J Supercomput (2018) 74:4623–4646
https://doi.org/10.1007/s11227-017-2071-1

QoS-aware VM placement and migration for hybrid cloud infrastructure

Kamran[1] · **Babar Nazir**[2]

Published online: 3 July 2017

Abstract Virtual machine (VM) migration is a process of migrating VMs from one physical server to another. It provides several benefits to a data center in a variety of scenarios including improved performance, fault tolerance, manageability load balancing and power management. However, VM's migration leads to performance degradation and service-level agreement (SLA) violations which cannot be ignored, particularly if critical business goals are to be met. In this paper, we propose an algorithm for VM's placement and migration that considers different users quality of service requirement, in order to decrease energy consumption and SLA violations due to under utilization of data centers. The proposed work mainly focuses on a novel heuristics-based energy-aware resource allocation to allocate the user's tasks in the form of cloudlets to the cloud resources that consumes minimal energy. In addition to that, it is incorporated with load balancing and constraint-based scheduling mechanism. The proposed work is implemented using the service-oriented-based architecture, and the same has been simulated using the CloudSim toolkit. In this paper, we compared our work with non-power-aware (NPA), dynamic voltage and frequency scaling (DVFS), single-threshold (ST) policies and minimization migration policy (MMP). The experiment results indicate that our approach saves about 83% of power comparing to the NPA system and 77% comparing to a system that apply only DVFS. However, if we compare these algorithms, which allow dynamic consolidation of VMs such as ST, it saves 53%, and finally, if we compare to MMP, it saves power between 22 and 38%. Similarly if we

✉ Babar Nazir
babarnazir@gmail.com

[1] Department of Computer Science, COMSATS Institute of Information Technology, Kamra Road, Attock 43600, Pakistan

[2] Department of Computer Science, COMSATS Institute of Information Technology, University Road, Tobe Camp, Abbottabad 22060, Pakistan

compare number of VM migration comparing to ST, it reduces 23 and 73% compared to MMP polices.

Keywords Cloud computing · Power management · Quality of service · Virtual machine placement · Server consolidation · Service-level agreement (SLA) · Utility computing · VM migration · Energy efficiency · Data center

1 Introduction

Cloud computing is an emerging computing paradigm that provides services over the network. In this environment, end users get their services and application from data centers and pay accordingly. This model is known as pay-as-you-go pricing model, main idea behind this is to provide computing power as a utility, such as other utilities like water, gas or electricity [1]. The rapid growth in cloud computing leads to the development of large-scale data centers all over the world. Data centers consume enormous quantities of electrical energy causing in high operating expenditures and CO_2 radiations.

Furthermore, global data centers energy consumption has grown by 56% from 2005 to 2010 and, however, in 2010 is assumed to be between 1.1 and 1.5% of the whole electricity used in the world [2]. According to Gartner, CO_2 radiation of the IT industry is presently expected to be 2% of the overall CO_2 radiation in 2007 that was same as the radiations produced by aviation industry and considerably contributes to the greenhouse effect [3]. Energy consumption of data centers is 10–100 times more per square foot than traditional office buildings. In 2013, cloud market share will be worth $150 billion. Thus, the service provider in cloud environment must adopt measures to prevent performance deprivation and high energy costs [3].

In order to reduce the total energy consumption of the entire data centers by itself, a number of methods have been exploited and tried, including: (1) improvement of the chip manufacturing to reduce the hardware power consumption while keeping high performance; (2) server consolidation using virtualization techniques to reduce the number of active server nodes; (3) rebuilding the heat dissipation systems, e.g., using novel heat dissipation methods such as water cooling to reduce the power consumption [3]. In cloud computing, virtualization is a key technology which permits service providers to create multiple VMs dynamically. It is a widely accepted solution for managing resources on physical machines. Instead of having its own dedicated physical machine (PM), each logical server runs on the VM hosted on the PM. Server consolidation using virtualization technology has become an important technology to improve the energy efficiency of data centers. Virtual machine placement is the key in server consolidation.

However, this is a well-known NP-hard problem. In the literature, several heuristics are proposed to solve such type of problems [8–10]. In this paper, we propose a new strategy for VM placement and VM migration that considers user's QoS requirement without violating SLA agreement. Further our technique also considers two type of users budget constraint user and time constraint user. For budget constraint users, we design algorithm for energy efficiency that not only achieves energy conservation but

also provides services within user's budget. We apply load balancing mechanism for time constraint users. Instead of switching off the server, to save energy or migrating VM to balance the load, we consider QoS requirement of the users.

Our contributions of the paper are:

- An architecture to support coordinated dynamic allocation of resources and scheduling of budget and deadline-constrained users.
- A heuristics for VM placement and migration. The objective of VM placement and migration is to minimize number of physical hosts on the basis of current utilization, whereas idle nodes are turned off by migrating VMs from one host to another to reduce power consumption.

We evaluate proposed algorithm using CloudSim [11] simulation tool kit for analysis and verification of our result.

The remainder of this paper is organized as follows. In Sect. 2, related work is discussed. Section 3 presents an extended proposed system model, algorithm illustration and motivation behind this work. In Sect. 4, we present the utilized benchmarks and the experimental settings for carrying our live migration. Evaluation and analysis of the experimental results are presented in Sect. 5. In the final section, conclusions are made and future research directions are discussed.

2 Related work

In the literature, a lot of research work has been done on the energy-aware resources allocation, i.e., VMs placement and migration in the data centers. Beloglazov et al. [12] proposed several heuristics for "energy-aware resource allocation." They have designed an architecture for VM provisioning, and they used an adapted version of Best Fit Decreasing (BFD) heuristic for bin packing. The basic aim of their work is optimal and efficient utilization of data centers resources; they said that the minimum number of servers is not optimal solution for energy efficiency. Therefore, they have used as sorting criteria for the servers to decide which are to be filled; first, a VM is mapped to the server that considers least energy. Secondly, they also proposed heuristics for VM migration, for they defined upper and lower threshold for server utilization and keep VMs utilization under these thresholds. If either threshold condition violates VM migration event, call for migration in order to decrease SLA violations and power consumption. Nevertheless, only simulation-based results based on energy-cost models and simple migration are shown. Lastly, only one resource dimension (i.e., CPU) is examined.

Pinheiro et al. [13] presented a technique, in which power management has been applied first time at data center level. This technique was suitable in a heterogeneous cluster environment for the reduction power consumption of servers, which are serving multiple web applications. The main idea behind this technique concentrates the workload to the least number of servers and turning off underutilized servers. It also needs to handle with the power/performance trade-off, because when we perform workload consolidation performance of application degraded. Users QoS requirements are also considered in terms of execution time and throughput to maintain SLA. In this work, servers are divided into master and slave server, algorithm runs only on the master

server and monitor instantaneously utilization of slave server resources, namely CPU, network interface and disk storage, on the basis of their utilization, and it takes a decision of turning servers on/off in order to reduce power consumption and at the same time provide the necessary performance. The system does not handle the responsibility of load balancing; therefore, it is also responsibility of application to manage the loads of its resources. Since the algorithm runs only on the master server, it creates a single-point failure, which becomes a performance degradation issue in case of large heterogeneous environment.

