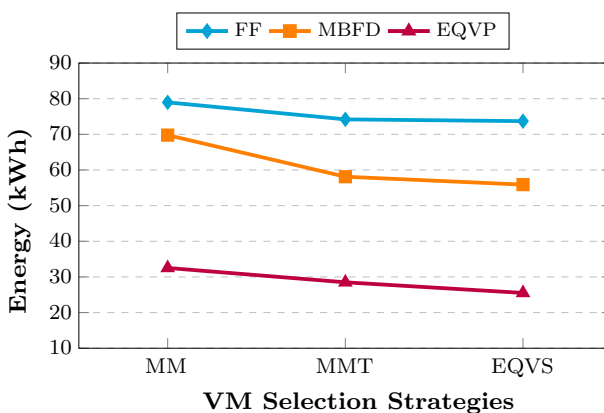


**Fig. 7** The ESV metric values when threshold (Thr)-based and prediction (Pre)-based over-utilized and under-utilized host detection strategies (PBOHD and PBUHD) are implemented with different VM placement-VM selection combinations

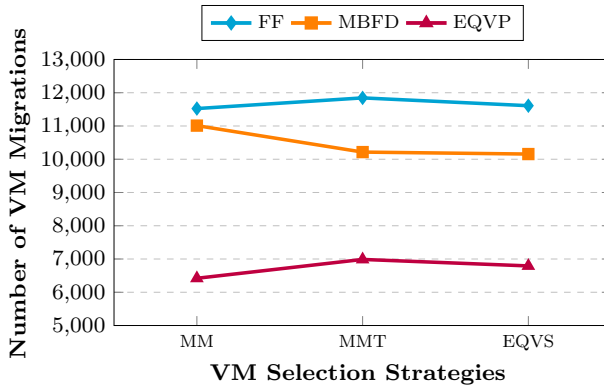
affect QoS. Thus, the service providers must set a proper threshold to maintain a trade-off between energy consumption and QoS requirements. It is evident from the results in Fig 7 that in comparison to the other threshold combinations, thresholds 80–40% generally provides better ESV. Also, it is evident that in every threshold combination, our prediction-based approach totally outperforms the threshold-based host detection strategy.

*Performance impact of VM placement strategy on VM consolidation* The results presented in Figs. 2, 3, 4, 5, 6 and 7 give us an idea about the relative efficacy of the VM placement and VM selection techniques. However, to achieve a clear understanding and perform better analysis, we have taken the VM placement strategies (FF, MBFD and EQVP) and the VM Selection strategies (MM, MMT and EQVS) and have implemented them in different combinations. In each case PBOHD (with  $Thr_{up} = 80\%$ ) and PBUHD (with  $Thr_{low} = 40\%$ ) have been used as the over-utilized and under-utilized host detection approach, respectively. The results are presented in Figs. 8, 9, 10, 11, 12 and 13.

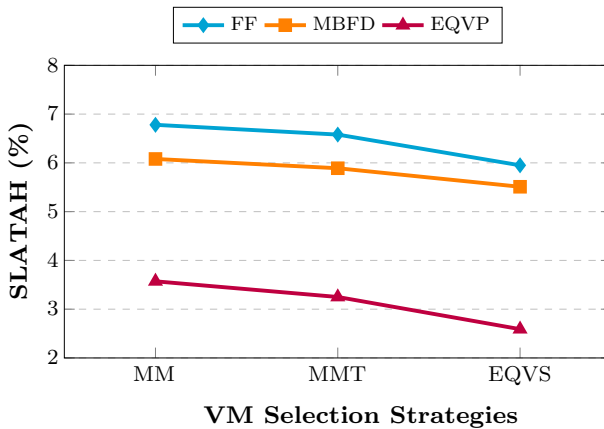
It is evident from the results in Figs. 8, 9, 10, 11, 12 and 13 that given a particular over-utilized and under-utilized host detection and VM selection policy, EQVP clearly outperforms MBFD and FF, when used as the VM placement policy. EQVP provides significant energy savings and reduces the number of VM migrations because it tries to allocate VMs on hosts having lesser *EQH* cost. It also reduces the time taken for VM migration which consequently reduces the value of the PDM metric. Moreover, EQVP effectively balances the load on the hosts by trying to allocate VMs on hosts having lesser CPU utilization. Hence, the value of the SLATAH metric is generally found to be low when EQVP is used. Also, the values of the SLAV metric (product of PDM and SLATAH) and the ESV metric (product of Energy and SLAV) are highly satisfactory when EQVP is used as the VM placement policy in a