Bobroff et al. [14] proposed a technique for VM provisioning and migration in a heterogeneous environment. The main objective of this work was to minimize power consumption of servers while sustaining SLA. In this technique, resources are allocated to nodes on the basis of estimation using history and usage of computing nodes. The algorithm uses time series analysis and historical data for demand forecasting to periodically decrease the number of hosts for the better power consumption while reducing SLA violations. However, this technique does not consider the number of VM migrations for new placement. They used First-Fit-Bin heuristics which, however, decrease the number of hosts used well, but it requires large amount of migrations. The similarity to their algorithm to ours both attempts to decrease the number of servers used, and reduce SLA violations, but we also consider user QoS requirements and also reduce the number of migrations in our algorithm.

In [15], authors have explored the issue of "energy-efficient resource provisioning" in virtualization environment Their technique increases the utilization of resources to reduce energy consumption. In order to make sure each user QoS requirements, they assigned priority to each application. In this technique each application can be deployed using multiple VMs instantiated on several servers. However, only RAM and CPU utilization are considered in resource management decision. In [8], the authors have addressed the issue of power-aware dynamic provisioning of application in a heterogeneous virtualized environment as continues optimization. In this work, algorithm check allocation of VM's at each time frame for the optimization of VM's placement to decrease power consumption and increase performance. To address the issue of application placement the authors have proposed a pMapper framework for application placement. In this work there are four main actors involve in this frame, three of them managers and an arbitrator which coordinates their actions and makes allocation decisions.The main responsibility of application behaviour and resizes VM's on the basis of current resource requirement and the SLA.The responsibility of power manager was to monitor the power state of hardware and applying power saving scheme DVFS. Migration Manager was in charge of migration-related issues, for the efficient workload consolidation of VMs. Lastly Arbitrator has a global view of the system and makes decisions for new VMs placement and search VMs on nodes, which should be migrated to achieve this placement. The authors have applied a heuristic for the bin packing problem with variable bin sizes and costs. To handle the bin packing problem, First-Fit Decreasing (FFD) algorithm has been improved to work for different sized bins with item-dependent cost functions. The proposed algorithms consider strict SLA requirement contrary to existing algorithm.

In [16], the authors have addressed the issue of VM consolidation, and they have proposed a technique for scheduling application in Grid environments, which mainly work using predicting the cost of communication events.

According to him, if the migration cost is lower than the estimated communication cost than they allow the communication process to migrate across network, ultimate objective of this approach is to reduce the overall execution time of communication process. The experiment results show that this method is suitable in grid environment; however, this approach is not suitable for virtualized data centers, because the VM migration cost is greater than migration cost.

In [17], the authors have explored the problem of energy consumption in data center and network architectures. However, this approach cannot apply dynamically, optimization of network applied only at the data center design. Kusic et al. [18] have studied the problem of power management in a heterogeneous virtualized environment as a sequential optimization and addressed it using Limited Look ahead Control (LLC). This method permits for multi-objective optimization under explicit operating constraints and is appropriate for computing systems with nonlinear dynamics where control inputs must be selected from a finite set. The main goal was to increase the profit of an infrastructure provider, decrease power consumption and also decrease SLA violations.

In [19], the authors have addressed the issue of "power-efficient resource management" in heterogeneous virtualized data centers. In addition to hardware scaling and VM consolidation, the authors have proposed a new technique for power management known as soft resource scaling. The main objective is to copy hardware scaling by provisioning a smaller amount of resource time for a VM by the virtual machine monitor's (VMM) scheduling capability. In this technique, authors discovered that a grouping of "soft and hard" scaling may provide larger power saving; this is due to the fact that hardware has limited number of states. In this research, authors also introduced a new architecture for resource management, which is distributed into local and global policies. At the local level, the system takes advantage of the guest OS's power management strategies, but such strategies may appear to be unproductive, as the guest OS may be a legacy or power unaware. The experimental results show that reduction in power consumption up to 17% by exploiting power management heterogeneity.

In [20], authors have studied VMs consolidation to enhance the usage of hardware resources and decrease power consumption, and they also presented four models, which are used to detect the performance noises among disk utilizations, CPU and the costs of VM migrations.

In [15,22], the authors address the problem of energy-efficient resources allocation and proposed a project known as "Green Cloud Project." The main objective of this project is provisioning of Cloud resources, while assuring QoS requirements characterized by the SLAs established through a negotiation between Cloud consumers' providers. The project addresses the issue of "power-aware allocation of VMs" in Cloud data centers for application services according to customers QoS requirements such as budget and deadline constraints.

Another work on energy efficiency, performance (i.e., SLA violations), virtualization overheads and proposed a "multi-objective profit oriented" VMs provisioning

algorithm. Likewise to [14], this work focuses on CPU only and its assessment is based on simulations.

Abrishami et al. [21] proposed two algorithms for cost-optimized, deadline-constrained execution of workflows in the Clouds. As explained in the next section, these algorithms do not consider all data transfer times during provisioning and scheduling, increasing the execution budget. Our proposed algorithm is based on one of such algorithms minimization migration policy and MBFD, but also accounts for data transfer times and Cloud resources boot time during the provisioning and scheduling process. Furthermore, it explores the possibility of tasks, deadline, budget and replication in order to increase the probability of meeting application deadlines.

In [22], the authors explored the problem of network overhead due to VM migration and proposed a headrest in order to decrease the number of VM migration and network overhead.

The authors use a combination of sever consolidation and migration control to decrease the number of VM reallocation with low SLA violations. They constructed an LP formulation and heuristics to control VM migration, which prioritizes virtual machines with a stable volume. Nevertheless, experimental results show greater improvement in power reduction and cost optimization than their approach.

In [23,24], the authors have investigated the performance and energy cost for live VM migrations from both theory and practice. They proposed an automated strategy for virtual machine migration in a self-managing virtualized environment, as well as an optimization model based on linear programming. In contrast, in our work we employ a two-threshold VM migration strategy to balance the loads across different physical nodes in the data center, in order to meet the varying demands of user applications.

In [25], the authors have proposed a technique known as "Cloud auto-scaling with deadline and budget constraints," the authors also proposed a heuristic for dynamic allocation of host task with a deadline, and it also takes VM start-up time.

Nevertheless, they focus on the allocation decisions rather than on job provisioning. Rather than the examined studies as shown in Table 1, in this paper, we proposed heuristics for of VMs provisioning which consider different users QoS requirement and applying VM migration on the basis of current utilization of resources and thus decrease power consumption. This technique can handle effectively strict QoS requirements, heterogeneous VMs and heterogeneous infrastructure. The proposed technique does not rely on a specific kind of workload and does not need any learning about applications executing on VMs.

3 Proposed strategy

In this section, the basic architecture of the proposed work and model definitions are presented. The core part of the system design lies at the broker layer. The broker acts as an interface between the users and the service providers. The broker may also be granted the rights to negotiate SLA contracts with cloud providers on behalf of the customer. The user forwards their job request to cloud infrastructure, and broker receives user request and takes appropriate decision to process job. The broker initiates the processing of the request, and later it submits the requests to other modules. It provides an interface for managing multiple clouds and share resources. Figure 1 demonstrates

Table 1 Classification of related work

Author	Techniques	SLA awareness	Energy efficiency	Load balancing	Budget time	Network load minimization
Antonet al. [26]	MBFD and VM migration	√	√	√	×	×
Pinheiro et al. [13]	Server power switching	√	√	×	√	√
Bobroff et al. [14]	VM provisioning and migration	√	√	√	×	×
Song et al. [15]	Resource throttling	×	√	×	√	×
Verma et al. [8]	DVFS, VM consolidation	√	√	√	×	×
Gyarmati et al. [17]	Limited look ahead control	×	√	√	×	√
Nathuji et al. [19]	Soft and hard power scaling	√	√	√	×	×
Sharifi et al. [20]	Workload-aware consolidation	×	√	√	√	×
Abrishamietal.[21]	Deadline-constrained workflow scheduling algorithm	×	√	×	√	×
Tiago et al. [22]	Server consolidation and migration control	√	√	√	×	√
Liu and Jong-Geun [23,24]	Virtual machine migration	√	√	√	×	×
Mao [25]	Auto-scaling mechanism	√	×	×	√	×

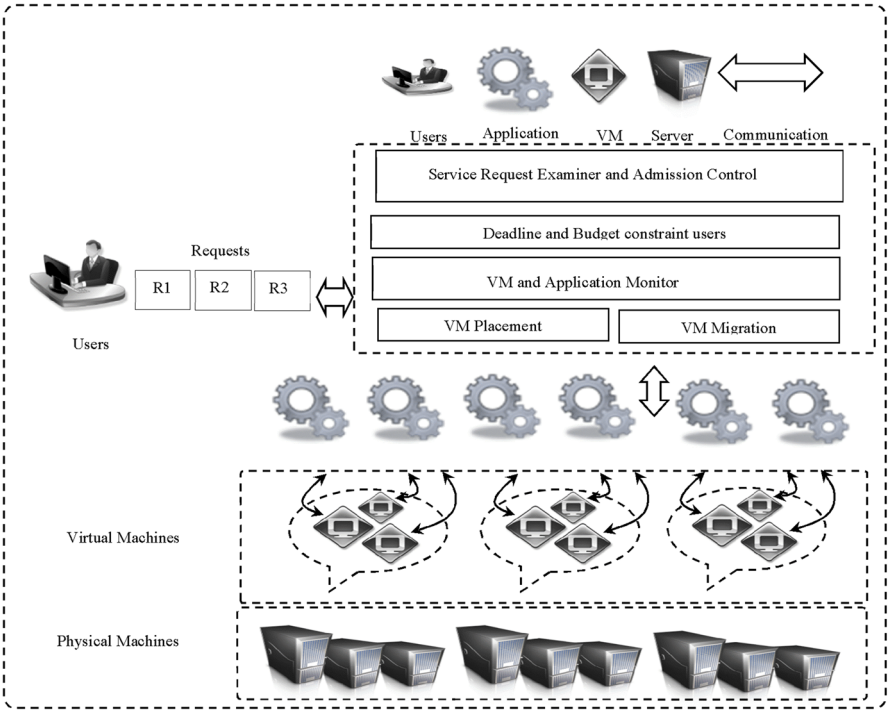

Fig. 1 Proposed system architecture

the proposed system architecture for supporting energy-aware resource provisioning in data centers. In following sections, first we discussed problem formulation after that our system model and their major components.

3.1 Problem formulation

We formulate the problem of QoS-aware VM placement and migration for cloud infrastructure. Assuming that there are N physical machines (servers) in a data center, represented by

$$\mathrm{PM} = \{\mathrm{PM}_i(\mathrm{PE}_i, \mathrm{MIPS}_i, \mathrm{RAM}_i, \mathrm{BW}_i) | i = 1 \ldots \mathrm{m}\} \tag{1}$$

and the set of VMs represented by

$$\mathrm{VM} = \{\mathrm{VM}_j(\mathrm{PEj}, \mathrm{MIPS}_j, \mathrm{RAM}_j, \mathrm{BW}_j) | j = 1 \ldots \mathrm{n}\} \tag{2}$$

$$\begin{aligned} \text{CPU utilization} = {} & ((\text{total CPU} - \text{CPU used by host} \\ & - \text{CPU used by VMs}))/\text{total CPU} \end{aligned} \tag{3}$$

Each server can host one or more virtual machines (VMs). The current CPU utilization of server PMi is calculated by using Eq. 3, and each virtual machine VM_j requires PE_j processing elements, $MIPS_j$, RAM_jMBytes of physical memory, BW_j Kbits/s of network bandwidth.

i. Users submit their request for application which is sent to the SaaS provider's (Cloud Broker) application layer with QoS constraints, such as deadline, budget. Then, the platform layer of SaaS provider utilizes the admission control and scheduling mechanisms to admit or reject this request.
ii. If the request accepted, a formal agreement (SLA) is signed between both parties to guarantee the QoS requirements such as response time.
iii. Cloud broker assigns the cloudlets to the available cloud resources, simply the VMs.
iv. If cloudlet successfully executed, its result submitted to the user.
v. The result of the job is submitted to user upon successful completion of the job.
vi. If the cloud resource over-utilize during execution of the tasks, due to the over-subscription and variability of the workload experienced by VMs, at the time t the overall demand for the CPU performance exceed the available CPU, the cloudlets is rescheduled on another resource which starts executing the job from scratch. This leads to more time consumed for the job than expected. Thus, the user's QoS requirements are not satisfied.
vii. To address this problem, the load balancing mechanism is used. Using load balancing, we migrate VMs from over-loaded host to another suitable host.
viii. If the host gets underutilize and there is no deadline cloudlet inside underutilized host, then we turn off this host and migrate all VMs from to current server to another suitable server. In this way, we can save energy.

3.2 System model

In the following sections, we will discuss system model of proposed work in detail.

3.2.1 Brokers/consumers

The main responsibility of this module is to provide a communication facility between users and service providers and provide services according to users QoS requirement, finalize SLA requirement and deploy services across Clouds.

This module is also responsible for examining user's QoS service requirements for instances CPU performance, RAM, network and bandwidth after that deciding whether to approve or discard the request according to updated energy information given from monitor module. By considering different users QoS requirement, we divide cloud users into two main categories, i.e., deadline and budget constraint users. Deadline users want the application results by a specific deadline, while budget constraint users want applications and services within their budget.

3.2.2 Service request examiner and admission control

When users submit their request for service, this module interprets and analyzes the submitted requests for QoS requirements before determining whether to approve or disapprove the request.

Further, this module accepts all the requests that do not have a deadline constraint. The requests have deadlines, and it makes the decision whether it has enough time and credit to run the job within deadline or not, if job completed within deadline, it is accepted otherwise rejected.

It guarantees that no SLA violation occurs by decreasing the chances of resource overloading whereby many service requests cannot be fulfilled successfully due to inadequate resources available. Therefore, it also requires updated information regarding the resource availability VM monitor and workload processing to make resource allocation decisions effectively. Then, it allots requests to VMs and determines resource entitlements for allocating VMs.

3.2.3 VM and application monitor

VM and application monitor is an important module of system architecture, and it is in charge of monitoring server and virtual machine. The main responsibility of this module is to control VMs and host status, CPU utilization, SLA violation and power consumption. VM monitor mechanism keeps track of the availability of VMs and their resource entitlements, while in the case of application software services, the performance is continuously monitored to identify any breach in SLA and send a notification trigger to SLA Resource Allocator for taking appropriate action

3.2.4 Physical machines (PM)

In a cloud environment, physical machine is a basic building block, and it is also the physical container of virtual machines. A PM provides the hardware infrastructure for creating VMs. A PM can hold multiple virtual machines. The number of VMs a PM can host is called the PM's capacity.