**Fig. 8** Comparison of different VM Placement and VM Selection strategies with respect to the energy consumption of the data center. In each case PBOHD (with  $Thr_{up} = 80\%$ ) and PBUHD (with  $Thr_{low} = 40\%$ ) have been used as the over-utilized and under-utilized host detection approach, respectively



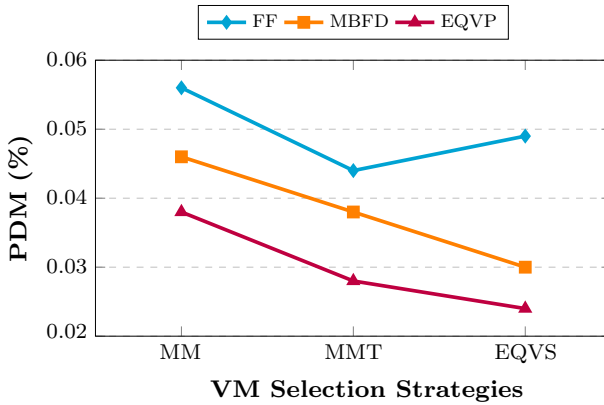
**Fig. 9** Comparison of different VM Placement and VM Selection strategies with respect to number of VM migrations in the data center. In each case PBOHD (with  $Thr_{up} = 80\%$ ) and PBUHD (with  $Thr_{low} = 40\%$ ) have been used as the over-utilized and under-utilized host detection approach, respectively



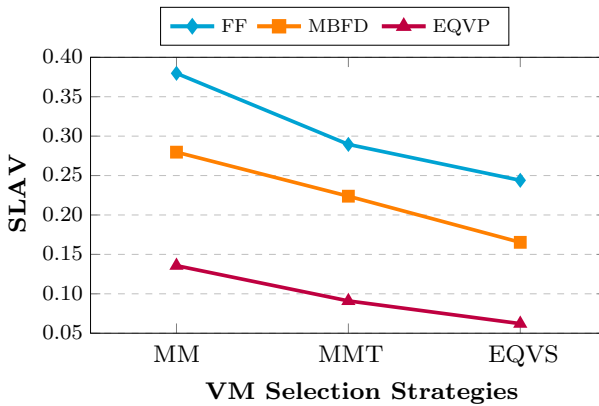
**Fig. 10** Comparison of different VM Placement and VM Selection strategies with respect to the SLATAH metric. In each case PBOHD (with  $Thr_{up} = 80\%$ ) and PBUHD (with  $Thr_{low} = 40\%$ ) have been used as the over-utilized and under-utilized host detection approach, respectively

VM consolidation approach. From the results it is evident that in comparison to FF, MBFD causes lesser energy consumption. Number of VM migrations, SLAV and ESV values of MBFD are also mostly lesser than that of FF.

*Performance impact of VM selection strategy on VM consolidation* The results presented in Figs. 8, 9, 10, 11, 12 and 13 help us to analyze the impact of the VM selection policy on the VM consolidation approach. For a given over-utilized and under-utilized host detection and VM placement policy, MM mostly causes lesser number of VM migrations but generates higher energy consumption, SLATAH, PDM, SLAV and ESV values in comparison to MMT and EQVS.

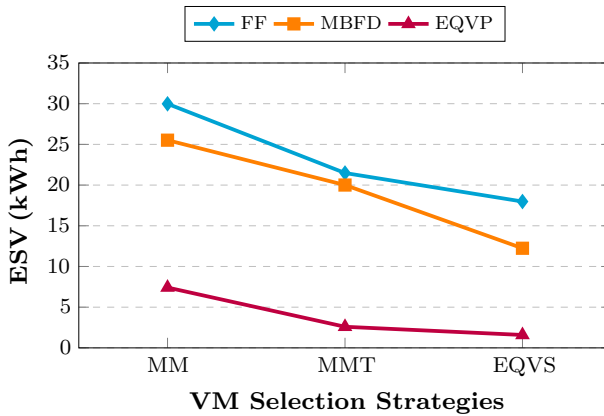


**Fig. 11** Comparison of different VM Placement and VM Selection strategies with respect to the PDM metric. In each case PBOHD (with  $Thr_{up} = 80\%$ ) and PBUHD (with  $Thr_{low} = 40\%$ ) have been used as the over-utilized and under-utilized host detection approach, respectively