3.2.5 Virtual machines (VMs)

VM is a logical representation of PM using software (known as VMware) that delivers an operating environment which can execute or host a guest operating system and can provision for several application environments. Different VMs may have different applications with diverse resource requirements an operating systems running on them. Migration is a process of transfer of VMs from one PM to another. Migration of VMs takes place to enhance the performance, hardware utilization and power utilization. VMs can be migrated using either static or dynamic, also called live migration (Figs. 2, 3).

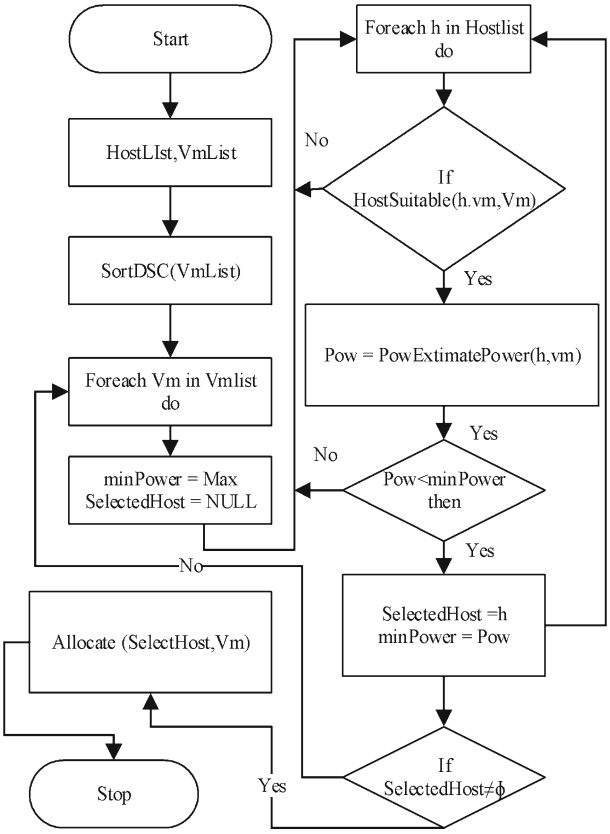

Fig. 2 Flowchart for VM placement

3.3 VM placement

VM provisioning in a heterogeneous virtualized data center is one of the main issues in cloud computing. VM provisioning can be considered as a "bin packing problem" with a different bin capacity and prices; here bins depict the physical hosts; items are the VMs that have to be assigned; bin sizes represent the CPU capacities of the hosts; and prices correspond to the power consumption by the hosts. VM placement can be divided into two phases, in first phase VM assign to suitable server when a request comes from the cloud, as shown in QoS-aware Best Fit Decreasing (QoS-BFD) algorithms, and in this algorithm we consider the time constraint user and budget constraint user. While in second phase VM placement is done for optimization, i.e., when servers are overloaded, during this phase VM migrates from an overloaded server to another, until server obtain normal load. The pseudocode for VM placement algorithm is presented in Algorithm 1.

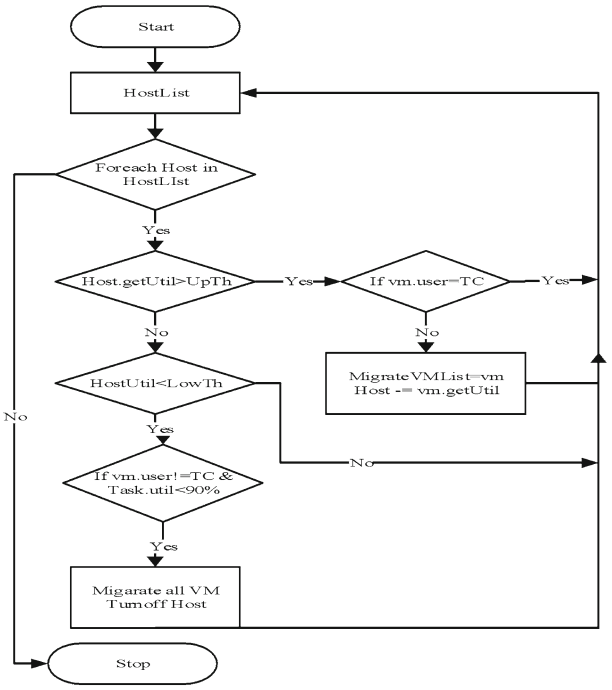

Fig. 3 Flowchart for VM migration

Algorithm 1 For Virtual Machine Placement (QoS-BFD)

```
1 Input: Vm_List, Host_ListOutput: VmPlacement
2  While Virtual Machine in VirtualMachineList do:
3   minimizePower = Interger.Maximum
4   AssignHost = ϕ
5          While Host in Host_List do:
6                 if (HostHaveEnougResoures()  && !HostNotCritial()) than
7                       power ←EstimatePower(Host,VM)
8                            if Power <minimizePower
9                                 AssignHost = Host
10                                minimizePower = Power
11                           endif
12                  endif
13            if AssignHost!= Null
14              Add(Host, vm)
15              break
16            endif
17    endfor
18   endfor
```

3.4 VM migrations

VM migration is an optimization step, in which VMs migrate from one server to another server according to server utilization, as shown in Algorithm 2, which is used for virtual machine migration. In order to decide which VM virtual machine is a candidate for migration, we presented dual-threshold VM selection policies and also considering time constraints and budget constraint users while selecting VM for migration. This algorithm has two parts, in first part we select VMs form overloaded server, and in this case we select only those VMs which are not critical, critical VMs those types of VMs in which time constraint task is not running and no job gets executed greater than 90%. The second part of algorithm searches underutilized hosts, once the underutilized host is found the next step is to migrate VMs form underutilized hosts to another. We further check QoS parameter while migrating VMs one server to another. To efficiently handle SLA requirement and decrease the number of VM migration, we restrict VM migration for time constraint users, for this if user jobs are in execution state, or it is 90% executed, in such conditions there is no need for migration.

Throughout VMs relocation process, additional network traffic is produced during the entire migration period since hypervisor needs to migrate memory states of the running VM to the designated host. Therefore, reliability VM relocation policy also depends on the number of required VMs migration. In QoS-MMP VMs placement and migration policies we also consider network bandwidth, the main objective is to minimize overall network traffic overhead in the data center. In order to accomplish, this objective is to place VMs with large amount of traffic communication in adjacent hosts.

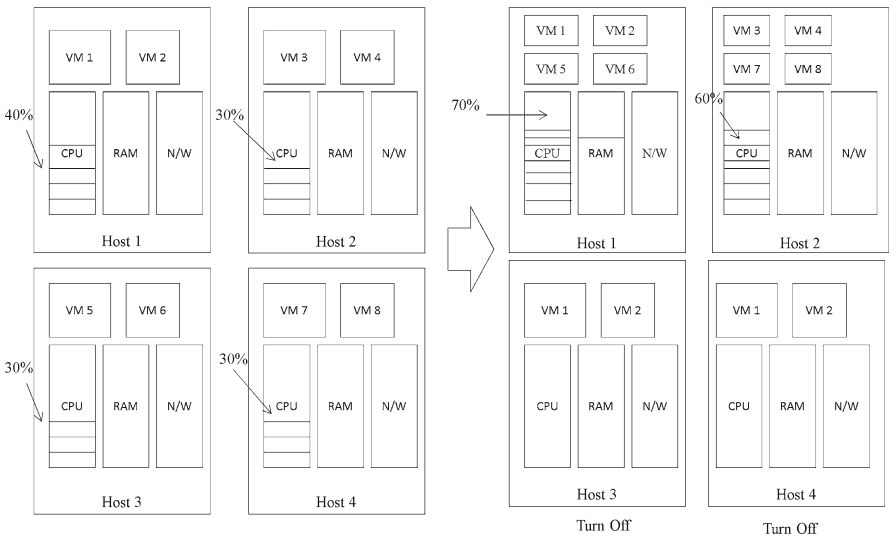

Fig. 4 Illustration of underutilized Hosts

Algorithm 2 VM Migration Algorithm (QoS-MMP)

```
1 Input: Host_List Output: Migration_List
2 for host in Host_List do
3vm_List←host.getVmList()
4 vm_List.sortDecendingOrderOfUtilization()
5 hostUtil←host.getUtil()
6 bestFitUtililization←MAX
7 while hostUtil> UPPER_THRESHOLD do
8                 If Vm.UserType==TimeContriant
9                   Cloudletstatus();
10                  endif;
11    t←Vm.getUtil() − hostUtil + UPPER_THRESHOLD
12  if   t <bestFitUtil then
13   bestFitUtililization ←t
14   bestFitVm ←vm
15   else
16   if bestFitUtililization = MAX then
17    best_Vm←vm
18    break
19    hUtil←hUtil − best_Vm.getUtil()
20    migrationList.add(bestFitVm)
21   vm_List.remove(bestFitVm)
22   If hostUtil< LOWER_THRESHOLD then
23                If Vm.UserType==TimeContriant
24                  Cloudletstatus();
25                endif;
26                If Vm.UserType==BudgetContriant
27                    migration_List.add(h.getVm_List())
28                    vm_List.remove(h.getVm_List())
29                endif
30          endif
31 return migration_List
```

3.5 An illustrative example

In order to show how the algorithm works, we trace its operation using the sample scenario as depicted in Fig. 4. We assume there are four available computation hosts (Host 1, Host 2, Host 3 and Host 4) which can be used to execute the different types of jobs. Algorithm 1 used for initial placement of VMs and secondly placement of VMs when VM selected for migration. Algorithm 2 used for VM migration. There are two situations for VMs migration in the first case when hosts are underutilized in this case algorithm migrate all migrate VM, since we take care of users QoS, we will not migrate VM which have deadline constraint jobs. In Fig. 4, Host 1 has 40% utilization and Host 3 has 30% utilization, both hosts are underutilized, we can efficiently utilize hosts if we migrate Host 3 VMs to Host 1. Similarly, Host 2 and Host 4 also underutilize

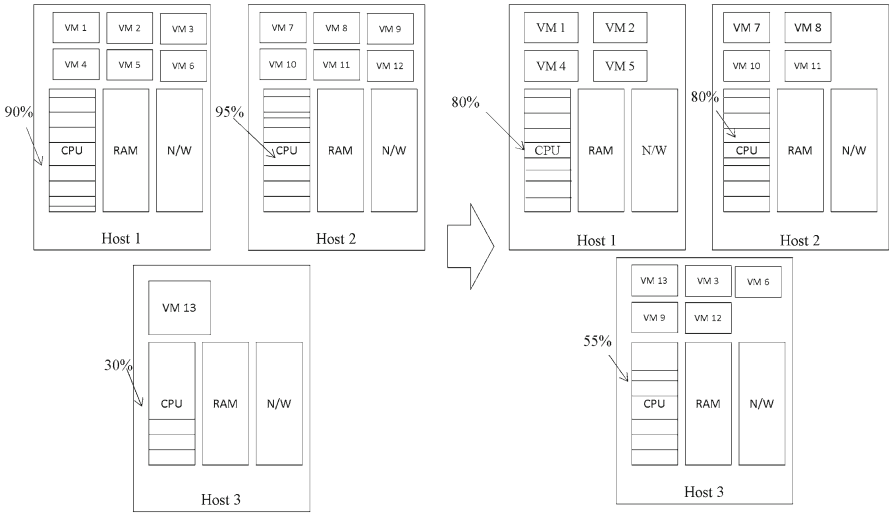

Fig. 5 Illustration of overloaded hosts

and we can migrate VMs of Host 4 to Host 2; in this way, we can efficiently utilize recourses.

Conversely, if hosts get overloaded, we should balance the load of servers, since overloaded server causes SLA violations due to SLA violations those users not able to get services or waiting for resources which hold by another resource. We can achieve load balancing via VM's migration. In Fig. 5, Host 1 has 90% utilization and Host 2 has 95% utilization, both hosts are overloaded, we can efficiently balance load of hosts, if we migrate VMs from Host 1 to Host 3 until host load reduces up to an acceptable range. Similarly, Host 2 also over-loaded and we can migrate VMs of Host 2 to Host 3; in this way, we can efficiently balance load of servers.

4 Results and discussion

This section is dedicated to the performance evaluation of our QoS-aware VM allocation and migration algorithm presented in the last section. First, we introduce the setup of the simulated setup (environment). After that, simulation results are presented from multiple aspects, including energy consumption, SLA violations and the number VM migrations.

4.1 Simulation setup

To analyze and compare the proposed algorithm, we simulated and implemented VM placement and migration algorithm in Java language-based simulator named as CloudSim. CloudSim is a simulation framework developed by the GRIDS laboratory of University of Melbourne which enables seamless modeling, simulation and experimenting on designing Cloud computing infrastructures. It can be used to model data

Table 2 Typical characteristics of hosts and VMs

Host	
Max power	250 W
Static power	0.7 70%
MIPS	(1000, 2000, 3000)
Storage	1,000,000 MB
Bandwidth	100,000 Mbps
VM	
MIPS rating	(250, 500, 750, 1000)
Number of CPU	1
RAM	128 MB
Bandwidth	2500 Mbps
VM size	2500 MB

centers, host, service brokers, scheduling and allocation policies of a large-scale Cloud infrastructure [11]. Simulation of the data center is performed which comprising of 400 physical heterogeneous nodes. As shown in Table 2, each of the hosts is demonstrated that consist of one CPU core in which the performance is equal to 3000, 2000, 1000 MIPS, 1 TB of storage and 8 GB of RAM. Similar to host, VM also requires one core for each VM which is equal to the performance of 1000, 750, 500 MIPS, 256 MB RAM is used for each VM. The request is sent by the user for the provisioning of the heterogeneous VMs; each VM runs any type of the application with the adjustable workload. In this simulation setup, each application takes 10 min to its execution; therefore, we assign 150,000 MI length to each application. At the beginning, each virtual machine is placed on behalf of demanding characteristic supposing 100% CPU utilization. To make experiment-based evaluation repeatable, it is significant to perform experiment using workload traces from a real system. These types of workload traces allow the experiments to be repeated as many times as needed. For the experiments, we used workload traces data provided as a part of the CoMon project, monitoring infrastructure of PlanetLab [27]. The traces include data on the CPU utilization collected every 10 min from more than thousand VMs deployed on servers located in more 500 places around the world.

4.2 Performance comparison parameters

To compare the efficiency of the algorithm, we use several metrics to evaluate their performance. We used following QoS parameters, to evaluate the performance of the proposed algorithm with an existing algorithm related to VM placement and migration.