**Fig. 12** Comparison of different VM Placement and VM Selection strategies with respect to the SLAV metric. In each case PBOHD (with  $Thr_{up} = 80\%$ ) and PBUHD (with  $Thr_{low} = 40\%$ ) have been used as the over-utilized and under-utilized host detection approach, respectively

Although the number of VM migrations in MMT is sometimes higher than that of MM, the value of PDM metric is lower. This is possibly because MMT selects VMs that require minimum time for migration, which reduces the performance degradation of VMs and thus PDM reduces. EQVS mostly provides least energy consumption. This is because while selecting VMs for migration, EQVS considers the  $(vm_{mips} \cdot vm_{util})$  value of the VM. The number of VM migrations are generally greater than MM and sometimes greater or lesser than MMT. As EQVS tries to select VMs with smaller memory (RAM), the duration of migration and performance degradation due to migration reduces. Thus, EQVS generates a PDM value which is lesser than MM. Although the SLATAH and PDM values



**Fig. 13** Comparison of different VM Placement and VM Selection strategies with respect to the ESV metric. In each case PBOHD (with  $Thr_{up} = 80\%$ ) and PBUHD (with  $Thr_{low} = 40\%$ ) have been used as the over-utilized and under-utilized host detection approach, respectively

generated by EQVS is sometimes higher than that of MMT, EQVS mostly leads to least energy consumption, SLAV and ESV values.

Thus, the efficacy of PBOHD and PBUHD, EQVS, and EQVP as suitable techniques for detection of over-utilized and under-utilized hosts in the data center, VM selection and VM placement, respectively, is established through this rigorous comparison. Also, the simulation results clearly show that our proposed EQC VM consolidation approach potentially outperforms other VM consolidation techniques by improving energy efficiency and QoS.

## 8 Conclusion and future work

Reduction in energy consumption of a Cloud data center while ensuring proper QoS to the users is one of the major challenges among the researchers. To address this problem, in this paper, we have presented an Energy and QoS-aware VM Consolidation approach (EQC) that can effectively consolidate the VMs among the heterogeneous hosts of a data center. EQC is developed by integrating the prediction-based over-utilized host detection (PBOHD), prediction-based under-utilized host detection (PBUHD), energy and QoS-aware VM selection policy (EQVS) and energy and QoS-aware VM placement policy (EQVP). Extensive simulations using CloudSim toolkit and PlanetLab workload data enable us to make the following conclusions. Firstly, the use of PBOHD and PBUHD in identification of over-utilized and under-utilized hosts in the data center substantially reduces the number of VM migrations thereby reducing SLA violations and improving QoS. Secondly, EQVS and EQVP clearly outperforms the existing VM selection and VM placement policies like MM, MMT, MBFD, etc., by significantly reducing energy consumption, number of VM migrations and SLA violations. Finally, the simulation results validate the efficacy

of EQC in achieving proper trade-off between two conflicting parameters—energy and QoS, in comparison to other state-of-art approaches.

In future, we will investigate the problem of consolidation of VMs in a Cloud data center by taking into account other system resources like memory and network bandwidth, as they also have an impact on the total energy consumption of the data center. We also plan to evaluate our proposed VM consolidation approach in a real Cloud environment like the OpenStack [44] in future.

**Acknowledgements** This research is supported by the UGC-NET Junior Research Fellowship (UGC-Ref. No.: 3610/(NET-NOV 2017)) provided by the University Grants Commission, Government of India and Visvesvaraya PhD Scheme of Ministry of Electronics & Information Technology, Government of India, being implemented by the Digital India Corporation (Ref. No. MLA/MUM/GA/10(37)C).