4.2.1 Power consumption

The first parameter is the energy consumption of physical resources of a data center caused by the application workloads. We use following model for the calculation of the power of a physical server.

$$P(u) = \mathrm{k} * P_{\max} + (1 - k) * P_{\max} * u \tag{4}$$

K is a constant and represents the power consumption of an idle server, where $P_{\max}$ represent the maximum power consumption when server is fully used, and c is the current CPU usage. For our simulations, $P_{\max}$ is assigned to 250 W, which is a normal value for recent servers. For instance, according to the SPEC power benchmark, for the fourth quarter of 2010, the average power consumption at 100% utilization for servers consuming less than 1000 W was approximately 259 W.

4.2.2 Number of VM migration

The second parameter is the number of VM migrations started by the VM manager during the adjustment of the VM placement.

4.2.3 Million instructions per second (MIPS)

MIPS represents the CPU load and total capacity of VMs and hosts; in CloudSim environment, Cloudlet is simply a task or job submitted to the cloud. The MIPS required by a task totally depends upon the length or size of a cloudlets. If we provide more MIPS to our task, then it will execute fast. Moreover, cloudlet length is the number of instructions that the processor is going to execute. If you have a cloudlet which length is 2500 and a VM with 250 MIPS, then it is going to be executed in $2500/250 = 10$ s. In this way, we can predict the time taken to complete the task.

4.2.4 SLA violation

The fourth parameter for evaluation of proposed algorithm is SLA violations, and SLA violation occurs when the user does not get their requested resources. In technical term, we can say SLA violations occur when VM cannot acquire the amount of MIPS that are requested. We use Eq. 4 for the SLA violations calculations. This metric displays the level by which the QoS requirements discussed among the consumers and resource provider are violated due to the energy-aware resource management.

$$\mathrm{SLA} = \frac{\sum(\text{requested MIPS}) - \sum(\text{allocated MIPS})}{\sum(\text{requested MIPS})} \tag{5}$$

4.2.5 Execution time

With the help of following formula, we find the execution time of cloudlet.

$$\mathit{Execution\ Time} = \mathit{cloudlet.getCloudletLength}()/(\mathit{vm.getMips}(); \tag{6}$$

We can estimate the current task execution with the help of task overall execution time; for this, we continuously monitor the ready queue. For instance, if cloudlet expected

execution time is 10 s, it already takes 9 s. Using this, we can estimate that task has 90% executed.

4.3 Comparison of energy consumption and SLA violations of single-threshold policy

The objectives of this experiment are to evaluate the performance of energy consumption and SLA violations with NPA and DVFS policies. For this scenario, experiment has the minimum values of the parameter settings with 400 hosts, 500 VMs, and 1000 Cloudlets. We simulated a Non-Power-Aware (NPA) policy, this policy does not perform any power-aware optimization, and it means that using NPA policies all servers execute at 100% and consume a full power even if the system is in idle position. DVFS is another benchmark policy that adjusts the voltage and frequency of the CPU on the basis of current utilization; however, DVFS does not perform any adjustment of assignment of VMs in run time. Single threshold (ST) is another VM selection policy for VM migration. This algorithm works by assigning the upper threshold for server and provisioning VM to each server within that specified threshold. In this experiment, the NPA policy consumes 35.3 kWh, while DVFS policy reduces this value to 21 kWh. To measure ST policy, we performed a number of experiments with several values of the utilization threshold. The simulation results are shown in Figs. 6 and 7, it shows that energy consumption can be considerably decreased compared to the NPA and DVFS policies 77% and 56% individually with 6.3% of SLA violations.

4.4 Performance comparison energy consumption

The objectives of this experiment are to evaluate the performance of energy consumption of the proposed algorithm with various power-aware algorithms. In this scenario, we used similar simulation setup which have 400 hosts, 500 VMs, and 1000 cloudlets.

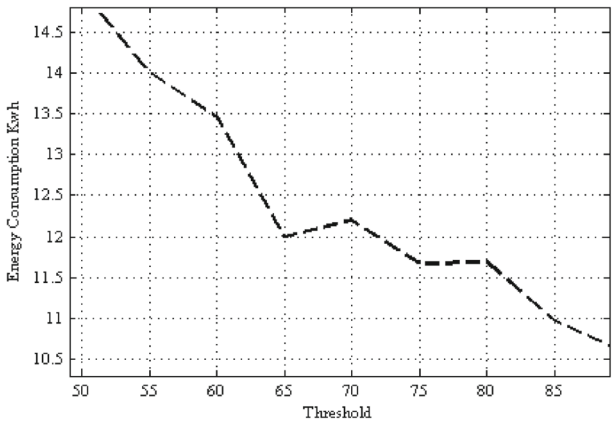

Fig. 6 Energy Consumption and Threshold comparison

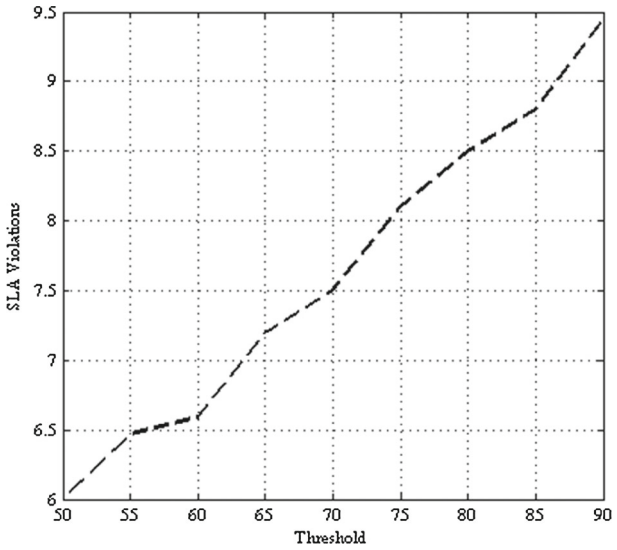

Fig. 7 SLA Violations and Threshold comparison

Figure 8 demonstrates the energy consumption comparison of power aware policies, it shows that energy consumption can be considerably decreased compared to the NPA and DVFS policies. In this graph, y-axis shows the energy consumption and the x-axis shows the different energy-aware algorithm. In this experiment, the NPA policy consumes 35.3 kWh, while DVFS policy reduces this value to 21 kWh. It also shows that MMP policy consumes 8.7 kWh, with 30% lower utilization and 70% upper threshold. At last, we compare our algorithm with MMP and other power-aware policies. We find that our VM placement and migration policies consume less power than MMP and all other policies. From the presented results, we can conclude that our policy provides