## References

1. Buyya R, Vecchiola C, Selvi ST (2013) *Mastering cloud computing: foundations and applications programming*. Newnes
2. Global warming: Data centres to consume three times as much energy in next decade, experts warn (2016) . <https://www.independent.co.uk/environment/global-warming-data-centres-to-consume-three-times-as-much-energy-in-next-decade-experts-warn-a6830086.html>. Accessed 13 April 2019
3. How to stop data centres from gobbling up the world's electricity (2018). <https://www.nature.com/articles/d41586-018-06610-y>. Accessed 13 April 2019
4. Zakarya M, Gillam L (2019) Managing energy, performance and cost in large scale heterogeneous datacenters using migrations. *Future Gener Comput Syst* 93:529–547
5. Dayarathna M, Wen Y, Fan R (2016) Data center energy consumption modeling: a survey. *IEEE Commun Surv Tutor* 18(1):732–794
6. Barroso LA, Hölzle U (2007) The case for energy-proportional computing. *Computer* 12:33–37
7. Fan X, Weber WD, Barroso LA (2007) Power provisioning for a warehouse-sized computer. *ACM SIGARCH Comput Archit News ACM* 35:13–23
8. Beloglazov A, Abawajy J, Buyya R (2012) Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Gener Comput Syst* 28(5):755–768
9. Boutaba R, Zhang Q, Zhani MF (2014) Virtual machine migration in cloud computing environments: benefits, challenges, and approaches. In: *Communication infrastructures for cloud computing*. IGI Global, pp 383–408
10. Ding Y, Qin X, Liu L, Wang T (2015) Energy efficient scheduling of virtual machines in cloud with deadline constraint. *Future Gener Comput Syst* 50:62–74
11. Li B, Li J, Huai J, Wo T, Li Q, Zhong L (2009) EnaCloud: An energy-saving application live placement approach for cloud computing environments. In: *2009 IEEE International Conference on Cloud Computing*. IEEE, pp 17–24
12. Speitkamp B, Bichler M (2010) A mathematical programming approach for server consolidation problems in virtualized data centers. *IEEE Trans Serv Comput* 3(4):266–278
13. Wang S, Zhou A, Hsu CH, Xiao X, Yang F (2016) Provision of data-intensive services through energy-and QoS-aware virtual machine placement in national cloud data centers. *IEEE Trans Emerg Top Comput* 4(2):290–300
14. Ghribi C, Hadji M, Zeghlache D (2013) Energy efficient VM scheduling for cloud data centers: exact allocation and migration algorithms. In: *2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*. IEEE, pp 671–678
15. Calheiros RN, Ranjan R, Beloglazov A, De Rose CA, Buyya R (2011) CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Softw Pract Exp* 41(1):23–50

16. Beloglazov A, Buyya R (2012) Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. *Concurr Comput Pract Exp* 24(13):1397–1420
17. Piraghaj SF, Dastjerdi AV, Calheiros RN, Buyya R (2015) A framework and algorithm for energy efficient container consolidation in cloud data centers. In: 2015 IEEE International Conference on Data Science and Data Intensive Systems. IEEE, pp 368–375
18. Shuja J, Madani SA, Bilal K, Hayat K, Khan SU, Sarwar S (2012) Energy-efficient data centers. *Computing* 94(12):973–994
19. How data center free cooling works (2015). <https://www.masterdc.com/blog/what-is-data-center-free-cooling-how-does-it-work/>. Accessed 05 Feb 2020
20. How data center cooling is changing (2018). <https://www.colocationamerica.com/blog/cooling-innovations-for-data-centers>. Accessed 05 Feb 2020
21. Zheng K, Wang X, Li L, Wang X (2014) Joint power optimization of data center network and servers with correlation analysis. In: IEEE INFOCOM 2014-IEEE Conference on Computer Communications. IEEE, pp 2598–2606
22. Verma A, Ahuja P, Neogi A (2008) pMapper: power and migration cost aware application placement in virtualized systems. In: Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware. Springer, New York, pp 243–264
23. Murtazaev A, Oh S (2011) Sercon: server consolidation algorithm using live migration of virtual machines for green computing. *IETE Tech Rev* 28(3):212–231
24. Tarafdar A, Khatua S, Das RK (2018) QoS aware energy efficient VM consolidation techniques for a virtualized data center. In: 2018 IEEE/ACM 11th International Conference on Utility and Cloud Computing (UCC). IEEE, pp 114–123
25. Gao Y, Guan H, Qi Z, Wang B, Liu L (2013) Quality of service aware power management for virtualized data centers. *J Syst Archit* 59(4–5):245–259
26. Farahnakian F, Ashraf A, Pahikkala T, Liljeberg P, Plosila J, Porres I, Tenhunen H (2015) Using ant colony system to consolidate vms for green cloud computing. *IEEE Trans Serv Comput* 8(2):187–198
27. Zhang X, Wu T, Chen M, Wei T, Zhou J, Hu S, Buyya R (2019) Energy-aware virtual machine allocation for cloud with resource reservation. *J Syst Softw* 147:147–161
28. Wu Q, Ishikawa F, Zhu Q, Xia Y (2016) Energy and migration cost-aware dynamic virtual machine consolidation in heterogeneous cloud datacenters. *IEEE Trans Serv Comput* 12(4):550–563
29. Laili Y, Tao F, Wang F, Zhang L, Lin T (2018) An iterative budget algorithm for dynamic virtual machine consolidation under cloud computing environment. *IEEE Trans Serv Comput*. <https://doi.org/10.1109/TSC.2018.2793209>
30. Chaisiri S, Lee BS, Niyato D (2011) Optimization of resource provisioning cost in cloud computing. *IEEE Trans Serv Comput* 5(2):164–177
31. Alnowiser A, Aldahiri E, Alahmadi A, Zhu MM (2014) Enhanced weighted round robin (EWRR) with dvfs technology in cloud energy-aware. In: 2014 International conference on computational science and computational intelligence, vol 1. IEEE, pp 320–326
32. Arroba P, Moya JM, Ayala JL, Buyya R (2015) DVFS-aware consolidation for energy-efficient clouds. In: 2015 international conference on parallel architecture and compilation (PACT). IEEE, pp 494–495
33. Tang M, Pan S (2015) A hybrid genetic algorithm for the energy-efficient virtual machine placement problem in data centers. *Neural Process Lett* 41(2):211–221
34. Zhang X, Lu JJ, Qin X, Zhao XN (2013) A high-level energy consumption model for heterogeneous data centers. *Simul Model Pract Theory* 39:41–55
35. Galloway JM, Smith KL, Vrbsky SS (2011) Power aware load balancing for cloud computing. *Proc World Congr Eng Comput Sci* 1:19–21
36. Markov chains and prediction (2008). <http://www.bandgap.cs.rice.edu/classes/comp140/f08/Module%206/Markovchainsandprediction.pdf>. Accessed 16 April 2019
37. Chapter 11: Markov chains (2016). [https://www.dartmouth.edu/~chance/teaching\\_aids/books\\_articles/probability\\_book/Chapter11.pdf](https://www.dartmouth.edu/~chance/teaching_aids/books_articles/probability_book/Chapter11.pdf). Accessed 16 April 2019
38. Beloglazov A, Buyya R (2013) Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints. *IEEE Trans Parallel Distrib Syst* 24(7):1366–1379

39. Jung G, Hiltunen MA, Joshi KR, Schlichting RD, Pu C (2010) Mistral: dynamically managing power, performance, and adaptation cost in cloud infrastructures. In: 2010 IEEE 30th International Conference on Distributed Computing Systems. IEEE, pp 62–73
40. The SPECpower Benchmark (2011) . [https://www.spec.org/power\\_ssj2008/](https://www.spec.org/power_ssj2008/). Accessed 03 May 2019
41. SPECpower\\_ssj2008 Results (2011). [https://www.spec.org/power\\_ssj2008/results/res2011q1/](https://www.spec.org/power_ssj2008/results/res2011q1/). Accessed 03 May 2019
42. Amazon EC2 instance types (2018). <https://aws.amazon.com/ec2/instance-types/>. Accessed 30 June 2018
43. Moghaddam SM, Piraghaj SF, O’Sullivan M, Walker C, Unsworth C (2018) Energy-efficient and SLA-aware virtual machine selection algorithm for dynamic resource allocation in cloud data centers. In: 2018 IEEE/ACM 11th International Conference on Utility and Cloud Computing (UCC). IEEE, pp 103–113
44. The openstack cloud computing platform (2019). <https://www.openstack.org/>. Accessed 09 May 2019

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.