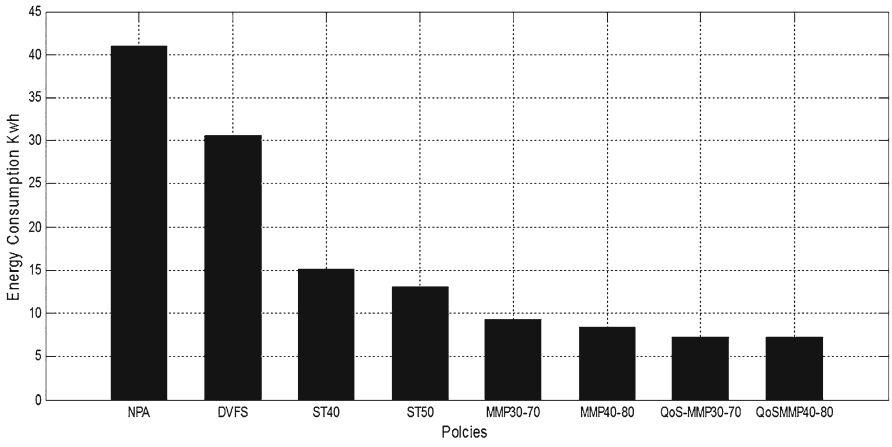

Fig. 8 The energy consumption

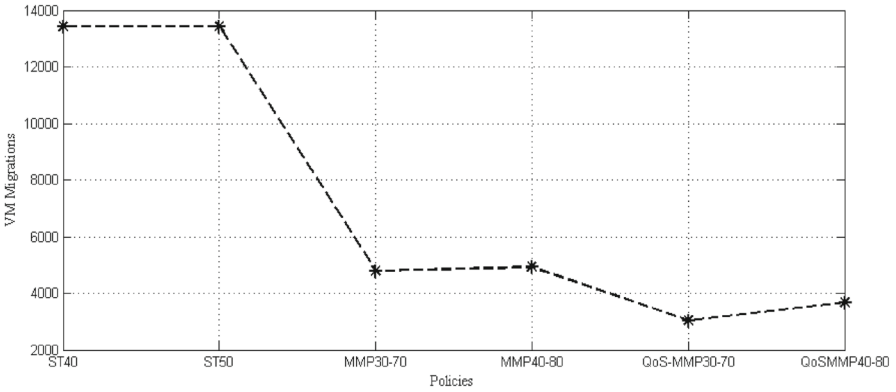

Fig. 9 The number of VM migration

the best energy savings with the least SLA violations and number of VM migrations among the evaluated policies for the simulated scenario, and it is due to fact that we migrating VM which have number of VM in running stage.

4.5 Performance comparison of the number of VM migration events

This experiment is designed to compare proposed algorithm number of VM's migration with various existing algorithms. Secondly, our ultimate objective of VM migration is to balance the load of sever by migrating VM from an overloaded server to another suitable server; further, we can achieve energy efficiency by turning off idle server before migrating VM from underutilized server. For this scenario, experiment has the minimum values of the parameter settings with 400 hosts, 500 VMs, and 1000 cloudlets. The y-axis on the graph shows the number of VM migrations, and the x-axis shows the various algorithms.The number of migrations performed by the proposed algorithms and other algorithm is presented in Fig. 9. Experimental results show that a QoS-aware MM algorithm leads to a significant decrease in the number of VM migrations and performs better than existing algorithms, viz. ST and MMP. The reason behind this is that the QoS-aware MM algorithm reduces number of VM migration; in our proposed algorithm, VM migration event is called only if there is no running cloudlet or no deadline cloudlet inside VM.

4.6 Performance comparison SLA violations

To compare the efficiency of our algorithm, we evaluated performance using several metrics such as SLA violations with other researchers' proposed algorithms. We define that SLA violation occurs when a given VM cannot get the amount of resources that are requested or the services fails to finish within deadline.

For this scenario, experiment has the minimum values of the parameter settings with 400 hosts, 500 VMs and 1000 Cloudlets. Figure 10 demonstrates the SLA violation percentage comparison of various power aware policies, in this graph y-axis shows the

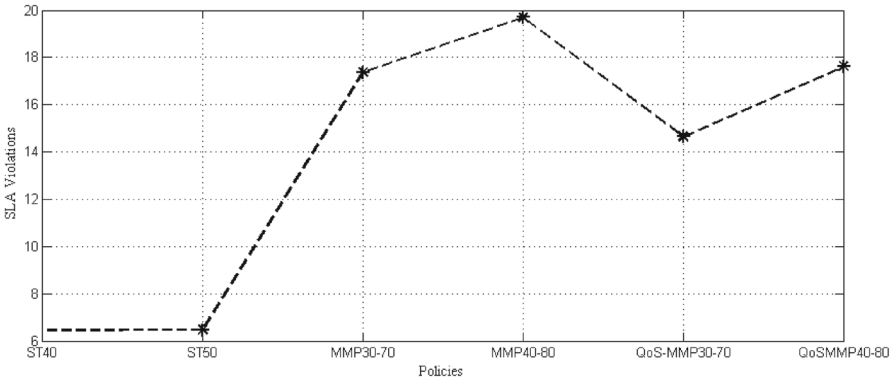

Fig. 10 The percentage SLA violation

Table 3 Summary of comparison results (energy consumption, VM migrations, SLA violations)

Technique	Energy KWh	SLA violation	VM migrations
NPA	40.95	–	–
DVFS	30.66	–	–
ST 50	15.05	6.44	13,433
ST 60	13.03	6.47	13,433
MMP 30–70	9.28	12.37	4801
MMP 40–80	8.44	16.7	4931
QoS-MMP 30–70	7.23	19.65	3034
QoS-MMP 40–80	7.27	23.62	3675

SLA violation percentage while x-axis shows the various algorithms. Experimental results show that QoS-aware MMP significantly decreases SLA Violations compared to the other policies such as MMP policies. Although ST policy has less SLA violation than our algorithm, this is due to the fact that ST energy performance trade-off. In this experiment, MMP 30–80 has highest SLA violations. Threshold (ST 50) and ST 60 have closure results to each other; however, MMP 30–70 has more SLA violation then both single-threshold (ST) VM selection policies. The results indicate that QoS MMP reduced the percentage SLA violation more efficiently than the other approaches. This is due to the fact that QoS MMP prevents the VMs from migration in which deadline cloudlets are running or cloudlets which deadline almost finish, secondly, we take care of those cloudlets whose execution almost completed, and in other words, if we migrate cloudlets which have 90% execution complete, then it should be re-executed some other hosts and then it consumes more budget and time (Fig. 9).

The overall simulation results (energy consumption, SLA violations, number of migration) are presented in Table 3.

Firstly, if we compare the energy consumption of the proposed algorithm (QoS-MMP) with DVFS, the proposed algorithm drastically reduces energy consumption, and results shown that the proposed algorithm on average leads to 23 kWh less energy consumption than DVFS. According to the results, an algorithm which pro-

vides the facility of dynamic reallocation of VM on the basis of current utilization is best for energy saving in contrast to the static resources provisioning algorithm, i.e., NPA and DVFS. Moreover, the proposed algorithm leads to more than 2 times less migration than double-threshold policy and almost 4 times less VM migration than singe-threshold policy (ST 60%). After evaluating proposed VM allocation and migration policy, we can say that the usage of the suggested algorithm offers the best energy utilization with the least SLA violations and number of VM migrations between the evaluated policies for the simulated scenario (Fig. 10).

5 Conclusion and future work

In this paper, we presented models and algorithms for virtual machine placement and virtual machine migration, which efficiently allocate resources by considering the various users QoS requirement and in other words considering users QoS requirements such as budget and deadline while migrating VM from host to another host. Further, we address the issue of load balancing which is one of the reasons of SLA violation, and secondly, we also handle the issue of server underutilization which is also one of the reasons of energy inefficiency. The simulation results show that QoS-aware MM algorithm not only maintains better SLA, rather it demonstrates rigorous reduction in power consumption, reduce number of VM migrations and decrease SLA violations.

The main limitation of our technique is network overhead, and this is due to migration of VMs from one host to another. In the future research work, we will implement intelligent techniques to manage the network resources efficiently. One of the ways to accomplish this for virtualized data centers is to constantly optimize network topologies established among VMs, and thus reduce network communication overhead and load of network devices. We will like to extend our model to consider the aspect of fault-tolerant VM migration. The data center may be very alarmed about the reliability of the live migration, because the failure of the VM migration will affect the usability of the application in the VM and possibly require the intervention of the VM users to recover the VM. As a result, the faulty migration rate should be considered while saving energy.

References

1. Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I (2009) Cloud computing and emerging it platforms: vision, hype, and reality for delivering computing as the 5th utility. Future Gener Comput Syst 25(6):599–616
2. Jonathan GK (2011) Growth in data center electricity use 2005–2010. A report by Analytical Press, completed at the request of The New York Times
3. Uddin M, Rahman A (2010) Server consolidation: an approach to make data centers energy. Int J Sci Eng Res 1(1):1–7
4. Beloglazov A, Buyya R, Lee YC, Zomaya A (2011) A taxonomy and survey of energy-efficient data centers and cloud computing systems. Adv Comput 82(2):47–111
5. Mishra M, Sahoo A (2011) On theory of vm placement: anomalies in existing methodologies and their mitigation using a novel vector based approach. In: 2011 IEEE International Conference on Cloud Computing (CLOUD), pp 275–282
6. Meng X, Pappas V, Li Z (2010) Improving the scalability of data center networks with traffic-aware virtual machine placement. In: INFOCOM, 2010 Proceedings IEEE, pp 1–9

7. Le K, Bianchini R, Zhang J, Jaluria Y, Meng J, Nguyen TD (2011) Reducing electricity cost through virtual machine placement in high performance computing clouds. In: Proceedings of 2011 International Conference for High Performance
8. Verma A, Ahuja P, Neogi A (2008) pMapper: power and migration cost aware application placement in virtualized systems. In: Middleware. Springer, Berlin, pp 243–264
9. tillwell M, Schanzenbach D, Vivien F, Casanova H (2010) Resource allocation algorithms for virtualized service hosting platforms. J Parallel Distrib Comput 70(9):962–974
10. Yao C-CA (1980) New algorithms for bin packing. J ACM 27:207–227
11. Beloglazov A, 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
12. Beloglazov A, Buyya R (2010) Energy efficient allocation of virtual machines in cloud data centers. In: 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid)
13. Pinheiro E, Bianchini R, Carrera EV, Heath T (2001) Load balancing and unbalancing for power and performance in cluster-based systems. In: Proceedings of the Workshop on Compilers and Operating Systems for Low Power (COLP)
14. Bobroff AN, Kochut A, Beaty K (2007) Dynamic placement of virtual machines for managing SLA violations. In: Proceedings of 10th IFIP/IEEE International Symposium on Integrated Network Management IM'07, pp 119–128
15. Song J, Li T-T, Yan Z-X, Na J, Zhi-Liang Z (2012) Energy-efficiency model and measuring approach for cloud computing. Ruanjian Xuebao J Softw 23(2):200–214
16. Dodonov E, Rodrigo FM (2010) A novel approach for distributed application scheduling based on prediction of communication events. Future Gener Comput Syst 26(5):740–752
17. Tuan AT, Gyarmati L (2010) How can architecture help to reduce energy consumption in data center networking?. In: Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking, pp 183–186
18. Kusic D, Jeffrey O, Kephart J, Nagarajan HEK, Guofei J, Kusic D, Kephart JO, Hanson JE, Kandasamy N, Jiang G (2009) Power and performance management of virtualized computing environments via lookahead control. Cluster Comput 12(1):1–15
19. Nathuji R, Schwan K (2007) Virtualpower: coordinated power management in virtualized enterprise systems. ACM SIGOPS Oper Syst Rev 41(6):265–278
20. Sharifi M, Salimi H, Najafzadeh (2012) Power-efficient distributed scheduling of virtual machines using workload-aware consolidation techniques. J Supercomput 81(1):46–66
21. Abrishami S, Naghibzadeh M, Epema D (2013) Deadline-constrained workflow scheduling algorithms for IaaS clouds. Future Gener Comput Syst 29(1):158–169
22. Ferreto TC, Netto MAS, Calheiros RN, De Rose CAF (2011) Server consolidation with migration control for virtualized data centers. Future Gener Comput Syst 27(8):1027–1034
23. Haikun L, Hai J, Cheng-Zhong X, Xiaofei L (2011) Performance and energy modeling for live migration of virtual machines. In: Proceedings of the 20th ACM International Symposium on High-Performance Parallel and Distributed Computing, pp 171–181
24. Jong-Geun P, Jin-Mee K, Hoon C, Young-Choon W (2009) Virtual machine migration in self-managing virtualized server environments. In: Proceedings of the 11th International Conference on Advanced Communication Technology (ICACT '09), pp 2077–2083
25. Mao M, Li J, Humphrey M (2010) Cloud auto-scaling with deadline and budget constraints. In: Proceedings of 11th ACM/IEEE international conference on grid computing, 25–28 Oct 2010
26. Beloglazov A, Jemal A, Rajkumar B (2012) Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future Gener Comput Syst 8(25):755–768
27. Park KS, Pai VS (2006) CoMon: a mostly-scalable monitoring system for Planet-Lab. ACM SIGOPS Oper Syst Rev 40(1):65–74
28. Viswanathan H, Lee EK, Rodero I, Pompili D, Parashar M, Gamell M (2011) Energy-aware application-centric vm allocation for hpc workloads. In: Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), pp 890–897
29. Goiri Í, Berral JL, Fitó JO, Julià F, Nou R, Guitart J, Torres J (2012) Energy-efficient and multifaceted resource management for profit-driven virtualized data centers. Future Gener Comput Syst 28(5):718–731
30. Buyya R et al (2010) Efficient management of data center resources for cloud computing: a vision architectural elements and open challenges. In: Proceedings of the 2010 international conference on parallel and distributed processing techniques and applications, pp 1–12

31. Pettey C (2007) Gartner estimates ICT industry accounts for 2 percent of global CO_2 emission. http://www.gartner.com/it/page.jsp?id=503867. Accessed 11 Mar 2016
32. Borgetto M, Casanova H, Da Costa G, Pierson JM (2012) Energy-Aware Service Allocation. Future Gener Comput Syst 28(5):769–779
33. Yue M (1991) A Simple Proof of the Inequality FFD(L)≤11/9 OPT (L) + 1 for All L for the FFD Bin-Packing Algorithm. Acta Math Applicatae Sinica (English Series) 7:321–331
34. Chase JS, Anderson DC, Thakar PN, Vahdat AM, Doyle RP (2001) Managing energy and server in hosting center. ACM SIGOPS Oper Syst Rev 35(5):102–116
35. Elnozahy EM, Kistler M, Rajamony R (2003) Energy-efficient server clusters. In: Power-aware computer systems, vol 2325, pp 179–197
36. Srikantaiah S, Kansal A, Zhao F (2008) Energy aware consolidation for cloud computing. In: Proceedings of the 2008 USENIX Workshop on Power Aware Computing and Systems (HotPower), pp 1–5
37. Nathuji R, Isci C, Gorbatov E (2007) Exploiting platform heterogeneity for power efficient data centers. In: Proceedings of the 4th International Conference on Autonomic Computing (ICAC)
38. Kusic D, Kephart JO, Hanson JE, Kandasamy N, Jiang G (2009) Power and performance management of virtualized computing environments via lookahead control. Cluster Comput 12:1–15